ICOT TRIP REPORT

Alan Bundy
19.11.84

1. INTRODUCTION

From the 5th to the 22nd of November I visited Japan. The 5th and 22nd were spent in travelling; the 6th to the 9th at FGCS-84; the 12th on a technical tour to NEC and ETL; the 13th at the ICOT open house; the 14th at the Institute of Physics and Chemistry at Riken; the 15th and 16th at the University of Kyoto; and the 19th to the 21st at ICOT. This is my trip report. It tries to serve simultaneously for ICOT and my own group, and will probably satisfy neither.

I gave four talks altogether: my incidence calculus talk as a contributed paper at FGCS-84; a survey of our group activities in mathematical reasoning at Kyoto; the ECO ecological modelling IFE as my first ICOT talk; and the PRESS/IMPRESS equation solving system as my second ICOT talk.

2. AN IFE FOR ECOLOGICAL MODELLING

I described ECO, an intelligent front end for ecological modelling. ECO takes as input the description of an situation using ecological terminology, builds a customised FORTRAN program to simulate that situation, runs the simulation and outputs the results. It is intended as an intelligent interface for an ecologist who could benefit from computer simulation, but who is prevented from doing so by his ignorance of the necessary programming and/or mathematical skills. The main contribution of Artificial Intelligence is in the representation of the user's problem in a logical description, which can be transformed into a FORTRAN program.

ECO is an example of a growing body of 'intelligent front ends' (IFEs), which make computers available to people who could benefit from its use but are denied access due to ignorance of the appropriate terminology or skills, or just because of the inherent complexity of the packages they want to use. IFEs are an important application area for fifth generation computers because they will increase the usage of computers and make computer power more widely available. This area is a research theme of the UK, Alvey/SERC Intelligent Knowledge Based System Programme. I am the academic coordinator for the theme.

My talk provoked a lot of questions, which I recorded and summarise below.

Q - Is there any way to prove the correctness of the Fortran simulation program?

A - ECO can be viewed as a program transformation program in which the users specification of his problem is transformed from ecological terminology into FORTRAN. It would be interesting to show that that transformation was always correct, but we have not done so.

Q - Can we check the correctness of the user's specification?

A - The ultimate responsibility for correctness of the specification must lie with the user. We can only check the internal consistency and completeness, and interpret the results of the run which might debug some misconceptions that the user has built up about the significance of the model.

Q - Do our results extend to other areas.

A - The actual ECO program is only built for ecology. The same simulation techniques apply to other areas, e.g. economics, and hence ECO might be adapted for them by replacing the knowledge base and changing the dialogue handler. More generally we hope that we can generalise from our experience with ECO and other IFEs we have built, e.g. for statistics and mechanics, to provide general purpose tools for IFE building.

Q - How do we handle time-stamp and time-delay?

A - We do not anticipate that the knowledge base will change dynamically during the run of the program, so time-stamp is not an issue. Time-delay should be handled by the relevant module giving constant output during a sequence of time steps, and by the nesting of time intervals, e.g. where modules associated with a yearly time step and not called during a daily time step.

Q - Do we see any relation to de Kleer type envisioning?

A - There is a potential application of envisioning in ECO, namely to allow a symbolic simulation run. This could enable a user to use ECO to predict the qualitative behaviour of a situation without committing himself to quantitative initial data.

Q - What is the key idea of ECO?

A - There is no single key idea, but important ones are: the explicit representation of the users problem in the task specification and the transducers between this and (a) the user and (b) the FORTRAN code, namely the dialogue handler/knowledge base browser and the code generator.

Q - Can the system check the correctness of the users problem description?

A - To a limited extent. The system checks the types of the input and output provided to the nodes of the graphs. The system checks the completeness of the model in the sense that all node inputs required must be provided. Each module introduced will bring with it further requirements for inputs.

3. PRESS/IMPRESS

I described the PRESS equation solving system and the IMPRESS program correctness proving system. PRESS consists of a series of 'methods', each of which manipulates equations to produce a specific type of effect, e.g. making all occurrences of the unknown occur within identical subterms. These methods are chained together - each method achieving the preconditions of the next - to solve the equation. These methods guide the search for a solution and avoid a combinatorial explosion.

