# PROCEEDINGS OF INTERNATIONAL CONFERENCE ON FIFTH GENERATION COMPUTER SYSTEMS

October 19-22, 1981

FGCS INTERNATIONAL CONFERENCE

JAPAN INFORMATION PROCESSING DEVELOPMENT CENTER

*Printed by SEIBUNSHA, Co., Ltd.*

8

# TABLE OF CONTENTS

# I KEYNOTE SPEECH

## Challenge for Knowledge Information Processing Systems (Preliminary Report on Fifth Generation Computer Systems)

T. Moto-oka et al.

# 1. SUMMARY

## 1.1 Background and significance

(1) Social requirements expected of computers in the 1990's

In the 1990's when it is expected that fifth generation computers systems will be in wide use, information processing systems will be central tools in all areas of social activity to include economics, industry, art and science, administration, international relations, education, culture and daily life and so forth. Such information processing systems will be required to meet those new needs generated by environmental changes and will not only be expected to play active roles in the resolving of anticipated social bottlenecks but also to advance society along a more desirable path through the effective utilization of their advanced capabilities.

Information processing systems in the 1990's will be expected to play the following roles:

1) To increase productivity in low-productivity areas.

Although product quality and productivity in the secondary industries have been greatly improved through the adoption of computer controlled manufacturing processes and assembly lines, productivity in the primary industries such as agriculture and fishing and also in the tertiary industries such as goods distribution and public services, has remained little changed. This fact has been the cause of serious social imbalances. Cost reductions via increased efficiency as represented by industrialization and office automation can be expected in these fields as well, as a result of the effective employment of advanced computer systems.

2) To meet international competition and contribute toward international cooperation.

Suffering from a shortage of land and natural resources, it is impossible for Japan to be fully self-sufficient in food, and her ability to supply her own energy and oil needs is the lowest among the developed countries. On the other hand, we do have one precious asset, that is, a highly educated, diligent and top quality labor force, our human resources. It is desirable to utilize this advantage to cultivate information itself as a new resource comparable to food and energy, and information – related knowledge – intensive industries should strongly be promoted to make possible the processing and management of information at will.

Such an effect would not only serve to help our country meet international competition, but would also enable us to make international contributions through knowledge-intensive technology.

3) To assist in saving energy and resources.

One of the most important tasks facing mankind in this century is how to use our worlds finite resources effectively. Paralleling the realization of minimization and optimization of energy consumption, improvement of energy conversion effficiency and simulators for use in developing new sources of energy through the use of computer technology, even the industrial system itself could be expected to change into a knowledge-intensive type information industry which would be typically non-energy consuming.

4) To cope with an aged society.

Our society is aging at an unprecedented rate. Rapid increases in medical expenses and welfare costs together with the relative reduction in the labor force resulting from this aged society could lead to big social problems. Accordingly, utilization of fifth generation computers to prevent the occurrence of or to cope with such problems by way of developing streamlined medical and related information systems, health management systems and lifetime education systems for the aged etc. will be necessary.

(2) Technological background

Computer technology has, from its birth on, consistently and emphatically been aimed at high-speed operation and large capacity, and has been developed mainly for processing numerical calculations. As a result, computers have had significantly limited functions in terms of input and output processing that restricted their applications, and this has caused considerable inconvenience. As applications for computers have become widespread, from the initial scientific and technical computations to the more recent business data processing, there has arisen a strong need for freer input/output capabilities such as by speech or voice, images, graphics and the like, all of which are natural forms of information transmission for man.

The high cost of hardware up to now has not only minimized the number of functions capable of being carried out by it, but has also gradually increased dependence upon software, the proliferation of which has led to a situation called the "software crisis". This problem has had an undesirable side effect in that computer architecture has become stiff and inflexible due to the continued reliance on existing software and is believed to be unable to meet new applications as long as it continues to rely on existing technologies.

From the standpoint of seeds for the development of new technology, a technological basis permiting new architectures and new functions such as improved computer intelligence has matured. This includes VLSI technology which has rapidly advanced in the past few years, the realization of larger capacity memories, increased possibility for developing high-speed elements, promotion of research into artificial intelligence and pattern recognition technology and the technological fusion of communication and information processing among others.

Judging from the relationship between needs and seeds, it is quite natural to hope that information processing systems based on new conceptions and architectures which would prove to be a quantum leap in the computer technology of the past thirty years will appear in about ten years.

(3) Significance of the project

Japan has come to be considered an "economic power" by the other countries of the world. Thus, if we consider the direction in which our industries should proceed, it becomes clear that we no longer need chase the more developed countries, but instead should begin to set goals of leadership and creativity in research and development and to pioneer the promotion of such a project throughout the world.

The significance and effect of the research and development project for fifth generation computers is summarized as follows:

1) By promoting this project, Japan is playing a leading role worldwide in the field of computer technology development. This effort will not only help our computer industry foster more creative technology, but will also provide our country with a means of bargaining power. We can also fulfill our duty as an economic power expected to assume international responsibility by investing in the development of this leading field.

2) In addition to making our society a better, richer one by the 1990's, this project will also prove influential in other areas.

The fifth generation computer is expected to be beneficial in the solution of social bottlenecks such as the energy problem and problems related to an aging society etc. It is also expected to serve as a prime mover in the field of industry by helping those industires experiencing difficulty to improve their efficiency and thus their productivity. Further, society as a whole will become more affluent as computers are applied to increasingly new fields and areas.

3) Developments in heretofore unexplored fields will actively contribute to the progress of all humanity.

Through the promotion of research into artificial intelligence, a better understanding should be gained of the mechanisms of life which future research and development will then concentrate on. The realization of automatic translation into multiple languages will help promote mutual understanding between and among peoples of different tongues and thus aid in reducing trouble due to misunderstandings and ignorance.

With the construction of a knowledge base made possible, the knowledge which man has accumulated over the ages can be stored and effectively utilized, and new kinds of insights and perceptions can be more easily obtained by man with the aid of computers.

4) Experiments for advanced research and development organizations.

It is of great importance to conduct national tests of research and development organizations which have been in existence for a long period of time. The majority of national projects from now on will be required to be carried out by organizations for advanced research and development such as these. Thus tests of these organizations at the national level could be considered a kind of experimental project for future projects. Promotion of this kind of testing project will create an environment which will produce original research based around it.

## 1.2 Functional requirements

Fifth generation computer systems will be required to have an extremely wide variety of sophisticated functions to solve the numerous problems which todays computers have and to meet the social needs of the 1990's during which decade computerization is expected to find many more applications than nowadays.

As a whole, functions required of fifth generation computer systems will be as follows:

(1) Increased intelligence and ease of use so that they will be better able to assist man.

1) Functions which enable inputting and outputting of information via speech or voice, graphics, images and documents.

Enhancement of input/output functions which serve as the interface between man and computer is of prime importance in making computers easier to use.

In particular, since current computers are quite limited in their input/output functions, the ability to input and output information in a wide variety of forms such as speech or voice, graphics, images, documents and the like man's daily means of transmitting information will be of utmost necessity.

2) The ability to process information conversationally using everyday langauge.

As computers penetrate further and further into every field of our society, there will be more opportunities for laymen to operate them and thus gain direct access to needed information. Therefore, the ability to communicate conversationally with a computer using everyday language will prove most beneficial.

3) The ability to put stored knowledge to practical use.

The ability to handle information in speech, graph or natural language form does not end with the input function, but rather such a computer can only fulfill its purpose if backed up with the knowledge to comprehended that input information.

In order to be able to utilize computers more effectively as tools for solving various problems, they will have to be equipped with specialized knowledge i.e. knowledge bases, related to the fields in which they are employed. Then by putting these knowledge bases to practical use computers will be better able to lessen the burden on their human operators as well as serve a role as consultant systems for all mankind.

4) The functions of learning, associating and infering.

So that computers have knowledge and can sufficiently use it for a desired purpose, they should be given in one form or another abilities of learning, associating and infering just like ours. With such abilities, computers would be able to clarify even vague requests given by man and using their vast ability to store information achieve new judgement facilities of their own which will help expand the capabilities of we humans as well.

(2) Lessening the burden of software generation.

1) Automated processing based on the input description of requirement specifications.

The cost of the development of software is now greater than that of hardware, and there will be increasing needs for software in the future. In such a situation, it is necessary to raise the proportion of automated programming in software development.

For example, an ideal process is one where a computer processing procedure is synthesized directly from requirement specifications described in a natural language, generated and performed.

2) Realization of a language capable of program verification and a suitable architecture.

A programming language is a direct interface between man and machine in the development of a program. The ease with which the programming language can be used and language specifications functions greatly influence the degree of difficulty of software development. Moreover, it is desired that in future ultra high-level languages with a high degree of verifiability to enhance reliability of software appear, and a machine be realized which has architectures suitable for processing such languages.

3) Improvement of environments for programming and realization of intelligent interfaces.

To improve programming productivity, not only language would be improved, but also programming environments should greatly be improved to provide intelligent interfaces between users and systems. Such requirements are common in every form of computer access such as data base access for retrieving desired information out of a great amount of information, or a knowledge base access for gaining a new perception to cope with an unknown problem.

4) Utilization of existing software assets.

As computer functions and performance are improved, new applications will increase to the point where conventional computers will not be able to deal with them. However, it will be desirable to utilize software developed to date as much as possible. To this end, systems will be preferred which are flexible enough to run software based on conventional architectures.

(3) Improved overall functions and performance to meet social needs.

1) Improved cost/performance.

It is a common principle to all industrial products that technological progress improve the cost/performance of products. In the 1990's, the cost/performance of hardware and software combined should be improved significantly.

2) Light, compact computers.

Computers are expected to be lighter and smaller as the technology of integration of devices progresses. The 1990's should find portable, high-function computers, multilingual translating machines, and industrial products equipped with high-performance computers.

3) High-speed, large-capacity computers to meet new applications.

It can be said that demands on speed of processing and memory capacity as basic computer abilities are and will be limitless. Many problems that are considered unprocessable by conventional machines will appear as new applications as processing speed and memory capacity increase. Realization of the intelligent system, described above, is dependent on great improvements in these basic performances. With these improvements, unknown situations can be simulated with high precision to assist in widening our ability to solve problems.

4) Increased diversification and adaptability.

Up to now general-purpose computers with fixed hardware have been in the mainstream, but computer systems in the 1990's will be required to have much wider diversification and purpose-oriented adaptability and flexibility. Hardware and software both should have their basic components modularized for free system adaptability and rearrangeability to suit various purposes.

5) Highly reliable functions.

As computers find their way into more and more field area of our society, they are likely to cause much more damage when they get malfunction. Therefore, constructing highly-reliable systems is an absolute requisite for future society. Computer systems should not only be equipped with functions to automatically detect and repair their own malfunctions, but should also be capable of preventing the danger of a runaway computer by means of malfunction prevention devices and the aforementioned improved machine intelligence.

6) Sophisticated function for the protection of secrets.

Social computer systems will be largely expected to serve as social utilities and thus will have to be equipped with sophisticated functions for the protection of secrets. These systems will also be required to have built-in mechanisms for preventing computer crimes and unauthorized use of computers.

## 1.3 Objective and image

The Fifth Generation Computer Systems will be knowledge information processing systems based on innovative theories and technologies that can offer the advanced functions expected to be required in the 1990's, overcoming the technical limitations inherent in conventional computers.

