1

ONRFUR Report

90-4-C

The U.K. Information Technology 1990 Conference

J.F. Blackburn

29 June 1990

Approved for public release; distribution unlimited

Office of Naval Research European Office

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION/DOWNGRADING SCHEDULE

3 DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
90-4-C

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
Office of Naval Research
European Office

6b OFFICE SYMBOL (If applicable)
ONREUR

7a NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Box 39
FPO NY 09510-0700

7b ADDRESS (City, State, and ZIP Code)

8a NAME OF FUNDING/SPONSORING ORGANIZATION

8b OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
The U.K. Information Technology 1990 Conference

12 PERSONAL AUTHOR(S)
J.F. Blackburn

13a TYPE OF REPORT
Conference

13b TIME COVERED
FROM ________ TO ________

14 DATE OF REPORT (Year, Month, Day)
29 June 1990

15 PAGE COUNT
17

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP
02	25	

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Information Technology

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The U.K. Information Technology (IT) 1990 Conference took place at the University of Southampton, March 19-22, 1990. The conference was organized by the Computing & Control Division, Institution of Electrical Engineers, along with the British Computer Society, the Department of Trade and Industry, and the Science and Engineering Research Council. To some degree, it replaces the Alvey Conferences held annually between 1983 and 1988.

The main topics covered in this conference were each supported by several technical papers. The main topics covered were: The Cycle of Software Engineering; Knowledge, Acquisition, and Development Methodology; Security and Fault Tolerance; Systems, Concurrency, Real Time; Vision; Control Systems & Environments; Formal Modeling Methods; System, Quality, Metrics; Parallel & Distributed Architecture; VLSI; Experimental Systems; Formal Aspects; System Design; Tools; Instrumentation & Intelligence; KBS; Speech Applications; Object Oriented Systems; KBS with Other Technologies; and Speech Language and Vision. (KR) ←

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
Ms. Connie R. Orendorf

22b TELEPHONE (Include Area Code)
(44-71) 409-4340

22c OFFICE SYMBOL
310

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

☆ U.S. Government Printing Office: 1986—607-044

UNCLASSIFIED

Contents

LIST OF TABLES

Table Page

The U.K. Information Technology 1990 Conference

Introduction

The U.K. Information Technology (IT) 1990 Conference took place at the University of Southampton, March 19-22, 1990. The conference was organized by the Computing & Control Division, Institution of Electrical Engineers, along with the British Computer Society, the Department of Trade and Industry, and the Science and Engineering Research Council. To some degree, it replaces the Alvey Conferences held annually between 1983 and 1988.

The main topics covered in this conference were each supported by several technical papers (see Table 1).

Table 1. Conference Main Topics

- The Cycle of Software Engineering
- Knowledge, Acquisition, and Development Methodology
- Security and Fault Tolerance
- Systems, Concurrency, Real Time
- Vision
- Control Systems and Environments
- Formal Modeling Methods
- System, Quality, Metrics
- Parallel and Distributed Architecture
- Very Large Scale Integration (VLSI)
- Experimental Systems
- Formal Aspects
- System Design
- Tools
- Instrumentation and Intelligence
- Knowledge-Based Systems (KBS)
- Speech Applications
- Object Oriented Systems
- KBS with Other Technologies
- Speech Language and Vision.

Summaries

The following paragraphs present summaries of the papers.

The Cycle of Software Engineering

Two papers were given in this section dealing with software engineering and instrumentation and what was termed 'he "death cycle."

Engineering-in-the-Large: Software Engineering and Instrumentation, A. Finkelstein and T. Maibaum, Imperial College of Science, Technology, and Medicine, U.K., and City University, U.K.

In this paper, the authors attempted to establish the conceptual foundations on which technology transfer can be built between complex instrument systems design and software design.

They examined current practice in these areas and illustrated significant differences in the way in which the different disciplines regard the development process and the organization of automated support for that process. These differences constitute an important barrier to technology transfer between disciplines. The differences also may be a major barrier to establishing a coherent approach to system engineering.

The authors outlined some techniques that appear appropriate for technology transfer: formal specification, systematic software process modeling, explicit value modeling, and requirements and concept generation. Despite some problems, they suggested that these techniques may provide real benefits for both software engineering and instrument design.

The Software Death Cycle, P.J. Rigby and M.T. Norris, British Telecom Research Laboratories, U.K.

Software is now one of a company's most important assets. As a result, it is vital that it is considered on the same basis as other assets. This paper showed how the costs of software ownership can be modeled and from this, how the viable life of a piece of software can be determined.

The basic premise that underlies the model is that quality maintenance is costly and must be offset against the benefits that accrue from that software. At some point, ownership cost begins to outweigh benefit and this is the point at which net income becomes negative. At this point, the software should be retired.

The ideas developed in this paper bring together experience in managing software maintenance and quality, and relate the two on a cost basis. The result is a model that matches observed reality across a wide range of research and development (R&D) projects. Although the authors attach no scale to the cost curves developed in the paper, they used a short example abstracted from a real situation to illustrate the practical application.

In explicitly modeling the cost aspects of software over its whole lifetime, concepts (like its depreciation) become apparent. Because of the ever-increasing reliance on software within high technology industries, this modeling will surely become a future commercial necessity.

by J F. Blackburn. Dr. Blackburn is the London representative of the Commerce Department for Industrial Assessment in Computer Science and Telecommunications.

Knowledge Acquisition and Development Methodology

This section dealt with knowledge elicitation of manufacturing and production skills and with a knowledge-based approach to flexible manufacturing system (FMS) scheduling.

An Approach to Knowledge Elicitation and Manufacturing Skills and Production Behaviors in an Industrial Environment, S.M. Hajsadr and A.P. Steward, Sunderland Polytechnic, U.K.

The authors believe that the root concept of all expert systems is the communication of knowledge. They illustrated two possible concepts for knowledge communication: (1) a unidirectional communication channel which illustrates an environment commonly called a single expert domain, and (2) a bidirectional communication channel illustrating an environment where expertise is held by multiple experts.

In the project described, the authors adopted the second type of communication channel. Thus, they benefitted from greater user involvement. The editor that was provided as a bulletin board encouraged feedback between users and between users and knowledge engineer. They acquired knowledge using a group of operators who were committed to and supported by the knowledge engineer, participating in all factory floor activities and demonstrating the advantages of the system. They also gathered the knowledge without disrupting the production factory schedule ard without using costly overtime.

Most manufacturing problems require involving a range of functions within the enterprise for their solutions. A knowledge-based approach that employs the single expert in a manufacturing context may fail. The single expert model has been used in many expert system projects; this may explain their lack of penetration in manufacturing. The notion of problem solving tasks to be handled by teams is not a new one in manufacturing; quality circles and just-in-time teams are well established in many companies. At the same time, there is interest in using computer technology in solving problems within small groups.

The authors suggested that the work of these problem solving groups is linked to knowledge acquisition in two ways. First, the initial deliberation is often a knowledge-building exercise to produce something akin to what they call the intermediate model. Software shells can help by providing a domain-independent structure in which the expertise of the group can fit. Second, expert systems technology (knowledge acquisition in particular) can learn from group problem solving experiences. If problems are naturally solved in groups, then the methods should reflect the working of the group. Methods for problem representation such as Issue-Based Information Systems (IBIS) can contribute to developing knowledge-based software which is less constrained than in present expert system shells.

A Knowledge-Based Approach to Flexible Manufacturing System (FMS) Scheduling, D.H. Walburn and E.T. Powner, British Aerospace Ltd., U.K. and University of Manchester Institute of Science and Technology, U.K.

The authors developed a knowledge-based scheduler based on an object-oriented model of the British Aerospace Small Machine Parts FMS. To determine the suitability of the problem for artificial intelligence (AI) treatment, they investigated the domain of the FMS and the use of knowledge-based techniques in scheduling. They decided that the topic provided a suitable complex and well-defined problem and also provided practical application of the developed techniques.

Accordingly, they established the relationship of the cooperating rule bases to the object-oriented FMS model before describing the extent of the representation and its functionality. Then, they attempted route generation in an incremental fashion. They attempted to schedule jobs one by one, according to a rigid job template. If the allocation of a particular machine at a particular time failed, they produced an exception. They handled the exception using rule bases that analyzed the nature of the contention and determined an appropriate rescheduling action.

As background to the development environment, they introduced the tools and techniques. Object-oriented and rule-based approaches provided the tools, and the techniques were inspired by the Muse development environment. Muse is an AI toolkit for designing real-time intelligent KBS. Object-oriented programing led to developing an object-oriented causal model. Eventually, providing independently operated knowledge sources led to distributing analysis and rescheduler rule bases.

