TM-0074

Progress in the Initial Stage of the FGCS Project

by

Kinji Takei

September, 1984

©ICOT, 1984

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Progress in the Initial Stage of the FGCS Project

Kinji Takei

Research Center

Institute for New Generation Computer System

ABSTRACT

The Japanese Fifth Generation Computer Systems Project, which will span about ten years, is divided into three stages: the initial stage (three years), the intermediate stage (four years), and the final stage (three years). The initial stage began in April of 1982, and is mainly devoted to the development of basic elements of computer technologies that will contribute to knowledge based information processing. This stage of the project is now approaching its conclusion

The R&D themes targetted for study in the initial stage include basic technologies for parallel inference and knowledge base machines, basic software systems, and the design and implementation of a sequential inference machine for software development.

In this presentation, we survey the main results of these initial R&D efforts. We give an overview of the project, including its approaches and basic plans for the next seven years, discuss various points concerning each R&D theme, and report our main results up to the present.

1. MOTIVATION OF FGCS PROJECT

The aim of the Japanese Fifth Generation Computer System (FGCS) Project is to create prototypes for the computers of the 1990s. computers are now used in many fields of industry and society. They will play a central role on the socio-economic front in the years ahead. However, if they are to serve as a tool for solving the diverse problems anticipated for computer-application in the 1990s, their basic technology must be entirely re-examined.

As post-industrial world moves closer to becoming a sophisticated "Information Society", it is foreseen that, in the 1990s, information and knowledge will have an increasingimpact on individuals, enterprises, and nations. The winners will be those who make more effective use of information and invent original knowledge. Here, the decisive factor is "intelligence" or one s intelligenct productivity, and the expected role of the computer is that of "intelligence-amplifier" as it has been called by Prof. W.R.Ashby.

current computers, regarded as intelligence-amplifiers, are very narrowly bounded. Needless to say, they are extremely efficient in straightforward processing, such as numeric calculation and routie business applications. In such processing, computers manipulate numbers or words without understanding their meanings like blinkered horses. H0wever, we know that human intelligence enables us to make inferences, solve problems, and is responsible for language, vision, hearing, and so on. If we expect computers to even partially possess such mental faculties, we must get them to understand situations, and to make choices about what to do, for themselves. This is the job of "Artificial Intelligence" (AI). We look to the results of AI efforts for more general-purpose intelligence-amplifiers that can fully support the new applications expected in 1990s. However, it is well known that AI programs tend to be too big and complex to implement and execute on conventional von Neumann computers, which are primarily oriented toward numeric calculation. We are in urgent need of new computers capable of thinking what to do for themselves.

2. AIMS OF FGCS PROJECT

The ultimate purpose of the FGCS project is to implement computer systems that works as wide-band intelligence- amplifiers. We expect this purpose can be attained through realization of an advanced information processing system called KIPS (Knowledge Information Processing System). KIPS will be an information processing system capable of understanding the meaning of the information it processes, and of controlling its own processes using information it contains. This is a case of knowledge processing using knowledge (including meta-knowledge) and is a typical goal of applied AI.

The point of developing the KIPS system is to systematically and coherently integrate various AI technologies that have been separately investigated by individual AI researchers. We can foresee many difficulties in this integration effort. Furthermore, there remain many undeveloped aspects of AI technology. It is necessary to try to fill these gaps.

AI is actually a highly specialized branch of computer science. AI programs are apt to get too big and complex. We are faced with software development problems. We expect the FGCS project to have a strong impact on the reduction of these problems. Our tentative approach is as follows: first of all, new programming languages, in particular, new declarative languages, should be introduced. It is well known that declarative languages have strong advantages for implementing highly-parallel AI programs. Secondly, sophisticated programming environments should be provided. Last but not least, automatic programming should be explored.

If we can succeed in writing big and complex programs, the machines that execute them will need to have very high performance. It is estimated that the machines require a 1,000-fold increase in their performance rate in order to do AI jobs rather than solving so-called 'toy problems' The introduction of highly parallel architectures is the way to solve this problem. It goes without saying that these architectures must be adaptable to the newly introduced programming languages, and vice versa. This approach will be implemented using VSLI technology, which has evolved rapidly in recent years.

3. PLAN OF FGCS PROJECT

It is very risky to drastically change a computer's architecture and language at the same time time. Moreover, there are a lot of unknown factors to be investigated. Therefore, it is desirable to proceed using step-wise refinement, that is, to repeat trial-and-error cycles as many as possible during the course of development.

The FGCS project, beun in 1981, will span ten years, and is devided into three stages: the initial stage (three years), the intermediate stage stage (four years), the final stage (three years). Each stage will span constitute one trial-and-error cycle in which a basic language and its accompanying architecture will be developed. The initial stage has concentrated on the development of basic concepts and technologies as well as the implementation of development tools that will be used in the subsequent stages. By the way, the major tool developed so far is a super-personal computer system, which constitutes the first step toward the creation of an overall logic programming environment for the project

The intermediate stage will be mainly devoted to improvéng and extendingtthe results of the initial stage, and integrating them into inference and knowledge-base subsystems. In addition, research and development in this stage will focus on establishing computational models well-suited to highly parallel processing, and that will promote knowledge-based programming. The hardware subsystems to be built will consist of about a hundred processing elements. In this stage, severalaapplication systems will be developed as experiments to check the soundness of the basic approaches in the R&D efforts.