### 1.3.1 Basic concept

The Fifth Generation Computer Systems will be knowledge information processing systems having problem-solving functions of a very high level. In these systems, intelligence will be greatly imporved to approach that of a human being, and, when compared with conventional systems, man-machine interface will become closer to the human system. Figure 1-1 shows a conceptual diagram of the Fifth Generation Computer Systems. As shown in the figure, a powerful problem-solving mechanism based on problem understanding and inference functions, knowledge bases, etc. are found between the human system and the conventional machine functions. In addition to that models will be realized principally through software, and the machine principally through hardware, and the Fifth Generation Computer Systems will perform the following functions as integrated capacities:

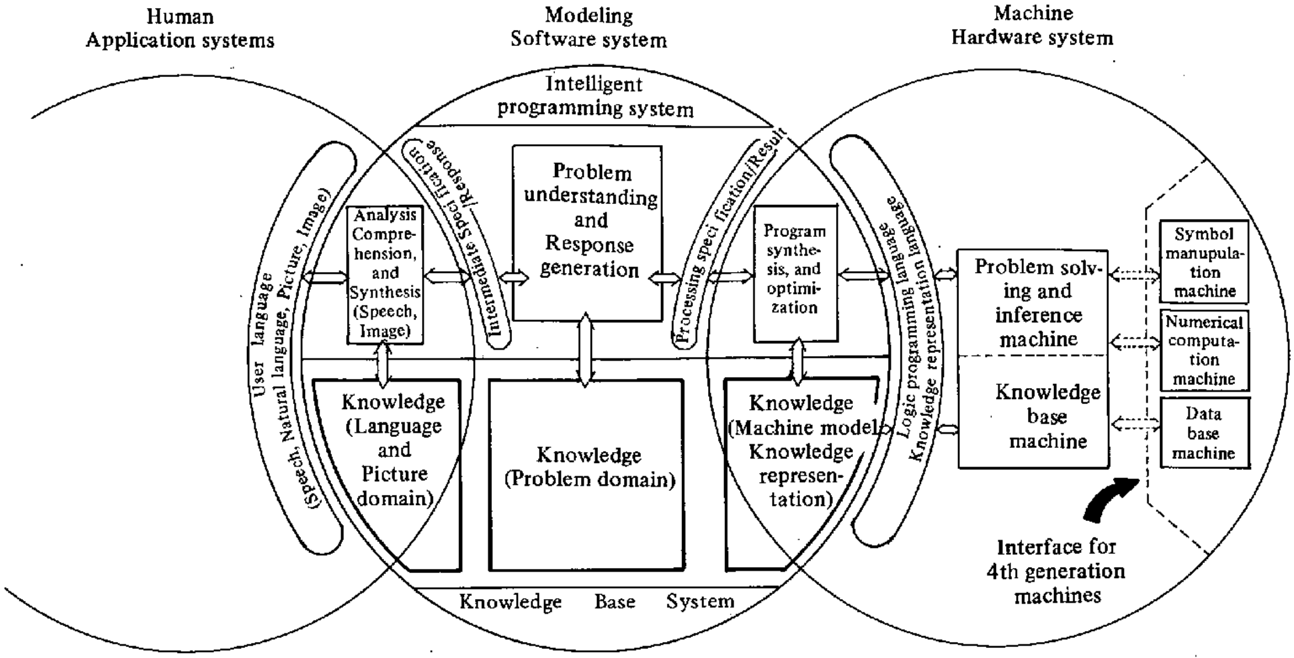



**Fig. 1-1 Conceptual diagram of the fifth generation computer systems**

1) Understanding of problem description and requirement specifications
2) Synthesizing processing procedures
3) Optimization between machine system and processing procedures
4) Synthesizing response based on outputs from machine system
5) Intelligent interface functions capable of understanding speech, image and natural language, etc.

The knowledge bases that support the above functions will contain knowledge of the following types:

1) Knowledge of the languages to be used for man machine communication
2) Knowledge on the problem areas to be solved
3) Knowledge on the machine systems

### 1.3.2 Constituent elements of software system

Figure 1-2 shows the system configuration image.

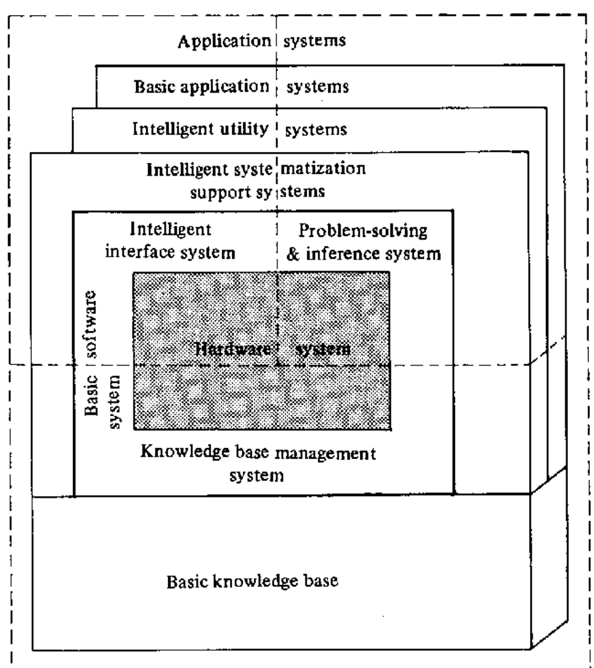



**Fig. 1-2 Configuration of the fifth generation computer software system**

These constituent elements perform the following functions:

(1) Basic software system

The basic software system forms the core of the software system, and consists of the three subsystems of Problem-solving and inference, Knowledge base management and Intelligent interface which are the basic functions of the Fifth Generation Computer Systems.

(2) Intelligent systematization support system

The system provides the human designer intelligent functions to strongly support systematization work based on the knowledge base contents and it consists of three subsystems of Intelligent programming, Knowledge base designing and Intelligent VLSI designing.

(3) Intelligent utility system

This system has such functions as to enable the user easy use of the entire computer system and make the system highly reliable.

These include programs to support the portability of software and database from other machines, user guidance functions, and automatic inspection and repair functions for the prevention and detection of failures.

(4) Basic knowledge base

The basic knowledge base supports the operation of the system itself in addition to containing the accumulated valid and universal knowledge necessary to the user. Generally, there are three types; the general knowledge base that mainly relates to the understanding of natural languages, the system knowledge base related to the system itself and the applied knowledge base containing specialized knowledge for various applications.

(5) Basic application system

The following types of basic application systems can be cited:

. Machine translation system
. Question-answering system
. Applied speech understanding system
. Applied picture and image understanding system
. Applied problem solving system

(6) Application systems

The following systems can be thought of as examples of knowledge information processing application systems:

. Intelligent CAE/CAD system(Note 1)
. Intelligent CAI system
. Intelligent OA system
. Intelligent Robot

### 1.3.3 Configuration image of the hardware system

The Fifth Generation Computer Systems should consist of all levels, from small to large scale machines, in order to process diversified applications. The machines of all performance levels must have common languages as well as the following three basic functions. The remarks in parentheses indicate the correspondence with conventional computer systems.

1) Problem-solving and inference machine (Central processing unit)
2) Knowledge base machine (Main memory with virtual memory facilities and file system)
3) Intelligent interface machine (Input/output channels and Input/output devices)

There will be machines of several performance levels in each of these small to large computer systems, to permit system configurations which emphasize any of the several

(Note 1)

| | |
|---|---|
| CAE: | Computer Aided Engineering |
| CAD: | Computer Aided Design |
| CAI: | Computer Assisted Instruction |
| OA: | Office Automation |

functions by application or purpose of use.

Figure 1-3 shows a configuration image of the Fifth Generation Computer Systems. The machines are to be structured according to function on various new architectures, including a data flow machine, which are based on VLSI architecture and each system is to be a combination of machines suitable for various individual applications or needs.

Furthermore, from a macro configuration point of view, having the system shown in the figure below as one of the principal elements, a multiple system form of usage where this would be connected to a local or global network and the whole network then be utilized as a large-scale distributed processing system, is also being envisioned.

## 1.4 Themes in research and development

Themes in research and development of the fifth generation computer systems are shown in Table 1-1, which contain seven groups and twenty six themes. The research and development are classified into four divisions as shown in Figure 1-4. The program will be affected at early, intermediate and final stages during which times achievements will be continuously re-evaluated with new trends in technology in view. It is planned to complete proto-types of the fifth generation computer system in the target year of 1990.

For the smooth accomplishment of this research and development program, it is indispensable also to develop and complete as soon as possible powerful support systems such as software-developing tools, high-function personal computers, VLSI-CAD, computer network systems, and the like.