The authors were able to describe Muse techniques in more detail by unfolding the system object and describing the nature of implementation of the knowledge sources. They introduced the architecture of the knowledge-based scheduler from a consideration of the flow of information between the model, the analyzer, and the rescheduler. This led to describing the implementation using Muse structures from the perspective of information flow. They showed information flow for each type of exception for both the knowledge source and the rescheduler. Identified advantages of the knowledge-based approach taken relate to complexity, rapid prototype developing, maintaining, distributing, embedding, and deep reasoning.

The authors addressed speed performance of the development techniques through consideration of the memory requirement/speed tradeoff. Improvements in speed by recoding at a lower level and the investigation of alternative architectures for implementation were suggested. For example, it would be possible to construct a development environment where the knowledge sources run in parallel. Different exceptions could be resolved simultaneously if they preserved the present implementation. A finer-grained structure would allow resolution of the implementation to be further increased as appropriate; e.g., through concurrency in the represented objects.

The constraint of static knowledge sources is recognized. The use of a system to improve automatically the knowledge sources of the system was proposed. Researchers at this university developed a system for automatic knowledge acquisition. The prototype demonstrated the usefulness of the techniques used and their relevance to the FMS marketed by British Aerospace. The application of the techniques in practice is currently under investigation.

Security and Fault Tolerance

Three papers in this segment covered authentication using smart cards, network security, and fault diagnosis.

Authentication in a UNIX Network Environment Using Smart Cards, S.T. Jones and M.J. Clark, British Telecom Research Laboratories, U.K., and University of Essex, U.K.

Experimental implementations show that an intelligent card can easily be incorporated into the existing UNIX login system. The resulting configuration potentially offers a greater level of security. In a network of workstations sharing access to common resources, including fileservers, this becomes important. Most workstations have spare serial ports on which a smart card reader may be interfaced through windows on the console screen, not interfering with the user's interface.

The performance of local logins is acceptable, but adding a smart card allows greater control with improved authentication methods. Because the existing weakness of unencrypted passwords on the network is removed, smart cards enhance the performance of remote logins. The system's performance was very satisfactory; the increased login time was far less than the variation in login times caused by changes in system and network loadings. The choice of the authentication technique used between the systems and the card is the subject of further research; however, the method used in implementation is effective but vulnerable to a masquerade attack. However, the use of some more secure techniques involving encryption and keys will be able to overcome this deficiency.

Extending card-based authentication to applications allows a greater range of facilities to be provided. Using a secure remote procedure call mechanism, an application can perform complex authentication and access control operations securely and efficiently without interaction with the user. The smart card also can be used to store data concerning the users access to resources and can be applied to billing procedures. One could extend the authentication mechanisms described previously to encompass resource usage logging or to provide a mechanism for providing credits toward more expensive resources.

Design of a Network Security Policy Model, V. Varadharajan, Hewlett-Packard Laboratories, U.K.

In this paper, the author considered the development of an abstract, multilevel network security policy model. Recently, the question of the design of a security policy model has been intensely scrutinized. This is because the Department of Defense published the Trusted Computer System Evaluation Criteria and the Trusted Network Interpretation. The model addresses some of the access control and information flow control requirements of a multilevel secure network. The network access control policy determines the requirements for establishing connections between network components, and information flow policy regulates the flow of information between the network components. The model formally proves that the access control requirements are not violated and the information does not flow from higher security classes to lower ones as a result of network operations.

In developing network security policies, you must relate them to specific layers of the network architecture. In particular, whether a security policy makes sense or not depends on the layer to which it is applied. Varadharajan discussed this by looking at another security policy model with different network subjects and objects.

In addition to these access control and information flow requirements, the communications must be protected. Hence, appropriate cryptograhic techniques to provide confidentiality, integrity, and authenticity of communications are necessary. The author did not consider such cryptographic aspects.

Fault Diagnosis in Embedded Software, D.J. Lumby and A.G. Stoddart, British Telecommunications Laboratories, U.K.

Real-time software engineering is a relatively new discipline. As practiced, the subject perpetuates the concept that testing in the laboratory finds all faults in software. In real-time systems where extensive use is made of decision tables or loop constructs, software fails. The more decisions made, the more paths there are through the system, and the less likely it becomes that all combinations can ever be tested. Real-time systems, no matter how well designed, do suffer from occasional data corruption and therefore testing cannot ever guarantee reliability.

The authors outlined some of the models of failure observed in practice. They discussed several techniques for dealing with in-service problems. The key observation from their work was that some mechanism for trapping and reporting the inevitable shortcomings of the initial design is an important factor in easing subsequent maintenance. Not only does such an approach make it easier to find and fix faults, it also helps in understanding what questions to ask when checking an initial design. "Right first time" is a noble aim, but in practice, techniques to protect against mistakes and erroneous events are a key part of good software practice.

Systems, Concurrency, Real Time

The two papers in this section covered specification of concurrent systems and real time AI.

Definitive Specification of Concurrent Systems, W.M. Beynon and M.D. Slade, University of Warwick, U.K., and M.T. Norris and R.A. Orr, British Telecom Research Laboratories, U.K.

In this paper, the authors described a new approach to the modeling and simulation of concurrent systems, characterized by the use of sets of definitions to represent states and transitions. They simulated system behavior from a specification of possible agent actions by considering relative speeds of response and operation. A distinctive feature of the approach is that the perceptions and capabilities of agents are explicitly modeled.

The paper described real progress toward understanding the problems of relating the interaction-oriented and event-oriented perspectives on concurrent system modeling, as they are identified in the context of a definitive approach to programing. The solutions proposed in the paper are not sufficiently well-developed to meet the ultimate objective of relating the abstract behavior of the system to the roles played by the participating agents as perceived by the designer. Nevertheless, they have provided the basis for prototype systems that can assist the designer in analyzing requirements and developing a formal specification. To guide future work on tools, more research must take place with the proposed design method on abstract programing and practical experience.

Real-Time Artificial Intelligence: A Practical Solution to a Difficult Problem, C.B. Marshall and P.R. Simmons, Philips Research Laboratories, U.K.

Data often has to be interpreted as it arrives, in order to build up and maintain an accurate and up-to-date model of the environment. Data to be interpreted can be coming in from one or more sources--sensors, performance, miscellaneous data--depending on the application.

In this paper, the authors started with an introduction to the problems of Electronic Support Measures processing, which exemplifies clearly the problems of real-time data interpretation. The authors examined the features of the approach followed by an operator to this task, and discussed the implications of the necessity for a real-time solution. They described the software architecture used to reflect these features, and discussed the tools used to implement the software.

The paper then presented an assessment of the software performance, combined with comments of the implementation path they now are following. Finally, they provided some conclusions.

The blackboard architecture-based approach proved it can provide a dramatic improvement in quality of performance for the difficult task of interpreting radar pulse data. Through the flexible approach, experts solve problems as they arise; this is extremely effective. The low-level tools provided excellent results through using, in a practical way, the concepts and features illustrated in the work described: modeling the approach used by the operator, and in the basic design approach, considering the limitations inherent in real-time operation.

Vision

Six papers in the vision section covered image analysis, computer vision, pattern classification, self learning, and performance improvements.

Real-Time Image Analysis for Vision Guided Vehicle Control, B.T. Thomas, et al., University of Bristol, U.K.

This paper presented an overview of a 3-year investigation into vision-guided road vehicle control conducted by a University of Bristol team. The goal was to develop a parallel computer architecture, based on the Inmos transputer, that can analyze in real time video images obtained from a camera mounted on the front of a vehicle. The authors based much of the work on analysis of prerecorded videos of typical road scenes. One version of the system is battery powered and small enough to reside entirely on a small electric vehicle used as a demonstrator. They paralellized the algorithms used in the investigation enough to allow them to operate at near-frame rates (10-20 frames/sec). The two principal approaches adopted are road edge detection and surface segmentation.

The first approach to edge detection attempts to locate edges in the raw image by passing several oriented edge detectors across the image in localized regions inferred from a model of road boundaries. The detected edges are then used to update the model. The model consists of two concentric arcs in the real world horizontal plane. An accurate knowledge of the camera geometry transforms from the image plane to the real world environment. The model is determined by sorting the edge candidates into two groups, corresponding to the primary road edges. These are then fitted with second order polynomials, from which the center and radii of the two arcs can be inferred. The authors treated straight road segments as arcs with large radius of curvature. They steered the vehicle toward the center of the road so that displacements decrease exponentially with time.