The methods can be incorporated into a mathematical theory: the meta-theory of algebra. We can reason in this theory either for the side effect of solving equations or, in its own right, to prove the correctness of the methods. The IMPRESS program was built to guide the latter kind of reasoning. It incorporates a method, adapted from the ideas of Boyer, Moore and Darlington, for guiding the induction technique.

I record below the discussion provoked by this talk.

Q - How are the conclusion and hypothesis part of an implication separated before input to IMPRESS?

A - This is done by the learning program that conjectures the implication. This is an excellent example of how problems that appear insuperable when only a narrow aspect of reasoning is considered, become tractable in a wider context.

Q - Would a different search strategy at the meta-level prevent the object-level search from working?

A - I don't think so, although a better meta-level strategy might improve the object-level search.

Q - What work have we done in learning?

A - The LP program (Precondition Analysis), the Focusing program (inductive inference), and the IMPRESS program (checking conjectured information).

Q - Was Leon Sterling's recent paper influenced by our work?

A - Early versions of that paper were written while Leon was in our group and summed up what he saw as the philosophy of the group.

Q - Is there search at the object-level?

A - Yes, although it is at most less than that done at the meta-level, because it is implicitly part of that search.

4. ICOT

ICOT is organised into three groups, which can be roughly classified as: 1. 5th generation computing, 2. high level software, 3. 4th generation computing. My visit was organised by Koichi Furukawa, the leader of the second group.

During its first phase ICOT has concentrated on building the logic programming based, but still essentially 4th generation, infrastructure for its work, e.g. the PSI machine with KL0, ESP and SIMPOS. It is now concentrating on 5th generation (i.e. parallel) infrastructure and applications software.

4.1 Fourth Generation

The Personal Sequential Inferential (PSI) Machine is a SUS of the same sort of power as a Symbolics Lisp machine, with 16 MBytes of memory, a bit mapped display, local area net, etc. Its 'machine code' is Kernel Language 0 (KL0), which is essentially DEC10 Prolog with knobs on, attaining 33k LIPS. ESP is a knowledge representation language, written in KL0, which gives an object-oriented programming facility and other things familiar from the Lisp-based knowledge representation languages. SIMPOS is the operating system of the PSI, and is written in ESP! DELTA is a data-base machine which can be accessed via the PSIs using Prolog. It provides large memory capacity on disk and RAM.

This experiment of using Prolog, rather than Lisp, as the basis of a SUS environment, is both interesting and, apparently, successful. I was pleasantly surprised by the ease with which things like an operating system, graphics and object-oriented programming facility could be provided, efficiently, in Prolog. The logic programming philosophy underlies everything done at ICOT and is a powerful unifying factor in the work.

4.2 Fifth Generation

A major payoff of this logic programming philosophy is the way it facilitates work on parallel processing. Shapiro's Concurrent Prolog has been taken as the parallel equivalent of DEC10 Prolog. It will be the new kernel language, KL1. The replacement for ESP is Mandala. The parallel hardware is the Parallel Inference Machine (PIM). Work on all these is reported in the proceedings of FGCS-84.

Shapiro has a particularly cunning way of doing object-oriented programming in Concurrent Prolog, where each object is a perpetual process one of whose arguments is a difference list of messages. The process removes the first of these messages, executes it and then calls itself on the rest of the list. Another perpetual process is busy instantiating the difference list. Someone at the conference showed how to implement arrays using this idea. In his invited talk, Bobrow showed some esoteric areas of object-oriented programming that could not be handled this way, but Shapiro claims to have the answer.

4.3 Applications Software

Work on applications seems to have taken third place to that on 4th and 5th generation infrastructure. This is fine for the initial phase, but impressive applications must be the ultimate basis for judging the success of the 5th generation programme, so they should eventually take first place.

Initial work on applications includes: a VLSI design aid, a Japanese language comprehension system, a Japanese chess player, a Japanese text editor, a CAI system for algebra, and an expert system for law and tax. Some of these projects involve people seconded to ICOT from industry. The language comprehension project is (anomalously) included in the third group, but all the others are in the second group.