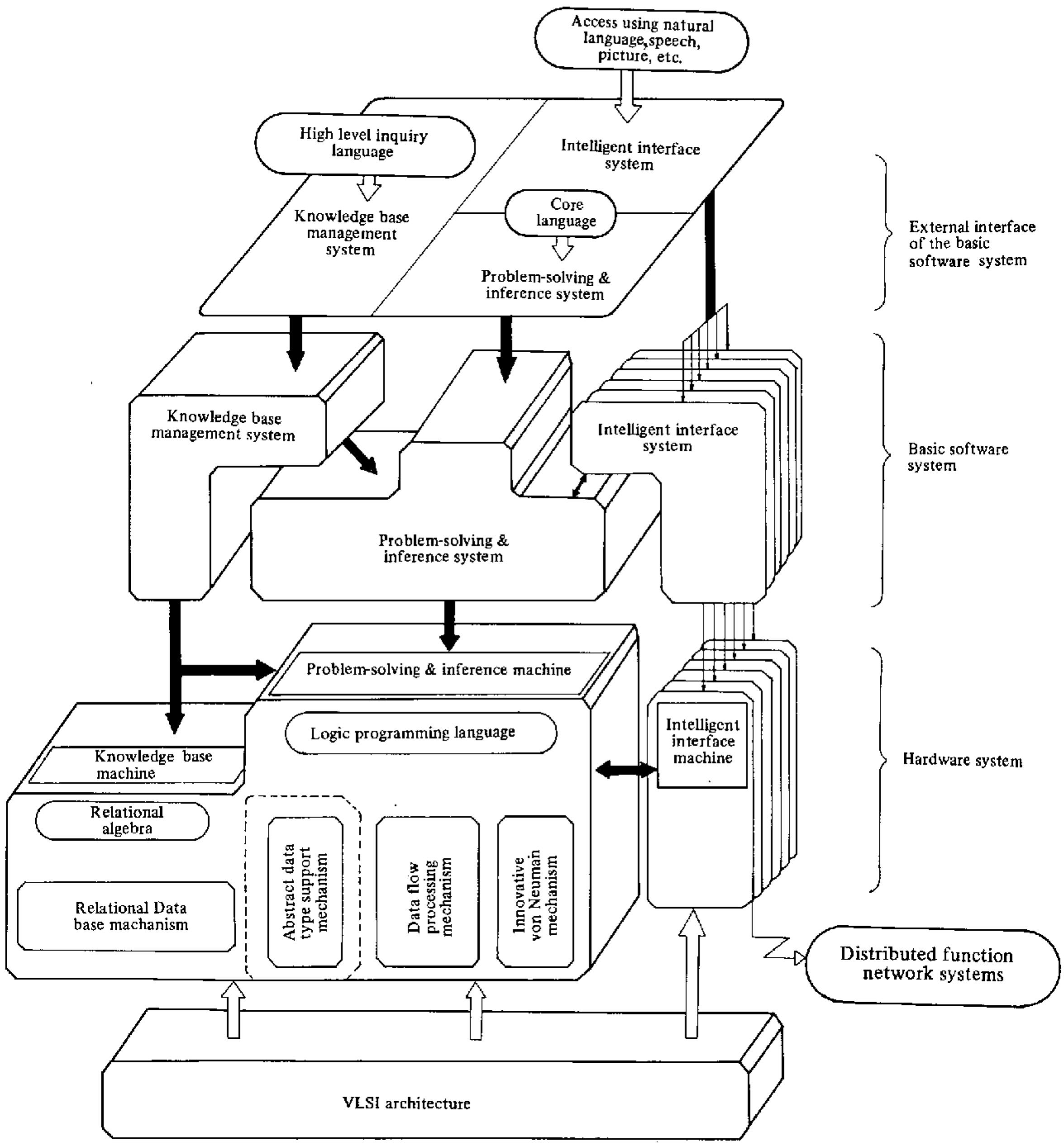



**Fig. 1-3 Basic configuration image of the fifth generation computer systems**

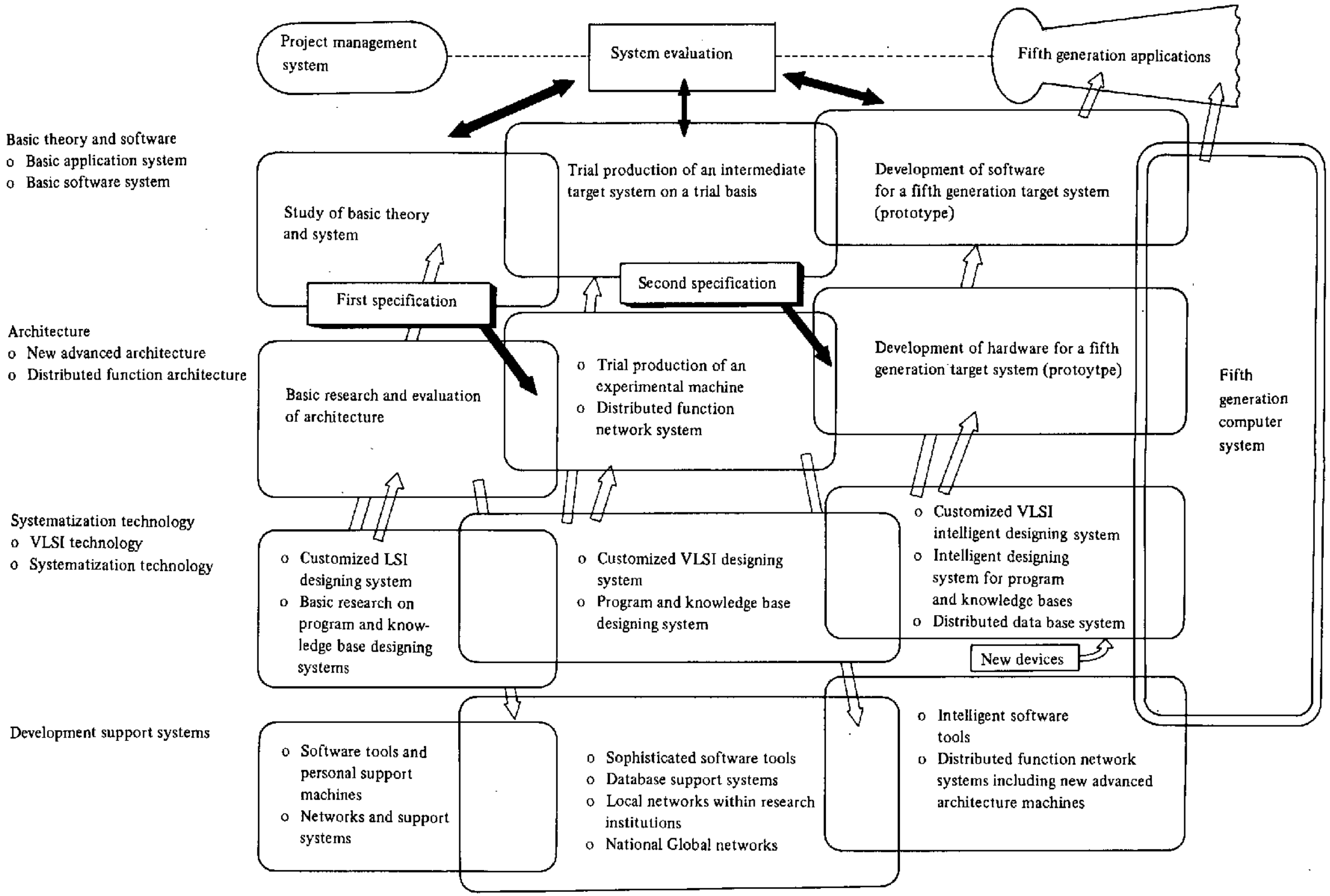



**Fig. 1-4 Concept diagram showing how research and dvelopment are to progress**

**Table 1-1 Themes in research and development of the fifth generation computer system**

| | |
|---|---|
| Basic application systems | 1–1) Machine translation system<br>1–2) Question answering system<br>1–3) Applied speech understanding system<br>1–4) Applied picture and image understanding system<br>1–5) Applied problem solving system |
| Basic software systems | 2–1) Knowledge base management system<br>2–2) Problem solving and inference system<br>2–3) Intelligent interface system |
| New advanced architecture | 3–1) Logic programming machine<br>3–2) Functional machine<br>3–3) Relational algebra machine<br>3–4) Abstract data type support machine<br>3–5) Data flow machine<br>3–6) Innovative von Neumann machine |
| Distributed function architecture | 4–1) Distributed function architecture<br>4–2) Network architecture<br>4–3) Data base machine<br>4–4) High-speed numerical computation machine<br>4–5) High-level man-machine communication system |
| VLSI technology | 5–1) VLSI architecture<br>5–2) Intelligent VLSI CAD system |
| Systematization technology | 6–1) Intelligent programming system<br>6–2) Knowledge base design system<br>6–3) Systematization technology for computer architecture<br>6–4) Data base and distributed data base system |
| Development supporting technology | 7–1) Development support system |

## 2. BACKGROUND AND SIGNIFICANCE OF THE DEVELOPMENT PROGRAM FOR FIFTH GENERATION COMPUTERS

### 2.1 Social requirements of computers in the 1990's

Our society is about to enter a transition period in various meanings of the term. It is an age of changes in internal and external environmental conditions such as the energy situation, and together with building a wealthy, liberal society, and overcoming the constrains of resources and energy, we must at the same time make international contributions as an economic power.

In making our way through this new age, informationization and the information industry which centers around computers are expected to play a big role. In 1990's when fifth generation computers will be widely used, information processing systems will be a central tool in all areas of social activity to include economics, industry, science and art, administration, interantional relations, education, culture, daily life and the like, and will be required to meet those new needs generated by environmental changes. Information processing systems will be expected to play an active role in the resolving of anticipated social bottlenecks and the advancing of society along a more desirable path through the effective utilization of their capabilities.

If we try to form an image of how society ought to be in the 1990's, information processing systems will be expected to play the following roles to achieve such a society:

(1) To increase productivity in low-productivity areas

Although product quality and productivity in the secondary industries have been improved through the adoption of computer controlled manufacturing processes and assembly lines, and a further improvement of productivity in the secondary industries can be expected with the use of industrial robots, productivity in the primary industries such as agriculture and fishing and the tertiary industries such as goods distribution and public services have remained almost the same. Examples of low-productivity areas are documents processing, office management, and decision making in management, and the ultimate purpose of office automation can be said to be increasing productivity in such areas.

Future images of office automation are:

1) Japanized office automation capable of processing the Japanese languages in a natural way;
2) Irregular or non-fixed job processing systems capable of freely handling non-numerical data such as documents, graphics, images, speech and voice, etc.;
3) Consultation and expert systems having inference and learning mechanisms of their own and capable of storing knowledge and providing adequate information as desired; and
4) Various data bases for providing high-level information necessary for decision making, and man-machine interfaces supported by artificial intelligence technology for making and supporting decisions.

(2) To meet international competition and contribute toward international cooperation

Japan, which has shortage of land and a population density about 40 times that of the United States cannot attain self-sufficiency in food, and her rate of self-sufficiency in energy is about 15% and that of oil about 0.3%. On the other hand, we have one precious asset, that is, our human resources, Japan's plentiful labor force is characterized by high degree of education, diligence, and high quality. It is desirable to utilize this advantage to cultivate information itself as a new resource comparable to food and energy, and to emphasize the development of information-related knowledge-intensive industries which make possible the processing and managing of information at will.

Such an effort would not only serve to help our country meet international competition, but would also enables us to make international contributions through knowledge-intensive technology.

We have to be internationally competitive and at the same time cooperative in the following subjects:

1) Construction and maintenance of various data bases;
2) Smoothing international exchanges through the development of translation-assisting and interpreting systems;
3) Improvement of productivity with the aid of intelligent robots; and
4) Accelerating research and development by using intelligent CAD systems.

Although we have mainly followed the head of other countries in computer technology up to now, it's time for us to break with this outmaded tradition, and center our efforts on the development of new computer technology based on our own conceptions, so that we can provide the world with new technology with a view to promoting international cooperation.

(3) To assist in saving energy and resources

One of the most important tasks facing mankind in this century is the effective use of finite resources. The information industry should cope with such problems by way of:

1) Minimization and optimization of energy consumption;
2) Improvement of energy conversion efficiency;
3) Simulators for use in developing new sources of energy;
4) Reduction of energy consumption in production through CAD/CAM*;
5) Extension of product service life through damage detection and automatic recovery; and
6) Reduction of movement of people through propagation of distributed systems.

Needless to say, the information industry itself is a typical knowledge-intensive non-consumptive industry.

(4) To cope with an aged society

People 65 years of age or older will make up 12% of the entire population of our country in 1990. Our society is aging of an unprecedented rate. Accordingly, a great increase in medical expenses and welfare costs together with a relative reduction in the labor force could become big social problems. Active contributions should be made to prepare for such problems by way of:

1) Improvement and streamlining of medical and realted information processing systems and helath management systems;
2) Development of systems for helping the physically handicapped get active;
3) Development of CAI system for the lifetime education of the aged; and
4) Development of distributed processing systems for enabling people to work at home.

As the society becomes more and more information-centered, computers and society will be related to each other in more complicated and diversified ways. Computers must be tools that can coexist with human beings. It is important to develop the information industry with meticulous care so as not to allow the rulers of countries to use computers as a tool for governing people and also not to let computers turn against mankind. Fifth generation computers should therefore be developed with a view to making them both usable and likable.

---

* CAM: Computer Aided Manufacturing

## 2.2 Problems with today's computer systems

Computer technology-has, from its birth on, almost consistently been aimed at high-speed operation and large capacity, and has features enumerated below:

(1) Computers are designed mainly for carrying out numerical calculations irrespective of whether they are of scientific use or business use, and have a minimized function to process character or image data and the like.

(2) Since the cost of hardware has been high as can be seen from sequential control of stored program systems, the basic design idea is that the functions are minimized and put to use at a high rate of efficiency.

(3) To improve cost performance ratio has been to centralize processing and make systems more huge, because high-speed operation and large capacity are indispensable.

(4) In order to operate a huge system and use it efficiently, the proportion of the cost of developing software has gradually been increased, resulting in unification of architectures.

From the standpoint of recent technological advancements surrounding computers and changes in user demands, distinguishing characteristics associated with conventional technology can be described as follows:

(1) Today's computers are not equipped with the necessary functions and voice to process non-numerical data such as sentences, symbols, speech and voice, graphics, and images, etc. However computers are expected to be developed which will be capable of associative and inference processing such as pattern matching functions which deserve the name of artificial intelligence. Computers such as these will be required to widen the areas in which information processing can be applied, diversify the forms of processing, and realize information systems that have a high level of intelligence. Compueters with new architectures are also hoped for which not only have increased processing capabilities, but also can put to practical use information management such as data bases and knowledge bases. Computers with new processing functions are desired to improve man-machine interface by developing easy-to-use computers capable of being good assistants for man and to effectively extend the range of the human senses.

(2) The performance of conventional computers has been increased largely through improvements made to their separate elements, and efforts to improve the system itself have thus far proven fruitless. However, since the high-speed operation of elements themselves has a limitation imposed by the speed of light, combined efforts from the standpoint of both elements and systems should be made from now on to improve the performance of computers of effort made thus far in terms of systems is parallel processing. This is not only essential for large-scale numerical calculation such as partial differential equations and for simulators for various systems, but is also needed for speeding up inference and associative processings. Various parallel control systems should be put to practical use which include proposed data flow control that is basically different from conventional sequential control.

(3) Due to the diversification of fields of application and advancements in LSI technology, the merits of distributed processing have come to be looked at in a new light. Distribution of processing can roughly be grouped into two categories. One is a regional distribution form in which processing functions and data bases are located near persons in charge so that various resources such as hardware, software, database and the like can be shared by many through communications lines. The other system comprises distributed functions wherein processors of different kinds designed for dedicated uses are con-

nected to each other via high-speed buses and the like, thus replacing a system having a relatively small number of processors of one kind connected to a common main memory. The former should serve as a means for realizing a huge information system designed from the standpoint of users, and the latter should be put to practical use as a means for realizing systems which meet diversified demands.

(4) The cost of software development is ever increasing, but many difficulties have been encountered in improving the productivity of said. While architectures have been proposed which can accept high-level languages with ease, and attempts have gradually been made to change OS into firmware, emphasis is still placed on the utilization of software heretofore accumulated, and old and inconvenient architectures are followed. Efforts are required to prepare environments in which architectures suited for the new age centering around new applied fields or areas can be introduced. When software can be programmed with increased productivity as a consequence, diversified architectures will also be made possible thus opening up prospects for future computer science and engineering.

## 2.3 The maturing technological foundation and expected new technologies

Technologies surrounding computers, including LSI, have achieved epoch-making advancements during the past ten years in various fields, and can, moreover, be expected to attain even greater technical advancements during the next ten years.

Principal seed technologies to be introduced for the development of computers in the 1990's are set forth below:

(1) VLSI technology

To achieve a quantum leap in the computer technology of the future, it is essential to introduce LSIs and VLSIs fully into computer technology. Although these devices have been smoothly introduced into memories, evolution of storage systems in which a logic and memory are combined, such as an associative, will be of great importance.

Microprocessors were the first step toward the introduction of logic devices and it will be possible within a few years for a current large-size processor with about one hundred thousand gates to be produced on a single chip by way of VLSI technology. Such a possibility will undermine conventional computer technology which has advanced via the effective utilization of simple logic (circuits) serving as a central standard for evaluation. Stated otherwise, a technological foundation is in the process of being matured which will allow computers totally different from those existing today, something similar to artificial brains, to appear.

On one hand, individuals will be able to have personal computers which are comparable in function and performance with present day large-size computers and, on the other hand by reevaluating package systems of various functions which have thus far been considered impractical new computers having advanced functions and performance will make possible the opening of new fields of application.

VLSIs are not omnipotent, but rather have the following limitations:

1) Integration on a small area is required;
2) Wiring areas needed for connection are almost as important as the device area; and
3) Although repetive patterns can be designed with ease, many difficulties arise in designing error-free completely random patterns for large-scale systems.

To get the most out of VLSI technology, it is necessary to back to algorithms for a logic arrangement suitable for the two-dimensional structure of VLSIs, and to realize an integrated VLSI.CAD system including an evaluating simulator and a test data preparation system for a design free of errors. One ideal is to construct an intelligent CAD system centered around a knowledge base that will enable the smooth reutilization of data used in past designing and provide the designer with a supplementary source of knowledge.