The second edge detection technique is based entirely on real-world analysis. Initially, the authors smoothed and subsampled the raw image to reduce the total quantity of data. They based the position of each subsampled pixel on a transformation between the image plane and a birds-eye view of the road surface. One must have accurate knowledge of the camera geometry to effect this transformation, but the orientation of the vehicle with respect to the road edge is left as a free parameter. The primary idea behind this technique is that the correct orientation (or required steering correction) is the one that reduces the road edges to a pair of vertical boundaries in the real-world scene. The technique is quite sensitive to errors in the orientation, which leads to slanted edges in the transformed image. One uses a simple, vertical edge detector to locate the road boundaries in the resulting plan view image. Then one links them into long near-vertical segments. The correct orientation is assumed to be that which leads to the strongest and most vertical edges, any residual tilt being used to determine a more precise steering direction. A further safeguard in the technique is based on a real-world model which predicts the position and separation of the two road boundaries, updated from frame to frame. The transformation itself provides approximate steering direction, but the precise location of the boundaries can be used in adjusting the vehicle's position on the road.

The surface segmentation algorithm being used is based on grey level and texture. The algorithm does not use a simple seeding technique but dynamically thresholds the image to locate a connected area of similar pixels directly in front of the vehicle, occupying an area of the image appropriate for the road surface. A drawback to this technique is that the segmentation algorithm is not perfect, particularly near the boundaries of the road. Therefore, the authors implemented an extension of this technique based on the principle of fitting a variable width, road-shaped template to the segmented region. This technique is fast and efficient on reasonably consistent surfaces, but relies on the vehicle pointing approximately along the road. The authors are now working on methods to fuse the information from each of the above methods using a common model of the environment, thus producing a more reliable and robust system.

Computer Vision Based Automatic Construction of Three-Dimensional (3-D) Geometry Scene Models, A. Sharma and S.A.R. Scrivener, Logica, U.K., and University of Loughborough, U.K.

This paper described an approach to automatic construction of 3-D object models using images of the scene. The method makes use of a visible surface scene representation called the 2-D. The implementation uses multiple simulated 2-D sketches to construct a 3-D voxel-based geometric model. The depth information from each 2-D sketch is used for "chipping" voxels from a 3-D array of voxels. This process is analogous to a sculptor chipping a stone block, using spatially displaced views to create an imitation of the scene. The geometric model forms a basis for deriving a more complete and visually realistic 3-D graphic model. Pictorial results show that the reconstructed models are visually similar to the original models. A similarity exists even when simulated discrepancies are introduced into the 2-D sketches used; only three views are used for reconstruction.

Integrated Computer Vision with the Associative String Processor, A. Krikelis, Aspex Microsystems Ltd., U.K.

Based on results from research into parallel computing technology at Brunel University and developments by Aspex Microsystems, Associated String Processor (ASP) modules comprise highly versatile parallel processor building blocks for the simple construction of fault tolerant Massively Parallel Processors (MPPs).

An ASP module comprises a Multiple control of a Multiple-Single Instruction stream, Multiple Data stream (MIMSIMD) parallel processing structure of intercommunicating ASP substrings, each supported with an ASP Data Buffer (ADB) and an ASP Control Unit (ACU).

The use of the ASP architecture has been proposed for the implementation of integrated computer vision systems. The simple configuration of ASP modules simplifies the development of MIMD/SIMD MPP systems which are well matched to the functional requirements of computer vision and capable of delivering the very high performance necessary for real-time system implementation.

Implementation and Evaluation of Pattern Classification Architectures Using Transputer Arrays, M.C. Fairhurst and P.S.J. Brittain, University of Kent, U.K.

The aim of a pattern classification algorithm is the assignment of an unlabeled pattern represented as a feature vector X to one of k possible classes, where for each class the recognition system must compute a class conditional discriminant function. An assignment decision is then made by appropriate selection among the k functions thus computed. The implementation of a pattern recognition algorithm is inherently a parallel task. Although each discriminant function is a function of X, yet each is typically independent of the others, so that at the implementation level these computational tasks can be divided in a very natural way among a set of parallel processors. This is a particularly important consideration when high-speed operation is required in some specific application. This paper covered some important issues related to implementation: (1) the specification of the classifier itself and (2) the practical evaluation of the variety of configurations possible within the basic framework defined.

The paper focused on the problem of transputer-based implementation of algorithms for pattern classification. Information revealed that a very close relationship can be established between the conceptual level at which a classifier architecture is conceived, and the physical level at which implementation is achieved. This relationship is to the extent that novel conceptual structures can be evolved directly from a consideration of the computational characteristics identified at the physical level. The principle was demonstrated with respect to a transputer-based infrastructure that can directly exploit the inherent parallelism in most classification algorithms. Such an infrastructure can efficiently support other analytical and evaluation tools to provide the integrated environment for design and implementation. The infrastructure can offer great potential for the specification of high-performance systems for application in a wide variety of pattern recognition tasks.

A Self-Learning Algorithm Applied to Object Recognition, J.P. Cubillo and C.J. Bland, Coventry Polytechnic, U.K.

This paper briefly described the design of a low-cost, binary image, vision system that fits into the slots of a standard personal computer (PC). The authors also presented associated software for image processing and object recognition. They compared various techniques for object recognition and contrasted with that of a self-learning method.

The authors described a PC-based system that enables both real and artificially generated images to be captured, displayed, and processed. They implemented and tested three types of object recognition systems. The least squares of errors and correlation techniques achieved a high degree of success, but their limitations were also noted. The self-learning algorithm described achieved 100 percent success rate when asked to recognize a set of pretaught shapes.

Performance Improvements to the Hough Transform, Luciano da Fontoura Costa, Doron Ben-Tzvi, and Mark Sandler, Kings College, University of London, U.K., and University of Sao Paulo, Brazil

This paper described a new algorithm for the Hough transfo^rm, the Combinatorial Hough transform. The algorithm provides an improvement to the Hough space accumulator which gives greater resolution for a fixed size, and a new method for detecting line end points using the transform.

Also, the paper introduced a method for analyzing the performance of Hough transforms and exercised on the new algorithm, comparing their error performance with that of the standard Hough transform algorithm. The new algorithm with the parameter space improvements offers some real advantages over the standard and combinatorial algorithms, although this is dependent on the precise details of the implementation. The authors described the error analysis procedure in some detail, and discussed its effectiveness, together with desirable enhancements.

Control Systems and Environments

The three papers in this section dealt with manufacturing control systems, real-time adaptive control, and flexible parallel control.

An Environment for the Specification, Design, Operation, Maintenance, and Revision of Manufacturing Control Systems, P.J. Gawthrop and L. Smith, Engineering Design Research Centre, U.K., and Eurotherm Ltd., U.K.

The Science and Engineering Research Council at Glasgow University created the Engineering Design Research Centre to conduct fundamental and strategic research into engineering design in all its aspects. As one aspect of this, the Engineering Design Research Centre and Eurotherm Ltd. sponsored a project jointly that was initiated to:

- Conduct fundamental research into the modeling, simulating, and designing industrial control systems
- Produce a product, based on this research, to assist in the specifying, designing, operating, maintaining, and revising industrial control systems.

The paper surveyed the project, together with a report on initial progress in system modeling and representing. The project is still in its very early stages.

The project developed a prototype bond graph modeling tool, based on numerical solution of the bond graph, using Pascal, a procedural implementation language. They entered the parameter values in numeric form, which simplified implementation, and produced the values of the coefficients of the model state equations. Use of this tool highlighted the limitation of this approach-the solution for nonlinear systems is clumsy, and the final results give no insight into the physical significance of the individual bond graph elements.

Later, the researchers rewrote the tool using a symbolic approach, resulting in a much more complex implementation, but it provides the benefit of state equation coefficients output as functions of symbols representing the constitutive relations of each bond graph node. There are five modules in the tool to provide maximum flexibility: user input, causality analysis, state variable calculation, symbolic processor, and output formatter. They designed the modeling tool to operate in either a standalone mode or as a frontend to produce mathematical models for analyzing or simulating tools.

A Systolic Parameter Estimator for Real Time Adaptive Control, F.M.P. Gaston and G.W. Irwin, The Queen's University of Belfast, Ireland

Systolic arrays were proposed originally for high-speed VLSI signal and information processing applications where regular, matrix-type operations must be performed in real time on continuous data streams. The original concept has now evolved to encompass systolic computation, where the systolic architecture does not imply a VLSI realization, but rather provides a fundamental description of the data flow and computation of a complex algorithm.

The paper described a new systolic parameter (or constant state) estimator based on the square-root covariance Kalman filter algorithm. Kalman filtering is a fundamen' .l technique for both signal processing and state feedback control. Parameter estimation is a fundamental component of adaptive controllers and explicit self-tuning regulators. Recently, at an International Conference in Mathematics in Signal Processing, Warwick, England, a new architecture was proposed, and this one was compared with it. Gaston and Irwin showed it to be potentially more attractive for parallel control applications.

Flexible Parallel Control, B. Minbashian and K. Warwick, University of Reading, U.K.

