REVISITING ORIGINAL PHILOSOPHY OF FIFTH GENERATION COMPUTER SYSTEMS PROJECT

Kazuhiro Fuchi
Director
ICOT Research Center
Tokyo, Japan

Looking back, it was three years after the last conference in October, 1981 and two and half years ago we embarked on our Fifth Generation Computer Systems (FGCS) Project. The various experiences we have had in the past two and a half years reinforce our confidence in the basic concept we formulated at the start of the project.

Underlying our concept is the recognition that a new phase of computer history is coming. The project represents our commitment to prepare for the new era. Many people have pointed out that there is a limit to the current framework of computer technology; what we want to point out is that the possibility of a new technological framework is in sight.

The conceptual framework we proposed is a possibility and a hypothesis. To prove that it is viable is the aim of the project.

Needless to say, this framework does not have the support of all people since it is a hypothesis. But we cannot wait until we have the support of everyone before we begin preparations for a new age. At the same time, we must realize that if the basic concept is wrong, we will easily lose the confidence of people.

I am happy to say that, as we see it now, our own research experience in the past two years or so is endorsing and reinforcing our basic hypothesis. Furthermore, these three years have seen a growing understanding of our basic concept all over the world.

It is true that our proposal made a sensation throughout the world. This was presumably because it basically appealed to people as something presaging a new era, though partly misunderstood. The fact that it did not end in a mere short-lived sensation but has led to the start of similar projects and new research efforts in various parts of the world seems to corroborate that our basic concept is right.

We explained our conceptual new framework for computer technology at the last conference and on other occasions, but let me recapitulate it now.

The conceptual framework is, in a nutshell, intended to reconstruct the body of computer hardware and software on the basis of a logic system called predicate logic. In other words, it assumes the next-generation computer as a predicate logic machine. We might call it an inference machine because the basic operation of predicate logic is inference.

Current computers have a language called a machine language. The machine language defines the architecture of a computer. And software is basically built on the machine language. The characteristics of so-called von Neumann type machines are intensively represented in the machine language.

In our conceptual framework we call our machine language equivalent the kernel language. We contemplate formulating a predicate logic-based language, as this kernel language, or a new machine language.

Assuming a new machine language level, we naturally need new hardware and software architectures.

The hardware will be basically designed for parallel operation or associative search since its basic function will be inference. Present von Neumann type computers are basically built for sequential operation and address search. Therefore, a parallel, associative inference machine will be a sort of non-von Neumann computer.

On the other hand, the software will be built by combining the basic inference functions provided by the hardware.

Our kernel language, a sort of nonprocedural language, falls under the category of very high-level language in the current technological framework. Since it will become a machine language, we can start from a far higher level than at present in software construction. The capability of the language will be utilized to achieve sophisticated functions such as knowledge information processing and natural language processing.

I would add here that the kernel language is not a user language. Higher-level languages than the kernel language will be formulated one after another as user-oriented languages. What are now called knowledge representation languages could be positioned at that level, though the ultimate form of high-level user languages will be natural languages.

I have given a very brief description of our conceptual framework.

Our concept including the assumption of a predicate logic language as the kernel language is a hypothesis, a working hypothesis for carrying on the project. But it is not a random choice. It is a vision based on an overall analysis of the past and future progress of research and technology in the information processing world.

I explained this in detail at the last conference. Subsequent developments do not seem to indicate any need to revise our concept basically. Rather they may be said to reinforce it.

The research fields we addressed are broadly divided into artificial intelligence, software engineering, architecture, and supporting devices. Also studied were the social image and needs of the future.

As for the field of artificial intelligence, our project was predicted on the prospect that applications of artificial intelligence would become the main stream of future information processing. That is, artificial intelligence will not only create unique application areas but have a close bearing on the sophistication of conventional application areas.

Knowledge engineering and expert systems as its applications are emerging. We see this as a presage of the future.

