A VIEW OF THE FIFTH GENERATION AND ITS IMPACT

Technical Note 265

July 1982

By: David H. D. Warren, Computer Scientist

Artificial Intelligence Center
Computer Science and Technology Division

To be presented at the Pergamon Infotech State of the Art Conference on "Japan and the Fifth Generation" in London, England, 27-29 September 1982.

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

ABSTRACT

In October 1981, Japan announced a national project to develop highly innovative computer systems for the 1990s, with the title "Fifth Generation Computer Systems". This paper is a personal view of that project, its significance, and reactions to it.

CONTENTS

I PREFACE

This paper presents a personal view of the Japanese Fifth Generation Computer Systems project. My main sources of information were the following:

- * The final proceedings of the International Conference on Fifth Generation Computer Systems, held in Tokyo in October 1981, and the associated Fifth Generation research reports distributed by the Japan Information Processing Development Center (JIPDEC).
- * Presentations by Koichi Furukawa of Electrotechnical Laboratory (ETL) at the Prolog Programming Environments Workshop, held in Linkoping, Sweden, in March 1982.
- * Talks given by Ed Feigenbaum of Stanford University, and conversations with him.

The information available to me at the time of writing (June 1982) is not as complete as I would have liked, and is partly secondhand. I apologise for any mistakes or misinterpretations I may therefore have made.

II THE FIFTH GENERATION PLAN

In late 1978 the Japanese Ministry of International Trade and Industry (MITI) gave ETL the task of defining a project to develop computer systems for the 1990s, with the title "Fifth Generation." The project should avoid competing with IBM head-on. In addition to the commercial aspects, it should also enhance Japan´s international prestige. After various committees had deliberated, it was finally decided to actually go ahead with a "Fifth Generation Computer Systems" project in 1981. The project formally started in April 1982.

The technical content of the plan seems to be largely due to people at ETL and, in particular, to Kazuhiro Fuchi. The overview given by Furukawa was:

```
VLSI --> "dataflow"      --> logic        --> "knowledge
           architectures     programming       info. processing"
```

"Dataflow" is taken to include advanced highly parallel architectures in general, and is not restricted to what is usually meant by dataflow. "Knowledge information processing" basically means applied artificial intelligence (AI); it includes "expert systems" and natural language interfaces. Thus VLSI technology is to be exploited to build advanced parallel architectures for AI-type applications, where the basic machine language will be an extension of the logic programming language, Prolog. So logic programming and Prolog play a crucial role in the systems envisaged. According to Furukawa, the decision to go for "dataflow" won out over more conservative ideas for parallelism.

The first step to be taken, which is intended to serve as a "springboard" for the rest of the project, will be to develop a high-performance personal Prolog machine. The ultimate aim is to achieve, through highly parallel hardware, a performance of "one gigalips," meaning one billion logical inferences per second, i.e., one billion Prolog procedure calls per second. This would be equivalent to something like 10,000 to 100,000 times the power of a DEC KL-10.

III PROLOG

Since Prolog plays such a central role in the Fifth Generation, what exactly is it?

Prolog [15] [2] is a general purpose programming language based on logic. It is a practical realization of the concept of "logic programming", due to Robert Kowalski [6].

Prolog can be viewed either as an extension of pure Lisp, or as an extension of a relational database query language. It was first conceived in 1972, by Alain Colmerauer at the University of Marseille. Since then it has been used, mainly in Europe, for a wide variety of applications, including natural language processing, algebraic symbol manipulation, compiler writing, architectural design, circuit design, and expert systems. The work on natural language processing [3] [4] [7] [14] is particularly notable (especially in the light of the Fifth Generation), as are the large-scale expert systems that have been implemented in Hungary [10], including one which assists organic chemists in the pharmaceuticals industry.

The Edinburgh DEC-10 Prolog system [12] [9] includes a compiler which generates code comparable in efficiency with that produced by current Lisp compilers [11]. Other work [13] indicates that, for queries over small databases, DEC-10 Prolog's speed is comparable with or better than current relational database systems.

Prolog is radically different from most programming languages, in that it does not presuppose a von Neumann architecture and does not have assignment as the basic underlying operation. Instead, Prolog is based on symbolic logic. A Prolog program consists of a set of statements which can be read declaratively as well as procedurally. In order to be sure that a Prolog program is correct, one just has to satisfy oneself that each statement is "true." For example, here are some Prolog statements:

```
european(europe).
european(X) :- partof(X,Y), european(Y).

partof(london,britain).     partof(britain,europe).
```

which can be read (declaratively) as:

Europe is European.
For any X and Y, X is European if X is part of Y and Y is
  European.