The authors introduced a flexible parallel control scheme aimed directly at restructuring feedback control that is immediately applicable within a parallel processing environment. Thus, it uses the advantages offered by parallel architecture. The overall computation time for a single period controller evaluation is, in the worst case, only equivalent to the most time-consuming feedback loop. The parallel control scheme is also useful at making much more of the available computing power while adding flexibility to the controller, allowing for a hierarchical control arrangement. Using a computer within a computer control scheme, simply for digital control, is really overkill. Within a parallel control implementation, it is possible to switch to a simple feedback loop if desired, or to achieve robust fault-tolerant control. These facilities are basic offshoots from the scheme, and arrive as a natural way of creating control. They can be regarded as special cases of the general method.

An advantage one can obtain from the implementation of a parallel controller includes the phasing in and out of different controllers, rather than direct switching of them. This reduces the possibility of transfer bumps occurring, thereby reducing transient spikes in the control and output signals. Also, the method allows for the cost-effective, easy introduction of modern control algorithms, such as self-tuning control, by gradual familiarization so that confidence can be built up without risky shut-down operation. A third point is that the actual cost of implementing a parallel controller scheme is fairly low, as control algorithms can be stored in software, providing a portable and readily modifiable controller base.

Formal Modeling Methods

The two papers in this section covered knowledge-based architecture for project planning and control, and formal methods for designing distributed KBS.

A Knowledge-Based Architecture for Project Planning and Control, J.T. Boardman and G. Marshall, Brighton Polytechnic, U.K. and British Airways Information Management, U.K.

To enable an AI system to perform comparably to a human, a model of the domain concepts must be developed, describing their interaction and form, and explicitly laying out the knowledge that is available. In general, the AI system is in no position to amass this experimental model unaided. An AI model might be considered as a representation of the salient information that must be assessed to appreciate problem structure and to postulate solutions, and experiential knowledge describing how analysis should proceed. In order to establish the requirements of an AI model, the concepts involved in project administration, and the process of project planning must first be introduced. In doing so, the key features that an AI system must have are drawn out and the foundations for the development of the AI model are laid.

The project was coordination and administration of several actions to achieve a certain product goal. The paper described a project planning support system that reflects appreciation of the industrial project planning domain. The heart of the project model presented was a representation of the structure that a product may assume. Just as there are various ways in which one may approach the planning of a project, there are alternative means of representing and processing the plans developed. A project representation form used in a realistic situation will benefit from its ability to exist in several abstract forms. The ideal project support system should provide a mechanism for abstraction to suit the needs of higher management as well as functional or line management.

The mainstay of the project model consists of an object hierarchy describing the component relationships of the product elements. Objects built into a component structure, in an associative network that has the components as nodes, connected by a relational link. Each object included in the model will contain a description of all the real world properties that will be products, effecting a library of component structures from which the product configuration may draw.

Boardman and Marshal implemented the generic project model in traditional expert systems schemes. They implemented the methodology using an expert systems environment known as the Rule-Based Frame System. This imposes a distinction between domain information and domain knowledge. Information is the structure describing the possible forms of problems and their solution. Knowledge is the means for assessing a problem and establishing its solution within the scope set by the information. Domain information is represented using a frame-based organization technique, effecting an associative network model of a domain. Associative network models may be generated graphically, while in parallel the internal frame model is maintained.

The salient conclusions from the work are: (1) Symbolic representations are powerful, and associative network representations provide a conceptually sound and receptive medium for modeling the interactions between the essential project concepts. (2) Production rules provide an easily derived mechanism for the representation of experiential knowledge. (3) Inheritance of properties reflects common class assumptions, and provides an efficient representation mechanism.

Some Applications of Formal Methods to the Design of Distributed Knowledge-Based Systems, M.C. Taylor, University of Keele, U.K., and University of Houston, Texas

The first generation of knowledge systems, based on expert system shells, required all facts to be resident in main memory, thus limiting their scope to small quantities of knowledge. Using a database to store the facts can overcome this limitation. The resulting knowledge base may be loosely coupled or tightly coupled. The latter requires putting both the rules and the data in a composite data/rule base. This allows the creation of an active system which can monitor particular data items and perform desired actions as soon as certain conditions arise.

The next step is to use the combined knowledge of several knowledge bases in answering user enquiries. This can be done by building a distributed knowledge base, which is analogous to allowing multiple experts to be consulted on the same problem. This paper focused on particular aspects for which formal methods provide valuable assistance in finding better solutions.

The approach to distributed query processing suggests that it would be valuable to define formally the semantics of the common model. The authors first defined constructs for a common model using the Vienna Development Method (VDM). They then investigated the validity of various transformations that might be applied to query expressions during query optimization. In establishing the redundancy or inconsistency of a set of rules, they used VDM specifically for defining functions. This approach reveals the precise conditions under which two expressions are equivalent. Developing language translators requires defining the local models as well as the common model. Although the authors did not develop such translators, they believe that using formal methods will make that development feasible.

The paper described a technique for formally defining the semantics of a knowledge base language. The technique has application in query optimizing and in maintaining knowledge consistency. The same approach can help in developing translators between different knowledge models. Also, the development of formal models of knowledge will help in integrating responses from different nodes, enabling the system to return a single coherent response to the user, based on knowledge contributed from several different sources.

System, Quality, Metrics

The three papers in this section discussed software maintenance, a theoretical framework for systems engineering, and quality tracking.

The Use of Cluster Techniques and System Design Methods in Software Engineering, D. Ince and M. Shepperd, Open University, U.K., and Wolverhampton Polytechnic, U.K.

Software metrics attempts to quantify features of a software product that can be used for activities such as production and standards setting. This paper described a novel use of such metrics. The metrics monitor the degree of structural degradation that occurs during the software maintenance process. The technique relies on a branch of statistics known as cluster analysis; it also employs algorithms from the field of classification.

The work is concerned with change. The majority of metric work at present is concerned with static measures--applying metrics to a design and providing information about factors such as maintainability at the time the measurement occurred. This paper also considered the dynamic behavior of a system in terms of change, and in terms of the effect of change on measures that measure the dissimilarity of system structures. The work stresses a dynamic view of a system's structure and it uses cluster analysis at a design level.

System Engineering: A Theoretical Framework, D. Tann, British Telecom, U.K.

Tann contends that systems engineering is in difficulty because current practice is built on inappropriate models of the engineering process that focus on the development of technology, rather than on the true needs of customers. The systems engineering process is more accurately modeled as a control system. Conversely, the control problem is primarily one of managing the evolution of a user's enterprise, by the developing technology to better meet the enterprise's goals. Systems engineers must focus on the needs of the enterprise that the technology is to serve rather than on the technology itself.

The models Tann introduces provide insight into the problems associated with adopting such a control/evolutionary approach to systems development, and establishes the need. For example, the management of evolutionary development will be very different from the management of traditional waterfall developments, partially because of the unfamiliarity of the approach and the lack of well-defined, measurable milestones. This situation will improve as soon as suitable metrics are developed. The models provide a framework within which to address the need for metrication. For a given enterprise, this could mean profitability; for another it could mean raw performance. By taking a broader, more holistic approach to systems engineering, one must address some issues that may have been side-stepped before. The models provide a framework for a dialogue between systems engineer and customer by illustrating the objectives of the dialogue--to establish a common understanding of the problems and define appropriate goals for the enterprise to solve those problems.

Automated Quality Tracking, M.J. Coleman and J. Allan, IBM United Kingdom Laboratories Ltd., U.K.

The automated tracking package described in this paper includes the following general elements:

- Controlling routine that inputs data from a driver plan file and a problem-tracking report summary file, and computes plotting information, which it passes to a graphical display routine for plotting
- Subsidiary routines to extract a raw problem-tracking report summary from the database for the product being tracked

- Subsidiary routines to define this summary into a list of functional verification problem-tracking reports
- A flat file-driver plan
- Files that provide the chart format, and data definition files which the graphical display manager uses to plot the chart.

At a more particular level, the package was designed to run under the Virtual Memory (VM) operating system on an IBM host machine with a PC link. These routines were written as VM Executive Routines. The problem-tracking report database was available to developers and testers to record and update problem-tracking report information. In addition, the routines were available to other parts of the organization for data retrieval. When computed, the plots were passed to the IBM Graphical Data Display Manager through its interactive chart utility interface. The plots could then either be output to the plotter or displayed on the PC screen.

Parallel and Distributed Architecture

In this section, the three speakers covered a standard for parallel program translation, an extensible UNIX multicomputer, and a programing model for parallel machines.

The VSA: A Standard for Parallel Program Translation, C.R. Jesshope, Southampton University, U.K.