The commercial business revolving around applications of artificial intelligence, notably expert systems and natural language processing, has been in the spotlight these few years. But we have no intention to get involved in the artificial intelligence business. The present artificial intelligence business concerns current-generation computers. Even if it makes economic sense for the time being, it will reach its limit before long.

We do not mean to ignore the significance of the artificial intelligence business. Rather our position is that if it is to achieve healthy growth in the future, it needs a new improved technological base. And the experience in applications of artificial intelligence accumulated until such base is established will contribute toward its growth in the future.

By the way, our project is not an artificial intelligence project or an expert system project as wrongly understood by some people. Artificial intelligence research, which is aimed at clarifying the mechanism of intelligence, is an open field requiring a very long period of sustained effort. Expert systems also offer unlimited potential for application. No project can cover all of them.

But our project is closely related to the fields of artificial intelligence and expert systems. The selection of the kernel language is also deeply related to the problem of knowledge representation languages. So, one of the reasons for choosing a predicate logic language as the kernel language is that predicate logic is a promising, if not the only, candidate for a knowledge representation base.

One of the research fields supporting our concept is software engineering. Improvement of software productivity is one of our greatest concerns. But we have no intention to deal directly with current-generation products such as ADA and UNIX, because our basic attitude is to free our thinking from the constraints of the present technological framework.

From a new standpoint we will be able to see the way clear to build on the past results of software engineering. In that sense we may call our project a software engineering project for a new age.

In software engineering, functional and object-oriented language loom large among predicate logic languages. As we see it, these languages can be organically integrated on the basis of a logic type language. Building an intelligent programming system on that basis is one of the primary objectives of the project.

It will integrate specification, verification, transformation, synthesis, debugging and so forth from a new standpoint. We may say it agrees in future direction with the concepts of rapid prototyping and fourth-generation languages now being addressed in software engineering.

In the field of software technology, it is desirable to organically integrate the programming world and the database world. The relational data model is a promising direction, and it is based on predicate logic.

A higher-level data model is also being studied on the basis of the relational model. It is approaching the problem of knowledge representation. We might say there is a major trend toward integration of software engineering research, database research and artificial intelligence research, which were conducted by separate groups for some time.

Our project is more than a software engineering project for the next generation. One of the lessons learned from software engineering, among others, concerns the present hardware architecture. To ensure that new ideas emerging in the software field become established, it is desirable to support them with hardware. Rather they will work only if they are combined with hardware.

New research in computer architecture, especially research in parallel machines, has come to be linked with functional languages and relational languages, that is, predicate logic and object-oriented languages.

Such architectural research is led by software concepts on the one hand and supported by forecasts of advances in device technology on the other.

Our basic concept is intended to grasp the progress and future direction of research in the information processing field and thereby organized them into a unified whole.

What I would like to stress here is we must have the overall picture and the basic philosophy behind it. In each of these various fields, there can be multiple options and solutions. But if looking in the perspective of overall picture, possible options or answers swill be much fewer. If we can see no philosophy to support the whole picture as a whole, it means mere continuation of the present framework.

Our argument is for the possible existence of a new framework. As long as it remains a possibility, there can be arguments against it. And different frameworks than ours may be proposed. For now, however, we may say there is no alternative presented to our conceptual framework as a whole, though there is much criticism of its details. While it is all right to search for alternatives, it will also be necessary to make efforts to verify the direction we believe we have found.

As you may know, the current technical arguments center around the kernel language. I would like to discuss this a little more.

One argument, though based on misunderstandings, concerns the positioning of the kernel language. It questions the appropriateness of the kernel language as a user language, because Prolog is currently used as a user language. But we position the kernel language as a machine language.

Another argument, also based on misunderstandings, maintains that we have chosen Prolog as it is and that this has its weaknesses and limits. From the outset we have clearly stated that while Prolog, a predicate logic language, is worth studying as a starting point, it needs to be improved and expanded.