(2) High-speed device technology

It is an important task to introduce into computer technology devices such as Josephson junctions or GaAs devices which can operate faster than silicon devices by more than one figure in unit time. Although the development of devices themselves is not the subject of the present program, (this task has been assigned to another project), this does not mean that we deny the importance of the development of devices. In order for these devices to be introduced into computers, VLSI technology utilizing such devices will have to be established. Since the establishment of such technology takes many years to achieve it appears impossible to consider fifth generation computer architectures premised on the technology of these devices because of the time factors involved, and hence these new devices themselves are excluded from the present program. However, the progress of the development of these devices should be watched so that they can be incorporated into the project at some intermediate stage of the fifth generation computer's development should these new devices prove sufficiently practical and capable of superior performance.

High-speed operation rendered possible by such devices tends to result directly in improved computer performance, and the construction of device-oriented systems is not nearly as difficult as for that of parallel systems. In addition to their high-speed operation, Josephson junctions are advantageous in that they require substantially no energy in storing information. Therefore, development of devices such as these should be carefully watched.

Optical technology should find a wide variety of applications in such areas as input and output devices and data transmission, and is also expected to prove useful in the area of peripheral memory technology, especially as a means for storing a knowledge base which requires no rewriting. Optical communication technology is suitable for high-speed data transmission, and can be utilized right now for high-performance local networks within a single building or premise. This technology will be a prime mover in the development of distributed environments described later on.

(3) Fusion with communication technology

The VLSI technology is oriented toward mass production, and calls for the extensive use of a single VLSI chip. If we consider VLSI's from the standpoint of architecture and systematization technologies, one possibility is load and function distribution due to miniaturization capabilities, and another is parallel processing which will be described below.

For the promotion of distributed processing technology, communication technology and computer technology will have to be more closely united.

It is necessary to establish a technoology which can connect a local network associated closely with a computer to a global network used for communication, and then to establish a system for allowing jobs and data bases to be distributed readily. Attempts have steadily been made to provide a foundation for realization of the foregoing technology and system, the technology of optical communication being an example. A wide variety of efforts ranging from technological research and development to standardization should further be made by those concerned.

One ideal to be realized toward and in the 1990's is either a nationwide or worldwide information system which utilizes a communication network for making correct and precise information readily available anywhere.

(4) Parallel processing technology

High-speed operation resulting from advanced devices is limited by the speed of light, and the paralleling of computers by means of pipe-line and SIMD* systems has been progressing. The fields in which high-speed operation has been most required are the area of large-scale numerical calculations such as solving partial differential equations and the field of simulators for large-scale systems. With the advance of LSIs, parallel computers have found a variety of applications and are expected to progress further, but control systems for these are limited by the fact that they can be commonly used for a wide variety of applications. Control systems based on data flow are now being widely accepted as systems capable of utilizing in its natural form, the parallelism possessed by algorithms. These systems are also considered capable of incorporating LSIs. Since such things as inference mechanisms, which will be described later on, are recognized as having an essentially large number of parallelisms, data flow machines are expected to become an an effective means of eliminating the many bottlenecks experienced with today's computers.

(5) Software technology

It has been pointed out for a long time that software is an impediment in the construction of information systems, and that its development and maintenance involve a lot of expense means of solving these problems have been researched as a part of software engineering, and many proposals have been made for the improvement of software productivity, some of which have been incorporated into high-level programming languages. Some examples of these are modularization, data abstraction, functional languages, non-procedural languages, and single assignment languages. Not only is reflecting the special features possessed by these languages in computer architectures necessary to efficiency process jobs described by these high-level language, but also, many functions are difficult to package without the aid of hardware.

The development of basic theories for programming and associated proposals for new computation models cannot be utilized without architectures and languages which have these as their premise. Thus, fixed architectures pose the danger of blocking the sound development of information technology as a whole.

On the other hand program verification and automatic synthesis technologies have steadily progressed. Although fully automated program rewriting to accompany architectural changes are impossible for the time being, it seems possible to construct systems for helping rewrite programs and greatly reducing the interposition of human hands. Such systems could be used to prepare new programs and effect program rewriting resulting from changes in specifications. In this way, the software crisis might also be avoided.

The realization of a highly intelligent support system for software generation is one of the ideals that fifth generation computers are aiming at.

(6) Artificial intellignece and pattern recognition technologies

Research into systems or highly intelligent robots which can understand the everyday conversation of human beings, know what it means, seek solutions and give answers together with machine translation and theorem proving as new computer applications, has been in progress from a relatively early stage. The study of some subjects, such as machine translation, though once considered readily realizable, has declined do to

* SIMD: Single Instruction Multi-Data stream

their proving to difficult to accomplish.

Languages and knowledge have been steadily studied as subjects of artificial intelligence, to produce many fruitful results. We have now reached the stage where the understanding of natural languages and the structuralization of knowledge data are nearing our grasp. However, this research still hasn't advanced beyond the basic level, being limited to the world of small vocabularies and narrow subjects. Problems which will be encountered when the foregoing technologies become practical have yet to be studied. The reasons for this are as follows:

1) Researchers of basic studies tend to be satisfied with solving problems in principle, and to have less interest in putting their subjects to practical use;
2) Computers at present have architectures designed mainly for numerical calculation, and as such have almost no functions, such as inference mechanisms, required for artificial intelligence. Therefore, processing takes a long time, and they are impractical for large-scale experiments; and
3) A lot of manpower is needed to prepare and input data and programs necessary for experiments.

Since the most detrimental of these reasons is the insufficiency of computer performance, it is important to smoothly promote the present program so that a computer system geared for artificial intelligence can be developed at an early stage of the project thus making its resources available to researchers of artificial intelligence.

The same can be said about the technology of recognition and understanding of patterns such as graphics, images, speech, voice and characters, etc. Computers with architectures suitable for pattern processing are necessary if we are to utilize computers to promote our research, realize a natural man-machine interface and further expand the realm of our senses. Since this will also be important as a facility for promoting basic research in stages prior to that, such a computer should be developed as a research-support system at an early stage of this project.

## 2.4 Objectives and significance of the fifth generation computer project

Fifth generation computers should be equipped with functions that will serve to eliminate social bottlenecks expected to appear in the 1990's, functions which today's computers lack. These can be summarized as the following four tasks:

(1) To increase the level of computer intelligence as well as their affinity for cooperation with man.

The five human senses can fulfil their functions only when backed up by the knowledge necessary to understand the information obtained through them. In order to raise the level of computer intelligence and increase their affinity for cooperation with man, it is absolutely essential to provide these computers one way or another with knowledge related to their respective fields of application and the means for putting these to practical use. It will also be necessary to develop a computer equipped with associative inference, and learning functions to process that knowledges more effectively.

Such requirements can be met by improving man-machine interface, and further researching understanding patterns such as speech, voice, graphics, images and objects, the comprehension of daily language, and knowledge bases.

(2) To process the ability to act on behalf of human beings as well as the ability to assist man in the development of unknown fields.

So that man and computers will be better able to share the burden of work related to environmental changes in our society, such as energy conservation and problems related to the aged etc., the intelligence level of computers will have to be increased to the extent where they can comprehend the environment. So as to expand the capabilities of our sensory organs with the aid of computers, development of sensor technology, and functions such as pattern collating abilities where a computer is connected to these sensors to extract the distinctive features of what is sensed, as well as a parallel processing ability for real time processing is necessary.

(3) To enable various forms of information to be made readily and easily available when necessary.

The information available through present information processing systems is highly limited with respect to the kinds, amounts and forms of information we come in contact with in our society. It is necessary to reduce the gap and facilitate instant access to a greater amount and wider variety of information. It is also important to develop a means of access which enables the easy and accurate retrieval of information needed at that time. Also important is a support system for clarifying the many vague requests made in the real world an essential technology for enabling computers to be applied to non-standardized jobs such as CAD and decision making support systems.

Computer networks which are capable of accessing dstributed data based, and knowledge bases capable of understanding the meaning of questions and giving answers are also important.

(4) Acquisition of new perceptions by simulating unknown situations.

It is expected that we will be able to acquire knowledge of unknown situations by means of large scale simulations in a variety of fields such as science and technology, management, administration, and society. Through realization of ultra-high-speed computers using high-speed devices and parallel processing, precise simulation will be made possible in fields where simulation had been impossible to date.

From the standpoint of the user, fifth generation computers should functions like the five which are enumerated below:

(1) Easy to use functions capable of being utilized even without professional knowledge.

Systems of this kind should be equipped with

1) functions for the inputting and outputting of information by way of sentences, speech and voice, graphics, images and the like,
2) functions for the processing of information in a conversational manner by means of daily language and graphs, and
3) functions for storing common knowledge as well as ones capable of utilizing the specialized knowledge for each field of application.

(2) Human substitute functions capable of judgement and decision making.

Ideally, judgements involving logic should be left up to the computer while the data necessary for important decision makings, is provided for man.

The following abilities should be developed:

1) functions which enable automatic retrieval of related information out of vast amounts of stored data in response to inquires,
2) functions which enable conclusions to be drawn from inferences based on stored data when an unknown problem is given, and
3) functions capable of learning and storing for subsequent use solutions to new problems.

(3) Functions capable of flexible configurations applicable to a wide range of jobs.

In order to be able to freely select efficient system configuratons responsibe to various non-standardized jobs in a wide variety of applied fields, the following are required:

1) functions capable of constructing system optimum for needs in question,
2) functions capable of handling large-scale computation processing and management of a large quantity of data as desired, and
3) functions that can easily be upgraded on a building block system to meet increased jobs.

(4) Functions for facilitating programming

Effective utilization of accumulated software and improvement of software productivity require:

1) functions enabling a computer to write and modify its own programs,
2) functions enabling a computer to judge and process matters of common sense without instructions from man, and
3) function able to cope easily with different types of computers as well as additions to existing equipment.

(5) System functions which are reliable and can be used expediently.

From the standpoint of system configurations, the following are necessary:

1) compact system functions having higher cost performance ratios,
2) system functions capable of sophisticated distributed processing between distant points,
3) highly reliable functions such as, functions able to recover automatically and minimize the adverse effects of malfunctions,- as well as functions to facilitate verification, and system functions of high maintainability, and
4) sophisticated functions to protect secrets.

Japan has come to be called an "economic power" thanks to the remarkable growth of our various industries. But when we considered the future course of these industries, it is important to stop playing "catch-up" with the more advanced countries and to set goals of leadership and creativity in research and development, and to search for a research and development system suitable for such an aim. Promoting a national project such as this in the computer industry which has a strong effect on various leading technologies, will probably greatly influences the way in which research and development systems will be made in other industrial fields.

The role and effects of the research and development project for fifth generation computers is summarized as follows:

(1) By promoting this project, our country will play a world leading role in the field of computer technology development. Our efforts will not only foster creative technology for our own computer industry, but will also provide our country with bargaining power. We also fulfil our duty as an economic power by investing in the development of such leading fields.

(2) This project will enrich the society of the 1990's and produce other effects.

This project is expected be beneficial in the solution of social bottlenecks such as the energy problem, the ageing of society and the like, and to serve as an active prime mover in all industrial fields by helping increase efficiency in those areas were increasing productivity has proven difficult, with the aid of CAD and management decision-making support systems. An increase in the intelligence level of industrial robots will result in improved product quality and energy saving. As industrial robots are applied to the primary industries, productivity in these fields will also increase, and workers no longer suffer from poor working environments should have less accidents.

Elimination of the software crisis will not only allow us to challenge the construction of more sophisticated systems, but will also serve to remove the negative aspects brought about by computerization, such as computer crimes.

(3) Development in unexplored fields can contribute actively to the progress of human society.

By promoting the study of artificial intelligence and realizing intelligent robots, a better understanding of the mechanisms of life will become possible. The approaching realization of automatic interpretation and translation will serve to help people of different tongues understand each other, to reduce troubles due to misunderstanding and ignorance, and to lead to further growth based on mutual understanding of cultures. With the construction of a knowledge base made possible, the knowledge which mankind has accumulated can be stored and effectively utilized, so that the development of culture as a whole can be rapidly promoted. Mankind will more easily be able to acquire insights and perceptions with the aid of computers.

(4) Experiments for leading research and development organizations.

It is extremely important to test at the national level the leading organizations for research and development which have been in operation for a long time. Almost all national projects from now on will be required to be carried out by leading research and development organizations such as this, and the present program can be looked upon as a kind of experimental project for future projects. Promotion of such a project creates an environment conducive to producing original studies based around it.

## 3. IMPACTS AND EFFECTS

### 3.1 Possible impacts of fifth generation computers on society

It is necessary for us to try to foresee every possible impact that fifth generation computers will have on functions and systems of our society.

(1) Elimination of social distortions resulting from differences between low-productivity and high-productivity fields

Increases in the cost of goods such as products, services and others is in inverse proportion to increases in productivity. Industrial fields in which no increased productivity can be expected, are producing goods at ever increasing costs which will reach an unbearable point. The result is that such industrial fields will either decline in or disappear altogether from society.