In this paper, Jesshope made a case for portable programing environments based on concurrent data and sequences of synchronized method applications. The Virtual Systems Architecture (VSA) is an example of such a system. This implementation must resolve the difference between the abstract code generated (based on the users perceived sets of concurrent data) and the target processor set. It automatically maps the users data objects onto the available processors in the system. The VSA is a multi-user environment, where resources are served to users. The resource set managed by a VSA implementation may well be dynamic.

Within Alvey project ARCH/001, implementations of the VSA have been developed for languages as diverse as Fortran plus and Lisp. The languages will be translated through the VSA into machine code for the AMY Distributed Array Processor and arrays of transputers. Both source and target provide extremes in parallel computing languages and architectures. In languages, one is symbolic and the other numeric; on architecture, one is SIMD and the other is MIMD. Prototype implementations of these translation routines exist at sites within this project.

TOPSY--An Extensible UNIX Multicomputer, P. Winterbottom and P. Osmon, City University, U.K.

TOPSY is a new message-passing multicomputer architecture, comprising a distributed UNIX-compatible kernel resting on a message software system which, in turn, relies on a new, high-speed, telecommunications hardware. The paper summarized the architecture's goals and the main design decisions. Also, the paper described the structure and major components of the operating system and the communications hardware, an evaluation, and discussion of future developments.

Based on TOPSY, it is feasible to build multicomputers that can be enhanced incrementally over a large range of sizes. The key features of the architecture are:

- Fully distributed network-based parallel kernel
- Distributed UNIX file system maintaining full UNIX semantics, and integrating transparently with existing UNIX tools
- Load balancing to distribute work dynamically among processors
- Cheap and fast communications hardware
- Easily extensible - from 1 to 256 processors at present, but essentially without limit.

The operating system--Meshix--appears like UNIX to users, but is completely new. Jobs are allocated by the operating system to the processors which are doing least work, thereby balancing the load among available processors. The operating system is truly distributed, with the majority of work involved in system calls being performed locally, or by distributed servers. Processes do not have to wait in queues for kernel service or access to resources.

The only means of communication within the computer between processors is the network Meshnet. This allows processors to be added incrementally. The custom communication chips allow any processor in the machine to talk to any other processor via intervening processor nodes. However, these intervening nodes need not interrupt their normal operation when communication is in progress.

The architecture is processor independent. There are two implementations to date. a four-processor prototype using Philips 68070 processors and first generation communications integrated circuits (ICs), and a 16-processor demonstrator using Motorola 68030 processors with virtual memory and faster second-generation communications ICs.

The New Programing Model for Parallel Machines, J.J. Florintin, Strand Software Technologies and Birkbeck College, University of London, U.K.

The new model discussed in this paper takes the well-known model of intercommunicating processes and refines it by:

- Identifying processes with the evaluation of a function, eliminating many of the difficulties resulting from identifying processes with procedural programs.
- Invoking process execution in two phases. (1) Pattern matching between a procedure call and a procedure definition is done. At this point, the procedure call can be placed on a suspend list. (2) Actual process execution begins. By careful choice of the matching conditions, both logically reasonable

synchronizing and performing efficiency can be achieved.

- Making it easy to express interprocess communication by ordered streams of data. There can be a long time delay between successive elements appearing on a stream.

Very Large-Scale Integration (VLSI)

The two speakers in this section discussed wafer scale technology and high-speed bipolar technology.

WASP: A Demonstrated Wafer Scale Technology, I P Jalowiecki, Brunel University, U.K., and S.J. Hedge and R.M. Lea, Aspex Microsystems Ltd., U.K.

This paper included a review of Wafer Scale Integration (WSI) Associative String Processor (WASP) 1 and WASP 2a and WASP 2b Massively Parallel Processor technology demonstrators, implemented in standard Complementary Metal Oxide Semiconductor (CMOS) technology. The latter devices are defect-tolerant arrays of 864 and 6480 processing elements and integrate 1.26M transistors (4cmx4cm) and 8.43M transistors (10cmx10cm). The two variants--WASP 2a and 2b--are examples of the ASP architecture, developed by Brunel University.

The WASP device is a highly versatile parallel processing computational architecture. The WASP hardware and software modules provide the potential to achieve step-functions in cost/performance over a wide range of computer vision tasks.

Results to date have demonstrated:

- Intermodule defect tolerance
- Selective power isolation
- Intermodule control and signal distribution
- Intermodule defect tolerance
- Basic configured processor functionality.

High-Speed Bipolar Technology, Peter Ashburn, University of Southampton, U.K.

This paper reviewed recent developments in silicon bipolar and GaAlAs/GaAs heterojunction bipolar technologies, and compared the performance of the two technologies for high-speed Emitter Coupled Logic (ECL) circuits. The gate delays of state of the art are estimated using an analytical equation, which expresses the gate delay as a linear weighted sum of all the time constants in the circuit. A common idealized transistor layout is used to eliminate geometric differences from the comparison, and highlight the role of the heterojunction and the fundamental material and device properties. Gate delays of 8.6 and 26.9 ps are predicted for 1 micron GaAlAs/GaAs and silicon bipolar technologies respectively. Scaling the emitter width to 0.4 micron reduces the gate delay for silicon technology to 17.2 ps, but gives only a small improvement to 8.1 ps for GaAlAs/GaAs technology. These results are explained by considering the limiting time constants for the two technologies.

Experimental Systems

Three papers in this section deal with multicast facilities in computer supported cooperative work, transputer based networks for parallel distributed sensor systems, and split-channel reservation for packet voice and data.

Initial Experience in Implementing Multicast Facilities in Computer Supported Cooperative Work, L.H. Ngoh and T.P. Hopkins, Manchester University, U.K.

This paper evaluated the performance of multicast communication facilities designed to support Computer-Supported Cooperative Work (CSCW). The CSCW systems allow a group of individuals to work together on a common pool of data and to make common decisions. These systems often involve dynamic group communications among the users in a combination of store-and-forward and real-time modes. A typical communication model for a CSCW supports the interchange of information from one CSCW user to the others. Multicast facilities are required to support such a one-to-many communication model.

Earlier work led the authors to design a set of multicast communication facilities that meet these requirements. A particular implementation of their design has been in operation for some time. The new communications facilities are implemented as both user processes and kernel level resident code in a UNIX environment.

The initial experimental results of the multicast communication facilities for CSCW systems show that the prototype implementation performs rather poorly compared to existing standard unicast communications. To improve the performance, the various components of the system that give rise to large overheads, were identified. The authors proposed two refinements to replace these components. They also proposed re-implementation. Initial results predicted that these two mechanisms could improve performance.

Transputer-Based Networks for Parallel Distributed Sensor Systems, M.J. Dudziak and J.L. Wilson, Inmos/SGS-Thomson Microelectronics, U.K.

With a transputer-based system, the hardware and software environments can take immediate advantage of the traditional virtues of parallel processing at the individual node level. This includes the ability to expand the number of processors at a node without affecting the performance of other nodes or requiring major changes to the programs operating at the affected node. The net result of such a design (employing transputers and Inmos links) affords the designer and operator of a wide area network the capabilities of runtime reconfiguration and fault tolerance, combined with relative simplicity in the architecture and in the operational requirements.

Using Reservation Virtual Time Carrier Sense Multiple Access (CSMA) with Split Channel Reservation for Packet Voice and Data, A.D. Malyan and R.L. Brewster, Aston University, U.K.

The Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol, originally developed for data transmission, is not very effective for the transmission of voice packets because of excessive packet delays at moderate to high loads. Users are very sensitive to voice packet delay, particularly variable delay. Using a variation of CSMA called Reservation Virtual Time CSMA (R-VT-CSMA) can reduce the delays problem. This protocol takes advantage of the periodicity of voice packets during a talk spurt to reduce the collision probability. At the same time, the protocol clocks out new packets onto the channel based on their packet arrival time to ensure fairness in delay.

In this paper, the authors proposed an extension of the R-VT-CSMA using split-channel reservation to improve channel throughput by splitting the available bandwidth into two channels--one for data transfer and the other for reservation. The advantage of this architecture is that contention resolution is performed in parallel with data transfer which results in a priori contention resolution giving an improvement in throughput.

Formal Aspects

The three papers in this section cover implementation of formal specifications, automatic program synthesis, and interactive program construction.

Functional Languages for the Implementation of Formal Specifications, P.M. Sanders and M.P. Johnson, British Telecommunications Research Laboratories, U.K.

This paper covered a formal implementation method based on the use of functional languages and transformation, rather than refinement followed by proof. Sanders and Johnson successfully applied the technique to a realistic problem, including justification of the transformation laws used. Use of transformation demonstrated major improvements in both space and time performance. They made these improvements, with the final program still meeting its specification.