The algebra CAI system has been built, in a couple of months, by Taizo Miyachi and Toshiaki Takewaki. It is inspired by PRESS, although I had difficulty finding out to what extent its 'methods' corresponded to PRESS's - not much I suspect. It was written in Prolog on a DEC20. The interface uses multiple windows (on a vt131) and is much nicer than PRESS's. They can solve a variety of problems taken from Japanese text books and our papers, mostly polynomials. The CAI part consists of showing the student user how to solve the problem with comments on the solution and a history of the rule applications. This would need considerable enhancement to become a useful CAI tool, e.g. allowing the user to solve the equation and offering advice when required. However, the program is very impressive for a few months work. It will be interesting to see how it evolves.

The Kaiser system uses Shapiro's model inference system to revise a faulty grammar and a faulty knowledge base. In the latter modifiable rules had to be marked as 'uncertain' to reduce the search space of changes. It was unclear whether this notion of uncertainty was meaningful outside the confines of the learning program, e.g. whether it was used by a uncertainty-based inference engine.

Hiroyasu Kondou is a semi-professional Shogi (Japanese chess) player. He is designing a Shogi player. The work is still in its early phases. He is collecting Shogi playing knowledge and thinking about the knowledge representation issues; there is no implementation.

Toshiyuki Tanaka is building an expert system for a small area of Japanese tax law involving deductions on medical grounds. He has a frame based representation of the knowledge required to describe a situation, written in Prolog on a DEC20, and is now designing the inference system.

Fumihiro Maruyama has built a VLSI design system in Prolog. The system is specified in Occam. It looked nice in the Open House demonstration, but as yet has only been applied to relatively simple circuits. Maruyama must have worked with the VLSI design people at Stanford because Ed Feigenbaum asked him how using Prolog compared with his previous experience there using Lisp. Maruyama gave a commercial for the relative virtues of Prolog for VLSI design. This shut Feigenbaum up.

DUALS is a question answering system for Japanese comprehension tests. It uses bottom-up parsing, an LFG grammar and situation semantics. The vocabulary is still small which limits its performance to a few test examples, but the system is under development.

5. FGCS-84

The first two days of of FGCS-84 were devoted to political speeches and fairly non-technical invited talks and summaries of ICOT research. The only exception to this was Alain Colmerauer's invited talk, so let me tell you about that.

Colmerauer has developed a theory of Prolog which accounts for the omission of the occurs check and the existence of infinite data-structures. This theory includes a version of the unification algorithm that permits unification of infinite structures without looping (unlike existing Prolog unification). An ironic consequence of this is that real unification (with occurs check) can be readily implemented by doing Colmerauer unification and then checking that the resulting terms and finite. This can cause looping in existing Prolog. Colmerauer's algorithm is 20 pay. Colmerauer has extended his algorithm, first to build-in 'inequalities' (really non-identities), and then to build-in the properties of rational numbers, strings and boolean algebra. The first extension is available in Prolog2 and the second in Prolog3. The built-in number unification uses the Simplex algorithm, and the Prolog3 extensions are fairly expensive. Prolog2, however, seems clearly worth the price.

The second two days of the conference consisted of contributed papers and panel sessions. Brief summaries, of the best of those that I attended, follow.

Joxan Jaffar, Jean-Louis Lassez and Michael Maher are working in a similar vein to Colmerauer. They prove the soundness and completeness of negation as failure for logic programming with generalised unification algorithms.

QUTE (Masahiko Sato and Takafumi Sakurai, Univ. of Tokyo) is an attempt to unite the ideas of functional and logic programming in a single language.

Yoav Shoham and Drew McDermott have a partial algorithm for inverting Prolog procedures defined for one calling pattern so that they work for another.

Kazunori Ueda and Takashi Chikayama show how to use Udi Shapiro's ideas about object-oriented programming to implement stream and array processing in concurrent Prolog.

Danny Bobrow gave an invited talk in which he described different programming styles that he thought should be supported in an AI language, and used this to criticise Prolog. He included an example which he claimed could not be handled in the Shapiro version of object-oriented programming in Concurrent Prolog.