London is part of Britain. Britain is part of Europe.

Given these statements, Prolog can determine, for instance, that London is European.

The declarative nature of the language, and the absence of assignment, are significant for two main reasons:

(1) Prolog makes the task of programming much easier. Turning an idea into a correctly running program is simply a lot less effort with Prolog. This is the main reason for the language´s popularity.

(2) Prolog is inherently well suited to parallel computation. The semantics of the language does not presuppose a strictly sequential execution; it is relatively easy to conceive parallel-processing strategies that are consistent with the semantics.

IV FUCHI´S OVERVIEW PAPER

What is the thinking behind the Fifth Generation? Why was Prolog chosen as the kernel language? The most widely circulated document on the Fifth Generation, the "Preliminary Report," is rather obscure on these questions. To obtain the answers, one has to turn to Kazuhiro Fuchi´s paper "Aiming for Knowledge Information Processing Systems" [5] in the full proceedings of the Fifth Generation Conference. In this paper, Fuchi gives a very clear explanation of the reasoning behind the Fifth Generation.

He begins by observing that present-day computers have a basic design that has not changed radically since the original conceptions of John von Neumann and others, and he states that "many voices are raised in dissatisfaction over present-day computers" for not being "truly handy." He then suggests that the time is ripe to make a "bold proposal" to realize what he calls "knowledge information processors"--computers with a nontraditional architecture that present a more human-oriented interface to the user.

The time is judged to be ripe because: "Reviewing the 1970s, research efforts into computing technology may be said to have split into a number of streams, and to have progressed through mutual competition. Along with this, interrelations between them grew in the latter half of the decade, and a trend toward mutual fusion emerged. This may be regarded as a valuable bud which will blossom in the 1980s, and an important legacy from the 1970s."

The main research achievements that Fuchi cites are:

(1) Proposals for new computer architectures, especially dataflow, and the related proposals for "single assignment languages," which, according to Fuchi, "resemble what has been derived from research on inference systems." In particular, Fuchi states that "it

is feasible to extend dataflow machines to inference machines," by applying the ideas of logic programming.

(2) Proposals for new programming styles and languages, aimed at achieving a clearer semantics, especially "structured programming," "functional programming," and "logic programming."

(3) Relational databases. Fuchi considers their philosophy to be closely related to that of logic programming. "At present it is common that databases and programming languages belong to different systems. This is not a desirable situation. Their unification appears to be quite feasible."

(4) Work on natural language understanding, especially that oriented towards logic.

(5) Results derived from artificial intelligence, especially the languages Planner and Prolog, and "knowledge engineering" applications. Fuchi says that "Prolog may be regarded as a logically reorganized Planner." He characterizes knowledge engineering applications as those requiring a "knowledge base plus inference engine."

As should be clear by now, the concept which Fuchi sees as drawing these five research areas together is logic programming. He goes on to explain why he thinks the programming language Prolog will form a sound starting point: "Excellent implementation techniques have been developed permitting its efficient execution." Prolog's base (logic) is "the same as formal specification languages, facilitating transformation of specifications into programs." "Prolog also has the same logical base as relational databases, and is suited as a base for integrating programming and database query languages." "Prolog is also intrinsically suited as a base for realising natural language processing and higher level inference functions."

Fuchi explains that the reason for choosing Prolog as the starting point rather than Lisp is "primarily that Prolog can be seen as an extension of Lisp." He says Prolog provides the extra functionality of "pattern matching and non-determinism," and is "capable of integrating interesting features of other languages such as Smalltalk, PS, and APL." But "will Prolog machines be feasible?" he asks. His answer: "Yes, if they follow the same lines as current Lisp machines, they are technically feasible even now." "It should in the near future be technically possible to achieve conversational Prolog machines, equipped with, for instance, 1 M-byte or greater main memory, several dozen M-byte disks, a high quality graphic display, etc., and to create environments in which they can be used personally."

So the great attraction of the Prolog approach is that a machine, technically still belonging to the fourth generation, can serve as a stepping-stone to the advanced architectures and applications envisaged for the fifth generation.