If an implementation must be achieved using an imperative programing language, then the functional program can be transliterated into an imperative language. Researchers have not yet attempted this, but it is one aspect of future work. An initial stage of refinement may be needed where specifications use an explicit style as the transliteration can only be applied to an explicit specification. Experience suggests that many specifiers naturally use an explicit style, particularly when using formal techniques.

The approach can be made viable using sophisticated support tools. Collaborative R&D work is in progress with York University. Much of the transformation and proof work used in this paper was completed using the Starship interactive transformation environment from York. Current research in this area is concentrating on developing a meta-language which will allow programers to create meta-programs that will perform transformation and proof of programs. This method offers a formal approach to deriving implementations from a specification. The program will satisfy the specification by using correctness preserving transformations. Experiments show that this approach can produce implementation standard programs from the specification.

Turning Eureka Steps into Calculations in Automatic Program Synthesis, A. Bundy, A. Smaill, and J. Hesketh, University of Edinburgh, U.K.

The paper gives a technique, based on theorem proving, for synthesizing programs from logical specifications. Programs can be synthesized from proofs that their logical specifications are satisfiable. A technique called "middle-out-reasoning" for the control of search in automatic theorem proving is used. Each proof step is also a program construction step. A naive use of this technique requires a human or computer to produce proof steps which provide the essential structure of the desired program. Justification for these steps at the time that they are made is difficult; the reason for them emerges only later in the proof. Such proof steps are often called "eureka" steps. These eureka steps to be produced automatically using middle-out-reasoning as a side effect of non-eureka steps.

Interactive Program Construction, M. Haggith, University of Edinburgh, U.K.

This paper dealt with the automatic construction of Prolog simulation programs in the domain of ecology. The paper identifies a problem with an existing computer system, called EcoLogic (EL), which builds simulation programs and demonstrates an experimental solution to this problem implemented in a new system called NIPPIE. This solution involves supplementing the program construction's search for model information by allowing it to ask the user for missing information, hence generating and controlling a dialogue with the user.

The program generation algorithm is good at generating programs but does not order the sequence of questions in such a way as to provide a natural dialogue with the users. This is because the ordering is that of a depth-first search for information. The experiment showed that control can be imposed on this algorithm, using metalevel information, consistency checks, and heuristics, to make it produce dialogue acceptable to users, while retaining its generative power. These control features effectively hide EL's depth-first search strategy behind dialogue strategies sensitive to some of the relationships between questions, like relevance, dependence, and level of detail. NIPPIE can take advantage of these without losing the guarantee of complete, well-structured program design provided by the underlying program construction algorithm In addition, using simple transformations, called idealizations, allows NIPPIE to enhance its use of ecological terminology.

System Design

The two papers in this section covered a tool set for designing Local Area Network (LAN)-based real-time systems and an environment for the design, analysis, and programing of digital systems.

Towards a Design Toolset for LAN-Based Real-Time Systems, **M. Merabti and D. Hutchison, Lancaster University, U.K.**

This paper presented a design environment that integrates modeling through analytical and discrete event simulation, and a KBS that helps in designing, implementing, and maintaining distributed systems based on LANs.

The LANs are increasingly used as a basis for distributing functionality in real-time industrial systems such as discrete parts manufacturing or in process control. These networks involve many options as well as several different communication protocols. To build and maintain correctness and efficiency in these systems involves coordination in the different stages of the design life cycle of the system.

Network systems are hard to design and understand, correcting a design mistake is costly. These systems are further complicated by their real-time needs. An issue central to the design of computer networks, including non-real-time systems, is that of performance evaluation for predicting system responses.

Much work has been done in requirements specifications, such as the Total System Design (TSD) framework proposed by G.C. Roman and M. Stucki, in 1984 in Computer, 17(5). This paper suggested six stages for the development of such systems: problem definition, system design, software design, machine design, circuit design, and firmware design. In the TSD framework, the problem definition stage is distinguished in two ways: (1) it is application oriented and (2) it involves no design activities. The sole purpose is to ensure that a clear understanding of the problem has been achieved, and that a statement of the system requirements has been generated before the start of system design.

The key to across-life-cycle integration of design activities rests with the ability to relate design and requirements specifications. This means that at design time, one needs to show that the system's components or whole system requirements are satisfied, especially for real-time systems. Also, a model of the design can be helpful in upgrading exercises as well as debugging in case of faults.

The design process discussed in the paper provides a KBS that encapsulates design heuristics and application domain characteristics to help the inexperienced as well as the experienced in understanding design implications. This is particularly important with designers who are not performance analysts and might not appreciate performance implications for real-time systems.

MATTRESS: An Integrated Environment for the Design, Analysis, and Programing of Digital Systems, C.C. Charlton et al., University of Liverpool, U.K.

The MATTRESS program is an integrated environment to support the hardware and software design of digital systems. The program is particularly, though not exclusively, oriented toward designing programable and microprogramable computer architectures. At the same time, its powerful features assist in developing and debugging low-level software or microcode. In digital system design, it may be used to study system behavior through simulation, from block diagram level down to gate level, and as a hardware/software development environment. Thus far, MATTRESS has been mainly a teaching aid in courses on computer architecture, low-level programing and microprograming, for which it offers features better than those provided by the native hardware.

The MATTRESS system is the kernel of a program of R&D aimed at creating software tools to support designing and programing digital systems. Developments in progress of planned systems include further exploiting the simulation method used, and enhancing the user interface to make greater use of graphics capabilities. Versions of MATTRESS have been configured to contain simulations of the 6502 microprocessor, the ORION microprogramable computer from high-level hardware, and the REKURSIV object-oriented architecture from Linn.

Tools

The single paper devoted exclusively to tools dealt with a shared-memory execution model for parallel functional programing.

A Shared-Memory Execution Model for Parallel Functional Programing, C.C. Lee and H.A. Fatmi, Lawrence Livermore Laboratory, U.S., and Kings College London, U.K.

In functional languages, a program consists of a set of function definitions that describe the computations without any side effect and only data dependencies constrain the order of execution. This makes the details of the underlying architectures transparent to the user. Also, the compiler can detect and exploit the parallelism of the underlying architectures and in the programs.

Traditionally, functional languages were developed simultaneously with the design of new architectures such as dataflow and reduction machines. This paper presented a general execution model to support a parallel functional language called Stream and Iteration in a Single Assignment Language (SISAL) on shared memory multiprocessors. Lee and Fatmi examined the implementation on two very different architectures: (1) 4-way Cray X-MP and (2) 32-way VAX Research Multiprocessor, M31. They derived the SISAL program primarily from the dataflow language VAL. However, unlike VAL, SISAL is developed for conventional sequential machines and varieties of parallel machines to study functional languages.

The general issues of mapping functional language programs onto an arbitrary multiprocessor was addressed by P. Hudak and L. Smith in "Parafunctional Programing, A Paradigm for Programing Multiprocessor Systems," Conference on Programing Languages, ACM, New York, 1986. Hudak and Smith described the runtime support as an extension of the previous runtime system developed for the sequent balance multiprocessor. They redesigned the runtime system to be ported on general shared-memory multiprocessor systems. The design started with the implementation on clustered VAX 11/784. In this paper, the authors described the internal workings of the general shared-memory execution model with implementat:on details on the Multi-VAX M31 and the Cray X-MP.

Instrumentation and Intelligence

In the two papers of this section, acousto-optic correlation techniques and the role of the engineer in a diagnostic task are discussed.

Acousto-Optic Correlation Techniques, C.D. Reeve, Royal Naval Engineering College, U.K.

The ability of coherent optical systems to perform signal processing operations (such as Fourier transforms, correlation, convolution, and matched filtering) have great potential for real-time operation. They involve inherently parallel processing systems in which the speed of processing is limited only by the speed of light. However, recently, with the availability of high-quality acousto-optic cells for use as input devices, true real-time operation has become possible.

This paper described the theory of acousto-optic devices and correlators. Reeve discussed two different operations of twin Bragg cell space integrating correlator/convolvers. In the first application, he employed the heterodyning technique in which he mixed light at different optical frequencies to give the convolution of the signals applied to the two acousto-optic cells. He configured the system to implement the first two stages of the multiply-convolve-multiply algorithm so the power spectrum of the input signal is obtained. The second system is a non-heterodyning correlator/convolver. Reeve showed that one can extract the full correlation function, including amplitude and phase information by measuring only the intensity of light at the output of the system. This is done by amplitude modulation of the signals onto the Bragg carrier frequency, using double sideband large carrier modulation.

The Cognitive Role of an Engineer in a Diagnostic Task, R.C. Thomas and J.G. Littler, University of Leeds, U.K., and G.B. Thomas, Central Electricity Generating Board, U.K.