Koichi Furukawa described Mandala, the knowledge representation system being developed at ICOT as a parallel processing replacement for ESP. It includes forward and backward inference, hierarchies, object-oriented programming, multiple windows, modules, etc. He claims that the logic programming approach leads to a more unified system than a Lisp based approach. A prototype runs (slowly) on the DEC10.

Rod Burstall described his ideas about modules and types in programming languages, and advocated their use in Prolog. Apparently Waterloo Prolog has a module system.

The panels I found disappointing. They were a series of rather formal, mini-lectures, with little contribution from the floor. However, after I had walked out of the parallel processing one to sleep off an acute case of jet lag, I was told it hotted up.

6. INCIDENCE CALCULUS

I presented my work on incidence calculus at FGCS-84. This is a mechanism for handling uncertainty in inference. It associates sets of incidents (situations, possible worlds) with formulae, rather than numbers, and in this way is able to capture a proper type of probablistic reasoning in which independence between formulae is not assumed. I made a mess of the question time, so the 'answers' that appear below are what I should have said rather than what I did say. The latter is only of historical interest. I also include conversations that took place after official question time.

Q - What is the relation to Montague, possible-world semantics?

A - Incidents can be thought of as equivalence classes of possible worlds, but that seems to be the limit of the connection.

Q - If we attach intensional incidences (i.e. expressions which evaluate to incidences) to formulae, how does this affect the inconsistency detection algorithm?

A - It would screw up. But why would we want to do that anyway?

Q - Why not use the existing theories of statistics to represent uncertainty?

A - Surely I am trying to precisely that, i.e. going back to the roots of probability theory to handle dependence. But I have to give a computational account, which involves adding to the statistical theory, e.g. by building algorithms.

to build, and sells for around 300 pounds. They have plans to transport it to the Sharp M25500, the IBM PC and the Apricot. It looks good for teaching. Hagino is currently a student under Rod Burstall at Edinburgh.

2 other students: Hagiya and Yuasa, have built a Common Lisp in C, which runs on a Data General Eclipse with impressive benchmarks - comparable to the Vax 780. Maybe it would port easily to the GEC series 63, and Alvey would be interested in buying it.

Nakajima has implemented a modular programming language called IOTA, with a built-in correctness prover of the Hoare logic type. Sakurugawa demonstrated this. The proof of a simple program took ages - even though the relation between the specification and the program was very close. It looked like an argument for logic programming, and the transformation approach to program correctness, to me.

At the EE Dept I met Makoto Nagao and discussed a pattern recognition program for aerial photographs and his machine translation project for two way translation of scientific abstracts between English and Japanese.

At the Information Science Dept, I met Shinji Tomita, and saw the parallel machines he is building for graphics and Prolog. I had demonstrations of: a system for finding different kinds of regions in newspapers (e.g. headlines, text, figures), a speech processing system for eliminating noise, and saw the big machines in the data processing centre - one 1.5 times the speed of a Cray 1.

8. MATHEMATICAL REASONING AND DISCOVERY

At RIMS I gave a talk surveying the work of the mathematical reasoning group at Edinburgh. I mentioned the work on theorem proving, learning, equation solving and incidence calculus. The talk produced the following questions and answers:

Q - How do you assign incidences given probabilities? Is this hard calculation done using correlations, triple correlations, etc?

A - I could use random assignment, within the correlation if given. This is to assume more information than is really provided and is subject ot error and back-up. I conjecture that back-up is not needed if the allocation for the rules is known to be correct and is only done dynamically for the facts.

Q - Why restrict yourself to mathematics as a domain?

A - (a) that is where my expertise lies, (b) it is good vehicle for studying inference, (c) actually we have wider interests in non-monotonic reasoning, psychological modelling and incidence calculus.

Q - Do we have trouble representing variables in Prolog?

A - It would be nice to have an occurs check for theorem proving work.

Q - Do we represent variables by Prolog variables?

A - In Press at meta-level we use Prolog variables and at object-level we use Prolog constants.

Q - Why not use meaningful incidents and have the user choose these directly?

A - I discuss this possibility in the paper. It would be nice if it could be arranged.

