Report to ICOT on my visit of 29 Nov - 3 Dec 1982.

by

E.A.Ashcroft
University of Waterloo and SRI International

INTRODUCTION

During my visit I engaged in the following activities. I gave three lectures, attended and participated in two "miniworkshops", engaged in discussions with several researchers, and gave an informal presentation of some ideas, due mainly to Bill Wadge, for making improvements to PROLOG, yielding a language which we call EPILOG. I will describe each of these activities in this report, and also summarise my impressions of ICOT.

LECTURES

The first lecture I gave was entitled

"Lucid, the dataflow programming language".

In this lecture I gave an introduction to the main ideas in the language Lucid, which is a language invented by myself and Bill Wadge, who is a lecturer at the University of Warwick in England. The language is quite unusual. It is technically a functional programming language, but to get the best out of it the programmer should think in terms of dataflow, by which I mean he or she should think in terms of "filters" processing streams of data values. (In fact there is a close

correspondence between Lucid programs and dataflow networks, and this was shown in the third lecture.) The language embodies the concepts of both recursion and iteration, making it natural to use by programmers of either persuasion (after they have got used to thinking in terms of filters). I brought several papers with me concerning Lucid, and the one with the same title as this lecture is a good introduction to these dataflow ideas.

The second lecture was entitled

"Lucid and the future of functional programming".

It gave a short, embarrassingly naive summary of the present situation vis-a-vis functional programming, how it has experienced a resurgence in popularity since the advent of VLSI and the decrease in the cost of hardware, and then showed how Lucid, even though it is a functional programming language, allows one to write programs that solve problems in an iterative way. It was maintained that this was an advantage, but the audience, mainly researchers in functional programming from all over Japan, were unconvinced, I think.

The third lecture was entitled

"Lucid and dataflow networks".

It showed how Lucid programs could be represented, very directly, as dataflow networks. These networks would have to be implemented on a demand-driven dataflow machine (for efficiency). The networks are quite standard, except for a construct to deal with subcomputations, which are handled in a very ad hoc way by most existing dataflow network languages. It was claimed that any network of this type could just as easily

be represented as a Lucid program, so that Lucid could justifiably be taken as the "machine code" of an abstract dataflow machine which "runs" such dataflow networks.

MINIWORKSHOPS

The first miniworkshop was with the PIM group of ICOT, that is with Rikio Onai, Noriyoshi Itou, Kanae Masuda and Hajime Shimizu. Itousan gave a presentation of the PIM-D project,explaining their ideas for designing a dataflow machine for PROLOG. Then Onaisan explained the PIM-R project, the design of a reduction machine for PROLOG. There followed a fruitful discussion of the different designs, mainly concentrating on clarifying my understanding of them. I did make a suggestion or two, primarily about PIM-D.

The second miniworkshop was with the PO group, namely with Toshio Yokoi and Toshiaki Kurokawa and three junior researchers, Takashi Hattori, Ko Sakai and Junichiro Tsuji. Firstly Yokoisan gave an overview of the Operating System project and then details were given, Hattorisan talking about the proposed basic constructs, Tsujisan talking about the the SIM Window System, one of the proposed sub-systems, and Sakaisan talking about the proposed programming system. There followed a discussion the whole project, with me expressing amazement that the project had to be completed in so short a time, by so few people. I was relieved to hear that various sub-tasks were going to be farmed out to the eight companies in Japan who are involved in supporting ICOT. Nevertheless the overall design

was to be the responsibility of the PO group, which I felt was a good idea. Too often, big projects are the responsibility of large heterogeneous committees, with disastrous results.

Kurokawasan then talked about the Intelligent Programming System Project, which to me is the core of the Fifth Generation Project. This is a longer term project than the Operating System Project, and at the moment is concerned with studying the present state of the art in specification systems and program synthesis and with deciding what else would be included in an intelligent programming system. Suggestions include program transformation, optimisation and verification, and various applications of logic programming, still to be settled upon.

DISCUSSIONS

The main discussion, apart from those in miniworkshops, was that following lecture three, with the functional programming researchers of Japan. Much of this centered on understanding Lucid, with me trying to explain that the infinite sequences in Lucid are not just infinite lists, that they are really transparent to the user and are a mathematical device that the user never really bothers about. I don't know how well this idea came across. The discussion moved on to reduction machines, and I expressed my opinion that, although they are a natural way of implementing functional languages and PROLOG, to me they represent a syntactic, operational way of expressing the semantics of a language, which leaves much to be desired if

questions of debugging, in particular, are considered. As a denotational semanticist, I think that the meanings of programs should be mathematical objects, not transformation rules. Such rules are useful, but they should come after the mathematical semantics, which should be used to verify that the rules are in fact valid.

EPILOG

I gave an informal presentation of some ideas of Bill Wadge on improvements to the language PROLOG. Those present included Yokoisan, Onaisan, Kurokawasan and several others whose names I have forgotten. I do not have space here to go into the detals of EPILOG, but one of its design objectives is that the semantics of the language should not be syntactic, that is, not specified by the results of symbol manipulation. (See my previous comments on reduction machines.) The details of EPILOG have not yet been all worked out, but some written material was left with Onaisan. I also talked a little about the possibility of combining EPILOG and Lucid into a single superset of both called Epilucid (or Lucilog?).

IMPRESSIONS

The main impression that ICOT has left me with is one of respect, even awe, for the scope of the whole thing. The project bears all the hallmarks of a carefully considered enterprise. The details are still a little vague, some would

say based on faith and unreasonable optimism, but the enthusiasm for the project is infectious and one can't help having a sneaking suspicion that it is actually going to work.

Most of the individual research projects I was given details of (and I was given far more details than I expected) seem to be soundly based. If I have one misgiving it is about the pervasive use of PROLOG in the system. Even if the language is made more reasonable, in, say, an EPILOG-like way, it will not be an efficient language for conventional programming tasks that do not need PROLOG's nondeterminism. It could be said that the proposed machines with large numbers of processors will make PROLOG efficient, but the fact remains that that large number of processors devoted to a language like Lucid (to mention one that unaccountably springs to mind) could be SUPER-efficient. It is possible to have both: if your PIM is designed to run Lucid, it will also run PROLOG (though slightly less efficiently than it runs Lucid perhaps) and the users of the machine can program in either - Lucid for determinate programs and PROLOG for intelligent programs. Just a suggestion.

Edward A. Ashcroft

office: Computer Science Department
University of Waterloo
Waterloo,Ontario N2 L3 G1
Canada
(519)885 1211 ext.2793

home: 159 University Ave.W
Waterloo,Ontario N2 L3 E8

Canada
(519)886 4226

Personal Data

Born 15th July 1944 in England, United Kingdom and Canadian Passports

Education

Ph.D. in compouter science, University of London 1970
B.A. in electrical engineering, Cambridge University 1966

Current Research Interest

The prcgramming language LUCID;
Dataflow machine;

Experience

1971-present
Professor,Computer Science Department, University of Waterloo.
1969-1970
Research Associate, AI.Project, Stanford University.

Conference and Workshop Presentations

IFIP Working Group 2.1 meetings Jan. andSept. 1982
IFIP Working Group 2.3 meeting Aug. 1979
IFIP 77 Conference, 1977
ACM Symposium of Principles of Programming Language, Jan. 1978

Thesis

Ph.D. "Mathmatical Logic applied to the Semantics of Programming Languages"

Publications

"Rx for Semantics," TOPLAS,1982
"A Generalised Setting for Fixpoint Theory,"TCS 1981
"Lucid, a Nonprocedural Language with Interation," CACM20,1977