The final stage will emphasize the optimization of both software and hardware system functions, and will involve final determination of the architecture for a full-scale fifth generation system. This full-scale system will consist of about one thousand processing elements.

4. SYSTEM OVERVIEW OF FGCS COMPUTER

The design philosophy of the FGCS computers is a language-based approach, that is, an approach to computer development that starts with a definition of a programming language and uses knowledge of the language structure to provide design specifications for the system components. We believe this is the vest way to free ourselves fro the constraints imposed by Neumann architectures.

The core programming language for FGCS computers is a logic programming language that is an extension of PROLOG. This is called the Kernel Language (KL) and serves as the machine language of the FGCS computer. Alogic programming language has many desirable features, that is, it is declarative and convienient for knowledge representation and problem solving, moreover, it is amenable to parallel processing.

The entire software system for the FGCS computer willevolve from the language. The system is divided into two parts: the basic software system, which corresponds to the operating system of a conventional computer; and the basic application systems. The development of actual applications is outside the range of the project. The basic software system consists of four components, each incorporating problem-solving and inference techniques, a knowledge-base, a natural-language interface, and semi- or full-auttomatic programming. The basic application systems will be used in bench-mark testing and will involve typical AI goals, such as machine translation, expert system, speech & image understanding, and advanced OA.

The entire hardware system will develop top-down under the ... kernel language (KL). The KL is base on predicate logic and its execution mechanism is inherited from logic proof procedures, that is, inference. The parallel inference machine will execute KL programs directly in parallel. Its basic operations are not arithmetic but syllogstic and its architecture is founded on the data-flow or reduction schemes. The knowledge-base machine corresponds to the database machine in a conventional system, but it will have a drastically different structure. In addition, the speech and image processing hardware will be included.

5. MAIN ACHIEVEMENTS IN INITIAL STAGE

The focus of effort in the initial stage has been on establishing the foundation of the FGCS computer, that is, on developing a basic programming language and a high-performance personal PROLOG machine to serve as a software development tool in the subsequent stages. But, leaving thes aside for present, let us surveythe overall situation.

Generally speaking, the progress of the project has been more rapid than expected in the R&D efforts that have taken a practical approach, and rather slow in those that have chosen a more sophisticated course. Also, the project involves many difficult research themes for which we have not yet found any definite solutions.

In the R&D effort on the parallel inference machine, four alternatives have been studied, among which a machine module based on the data-flow scheme is now under circuit design. In addition, several simulation programs have been developed and are being used to collect design data and for design evaluation.

As for the knowledge base machine, development of a relational database machine was proposed to develop the basic techniques necessary for developing a knowledge base machine in the intermediate stage. The relations in the relational database can be regarded as logic language expressions. The database machine, called Delta was completed in this spring.

In the R&D for the problem-solving and inference software, experimental implementation is be ing applied to three capabilities for increasing inference performance. these are: a parpllel inference capability that will take full advantage of a parallel execution environment; a meta-inference capability to manage the reasoning process using various types of control information; and a theorem-proving capability to enhance deductive power to the level of higher-order logics.

Regarding the knowledge base management software, alarge-scale relational database management system called KAISER, and a preliminary knowledge representation system based on logic, are being developed. The KAISER system is a logicdatabase system with a deductive question and answer facility. It differs decisively from conventional database systems in that it will have a knowledge acquisition capability.

The intelligent interface software consists of three programs, that is, a high-level syntactic analysis program, a semantic analysis program, and a dictionary system pilot program. The syntactic analysis program is a kind of a generator system that generates a bottom-up parser corresponding to a given grammar descrived in definite clause grammar. This program is already at work. The semantic analysis pilot program enables the computer to understand natural language. In this program, LFG and Situational Semantics are used as the linguistic framework.

The intelligent programming software module contains two programming systems. The advanced modular programming software will provide an advanced program development environment for object oriented logic programming. This software is now being debugged. The advanced verification system will support the user at each stage of programming, making use logic programming language formalism.

6. FGCS PROGRAMMING LANGUAGE

The basic programming language plays a central role in the development of the FGCS computer. Therefore, the design of the language requires the closest attaintion and needs as many trial-and-error cycles as possible. According to the project plan, three logic programming languages will be designed at the machine-language level, and two or three at the system-description-language level through the initial and intermediate stages.

KL0 (Kernel Language version 0) is the machine-language of the sequential inference machin, PSI. It is based on PROLOG with various extensions and some deletions. The main extentions are :

- -Extended control structure
- -Process switching
- -Operations with side-effects, and
- -Hardware-oriented operations.

Deleted features include:

- -Database management, and
- -Name-table management.

The features omitted in KL0 are supported in the system-description language, ESP. All the software for the PSI machine will be written in ESP.