Central Electricity Generating Board engineers accomplish work with the provision of intelligent front ends to a suite of turbine vibration programs. The authors provided a brief account of the domain and the numerical simulations that model it. Then, they gave an overview of the expert system interface that was written to do a matching task. They discussed some of its limitations when used by non-experts. In addition, they discussed the cognitive role of the engineers under the headings of Model Transformations, Associative Memory, and Perception. They then described the proposed simulation support environment.

The functions of the support environment should include:

- Matching as performed by the current interface
- Logging of all simulation runs in a manner suitable for later retrieval in a most flexible way
- Maintaining a history of all run-down curves from the actual machine
- Annotating actual and simulation data with pertinent facts such as misalignment
- Picking candidate peaks from run-down curves and then to let the machine match them as best it can
- Building a mode or run-down curve from its constituent parts and displaying the results for later analysis
- Running the rotordynamics programs in any manner the user thinks fit
- Extending the interface as new semistructured tasks are identified
- Pointing out previous instances of actual run-down data that are similar to the current one, and suggesting possible ways in which one can use them in solving the current problem
- Using high-performance graphical interface.

Knowledge-Based Systems

The three papers in this section dealt with knowledge representation in financial expert systems, model-based diagnosis, and management system research.

Knowledge Representations and Interfaces in Financial Expert Systems, S.C. Lambert and G.A. Ringland, Rutherford Appleton Laboratory, U.K.

This paper dealt with related themes -explanation and knowledge reuse. The authors treat them in the context of interfaces and knowledge representation. The themes are related because they both require explicitness of knowledge. Implicit knowledge results in poor explanations and restricts the potential for reuse.

The work described in this paper was conducted in the Paraflex project. The Paraflex project ran in parallel with an Alvey program community club project called Alfex. The work related in this paper was designed to explore the issues in a more research vein than in the Alfex project. The application which was used to define and exemplify the ideas is in the field of corporate finance. The Source of Finance Adviser determines the most suitable source of finance for a company with certain needs, based on the nature of the company, its state of health, and the need for finance. Several contributing factors influence each of the six finance options considered. These factors are derived from financial ratios, information about the need for finance, etc. The Source of Finance Adviser, while intended primarily as a demonstrator, does contain sufficient knowledge to be useful and realistic. The problem is a classification one; the knowledge is sufficiently complex and the reasoning sufficiently deep that to explain requires an effective interface. There is also considerable scope for knowledge reuse in the financial domain.

The paper describes the approach to graphical explanation and knowledge reuse. Lambert and Ringland describe the approach to graphical explanation on the project. By way of considering the generality of the approach, it leads to a consideration of aspects of knowledge reuse and a framework in which it may be realized.

REPAIR: A Model-Based Diagnosis System, M.H. Lee et al., University College of Wales, U.K.

The authors described the REPAIR project emphasizing research into model-based diagnosis. The REPAIR project was a 3-year project entitled "Qualitative Reasoning Systems for Physical Mechanism Diagnosis." The goal was to investigate diagnostic reasoning processes in the context of physical mechanisms drawn from industry and engineering. This goal was pursued by building experimental software simulators that facilitate the modular construction of device models and allow the simulation of realistic device behavior.

Experience suggested that heuristic-based diagnosis techniques have serious inherent limitations. Although there has been some success in building heuristic-based diagnostic systems, usually these systems are specially configured and tailored for given applications. In these cases, the success is achieved because the heuristic rules are highly tuned to the domain and any reasoning involves very little first principle analysis of the fault symptoms. In contrast, we must model directly the structural and functional aspects of mechanisms, to investigate how automatic reasoning tools may ascertain the causes of faults and failures when malfunctions occur.

The paper described the ideas of model-based diagnosis and outlined the requirements for suitable models. In particular, the features of qualitative reasoning are significant. The authors briefly described the REPAIR project model-based system. They considered the advantages and limitations of the techniques employed and examined using model-based reasoning systems as the basis of a more powerful general diagnostic architecture.

Aspects of Knowledge-Based Management System Research at the European Computer Industry Research Centre, Jorge Bocca and Michael Freeston, Federal Republic of Germany

The European Computer Industry Research Centre (ECRC) is a joint research organization founded by Bull, ICL, and Siemens to help in enhancing the future competitive ability of the European IT industry. The center's work focuses on advanced information processing technology for the next generation of computers. There are about 60 researchers at the center drawn from 21 countries worldwide, but predominantly from Germany, France, and Britain. The activities of the center are divided into computer architecture, logic programing, human/computer interaction, and knowledge bases.

Knowledge-Based Management Systems (KBMS) research is the main activity within the ECRC Knowledge Bases Group, which includes three British researchers among its 16 members. The present object of the Knowledge Bases Group is to develop a fully comprehensive KBMS. Bocca and Freeston's paper concentrated on aspects of this work in which they are actively involved, and was essentially an overview of theoretical studies and software development.

The report began with a state-of-the-art assessment of those areas of research from which the techniques for building a KBMS must be drawn. They then discussed the nature of the problems that must be solved before they can achieve the ultimate objectives. Finally, they provided an outline of the solutions that they devised in the context of a system currently under development at ECRC. The report pointed out the main obstacles to progress, showed how some of these have been overcome, and how it is expected to solve the remaining ones.

Speech Applications

Three papers in this section covered voice input, evaluation of a speech-driven interface, and speech synthesis in a computer-aided learning environment.

A Direct Voice Input Man-Machine Interface Strategy to Provide Voice Access for Severely Handicapped Speakers, A.G. Warner, R.D. Hughes, and R.A. King, Royal Military College of Science, U.K.

This paper described the initial stages in the development of a simple Time-Encoded Digital Speech (TES) Isolated Word Recognition (IWR) system incorporating Automatic Token Set Selection (ATSS) procedures to provide an effective man/machine interface for severely handicapped speakers. The authors emphasized three aspects of the investigation. (1) Automatic diagnostic routines to select and assess acoustical utterances produced by severely impaired speakers. (2) Characteristics of so-called sentence generator algorithms, developed to capitalize on the advantageous features of the TES coding format and ATSS. (3) First results of exercising a semi-portable testbed embodiment against a small number of severely impaired speakers.

The status of the current embodiment is limited to that of a vehicle to test concepts and to enable a first exposure of both concepts and hardware to therapists and patients. Many important issues must be resolved by interaction between patient, therapist, and researcher.

Evaluation of a Speech-Driven Interface, M.P. Zajicek, Oxford Polytechnic, U.K.

Zajicek described experiments performed on a prototype speech-driven word processor under development at the Intelligent Speech-Driven Interface Project based at the Hatfield Polytechnic. The objectives of the experiments were to:

- Determine whether there is an increased level of user satisfaction associated with a speech-driven interface
- Determine whether there is an enhanced learnability factor associated with a speech-driven interface
- Determine whether the dialogue provided by the speech-driven interface can instill user confidence in error retrieval
- Test the hypothesis that attitudes vary with different user groups
- Determine whether users are inhibited when using a speech-driven interface; if so, whether the degree of inhibition differs with the physical location of the interface
- Determine which evaluation criteria users consider important to evaluate a speech-driven interface.

A methodology was developed for user-dependent evaluation of speech-driven interfaces based on the results of experiments. Results of the experiments showed that attitudes to a speech-driven interface are user dependent. Also, choice of suitable evaluation criteria varies with the participant. Therefore, an evaluation profile should be developed using criteria selected by the proposed user.

Selection must be adjusted to the user's experience. For the less experienced user, the evaluation criteria can be presented as a simple list of system requirements from which one must select the subset that is considered most important.

Speech Synthesis in a Computer-Aided Learning Environment, E. Lewis, University of Bristol, U.K.

Computer-Aided Learning Environment (CAL) started in the U.K. in the sixties. In 1967, a working party on computers for education chaired by Professor Black was commissioned by the National Council for Educational Technology (NCET) to:

- Investigate the potential role of the computer as a component of educational and training systems in the U.K., considering experience and trends in other countries
- Outline a systematic program of applied R&D that would be desirable to encourage in this country, aimed at exploiting the computer to the best advantage in education and training.

Following the working party's report in 1969, the government funded the National Development Programme in Computer-Assisted Learning for a 5-year program beginning in 1972; its goals were to:

- Develop and secure the assimilation of computer-assisted and -managed learning on a regular institutional basis
- Recommend to appropriate agencies in the public and private sectors about possible future levels and types of investment in computer-assisted and -managed learning in education and training.

Other government initiatives have followed, including the present program for examining the use of text-to-speech in CAL. Last year a program was produced to introduce students to the basic graph facilities on a SUN workstation. The fact that one could produce graphics on the screen and talk simultaneously was a bonus. Students were shown how to call graphics functions and then test them by getting them to produce their own graphic programs, from within the CAL program, through editing a template provided by the system.