It is as a working hypotheses for establishing a new technological framework that we are assuming a predicate logic language as the kernel language. It is based on our belief but is not a definitive conclusion. Neither Prolog nor our current kernel language is a finished product. But it is our thinking that a working hypothesis is effective if it is clear.

It seems that people criticizing something as a finished product have the wrong idea that our project aims at making commercial products in the near future. We are rather looking at possibilities for the future.

The criticisms of our kernel language include such resonable questions as whether there is no better alternative to it and whether a predicate logic language has the power to support future commercial-scab software.

Among the candidates for a new language are functional and object-oriented languages. You will see from my speech at the last conference that we have studied these types of languages for a long time. Our position was that the advantages of these languages would be organically integrated into a predicate logic-based language. Needless to say, it was also a hypothesis.

Besides, we stated at the out set that it would be necessary to expand the language for high-level inference and problem solving and to develop a higher-level language.

Now let us review these opinions or misunderstanding in light of the progress made in the initial stage of our project.

As you are aware, we have developed sequential inference machines as tools for research in the intermediate stage and beyond. One of them is PSI.

We designed KL0 as a machine language for sequential inference machines. PSI is a KL0 machine. Basically, KL0 may be said to be an expanded version of Prolog. But it is a machine language. So we designed a more user-friendly language called ESP. This language incorporates modular and macro functions, among others. It may be called a macro assembly language for KL0.

ESP naturally realizes functional and object-oriented notations using its macro function. It may not be a final solution, but we believe it is an answer to the initial hypothesis.

The operating system for PSI is called SIMPOS. All SIMPOS programs are written in ESP. One criticism of a predicate logic language concerned the question of whether it could be used for writing control programs like operating systems. Our experiment will provide an answer to that question. Besides, I would like to point out that SIMPOS is large-scale software.

There are expected to be numerous difficulties in writing a new operaing system in a new language. But the use of ESP has proved effective in improving the productivity of software construction as well as system efficiency.

SIMPOS is still under development, and no large application program is yet to be run on it. Although I am not able to say anything definite at this stage, I may say that our experience with ESP and SIMPOS provides data to respond to much of the criticism to date.

KL0 was designed immediately after the start of the project. Therefore, it reflects the constraints involved in sequential execution.

To prepare for activities in the intermediate stage and later, we have studied KL1 as a language incorporating parallelism and other improvements over KL0. Concurrent Prolog, which emerged as an expanded version of Prolog, has been a great help to our KL1 design effort. Concurrent Prolog has potential for object-oriented programming and parallel event simulation based on stream parallel programming.

Incidentally, predicate logic languages including Prolog are based on a predicate logic subsystem called the Horn clause. This is a sort of restriction. While it is desirable to place a sort of appropriate restriction on the kernel language as a machine language, it alone is naturally insufficient from the standpoint of knowledge representation or semantic representation. Research in the direction of its extension is also necessary.

But it is desirable not to move it back to normal first order predicate logic or high-order predicate logic but to seek a different structure. From the outset we have pointed out the possibility of a multi-level structure incorporating something like a meta-inference function. We have conducted research on these lines for the past two years, and are reflecting the results in the KL1 design.

It may be in order to respond to the criticism that we use the term inference too easily. It stands to reason that inference in general is highly complex and involves operations yet to be clarified. When we say inference machines, for instance, we mean basic inference such as is dealt with in symbolic logic. Nevertheless, it is also inference. Maybe the question is whether a combination of such basic inference operations makes possible sophisticated inference, such as, for example, induction.

Though this depends on philosophical standpoints, our position is that induction is also complex inference and, if analyzed, is reduced to a combination of basic inferences. Of course, this is also a hypothesis and, therefore, an important research theme.

Among the questions often discussed in connection with the kernel language is whether to use Lisp or Prolog. As stated earlier, we do not feel uncomfortable with Lisp in the sense that it is a functional language. We believe that the two languages will be organically merged in the future, possibly overnight.