Q - (Russell Kirsch) I should be aware of the hostility that will be aroused in conventional statisticians because of my rejection of an arithmetic basis strikes at he heart of their paradigm. I must arm myself with statistical knowledge to counter and win this argument. He recommends: Bar Hillel & Carnap 'An outline of a semantic theory of information' 1952, which expresses similar ideas and Fellar 'An introduction to probability theory and its applications', for a basic introduction. John Toley at Princeton will have strong counter-arguments that I will have to meet. Never-the-less I should take heart because I am right.

Q - Doyle has a discussion of uncertainty using lattices.

Q - Pawlak, Poland has an incidence-like system called 'information system', which is published in the 'German literature'.

7. MY TRAVELS

During the week 12th-16th I was sent on various trips for my education and to keep me out of the hair of the ICOT people, who were busy with their Open House and recovering from that and FGCS-84. I had a bewildering series of demonstrations and talks during this time. I will summarise them briefly below, except where they are especially close to my interests.

At NEC we saw some promotional videos and a showroom of products. Few of the products were very advanced, but they were well packaged and cheap. Most interesting were a couple of 200-500 word, continuous speech recognisers, and a cheap SUS. The SUS was about 4000 pounds complete with bit-mapped, colour display, resolution 768x512, 128k memory, floppy disk drive and printer.

At ETL we saw a low level, formant-based, speech recogniser; a chinese character recogniser based on a distance heuristic guiding a polynomial fitter; and a logic program transformer. The latter was based on the work of Darlington, Clarke and Hogger, except that it preserved equivalence during transformation of a simple, inefficient program to a complex, but efficient one. The author was Taisuke Sato.

At Riken I saw a special purpose machine they are building for algebraic manipulation, which Arthur Norman and crew plan to access over a WAN. I was told about their work on Josephson junctions, which exploits an alternative to the transistor, called a parametron, as the basic device.

At Kyoto University, Alain Colmeraur and I visited: the Research Institute of Mathematical Sciences (RIMS), the Electrical Engineering Department, and the Information Sciences Department.

At RIMS, 3 of Reiji Nakajima's students: Sakurugawa, Shibayama and Hagino, have built a version of Prolog for CPM and MSDos machines. It is called KABA Prolog. KABA is Japanese for hippo, and an acronym for Kyoto Anti-Basic Association and Kyoto Artificial Brain Associates. It attains 1k LIPS on a NEC PC-9800 (a personal computer selling for a few thousand pounds), is of DEC10 standard and has a built-in screen editor, colour graphics, tail recursion optimization, and garbage collection. It only took about 2.5 months

9. COOPERATION

There was much talk during the conference of the value of cooperation between Japan and other nations on 5th generation research. Alvey and ICOT met after the conference and agreed in principle to cooperate. Everybody agrees with cooperation in principle, but in practice there are problems of national and commercial interest and barriers of language, culture and distance. I will leave the national and commercial problems to others, and address the other problems.

The language and cultural problems are still present, but are diminishing, thanks largely to the efforts of the Japanese. They nearly all speak excellent English and have become more 'western' in their approach to research interaction, i.e. more informal and out-spoken. That is not to say that communication is now perfect, but it is improving.

Problems of distance and time difference are not being solved so easily. For instance, very few Japanese labs are connected to international networks. There is a political problem with the ArpaNet; the, otherwise laudable, non-military policies of the Japanese prevent them from connecting. No such policy prevents connection to CS-Net, Use-Net or SERC-Net, and Alvey could particularly promote the latter. Distance also makes it hard for us to experiment with and extend the ICOT hardware and software - how could we maintain the hardware?

10. ACKNOWLEDGEMENT

I would like to thank ICOT for funding my trip to Japan and arranging a series of visits to research labs and discussions with researchers. It was an interesting experience for me. Everybody was very friendly and helpful and went out of their way to make my stay enjoyable and valuable.

Curriculum Vitae

Alan Bundy

Alan Bundy received the B.S. degree in mathematics from Leicester University, Britain in 1968, and PhD in Mathematical logic, also from Leicester University, in 1971. Since 1971 he has been a member of the Department of Artificial Intelligence, University of Edinburgh, first as a research fellow, then a lecturer and now a leader. His research interests are in the automation of mathematical reasoning and discovery. He has held a number of research grants on this topic. He is the author of two books on Artificial Intelligence. He was programme chairman of IJCAI '83.