V THE STORY BEHIND THE CHOICE OF PROLOG

Considering that Prolog was relatively unknown to the computing world before the announcement of the Fifth Generation, the choice of that language as the basis for a national project of the scale proposed was quite a bombshell! It certainly seems to have come as a complete surprise to the logic programming community outside Japan, which had previously had little inkling of the Japanese interest in Prolog and logic programming.

Outside Japan, there are perhaps 500 people actively interested in Prolog and logic programming. Japan must now have jumped to the "head of the league" of countries developing or using logic programming (the rest, to my knowledge, being USA, Britain, Hungary, France, Sweden, Canada, Portugal, Poland, Australia, Venezuela, Belgium, Norway, Denmark, Yugoslavia, New Zealand, Italy, Argentina, Finland, USSR, Ireland, Costa Rica and India, in rough order of the amount of work being done).

The story of how Prolog got to Japan is quite interesting. Apparently Fuchi had been interested in logic programming since reading Kowalski's 1974 paper. Prolog itself was first brought to Japan by Furukawa--ironically enough from SRI, where he was an international fellow in 1976, at a time when Prolog was virtually unknown in the United States. SRI, in the person of Harry Barrow, had acquired the original Marseille Fortran implementation of Prolog from me at Edinburgh (where I then was). Harry had not been able to get the system running himself, but gave the sources to Furukawa. Several institutions in Japan subsequently acquired copies of DEC-10 Prolog from me.

There evidently must have been quite a lot of politicking to get Fuchi's "bold proposal" accepted, but eventually, according to Furukawa, people were persuaded that in the 1990s "even cats and spoons will write

Prolog" (to use a Japanese expression). However, the project planners have been careful to hedge their bets; the research on "new advanced architectures" will include studies for a "functional" (Lisp) machine and an "innovative von Neumann" machine, according to the Fifth Generation Committee's preliminary report. Certainly, the influence of Lisp in Japan is still strong. The proponents of Prolog seem to be known as the "Basic Theory" group, and to come largely from ETL. A provisional name for the extended Prolog that will form the kernel language of the Fifth Generation is "Himiko," after a woman from Japanese history--presumably Japan's answer to "Ada."

VI THE FIRST THREE YEARS

The first stage of the Fifth Generation plan calls for a 3-year initial study costing $50 million. The main goals of this stage are:

* A high-performance personal Prolog machine (cf. the Lisp Machine).
* A relational database machine.
* Basic research studies to pave the way for the following four years.

The Prolog and database machines will be used as research tools, serving as a "springboard" for the rest of the project. A surprisingly specific specification of the Prolog machine has been reported [16]: 0.2 megalips speed, one million words of main memory, Winchester disk, bit-mapped display, compact size and "beautiful apperance." The fact that the word size is reported to be 36 bits suggests that some form of emulation of DEC-10 Prolog is intended. Since DEC-10 Prolog achieves at best 30,000 lips, the personal Prolog machine is targeted at being some 7 times faster than a DEC KL-10. I understand it will not be simply a microcoded implementation in the style of the original Lisp Machine. There are said to be no definite plans to sell the Prolog machine to the outside world, although it is possible that a company like Fujitsu might produce a commercial version.

VII PROGRESS REPORTED SO FAR

Furukawa [17] has described a Prolog interpreter, implemented in Simula, which breaks the execution up into two kinds of process, AND-processes and OR-processes. It is intended that a simple extension of this idea will permit a limited degree of parallel execution (OR-parallelism only). This study is apparently aimed beyond the Prolog machines envisaged for the first stage.

Also reported is a parallel logic programming system called "Paralog," developed by Aida and Moto-oka [1], which is already running on a parallel machine called Topstar-II. This machine consists of 24 Z-80s, comprising 16 "processing modules" and 8 "communication modules." Here again, only OR-parallelism is involved. It appears that entire resolvents (Prolog execution states) are copied and transmitted between processing modules. Performance tests have been made on Prolog programs for symbolic differentiation, natural language processing, and logic (i.e., circuitry!) simulation. Timings for the symbolic differentiation example showed execution speed to be proportional to the number of processing modules used.

In March 1982, a national conference on Prolog was held in Japan with about 80 people attending [17]. Of the 12 papers presented, 2 were on machine architectures, 2 on theoretical aspects, 4 on various software implementations, and 4 on applications. The applications papers describe Prolog programs for antibiotics counselling, game-playing, engineering design, and natural language (Japanese) understanding. Furukawa has also shown me a sophisticated program for solving Rubik's cube, which is based on a production system implemented in Prolog, and which displays its moves on a color terminal.