But the characteristic of this sort of discussion seems to be that it is based on the view that the kernel language as it exists now is a finished product. We are not directly-committed to arguments over which of the two languages will provide a better basis for commercial systems of current-generation computers in the near future.

Let me reiterate that our project is aimed at building a new framework. Our standpoint is based on what is effective as a working hypotheses for that purpose. From this standpoint our project is pursuing our choice; we do not intend to force this on similar projects in overseas countries. Nor can we do so. We wonder if any criticism based on misunderstanding of this basic position of ours is valid. Of course, constructive criticisms based on mutual understanding are important and useful. We believe we have attentively listened to them. For the world as a whole, it is desirable to make efforts to explore various possibilities, because they will complement each other to accelerate the arrival of a new age.

Through the experiences of the past three years, we have deepened our faith in the basic concept and framework of the project. Of course, this conference marks only the end of the initial stage of the project, and it is as yet too early to announce any conclusions; but I believe we are capable of saying this much.

What we have been engaged in throughout the past two years and more may be described as preparation for the full-scale research of the project in the intermediate stage and beyond.

These preparations include not only basic research, but also the construction of tools for use in research in the intermediate stage.

These "tools" are not simply tools; incorporated in them are the basic concepts of the project, for use in design.

The PSI sequential inference machine and DELTA relational database machine, developed in the initial stage, may be regarded as such tools. This development was, at the same time, a reaffirmation of the basic concepts we have adopted. These results may be witnessed here at this conference, and also through demonstrations held within the Institute.

Together with these results, I would here like to note the vigorous research activities underway not only in Japan, but in other countries as well. These activities grow with each passing year, bringing forth a variety of technical achievements. Of course, these efforts lie outside this project, but many of them fit well into our project framework. For us, this lends powerful support to our own work; taking a wider point of view, however, this may be seen as the beginning of a movement to a new age. Our project may then be ranked as one part of this broader movement.

Based on the results gained in the initial stage, the intermediate stage of our project will be initiated next year. This presupposes a reaffirmation of the basic strategy which we have established.

In the intermediate stage, we have planned more challenging research themes than were attempted in the initial stage.

Parallel inference machines and knowledge base machines are themes which will be taken up on fuller scale in the intermediate stage. In addition, work will begin with more emphasis on natural language processing and knowledge information processing. In this research, the tools developed in the initial stage are expected to be put to effective use.

In accumulating software to be run on sequential inference machines, attempts will be made to solve problems at the level of practical application. Through this, more people will gain an understanding of the approach we have adopted, thus we anticipate a new kind of culture in computer technology will emerge.

The approach we have put forward was meant to serve as a framework for future technology; it should not, however, act to constrain the diverse efforts being made in moving toward the future. Rather, it may be regarded as the key to a new world. This "new world" I speak of will offer us more freedom, more space in which to move. A large number of original ideas will be necessary; these ideas will not stand alone, but must be organically linked, to give shape to this new world. We believe that the present plan provides a viable framework for this.

Given a reaffirmation of our basic conception, what I would in closing like to stress is the problem of international cooperation in moving toward a new age. Our project is, in effect, an effort to create a new age for modern man. This in turn signifies that our project must not be exclusive or closed to outsiders. What I have called "the creation of a new age" is not possible for a single nation, through a single project. The new age will be brought about by the will of the people of the world to move into the future, and by the cooperative efforts of these people. Nor must these efforts be the result of prodding or coercion. Cooperation must be based on spontaneous efforts toward a common goal.

Our conception has been evaluated by some as a bold hypothesis. The inauguration of a project based on such a bold hypothesis is a first for Japan. But for Japan, which has developed its prosperity with the assistance of a number of other nations, contributions to the world community are possible only by efforts which entail a certain amount of risk. It is our fervent hope that, once this is understood, the circle of cooperation will spread throughout the world. For my part, I wish only that this conference may serve as a medium for fresh exchanges of ideas, and that it will lead to greater future development.