Fifth generation computers are expected to function extremely effectively in all fields of society. First they will take the place of man in the area of physical labor, and through the intellectualization of these advanced computers totally new applied fields will be developed, social productivity will be increased, and distortions in values will be eliminated. As an example, there are indications that the agriculture and fishery industries might be changed into food industries which would function effectively enough to give Japan full self-sufficiency in food.

Fifth generation computers are also expected to greatly improve the low levels of productivity being experienced in the fields of medical treatment and education, and to play a big role in the distribution of information.

(2) Expansion of man's abilities

Heretofore, increasing productivity has been accomplished only through improvements in the efficiency of man's labor. In the future, we should let machines do what they can do for increased productivity, and should concentrate our efforts on those things that only human beings can do. Fifth generation computers can play an important role in anplifying an intelligent ability which only mankind can have. Representative for performing this task are DSS*, CAE/CAD.

(3) Impacts on individuals

Today's informationalized society appears to be flooded with a great amount of information. Considerable efforts are necessary for people to obtain only that information they desire and to form unbiased judgments on it. Thus, there is no general guarantee that their conclusions will be reasonable ones. Such a situation is sure to lead people to doubt the merits of the utilization of information.

This tendency is quite disadvantageous to the progress of information technology. One of the expectations for fifth generation computers concerns the great progress which is likely to be made in the relationship between individuals and information. This will stem from the fact that anyone will be able to converse with computers even without a professional knowledge of them, even if everyday natural language is used, the computers will be able to understand our thoughts, and give us suitable answers. Although we have heretofore had to adapt ourselves to machines in order to get results, in the future machines will become much more 'human' in their ability to handle our requests.

(4) New Society

It is difficult to foresee exactly what form the forthcoming new society will take. In the 1990's, by means of the realization of fifth generation computers we would like to expect that the numerous tasks outlined by the Subcommittee for Research into Social Environmental Conditions will be capable of being

* DSS: Decision Support System

solved. But perhaps even more than this, it is felt certain that fifth generation computers will trigger the realization of developments and phenomena heretofore undreamed of.

## 3.2 Impact and effects in various applied fields

The following are some effects that fifth generation computers can be expected to have on main fields which will be rapidly systematized up to the 1990's and in which computer systems will produce great advantages.

(1) Effects of OA

By means of office automation systems (OA), the various functions of EDP and other departments will be structurally connected to allow managers at various levels to make handy use of data bases and results of various simulations via personal computers and terminals, thus permitting jobs to be done on a company-wide scale.

Planning, research design engineering and other departments will be able to use, in addition to conventional characters and numbers a data base system capable of handling a wide variety of forms such as images, graphs, speech and other mediums for giving flexible creativity a greater chance to blossom from a general point of view.

Systems utilizing less paper will be realized to enable electronic storage and exchange of information together with the utilization of multiple-function terminals, and the support of office jobs with automatic translation and various knowledge processings.

As a result, productivity by white-collar workers will be increased, the quality of management improved, and creative environment achieved, thereby realizing offices which are labor and energy savers as well as more internationally oriented. Offices will be required to have the right men in the right places, to be organized structurally and flexibly, to cope with an ageing society, to have an increased ability to develop technology, and to have better international exchange, and to be more open internationally.

In the 1990's, with the technology to achieve these objectives, Japanese who are capable of adapting themselves flexibly to new environments will be able to realize sophisticated office automation systems for processing information through various mediums including images, graphs, speech and the like, and become a world pioneer of OA.

(2) Effects of DSS

DSS provides high-level information and support thinking processes to decision making individuals and groups for increasing the validity and reducing the time required for making decisions, as well as reducing the costs involved in decision making.

With DSS, due to the fact that consistency in decision making processes is imporved, and group decision making is rendered more efficient and adaptable, more sophisticated decision making becomes possible, thus enabling industries to increase productivity rationally and smoothly, separate technologies to be integrated, and general knowledge industries to be developed to allow for more stable and sophisticated judgment and decisions to be formed in politics, administration, and industry. DSS is being promoted at the home and personal level where it can be used in a sophisticated way for planning family finances, designing lifestyles, and scheduling activities.

With these achievements, activities in all facets of society will be affected and within a margin of safety, more advanced, humane behavior will be possible, thus allowing for a more balanced society.

(3) Effects of CAE

There are complicated steps involved in all manufacturing processes, from the basic design up to completion of the product. By incorporating a CAE system into such processes, a multi-form database, advanced image processing system, a high-speed calculation system for design use, a system for automatically translating and preparing documents and a project management system will become the principal construction elements necessary for supporting the process until product completion.

These systems are indispensable for putting a company's know-how to practical use especially where vast numbers of blueprints and basic data are concerned, for effecting high-speed, flexible designing operation, for making bids overseas, for preparing a large quantity of documents such as contracts and instruction manuals, and for optimum construction of large buildings and plants. Of course, performing these jobs requires a man-machine interface, safety, and improved cost performance.

With completion of the fifth generation computer and the above-mentioned subsystems production processes in the building and manufacturing fields will be greatly rationalized for saving labor, energy and time, and safety which has been dealt with empirically up to now can be greatly strengthened. Also better designs and new technologies can be introduced. Furthermore, our country can establish a foothold in the world by performing its role in the international division of labor, furthering qualitative improvements in economics, applying new technologies to other fields, and cultivating new frontier industries.

Such accomplishments will make possible the evaluation of plans even for huge developments based on sufficient data and simulation evaluation of component blocks, avoiding risks associated with big, long-term investments.

The products of our country will be rendered unique and specialized in their respective fields due to their performance, design and knowledge-intensive qualities. These achievements will further serve as a foundation for promoting the true knowledge intensiveness of our industries.

(4) Effects of intelligent robots

It is believed that the 1990's will find robots possessing senses and high intelligence approaching those of human beings based on sophisticated back-end computing power.

Such robots will be able to handle more sophisticated requests from mankind by means of their high-level intelligent processing capabilities, increased responsiveness to humans, and facilitated operation due to their compact size and high power.

These robots will extend our spheres of activity to outer space, the ocean depths and mines deep within the earth to acquire resources, develop natural sciences, and carry out various investigations. They will also liberate man from such adverse working environments as those where he would be exposed to radioactivity and high temperatures. These robots will also find themselves working in fields such as agriculture, fishing, forestry, transportation, and nursing, releasing man from labor in a wide sense of the word. Production activities that will require less care on the part of man will extend from the primary to tertiary industries, and will allow us to engage in more sophisticated activities in those fields. This will lead to automated production in the secondary industries, which means that the rate of productivity per worker will be greatly increased.

It is also expected that general-purpose robots will become popular for automated production in small-scale manufacturing industires, causing their productivity to approach that of big companies.

The impact of intelligent robots will be especially great from the standpoint that the small, rugged systems for inputting and outputting speech, understanding graphics, and controlling systems, plus the inference machine modules in the fifth generation computer systems which will be utilized and completed by these can be used as parts for general machines to make the latter more sophisticated in operation.

# 4. CONTENTS OF RESEARCH AND DEVELOPMENT

## 4.1 Targets of research and development

Fifth generation computer systems will be designed to overcome technical limitations which conventional computers have had, and will be oriented toward processing knowledge intensive information based on innovative theories and techniques proposed to meet the sophisticated functions which are considered to be required in the 1990's.

Such fifth generation computer systems will have the following basic functions:

1) Problem solving and inference functions;
2) Knowledge base management function; and
3) Intelligent interface function.

These functions will be realized by software and hardware systems respectively, and will be aimed at maximum scales and performances such as those which follow:

The problem solving and inference function will be aimed at a maximum performance of 100 M – 1 G LIPS*.

The knowledge base management function will be aimed at a performance capable of effecting retrieval of a knowledge base required for inference within several seconds, with a core data base machine having a maximum capacity of 100 – 1,000 GB.

The intelligent interface system will be aimed at making conversation with a computer through the medium of speech, graphics, and natural languages etc., a possibility as well as enabling the exchange of information in a form which is natural for man.

These functions will be combined together into a single general-purpose machine having a system configuration which can meet various performances required in a variety of applied fields.

These functions may be arranged so as to serve as machines in which any one of the functions is reinforced, and as machines they will have a common programming language.

The fifth generation computer system will be aimed at sufficient general-purpose functions and performances required to realize systems for machine translation, question answering and utilization of speech, picture and images, systems which will be basic and common for a wide variety of applications in the 1990's.

The target performances of the basic application systems perceived here are shown in Table 4-1.

## 4.2 The image of the fifth generation computer system

The fifth generation computer system will be considered here from two different points of view in order to get as general an image as possible.

The first point of view is a conceptual view of a hierarchical structure including a human system, a modeling system and a machine system, and is centered on how the level of the man-machine interface will increase with respect to its present level.

The second point of view deals with the fifth generation computer system more specifically, and shows how components are combined into the system as software or hardware. Since it would be difficult to describe the system in its overall configuration, it will be divided into an application system, software

---

Note: * 1 LIPS (logical Inferences per Second) means one inference operation of syllogism per second. One inference operation on a present computer is considered to require 100 – 1,000 steps, and hence 1 LIPS is equivalent to 100 – 1,000 IPS (Instruction per sec). Machines of the present generation are of approximately $10^4 - 10^5$ LIPS.

Table 4-1 Themes and targets of basic application systems

| |
|---|
| **Machine translation system**<br>o Multi-lingual translation<br>o Word capacity: 100,000 words<br>o Translations should be 90% accurate, with the remaining 10% processed (edited) by humans.<br>o Should be an integrated system capable of taking part in each of the processes from compilation of the text to the printing of the translated documents.<br>o The entire cost of translation should be 30% or less than those made by humans. |
| **Question answering system**<br>o Should be a system prototype for answering questions in a variety of professional fields.<br>o Word capacity: 5,000 words or more<br>o The number of inference rules: 10,000 or more |
| **Applied speech understanding system**<br>o Phonetic (voice inputting) typewriter:<br>Should handle 10,000 words, possess a meaning analyzing function, be capable of correcting errors in speech by itself, and output sentences easy to understand.<br>o Speech-responding system: Should handle 10,000 words, be able to grasp the meaning of responses and thus be capable of natural conversation.<br>o Speaker identification system: Should be able to handle a few hundred people or more and identify speakers within a practical interval of time. |
| **Applied picture and image understanding system**<br>o This system should structurally store about 100,000 pieces of information in picture and image form so as to be usable for knowledge information processing. |

system, and a hardware system to provide an image of the configurations for each system.

The application system corresponds to part of the human system in the hierarchical structure described above, the software system corredponds mainly to the modeling system, and the hardware system corresponds primarily to the machine system.

The image of the fifth generation computer system can be grasped more clearly by combining the foregoing two points of view.

### 4.2.1 A conceptual image of a fifth generation computer system

The fifth generation computer system will be oriented forward processing knowledge information and will have quite a high logic capability. Its greatest feature will be that interface between man and computer will greatly approach the human level.

Conventionally, man-machine interface has been via procedural programming languages. To solve a problem with the help of a computer, man has first had to describe, model and program the problem. Humans and computers have been able to understand each other only through programs thus prepared.

With fifth generation computer systems, however, the description and modeling of a problem will take place at interface. In other words, computers will be able to understand problem descriptions and from that express a model, and synthesize a program based on such modeling. Man will be able to communicate with computers by using speech, natural languages, picture or images with a certain extent of freedom.