While KL0 and ESP, as direct descendants of PROLOG, are sequential languages, Concurrent PROLOG's descendants, KL1, Mandala and KL2, are highly parallel languages. KL1 and Mandala are currently under development, the results of which will be applied to the KL2 design in the intermediate stage.

ESP is designed for ease of writing systems software and is based on Kahn's thesis that a hybrid of object-oriented programming and logic programming is a strong alternative to a language based upon either concept alone. The ESP program is compiled into KL0.

KL1 provides enhanced parallel programming functions that have been proposed in Concurrent PROLOG and PARLOG. In addition, the concept of set expression is introduced to enable representaion of all solutions that satisfy a goal and to achieve higher-order expressions. Facilities for modularization support and meta-inference are also added. Mandala is to KL1 what ESP is to KL0.

7. SOFTWARE DEVELOPMENT TOOL (SIM)

The initial-stage has concentrated on the development of basic concepts and technologies as well as the implementation of development tools that will be used in the subsequent stages. The major tool developed so far is a high-performance personal computer system, which constitutes the first-step toward the creation of an overall logic programming environment for the project. The computer system is called SIM (Sequential Inference Machine), and consists of a minicomputer-scale von Neumann-type machine called PSI (Personal Sequential Inference Machine), and advanced software called SIMPOS (SIM Programming and Operating System). SIM will include a local area network system (LAN) to connect many PSIs in a distributed system.

PSI provides an execution speed comparable to that of DEC-10 PROLOG running on the DEC 2060. Memory capacity will be up to 16 M words (40 bits/word). PSI will be equipped with new I/O devices common to most high-performance personal computers, such as a bit-mapped display, a mouse, and a LAN interface. The relational database machine Delta is connectable to SIM, either via a local area network or directly.

Eight PSIs are currently in operation at ICOT.

The SIMPOS system has a hierarchical structure consisting of four layers. The layers, in order of their proximity to the hardware, are the resource management programm(supervisor), the I/O system, and the programming system. Each layer consists of several subsystems or modules. The functional contents of the subsystems or modules are similar to those of an advanced programming and operating system, such as those in the LISP Machine software environment.

The first version of the SIMPOS system is now being debugged Almost the entire system is written in ESP. Exceptions are a few short runtime-subroutines. It is too early to assess performance. However, it may be said, based on our experience in writing a full-scale system, that ESP has sufficient expressive power for systems programming.

8. PROPECTS FOR R&D IN THE INTERMEDIATE STAGE

In the intermediate stage, experimental small-scale FGCS computer s subsystem will be developed on the basis of the basic technological achievements of the initial stage. Through this development process, the viability of previous achievements will be evaluated. Moreover, the fundamental research items taken over from the initial stage will continue to evolve and will brought to fruition. A number of neww R&D items will be added in this stage including the development of preliminary application systems such as a machine- translation system and some kind of expert systems to test the performance of basic experimental software and hardware. Advanced development-support tools will also be developed, including a small-scale parallel machine that can be execute KL1 programs and a development environment consisting of a database and a computer network.

The R&D effort for the Kernel Language will focus on the development of the KL1 high-performance processor and the specification of the KL2 design. KL2 is characterized as embedded with " equality " and knowledge-base management support facilities. In addition, a programming support environment will be built with specia attention paid to a facility to realize highly parallel KL1 programs.

As to hardware, a parallel inference machine with about 100 processing elements will be developed to execute KL1 programs directly An architecture for large-scale parallel machines (1,000 processing elements) will also be investigated. The parallel configuration of the relational database machines will be combined with SIMs to evaluated their usefullnessas parallel knowledge base machines. Support mechanisms for a distributed knowledge base and an architecture for a large-scale knowledge base machine will be studied as well.

Many R&D software items will be pursued under the four main subjects, that is, problem-solving and inference software, knowledge-base management software, intelligent interface software,and intelligent programming software.

The focus of problem-solving and inference software will be on establishing the method of problem-solving in a parallel processing environment, on investigating the mechanization of inductive and analogical inference, learning, etc., and on building a pilot system for cooperative problem-solving involving multiple experts.

In knowledge-base management software, effort will be directed to the development of a knowledge representation and utilization system and a distributed knowledge base management program, and to the investigation of knowledge acquisition.

For intelligent interface software, research on natural language processing is focused on semantic analysis and, partly, on context analysis. In addition, language databases, such as dictionaries, will be compiled after the initial-stage effort and a preliminary natural understanding system willbe constructed utilizing them.. Moreover, construction of discourse models willbe attempted in order to realize mutual-understanding between man and machine, aiming at implementation of a discourse system capable of handling speech and images as written materials.

Intelligent programming software will focus on designing a specification description languagc, with an eyes to uniting the formal approach based on logic and the more natural approach to human language and images. As to design support facilities, the rapid prototyping approach has been chosen as the core of planned design support facilities. A prototype support system will be developed. Part of the R&D effort will be aimed at advanced modular programming, providing a database consisting of component programs and an expert system to aid in progamming.