VIII THE SIGNIFICANCE OF THE FIFTH GENERATION PROJECT

The Fifth Generation project is extremely bold and ambitious. If it succeeds, it will truly bring about a revolution in computing, superseding the von Neumann architectures and von Neumann programming languages which have remained essentially unchanged since the '50s. Japan is taking a calculated gamble. Either the project will fall flat on its face, or it will achieve a spectacular success. When asked about the risks, Furukawa seemed unperturbed. My guess is that those concerned feel they have already done enough groundwork to be confident of success. And, in fact, the goals for the first 3 years do not really involve any great advance. As Fuchi says, a Prolog machine along the lines of a Lisp Machine is indeed "technically feasible now."

It is interesting to speculate as to why the aims of the project have been revealed so openly. Some conjectures are as follows:

* The prestige aspect is indeed important.
* Because of the high degree of research involved, Japan wishes to be able to exchange ideas and information with other countries.
* Japan wishes to stimulate work on applications for which the proposed machines will be especially suited. There is a need to create a market for the product! It seems the project is concentrated mainly on the hardware side of the Fifth Generation, with Japan probably relying on the outside world to produce much of the software.
* Some degree of competition on the hardware side would probably be welcomed, to avoid the machines being seen as a uniquely Japanese product, inviting trade barriers.

How should the rest of the world respond? First, it would seem wise to find out much more about what the Japanese are actually doing. Then the decision must be made whether to compete or, if the Japanese are willing, to collaborate.

Fuchi says the route they are taking represents an "inevitable direction for the development of information processing technology." The only question was "whether to stand still or proceed, as there are no other paths to choose from." If he is right, other countries have to decide whether to follow or to sit back and leave it to Japan. Either way, it would appear Japan has already scored on the prestige front.

IX REACTIONS IN BRITAIN AND THE UNITED STATES

In Britain generally, although there has been considerable reaction at high levels, including rumors that the Department of Industry is contemplating responding to the tune of 250 million pounds, there seems to be little appreciation of the details of the Fifth Generation project, particularly the very concrete plans for the first three years. This is surprising, considering how much the Fifth Generation owes to work in Britain on logic programming and Prolog, and in view of Britain's strength in work on dataflow at Manchester and elsewhere.

This lack of awareness seems to be due to the following factors:

* None of the people from Britain who attended the Fifth Generation Conference in Japan knew much about logic programming and Prolog. (Kowalski was invited, but was unable to attend).
* Few people have read the full conference proceedings. Most people have only seen the "Preliminary Report" which is rather vague about how the different aspects fit together. In particular, when I visited Britain in March, no-one I met had read the paper by Fuchi which explains the "grand design."
* Much of the emphasis in the British response so far seems to have been on expert systems.

In the United States too, the general reaction is one of bemusement at the enigmatic picture of the Fifth Generation to be found in the "Preliminary Report." Reactions are further tempered by the fact that, although Prolog has been in existence since 1972, it is only recently that it has made much of an impact in the United States. Prolog is seen as somehow "un-American"--a European fad that has now been taken up by the Japanese. I have even heard it suggested that the only reason Japan has opted for Prolog, rather than Lisp, is that the Japanese do not wish to be seen to be copying American technology. While this aspect of Prolog may have a certain appeal in Japan, it can hardly explain why

Japan should risk so much money and prestige on a relatively unknown language.

Despite the lack of understanding of the details of the Fifth Generation, there is a widespread feeling that some kind of response is called for, although in what direction is not clear. In some quarters, the quandary that arises is how to respond without giving the appearance of simply following the Japanese. The Defense Advanced Research Projects Agency is believed to be considering injecting more money into research related to the Fifth Generation, and the US computer manufacturers' joint research company, MCE, is taking an interest in the hardware side of the Fifth Generation. Apart from that, there is considerable interest at a more grass roots level inside US companies, and one can be sure that these companies will take a pragmatic approach.

X PROLOG IN THE UNITED STATES

In view of its effect on reactions to the Fifth Generation, it seems worthwhile to review Prolog´s status in the United States. The main reasons for Prolog´s late arrival on the American scene seem to be the following:

- Prolog did not originate in the United States, and there was no published paper in English describing the language until 1977.
- Prolog was perceived as being purely an artificial intelligence language, and was pigeon-holed with the Planner family of "problem solving" languages, which had fallen into disfavor.
- Prolog´s "ecological niche," symbolic computation, is overwhelmingly dominated in the United States by Lisp, to a far greater extent than elsewhere. This has made it harder for Prolog to gain a foothold in that country.