To realize such sophisticated capabilities, both software and hardware should be functionally improved. Figure 4-1 shows a conceptual image of such a system in which the machine system indicates future hardware. It can be understood from this Figure that the machine system has functions much higher in level than those of conventional machines. If we compare the old with the new in terms

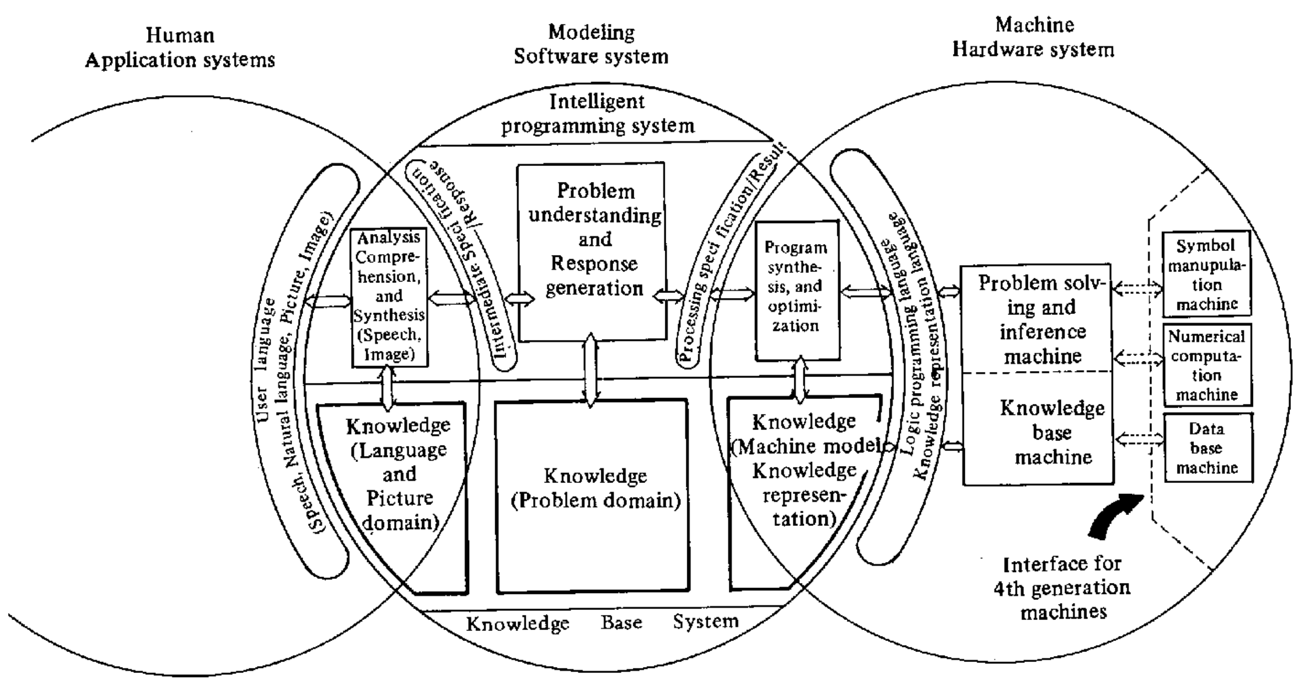



**Fig. 4-1 Conceptual diagram of a fifth generation computer system as viewed from the standpoint of programming**

of programming languages, while conventional machines use procedural languages on the basis of sequential execution, the new machine system will use logic programming languages or problem solving languages for trial-and-error logical inferences.

The modeling (software) system illustrated above will be highly effective software for such hardware and will serve mainly to perform meta-inference functions for problem solving such as understanding problems and synthesizing programs. Since the level of logic programming languages is quite high, the modeling system can be a man-machine interface during the period of transition before the final object is accomplished. However, input processing in the form of everyday language, picture, or images etc. In order to minimize the incompleteness and vagueness of inputs is indispensable if we wish to allow the next stage of development. Conversely, a function will be necessary to add some of vagueness and incompleteness to fully original responses for obtaining summarized outputs.

The modeling system includes an intelligent communication system capable of understanding speech, natural languages, pictures, and images at that point when it interfaces with the human system.

The intelligent communication system itself will be realized as a sophisticated knowledge information processing system having modeling and machines systems such as described above.

The fifth generation computer system will always utilize knowledges required in series processing, beginning with inputs such as speech, natural languages, picture or images from the human system, and extending to understanding these inputs, synthesizing and executing programs around them, and generating responses. These knowledges include a knowledge of languages, a knowledge about images, a knowledge about problem domains, and a knowledge about the mechanisms and data expression of the machine system, all stored in a knowledge base.

With the functions of the machine system being sophisticated and amplified by the modeling system in this way, our ability to process information will be greatly improved.

### 4.2.2 An image of the configuration of an application system

Structures common to various systems such as intelligent CAD, intelligent OA, intelligent CAI, and intelligent robots which will also be realized in the fifth generation computer system are shown in Figure 4-2.

| Interactive system | Processing system |
|---|---|
| Management system | |

**Fig. 4-2 System structure**

All application systems are composed of three subsystems, namely, interactive, processing and management systems. These three subsystems will be proportionally different from application system to application system. These subsystems are illustrated in Figure 4-3. (see next page.)

Showing their mutual relationship and internal operation to clarify the various functions they perform. Speech, natural languages, picture, images or their combinations are used to put a question to the system. The interactive system utilizes the knowledge inherent in languages or picture to analyze a structure (construction) and convert it into an internal (intermediate) expression such as an anatomical tree. Then, an analysis is made of that meaning in context and a description of the problem is extracted from that. This, however, is incomplete due to omissions and the like. A knowledge used here about context and background knowledge, which is one of the knowledges used at this time, is information related to the background and flow of the conversation taking place. The processing

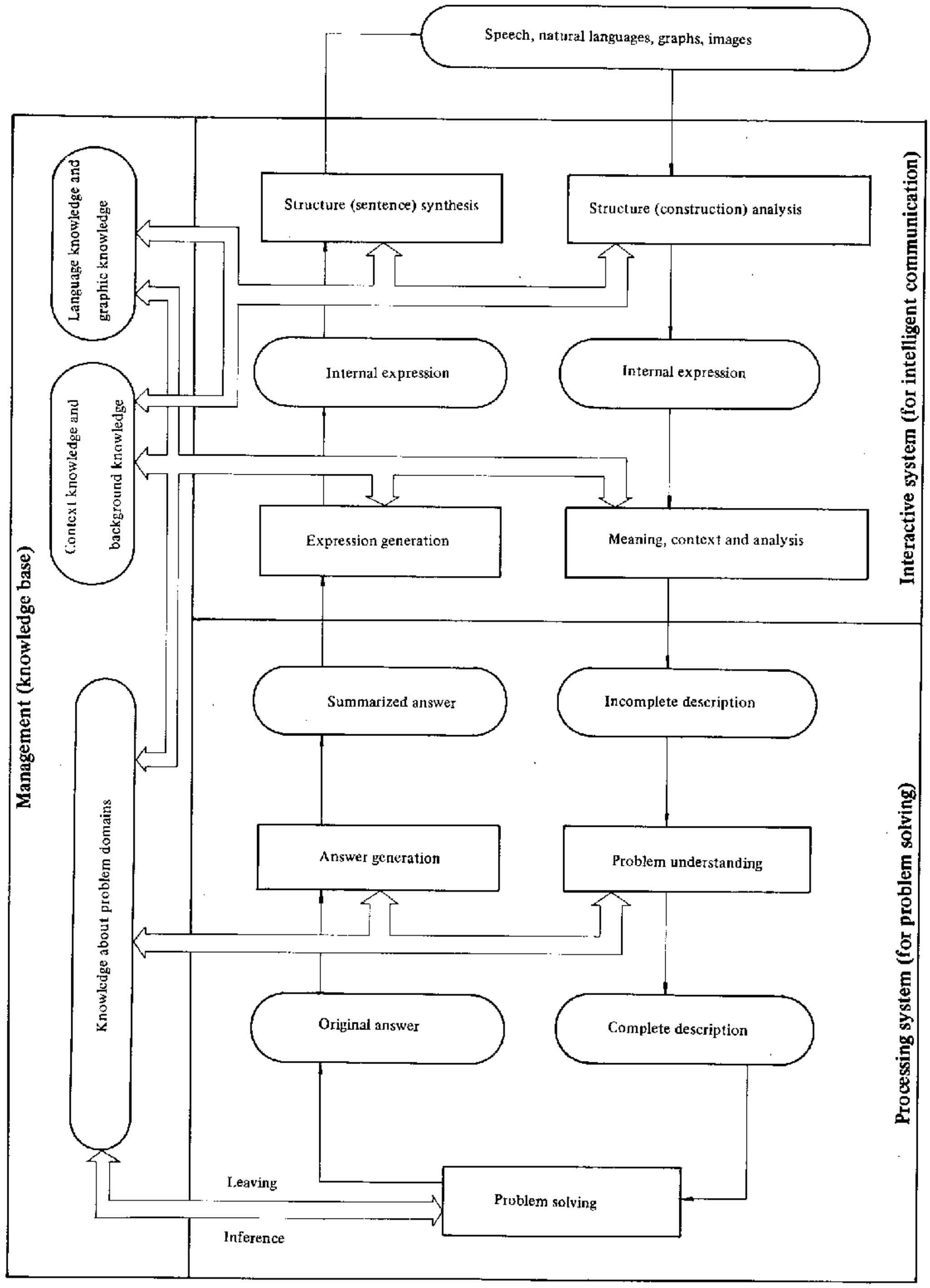



**Fig. 4-3 Mutual relationship and internal operations of the three subsystems in an application system**

system converts the incomplete description into a complete description using its knowledge about problem domains, and generates and answer to the description. At this time, opeations such as effective utilization (inference) of the knowledge about problem domains and storage (learning) or new knowledge are effected. The generated answer is then converted into a summarized answer by getting rid of unnecessary self-evident information. Thereafter, this summarized answer is converted by the interactive system into an internal expression, which in turn is converted into an external expression understandable to man. In this way, one conversational cycle is completed. During this cycle, the management system oversees a variety of knowledges for effecting common operations of inference and learning.

### 4.2.3 An image of the composition of a software system

An image of the composition of a software system for realizing various application systems is shown in Figure 4-4, the software system directly reflecting the structure of application systems.

(1) Basic software systems

These will be the core of all systems and consist of a problem solving and inference system, a knowledge base management system, and an intelligent interface system. These systems correspond respectively to the problem solving and inference machine, the knowledge base management machine, and the intelligent interface machine, and may be defined as those which cannot be constituted as hardware in realizing functions.

(2) Intelligent systematization support systems

These will be a group of systems which, in designing and producing (systematization) optimum information processing systems for various applications, will have knowledge about what is to be produced, production processes, and the like for greatly reducing the amount of work which man will do in systematization. These systems include subsystems which lead from a strict specification description language and a described specification to what is to be produced, or a subsystem for verifying correctness, and a subsystem for simulating operations, and the like. It also comprises three support systems, that is, an intelligent programming system for handling programs, a knowledge base design system for handling a knowledge base, and an intelligent VLSI design system for handling VLSI chips and computer architectures.

(3) Intelligent utility systems

These will be a group of systems which will provide sophisticated functions to facilitate utilization of the system itself. These will be comprised of a system for maintaining transferability to transfer stored programs and data bases from existing commercial machines to a target machine, a system-explanation and

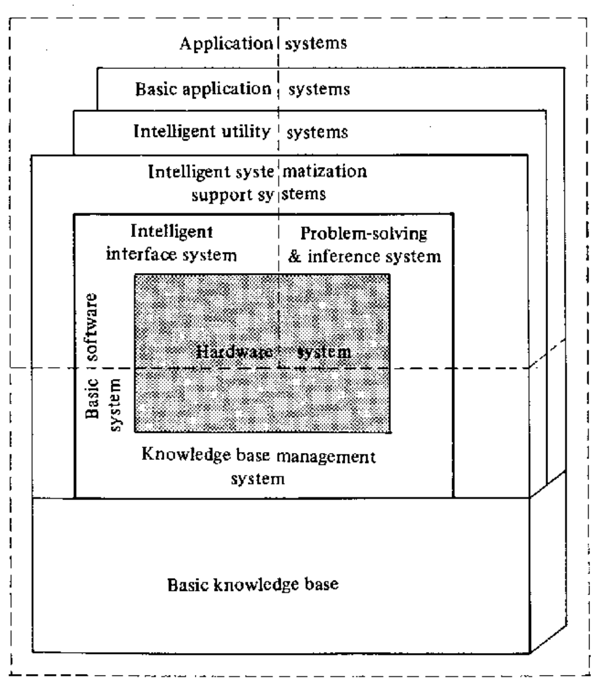



**Fig. 4-4 Conceptual diagram of the composition of a fifth generation computer software system**

education system for explaining the functions and use of the overall system and subsystems and for responding to user's consultation, an intelligent trouble diagnosis and maintenance system for automatic inspection and recovery and for guidance and consultation about inspection and repair of complicated troubles, and other systems.

(4) Basic knowledge bases

Universal knowledge used by the system itself and users will be arranged as basic knowledge bases which are components of the foregoing systems and are emplyable in application systems which users will make. There are largely three knowledge bases: a general knowledge base similar to common sense; a systems knowledge base which will gather knowledge related to systems; and an applied fields knowledge base which will gather together knowledge about certain applied fields. The general knowledge base includes bases of basic words of everyday use, basic sentence patterns and basic scripts, a base of dictionaries of various languages and sentence construction rules, and other bases related to natural languages. The systems knowledge base includes bases containing specifications for the system itself, such as a processor specification description base and an operating system specification description base, a language manual base, a program module base containing programs which are highly usable and other bases. The application knowledge base includes a VLSI design technology base, a computer architecture base, a basic program base, and other bases.

(5) Basic application systems

This group of systems which will be developed as basic application systems and have respective final target performances. These systems will be highly worth utilization and be a source of knowledge bases and sophisticated-function modules commonly usable by various application systems. The systems are largely classified as follows:

. Machine translation system
. Question answering system
. Applied speech understanding system
. Applied picture and image understanding system
. Applied problem solving system

### 4.2.4 Future of hardware system structure

(1) A profile of the fifth generation computer systems

Fifth generation computer systems, covering all sizes from the small ones for personal use to the large-scale computers, will find application in various fields. These will include machines for exclusive use as well, incorporating particular strengthened functions like the existing data base machines grouped into a community by a local network.

