SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

D I S T R I B U T E D C O M P U T I N G N O T E 5 0 8

VISITS

issued by
D A Duce

Notes on a visit to R J Cunningham
Imperial College 10th November 1981

16th November 1981

DISTRIBUTION: R W Witty
D A Duce
F Chambers (Logica Ltd)
Miss G P Jones
R J Cunningham file

CUNNINGHAM/KRAMER PROJECT

The visit was organised so that Fred Chambers might learn about Jim Cunningham's project.

Jim hopes he has now recruited a systems programmer. The new RA will be responsible for implementing prototype algebraic specification tools, using known techniques.

Jim gave a brief presentation of the project.

There are two steps to an axiomatic problem specification:

1. Define the environment e.g. axiom for integers.

2. Pose the problem e.g. for given integers a, b find an integer x such that: a + x = b.

Note that the traditional problem specification asks for a solution value rather than a solution process.

The specification of the solution process can be given in terms of an antecedent, consequent and local environment.

e.g. find process X where given any initial values for a, b (desired antecedent)
execution of X
ensures a + x = b (consequent)
where a, b, x are integers (local environment)

For more complex problems:

1. express assertions in predicate calculus

2. use a hierarchy of environments

e.g. process place - components
antecedent time
consequent for all x, placed (x)
where for component (x)
placed (x) means .....

To specify a system of processes entails the specification of

1. environment
2. system components
3. axioms of system

e.g. for a telephone exchange:

environment includes:

set of subscriber numbers

function dialled.

predicates off hook, enough- digits, conversation,.....

System includes:

process receive-digits

attempt - path - set-up

Axioms include:

NOT enough-digits(S) → FOR ALL t, NOT conversation (s,t)

Note that axioms are properties of local quiescence (steady state). This approach says nothing about how the system gets to a steady state.

Consistency requirements:

1. some process initiated if quiescence condition false
2. if a process is initiated its pre-condition is true
3. sequences of processes terminate with quiescence condition time
4. concurrent processes do not interfere - ie their proofs do not interfere.

There is a class of problem not specificable by this steady-state approach, e.g. an i/o module where input and output sequences have to be specified. There is a kludge representing sequences as data structures which gets round this, but more complex modules are probably more appropriate. Work so far indicates that the steady-state approach encompasses a very broad, useful, class of problem.

A set of axioms specify quiescence. If the system is not in a quiescent state, the conditions for initiating processes are obtained by negating these axioms. If the results are put in disjunctive normal form and the conjunction components are identified, it turns out the these correspond to particular situations and can be taken as antecedents for some process. Thus the specification can be decomposed into a set of processes.

An MSc student (Sylvania?) has developed a set of modules written in Prolog for performing these transformations on a set of axioms. The suite was briefly described.

Jim has an informal arrangement with GEC Hirst Laboratories who are very interested in these techniques. He has also given seminars in the STL Management Seminar series (knows Bernie Cohen!) He has been involved in a departmental project looking at an Ada Programming Environment Specification for Ada Europe.

Jim has discussed informally possible formal collaboration with STL or Hirst.

WOLFSON MICROPROCESSOR UNIT

Jim introduced us to Martin Cripps head of the Imperial College Wolfson Microprossor Unit. Martin is also a senior lecturer in the department, their tame hardware expert.

The role of the unit is to provide support and equipment (free) to anyone in the College, using microprocessors for research. The equipment provided is advanced support equipment that an individual department would not be able to afford. The motivation is similar in part to Common Base, the chemists should do chemistry.

The unit currently has HP 64000 series development systems with 20 different cross assemblers and Pascal. They provide full emulation for Z8000, 6801 etc. The unit also has a number of logic state analysers and one high quality oscilloscope.

The unit is staffed by 1½ RAS, one of whom is solely responsible to the Unit, the half post is a joint post with the college microprocessor teaching unit (so each unit knows what the other is doing!).

The Unit is a 5 year project, 1½ of which have elapsed. Wolfson Foundation grants are one off, and having been funded once you cannot go back a second time. The only string attached to the grant is that the money should be used for industrial applications of microprocessors.

The Unit does not support hardware construction - departments must use their own resources for this.

IMPERIAL TERMINAL NETWORK

Martin is also involved in the construction of low cost network systems and has been for the last 4 years. The project is driven by the need to install terminals in offices.

Martin stressed that the project is not a rival to Cambridge Ring, Ethernet etc. It is solely for the connection of terminal (V24 interfaces) at speeds of up to 19200 bps.

Martin and a research student designed a terminal network called Synchronet which finished in 1979. The design was based on a "string" with loop back. All data could be close packed, transmission was asynchronous but arbitration was synchronous. The design fitswell with fibre optic "strings", but relied on the availability of passive fibre optic taps which are not yet available with acceptable losses.

A node was about 40 IC's, too many to meet the target price of less than £100 per node. There was at that time no possibility of making chips.

A subsequent project, with another student, built upwards from current technology and uses an "intelligent" nodes, i.e. a 6802 emulates special purpose hardware.

The new design provides for 192 connections, 128 of which can be active at any time. Each is guaranteed a 2.4 K bps connection. The hardware allows speed to be traded against number of connections.

Cable is twisted pair and connections can be made without breaking the cable. A small driving/receiver is mounted very close to the cable, a ribbon cable connects that to the network communications unit. Data is phase encoded and phase recovered. Packets are 11 bits (start, stop, parity and 8 data).

The network sets up virtual channels, addressing overheads on individual characters would be too high. When one station transmits, the corresponding receiver knows when to receive. The network contains a name service which maps logical to physical address.

Each local connection unit contains a look-up table of 64 standard services (same in each unit) so that the name server does not have to be involved in setting up the most frequently used connections.

The name server can provide protection, validation etc.

There are no limitations on the distance between connections to the cable. The system was designed assuming a maximum cable length of 1 km. The present system on the bench uses 500 m. The system is scheduled for installation during the Christmas vacation this year.

Martin believes the system will meet the college requirements for the next 3 years, until Ring, Ethernet chips become cheap enough to use in this context.

Intermachine communication is a separate issue. At present they use ad hoc asynchoronous "string". They clearly are looking at Rings etc for the future.

NCB/SLOMAN PROJECT

Morris Sloman and Jeff Kramer were in Brussels, but we were given a demonstration of the NCB system by Jeff Magee.

The project has an operational 4 processor system (LSI 11/02). Each processor runs a kernel which handles message passing etc. Interprocess message passing channels can be set up dynamically. Process code cannot yet be down-line loaded, but a loader is being written.

The system is based on a loop of asynchronous connections.

Results of the project so far are really a set of concepts and a pilot implementation

The system assumes a computation is represented as a set of communicating modules. Such a representation is produced by the methodology Jim and Jeff have developed, hence the connection between the two projects.