McDermott´s SIGART article on the "Prolog Phenomenon" [8] contains an interesting perspective on these matters.

Since about 1977, Prolog activity in the United States has been steadily growing, roughly doubling each year. It started in the universities (especially Syracuse, Irvine, Kentucky, Yale, and Caltech), but has more recently spread to company research laboratories (Logicon/OSI, Digital Equipment Corporation, IBM, Hewlett Packard, Xerox, and others). In 1981, two Prolog workshops were held in the United States--one at Syracuse University, the other in Los Angeles (organized by the Operating Systems Division of Logicon), with about 50 and 60 participants, respectively. At the present time, to my knowledge, there are at least four groups contemplating building hardware for Prolog.

XI CONCLUSION

Japan has launched a far-sighted project for computer systems of the future, based on the ideas of logic programming and Prolog. The researchers involved show every sign of having the necessary expertise and judgment to make the project a success. The United States is unlikely to be left far behind, and work on Prolog-based machines and applications is already underway at several centers. It will be ironic if Japan and the United States lead in exploiting ideas originally conceived and developed in Europe (especially Britain).

REFERENCES

1. H. Aida and T. Moto-oka, "Performance Measurement of Parallel Logic Programming System ´Paralog´," Dept. of Electrical Engineering, University of Tokyo. Presented at the Prolog Programming Environments Workshop, Linkoping, Sweden (March 1982).

2. W. F. Clocksin and C. S. Mellish, *Programming in Prolog*, Springer-Verlag (1981).

3. A. Colmerauer, "An Interesting Subset of Natural Language," *Logic Programming*, Clark and Tarnlund (eds.), Academic Press (1982).

4. V. Dahl, "Translating Spanish into Logic Through Logic," *AJCL*, Vol. 7, No. 3, pp. 149-164 (July 1981).

5. K. Fuchi, "Aiming for Knowledge Information Processing Systems," International Conference on Fifth Generation Computer Systems, pp. 101-114, Tokyo, Japan, (October 1981).

6. R. A. Kowalski, "Logic for Problem Solving," Elsevier North Holland (1979).

7. M. C. McCord, "Using Slots and Modifiers in Logic Grammars for Natural Language," *Artificial Intelligence Journal*, Vol. 18, pp. 327-367 (1982).

8. D. McDermott, "The Prolog Phenomenon," ACM SIGART Newsletter 72, pp. 16-20 (July 1980).

9. L. M. Pereira, F. Pereira, and D. H. D. Warren, "User´s Guide to DECSystem-10 Prolog," Department of Artificial Intelligence, University of Edinburgh (1978).

10. E. Santane-Toth and P. Szeredi, "Prolog Applications in Hungary," *Logic Programming*, Clark and Tarnlund (eds.), Academic Press (1982).

11. D. H. D. Warren, "Implementing Prolog--Compiling Predicate Logic Programs," Research Reports 39 and 40, Department of Artificial Intelligence, University of Edinburgh (1977).

12. D. H. D. Warren, "Prolog on the DECsystem-10," *Expert Systems in the Micro-Electronic Age*, Michie (ed.), Edinburgh University Press (1979).

13. D. H. D. Warren, "Efficient Processing of Interactive Relational Database Queries Expressed in Logic," Seventh Conference on Very Large Data Bases, pp. 272-281, Cannes, France (September 1981).

14. D. H. D. Warren and F. C. N. Pereira, "An Efficient Easily Adaptable System for Interpreting Natural Language Queries," Research Paper 156, Department of Artificial Intelligence, University of Edinburgh (February 1981). [Submitted to AJCL.]

15. D. H. D. Warren, L. M. Pereira, and F. C. N. Pereira, "Prolog--The Language and its Implementation Compared with Lisp," ACM Symposium on AI and Programming Languages, pp. 109-115, Rochester, New York (August 1977).

16. T. Yokoi, et al, "Logic Programming and a Dedicated High-Performance Personal Computer," International Conference on Fifth Generation Computer Systems, pp. 151-156, Tokyo, Japan (October 1981).

17. Anonymous, Proceedings of Prolog Conference, Tukuba, Japan (15-16 March 1982). [In Japanese.]