Tests on the benefits of using speech were conducted with only a few people and over only a short time for the results to be positive proof of the value of speech, but they were definitely encouraging.

This year, the author and his colleagues are attempting to produce some speech-enhanced software that will be oriented more to the general public than DECtalk is. Speech is being added to a standard spreadsheet tutorial, and CAL packages are available to teach the highway code and basic first aid.

Object-Oriented Systems

The three speeches in this section covered software reusability, the TEN15 Project, and object-oriented Ada.

Software Reusability Using Object Oriented Programing, D.B. Anderson and S. Gossain, University of Essex, U.K.

The authors discussed a model for the software life cycle in an environment based on reuse. The paper dealt with three interlocking life cycles, at the levels of components, hierarchies, and applications:

- Component life cycle - interface is given. Changes involve work on the implementation of the component (perhaps to make it faster) and an alternative version might be provided with different properties such as the amount of checking or the total capacity. At this level, measurements may be made on components.

- Application life cycle - refers to the sequence of software products, which may be versions of a single application or a series of applications. Determination can be made as to which components are useful and which need modification. Amount of reuse in the application can be measured.
- Hierarchy life cycle - collection of components in the model is a rich structure that is central to the software process. New classes are added and the hierarchy is reorganized. This process needs to be tracked to gain experience in evaluating progress.

The TEN15 Project, Dr. N.E. Peeling, Royal Signals and Radar Establishment, U.K.

Peeling described the background and progress of the TEN15 project at the Royal Signals and Radar Establishment (RSRE). TEN15 provides an algebraic basis for software development, it has three important aspects:

1. Provides an interface that makes different computers compatible, thus ensuring the portability of all programs that are written on top of TEN15. Using techniques developed for high-level programing languages, such as compilation and strong type checking, means that this compatibility is achieved without compromising efficiency or integrity.

2. Provides a powerful set of facilities for implementing and integrating system components and databases within a distributed environment. Systems whose complexity or functionality makes them difficult to implement on conventional operating systems should be significantly easier to build on TEN15. Achieves this by providing facilities for dynamic and secure resource allocation in mainstore, filestore, and over a network, within a comprehensive type system that has a sound mathematical basis. Will support: high-integrity systems, secure systems, IPSE developments, expert systems, heterogeneous networks, fine-grain databases, and formal methods.

3 Allows users to preserve their existing investment in software. Runs efficiently on conventional computers and can coexist with existing operating systems and can share data with them. Efficiently supports most existing programing languages and provides flexible mechanisms for mixed language working.

The paper gave a background to the development of TEN15, a summary of what it offers to users, an analysis of the potential application areas, and concluded with its current state of development. There is available a prototype translator and kernel for VAX/VMS. PRAXIS plc is producing a kernel for UNIX, and researchers at the University of York are producing a translator for Motorola 6800. The RSRE is developing an evaluation system for TEN15.

Object-Oriented Ada for Knowledge-Based System and Neural Network Application, M.J. Corbin and G.F. Butler, Royal Aerospace Establishment, U.K.

The Ada language contains some support for object-oriented programing but has some notable deficiencies. The Ada object-oriented toolkit (toolkit) described in this paper was developed to provide facilities to create and manipulate objects, and to provide support for more general relationships between objects. Such features are especially useful when developing system elements using KBS or neural network techniques.

The paper described a new approach to the problem of developing and maintaining object-oriented software. The toolkit retains all the software engineering facilities for use with the object-oriented paradigm. These include support for temporary association between groups of similar objects and the definition of operations which act on all the members of a subset. Although intended primarily for use in KBS and simulation, the toolkit is not specialized and can be used to support any application for which Ada is appropriate. Initial applications have included an integration package for continuous and discrete simulations, an inference engine for an expert system, and a simulation of a neural network.

Joining Knowledge-Based Systems with Other Technologies

The three papers in this section dealt with KBS with embedded simulation components, a support environment for dynamic systems modeling, and an experimental neural network-based KBS.

Knowledge-Based Systems with Embedded Simulation Components, W. Walker et al., Sunderland Polytechnic, U.K.

In this paper, the authors provided a methodology KBS which has embedded Discrete Event Simulation capabilities. The paper included a brief review of some of the problems encountered when implementing nontrivial KBS and simulation modeling tools. The authors proposed the integration of KBS and simulation technologies as a solution to some of these problems. In addition, they presented a method for achieving this integration, and the paper concluded with two case studies that investigated the potential for exploiting KBS and simulation technologies in the manufacturing environment.

A Knowledge-Based Support Environment for Dynamic System Modeling, D.A. Linkens et al., University of Sheffield, U.K.

This paper described a prototype environment for dynamic systems simulation that was constructed using a mixture of commercial packages and inhouse software generation. The aim was to incorporate principles of AI in an attempt to introduce elements of intelligence into the skilled area of system simulation. This required careful consideration of knowledge representation styles for dynamic modeling, and of the important concepts of global and integrated databases. The need for multiple cooperating expert systems has become apparent. Therefore, the authors are pursuing some extensions to the system developed. These include a separate expert system module for model validation, alternative simulation languages drivers, and an HCI advisor for construction of graphics layout based on good ergonomic principles. To widen the usefulness of the system called Knowledge-Based Environment for Modeling and Simulation into areas such as the life sciences, a front-end modeler based on bond-graph methodology is also being investigated.

The hybrid tool kit runs on PC286 or PC386 systems under MS DOS, and uses a multiplicity of languages for its implementation. Most of the inhouse software was written in Modula-2.

An Experimental Neural Network KBS Using an Object-Oriented Processor, G. Whittington and C.T. Spracklen, University of Aberdeen, U.K.

This paper covered a variety of mappings between the neural network paradigm and the object-oriented programing model. For each of the proposed mappings, Whittington and Spracklen described the advantages and disadvantages with specific reference to the development time and runtime efficiency of such mappings. They presented experimental results for both a conventional machine architecture, a SUN Microsystems' Series 3 workstation, and an object-based computing system, the REKURSIV. These results indicate that the runtime penalty for using a compiled object oriented programing language is not significant in neural network applications. The paper covered, in more general terms, problems that might be encountered during the incorporation of a neural network based subsystem into a conventional KBS.

Speech Language and Vision

The three papers in this section covered a speech recognition system based on a transputer, recognition of handwritten arabic mathematical formulas, and computational hyphenation.

OSPREY: A Continuous Speech Recognition System Based on Transputer Parallel Processing, A. Sutherland, Y. Ariki, and M.A. Jack, University of Edinburgh, U.K.

The Centre for Speech Technology Research (CSTR) at Edinburgh University has developed a real-time constrained domain speech recognition system for application in ground movement control--OSPREY. The system is currently in prototype form; however, its technology and application structure defines the route that CSTR will follow in future systems development.

The transputer-based system uses advanced parallel processing architecture to perform real-time, continuous speech recognition. Using linguistic modeling further enhances the high recognition rate for particular application domains, the domain of airport ground-control movement is a demonstration.

A recent market trend has been that the most advanced processors now become commercially available on AT-compatible, plugin boards at a very early stage and this has been exploited in the OSPREY recognition system. A very sophisticated state-of-the-art recognizer has been rapidly and reliably assembled at relatively low cost from unmodified commercially available hardware.

Recognition of Handwritten Arabic Mathematical Formulas, Taalat Salem El-Sheikh, University of King Abdul-Aziz, Saudi Arabia

In this paper, the author describes a syntactic/hierarchical system for the recognition of online Arabic mathematical formulas. The segmentation of a mathematical formula is relatively simple since all characters are isolated and each consists of one stroke only, except for three symbols where each consists of two strokes. Thus, the mathematical formula recognition problem reduces, after segmentation, to a character recognition problem. The characters comprising formulas include 16 isolated letters, 10 digits, and 11 mathematical symbols.

The author used several features in the recognition process, where he used different features sequentially in different stages of the hierarchical system. These features for a certain character include its width, the ratio of its width to height, and the relative distance between the first and last points. The features also include the direction of the first part of the character and the distance between the first point and the points of largest x and y coordinates. Moreover, he used the number of minima and maxima in the horizontal and vertical directions and some related distances for recognizing characters.

The author developed a context-free grammar to recognize Arabic mathematical formulas. He used this grammar to obtain precedence relations between the different terminal and non-terminal elements. He also used an algorithm for parsing the formulas.

The author tested the recognition technique using a large set of formulas including about one thousand characters. The recognition rate exceeded 99 percent and the recognition time is very short.

Computational Hyphenation, J.A. Uren, Linotype Ltd., U.K.

This paper gave an overview of the role of hyphenation in a computational environment. The work is a summary of an existing application. Uren presented an overview of the language independent system and suggested possible extensions. The system is a multilingual hyphenation package that was developed at Linotype for use in the graphic art/desktop publishing industry.