The computers in this community may be classified according to their abilities, but in so far as they will share a common programming language, they may be looked upon as members of a new computer family.

From the standpoint of their basic software interface, these computers shall have three functional components. These are listed below side by side with the corresponding components (in parentheses) of the existing computer systems.

1) Problem solving and inference machines (CPU)
2) Knowledge base management machines (Memory and filing system with virtual memory)
3) Intelligent interface machines (I/O channels and devices)

These three components will form part of each and every computer system. A general-purpose fifth generation computer system will be equipped with each of these machines in substantially the same proportion, whereas a small system with the same structure will form a general-purpose fifth generation personal computer.

A computer system with enhanced problem solving and inference function will be referred to as a problem solving and inference computer. This will find application in fields like consultation requiring professional knowledge, calling for strong ability to infer. Systems with reinforced knowledge base management function will be called a knowledge base computers. Like the existing data base machines, they will be applied in fields requiring storage of 'Knowledge' in large masses.

Computers incorporating an enhanced intelligent interface function will be provided with an interface with various interactive media, speech, picture and image as well as those based on natural languages. It will be possible to use these machines independently or in combination.

Figure 4-5 projects a conceptual image of the general fifth generation computer configuration.

Computer functions will be available at various levels and their combinations will create a wide range of machines covering both the small personal computers and large-scale machines incorporating each function to its maximum extent.

(2) Profile of the structures of machines serving different functions

Hardware architectures shaping the functional components will be based on a combination of six machines. These are the six machines that are being studied as the likely candidates to establish the new architecture. The machines will be so combined by adopting the distributed function architecture, namely by applying the modularization, adoptation, and microprogramming techniques.

For the smaller computers of moderate performance, a firmware base architecture built up on innovative von Neumann technique will be adopted. Language interfaces will center on new languages of both the predicate logic and abstract data types. Thus, the results of study on both the logic programming machines and abstract-data type support machines will be made programming use of.

For the powerful large-scale computers, data flow machines including functional machines will serve as the core technology. For the problem solving and inference computers, the execution part in the logic programming machine will use a large-scale data flow mechanism for its execution, and the knowledge base will be processed by a small-scale high-speed relational algebra machine. The relational algebra machine will use a suitable data flow mechanism for its execution.

A large-scale knowledge base computer will use as its core a large-scale relational data base machine including a relational algebra machine. Results of studies on the abstract data type support machines will also be used.

The supporting hardware in the intelligent interface system will include a VLSI processor for exclusive use in speech and a signal processors. Data flow machines techniques, including functional machine techniques will be used frequently in high-speed operations.

The data flow machine will constitute the basic execution mechanism for high-speed processing, and hence should be taken up as one of the main subjects for research and development.

A number of customized VLSIs are indispensable for the machines, and, therefore, development of VLSI-CAD to produce such VLSIs in a short period is to be treated as the most important theme from the standpoint of packaging.

(3) Macro image of the fifth generation computer structure

Fifth generation computers will be linked to communication systems to form a global network suitable for various social organizations. Potentially, each node in such a global network, that is, each computer site, will

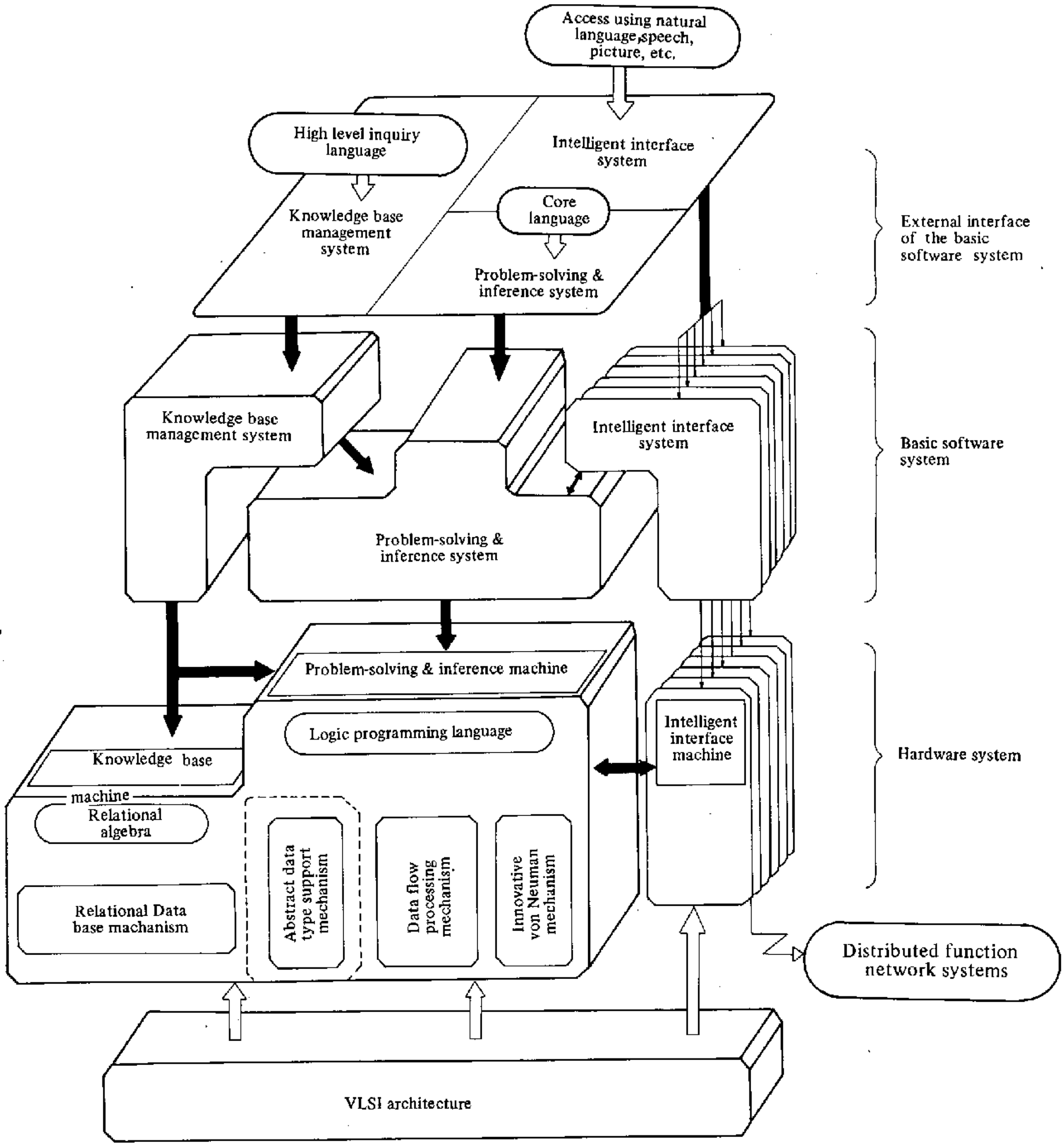



**Fig. 4-5 Basic configuration image of the fifth generation computer systems**

constitute a system connected by a local network to two or more computers. The local network, capable of high-speed data transfer, will connect together computers of different functions, including the smaller personnel computers, thus making up a single as a whole general-purpose group (community).

As the macro image suggests, a fifth generation computer system will be a collection of computers serving different functions: a small, general-purpose personal computer, a knowledge base computer, and a problem solving and inference computer, all connected by a local network.

In principle, the component computers will have a common programming language. These computers, therefore, will form a computer family linked by a common language even though they may be intended to serve exclusive purposes with one or the other of the functions enhanced.

A structure of this nature will help build up a flexible computer system suitable for the intended applications. As the above suggests, hardware and software research and development for fifth generation computer systems should be so carried out as to allow them to be connected by local and global networks.

### 4.3 Research and development themes

There are in all 26 themes on which research must be carried out to develop the fifth generation computers. These are grouped into seven in Table 4-2. These research and development themes are summarized in Table 4-3.

**Table 4-2 Items in research and development of the fifth generation computer systems**

| | |
|---|---|
| Basic application systems | 1–1) Machine translation system<br>1–2) Question answering system<br>1–3) Applied speech understanding system<br>1–4) Applied picture and image understanding system<br>1–5) Applied problem solving system |
| Basic software systems | 2–1) Knowledge base management system<br>2–2) Problem solving and inference system<br>2–3) Intelligent interface system |
| New advanced architecture | 3–1) Logic programming machine<br>3–2) Functional machine<br>3–3) Relational algebra machine<br>3–4) Abstract data type support machine<br>3–5) Data flow machine<br>3–6) Innovative von Neumann machine |
| Distributed function architecture | 4–1) Distributed function architecture<br>4–2) Network architecture<br>4–3) Data base machine<br>4–4) High-speed numerical computation machine<br>4–5) High-level man-machine communication system |
| VLSI technology | 5–1) VLSI architecture<br>5–2) Intelligent VLSI CAD system |
| Systematization technology | 6–1) Intelligent programming system<br>6–2) Knowledge base design system<br>6–3) Systematization technology for computer architecture<br>6–4) Data base and distributed data base system |
| Development supporting technology | 7–1) Development support system |

**Table 4-3 Contents of research and development themes**

Item group

- **Basic application system**

A basic application system representing functions like hearing, speaking, seeing, drawing, thinking, and problem solving will be studied and developed.

R & D themes

- Machine translation system

Results of researches in documentation techniques and artificial intelligence for knowledge utilization will be combined together to research and develop an integrated multi-lingual translation system.

R & D details

- Designing a machine translation system and its core.
- Development of the grammars for the languages
- Development of sentence generating grammars
- Development of an integrated machine translation system with room for operator intervention
- Development of a specialized terminology data base (knowledge base)
- Development of a machine for the specialized terminology data base
- Development of high-level word processing techniques

Targets and specifications

- Number of words to be handled: 100,000 words
- Machine must assure 90% accuracy, and the remaining 10% is to be processed by the translators
- The system must serve general purposes computerizing all the jobs including text compilation to printing of translated documents
- Translation cost must be 30% or less than that by translators

Item group

- **Basic application system**

Themes in R & D

- Question answering system

Research and development of a question answering system for various specialized fields including the intelligent CAE/CAD system, DSS, and intelligent robots.

R & D details

- Conversation analysis technique
- Graphic data I/O units
- Specialized data generating
- Input error processing technique
- Natural language subset designing and development of grammars and dictionaries
- Synthesis technique

Targets and specifications

- An interim target to be achieved in 5 years is to develop a trial question answering system for limited use in particular specialized fields.
    - Number of words: 2,000 words (Japanese)
    - The user must provide supplementary information to eliminate ambiguity.
    - Number of inference rules: 1,000

- The interim experimental question answering system will be evaluated, and prototype question answering systems will be developed for various specialized fields.
    - Number of words: 5,000 or more
    - Number of inference rules: 10,000 or more

Remarks

- Will be used for the development of an intelligent utility system.

| Item group |
|---|

- **Basic application system**

| Themes in R & D |
|---|

- Applied speech understanding system

Research and development of a speaker identification system as part of a general-purpose speech responding system for input and output in machine translation, a phonetic typewriter, and a telephonic inquiry system.

| R & D details |
|---|

- Development of a phonetic typewriter
- Development of a speech responding system
- Development of a speaker identification system

| Targets and specifications |
|---|

- Phonetic typewriter
  (1) Interim target: A system with of simple sentence construction data to handle several hundreds to several thousand words.
  (2) Final target: To handle about 10,000 words with simultaneous meaning analysis, automatic error correction during speech recognition, and generating as a whole comprehensible sentences.

- Speech responding system
  (1) Interim target: Handling several thousands words mainly through analysis and synthesis system.
  (2) Final target: Cap handling about 10,000 words, comprehending the meaning of questions to be answered, and developing a sophisticated structure to enable natural conversation.

- Speaker identification system
  (1) Interim target: Identifying fifty to sixty speakers.
  (2) Final target: Identifying several hundreds of speakers within a practicable interval.

Item group

- **Basic application system**

Themes in R & D

- Applied picture and image understanding system

Development of a system for structural storage of picture and image data and effective retrieval of such information to process intelligence.

R & D details

- Research on a picture and image data storage and retrieval system image information
- Development of a language system to retrieve picture and image data
- Development of a picture and image data base machine
- Development of a system to store and retrieve picture and image data

Targets and specifications

- A picture and image data base must contain about 100,000 retrievable picture and image data items.
- The system must store picture and image data including abstract delineations within a few seconds.
- Picture and image data must be retrievable within 100 m sec. on the average.
- Interim target in the stage: about 10,000 picture and image data items to be handled and processed at about half the speed aimed in the final target.

Item group

- **Basic application system**

Themes in R & D

- Applied problem solving system

Development of a formula understanding system outputting an "answer" to the "problem" input, thereby solving general advanced problems. Also development of a system playing the Go-game.

R & D details

- Research on a basic formula understanding system for mathematical expressions.
- Development of formula understanding system
- Research as a basic system to play the Go-game
- Development of a system to play the Go-game

Targets and specifications

- Formula understanding system
  (1) Interim target: System with a knowledge base combining the performances of the existing MACSYMA with inequalities and simple equations processing function.
  (2) Final target: A knowledge representation and problem solving system related to formula combining sophisticated formula manipulating algorithm.

- Go-game playing system
  (1) Interim target: A system having a playing standard equivalent more or less to amateur subgrade 10.
  (2) Final target: A system having a playing standard equivalent more or less to amateur grade 1.

Item group

- **Basic software system**

This will constitute out core of the fifth generation computer systems. A group of modules corresponding to basic information processing functions (management, processing, interaction) will be research and developed.

Themes in R & D

- Knowledge base management system

Research and development of intelligent system management techniques to format and store human knowledge in a computer to utilize it and support the user in solving problems.

R & D details

- Research on knowledge representation and utilization techniques
- Knowledge acquisition and learning
- Research on a large-scale knowledge base system
- Development of a knowledge base management system
- Development of a knowledge base machine

Targets and specifications

- Knowledge base management system
  - (1) Interim target: Simultaneous management of rules and data, to data base access optimization mechanism, a mechanism for eliminate inconsistencies, and interface with an inference machine.
  - (2) Final target: A multiple-world knowledge base, a distributed-knowledge base, learning based on inductive inference, fusion with an inference machine

- Knowledge base machine
  - (1) Interim target: Storage and retrieval of 2,000 rules and 1,000,000 data items ($10^3$ B per one item)
  - (2) Final target: Storage and retrieval of 20,000 rules and 100,000,000 data items (100 GB)

| Item group |
|---|

- **Basic software system**

| Themes in R & D |
|---|

- Problem solving and inference system

This will constitute the core of the processing functions in the fifth generation computers. Basic techniques will be research to develop a problem solving systems by establishing a processing model of the problem solving and inference systems, and to explain its processing ability theoretically.

| R & D details |
|---|

- Research on problem-solving and inference algorithm
- Development of a coding language to solve problems
- Development of an inference machine

| Targets and specifications |
|---|

- Inference machine
  (1) Final target: Performance of about $10^2 - 10^3$ Mega LIPS

- Coding language to solve problems must support functional and logic programming as well as object-oriented modular programming. Program modules generated will form a knowledge based software component for effective use in intelligent programming.

| Remarks |
|---|

- Logical Inference per Second

1 LIPS (Logical Inference Per Second) means one syllogistic inference per second.
One inference operation in the currency used computers is considered to require 100 – 1,000 steps, and hence 1 LIPS is equivalent to 100 – 1,000 IPS (Instructions Per Sec.)
Machines of the present generation feature approximately $10^4 - 10^5$ LIPS.

**Item group**

- **Basic software system**

**Themes in R & D**

- Intelligent interface system

Research and development of a technique for flexible conversational functions and elimination of the language (including natural languages, speech and image) gap between the user and his computer.

**R & D details**

Natural language and speech system

- Parsing (syntactic analysis)
- Semantic analysis
- Discourse analysis
- Sentence construction
- Construction of a language data base
- Natural langauge processor
- Multiple-language basic grammar generation
- Research on phoneme identification system
- Research on sentence understanding system
- Research on speech synthesys system
- Research on system to recognize the differences between individual speakers
- Development of a speech understanding system

**Targets and specifications**

- A man-machine communication technique based on natural language or speech data, providing an intelligent man-machine interface.

- The natural language and speech system with fulfil the following target:
  - (1) Vocabulary to be handled with cover computer and one branch of scientific and technological terminology, will include specialized as well as frequently used terms.
  - (2) The system must adapt itself to speakers and communicate with unspecified speakers.
  - (3) The system must be capable of speech output in Japanese and English.
  - (4) The system must identify speech signals on almost real-time basis.

Item group

- **Basic software system**

Themes in R & D

- Intelligent interface system

R & D details

Picture and image system

- Research on picture and image construction technique
- Research on picture and image generating algorithm
- Development of systematization technology
- Development of picture and image processors
- Research to interrelate natural languages, picture and image

Targets and specifications

(1) Development of soft- and hard-ware to enable smooth user interaction with the computer through picture and image media.

(2) Picture and image to be handled will be an complex as, respectively, medium small scale machine drawings, and photographs for medical use.

(3) High-speed information processing to allow smooth man-machine interaction.

(4) The interim target is to handle 70% as complex picture and image as are aimed in the final target. For the interim target emphasis will be laid on the method of processing rather than the processing speed.

Item group

- **New advanced architectures**

  Research will be made to enabled the fifth generation computer architecture to satisfy the knowledge data processing system requirements.

Themes in R & D

- Logic programming machine

  Study and development of the necessary architectures to support inferences and a computational model based on predicate logic with a power of expression approximating natural languages.

R & D details

- Development of and incorporating the predicate logic
  - PROLOG system
  - New language system

- Development of the basic technology
  - Research on parallel systems
  - Development of special-purpose mechanisms

- Logic programming machine
  (1) Firmware base machine ................ 0.1 Mega LIPS
  (2) Personal logic programming machine ...... 0.1 – 1 Mega LIPS
  (3) Parallel logic programming machine ....... 50 – 60 Mega to 1 Giga LIPS

Remarks

- Development of languages and a processing systems
  - Extended PROLOG language
  - New logic programming language

Item group

- **New advanced architecture**

Themes in R & D

- Functional machine

Development of architectures to support a functional model and programming language suitable for symbol manipulation, both based on theory

R & D details

- Development of a function language (including LISP)
- Method of coding for parallel processing
- Parallel computation model (data flow model)
- Firmware based LISP machine
- UHM for symbol manipulation and its VLSI
- Method of interconnecting the processors
- Associative processor and associative memory

Targets and specifications

(1) Personal LISP machine
To be two or three times a general-purpose computer (4 MIPS) in list processing capacity

(2) Parallel reduction machine
To be ten times on a general-purpose computer in list processing capacity

(3) Data flow function machine
To be several hundreds to several thousands times a general-purpose computer in capacity list processing

Item group

- **New advanced architectures**

Themes in R & D

- Relational algebra machine

Research to develop a machine architectures to handle, say, set operations, using relational algebra (constituting the core of future data base systems) as the interface language.

R & D details

- Development of an interface language and a processing system (relational logic, relational algebra, basic machine operations)
- Development of a data base management system
- Development of the algorithm for basic machine operations
- Development of machine architectures.
- Development of processor elements and connective hardware
- Building up a memory hierarchy system and development of memory devices

Targets and specifications

- Number of processor elements in the parallel processors
  - (1) Interim target: Not more than hundred
  - (2) Final target: At least five to six hundred

- Storage capacity
  - (1) Small capacity for high speed operations: 10 – 100 MB
  - (2) Medium capacity for medium and high speed operations: 100 M – 10 GB
  - (3) Large capacity for low and medium speed operations: 10 – 1,000 GB

Remarks

- To be closely related to data base machines and support the relational data base systems.

Item group

- **New advanced architecutres**

Themes in R & D

- Abstract data type support machine

Research and development of a memory structure and processor functions in the future computers to provide system architecture support to modularize the vast and complex software.

R & D details

- Arrangement and systematization of an abstract data type model, and research and development of a language system
- Mapping from the space handled to the physical resources
- Garbage collection
- Structured memory
- Abstract data type processors and architectures
- Parallel processing
- Others (I/O, relationship with the conventional languages, OS, DB, distributed processing)

Targers and specifications

(1) Development of about 100 parallel von Neumann abstract data type support machine

(2) Development of about 1,000 parallel non-von Neumann abstract data type support machine

Remarks

- Must be closely related to logic programming machines and functional machines.

| Item group |
|---|

- **New advanced architectures**

| Themes in R & D |
|---|

- Data flow machine

  Research on an architecture based on a data flow model oriented to parallel processing thereby achieving sophisticated parallel processing.

| R & D details |
|---|

- Design of a machine instruction set
- Design of a high-level language for data flow
- Determination of the overall structure of the meachines
- Configuration of the network connected
- Development of a structured memory
- Establishment of an activity control system
- Method of developing OS functions
- Countermeasures against troubles and protection plans
- Structure allowing combination with a conventional machines
- Development of a prototype data flow machine
- Development of a personal data flow machine
- Combination with the data base management functions

| Targets and specifications |
|---|

(1) Initial target: 16 processors with a memory of 8 MB (basic operating level.)

(2) Interim target: 100 processors with a memory of 100 MB and achieving 50 MIPS (practicable use).

- Processor networks: Structured to accommodate LSIs
  Consist of $10^3 - 10^4$ processors

(3) Final target: An extra high-speed data flow machine, $10^3 - 10^4$ processors with a memory of 1 – 10 GB and 1 – 10 BIPS.

- Personal data flow machines
  32 processors with a memory of 10 MB and achieving 10 MIPS.

| Remarks |
|---|

- Closely related to parallel logic programming machines, parallel functional machines etc.

| Item group |
| --- |
| • **New advanced architecture** |
| **Themes in R & D** |
| • Innovative von Neumann machine<br><br>Development of architecture with innovative von Neumann machines retaining their original advantages and with sophisticated VLSI. |
| **R & D details** |
| • Innovative VLSI von Neumann architecture with respectively one million and ten million transistors per chip<br><br>• Research of an architecture data base<br><br>• Development of micro 90 (object-oriented architecture) |
| **Targets and specifications** |
| (1) Interim target: A processor with one million transistors per chip for the innovative von Neumann machines<br><br>(2) Final target: A processor with ten million transistors per chip for the innovative von Neumann machines |

Item group

- **Distributed function architecture**

  Development of an architecture to combine the VLSI architectures with the new, advanced ones, using VLSI with importance attached to progressive architectures.

Themes in R & D

- Distributed function architecutre

  Development of a distributed function architecture consistently assuring high efficiency, high reliability, simple use and construction, easy adaptability to future technological improvements and the different machine/system levels, and sophisticated functions.

R & D details

(1) Development of a basic distributed function system
- Establishment of a logic model
- Establishment of various architecture system
- Dynamic architectures
- Implementation
- Special purpose machine development techniques

(2) Development of an experimental distributed function system
- Personal computer
- High-level language machine groups
- Local networks

(3) Development of an integrated system

Item group

- **Distributed function architecture**

Themes in R & D

- Network architecture

This architecture will be meant to loosely couple computer systems spaced apart. Development of the techniques to combine systems with a global network and build up a distributed information system based on the high-speed local network to be available to the fifth generation computer.

R & D details

- Standardization of network architectures
- Protocol coding, generating, and verifying technique
- Network OS development technique
- Multimedia processing technique
- Data security mechanism
- VLSI technology
- Optical fiber communication technology
- Satellite communication technology
- Local networks

Item group

- **Distributed function architecture**

Themes in R & D

- Data base machine

Development of special-purpose machine with an architecture suitable to process data bases and capable of high-speed accessing large-capacity data bases.

R & D details

- Research on new advanced architecture to process data bases (non-numeric)
- Research on data base machine with sophisticated functions
- Research on a man-machine interfaces
- Research on distributed data bases
- Research on techniques for conversion from or emulation of existing data bases
- Research on effective use new devices to be based on VLSI
- Collection and analysis of basic data to design data base machines
- Development of experimental machines
- Development of practically feasible machines

Targets and specifications

(1) Experimental machines
- Capacity: Up to 100 GB
- Processing ability: $10^3$ transaction/sec
- Data model: Relational

(2) Practicable machines
- Capacity: Up to 1,000 GB
- Processing ability: $10^4$ transactions/sec
- Data model: Relational

(To support conversion from and emulation of the data base of another model )

Item group

- **Distributed function architecture**

Themes in R & D

- High-speed numerical computation machine

  Development of special-purpose machine for high-speed scientific and technical computation for, say, numeric simulation to replace experiments.

R & D details

- High-speed logic devices
- High-density implementation technology
- Cooling technique.
- Architectures (logical specifications)
- High-speed numerical computation technique
- Special-purpose operating systems
- High-level language compilers

Targets and specifications

- Development of processor elements (40 – 100 MFLOPS), using new high-speed devices

- Processor elements of about 4 MFLOPS will be developed. Also, a parallel processing system will be developed to simultaneously operate 1,000 such processor elements to develop an overall performance of about 1 BFLOPS.

- Head per track disk of some fifty to sixty GB.

Item group

- **Distributed function architecture**

Themes in R & D

- High-level man-machine communication system

Development of a system to input and output characters, speech, picture and image and interact (intelligence) with the user

R & D details

- Devices to input and output characters (including Chinese characters)
- Device to input and output picture and image
- Device to input and output speech
- Development of an integrated system to input and output characters, speech, picture and image

Targets and specifications

- Character (including Chinese characters) in- and out-put system
  - (1) Interim target: A display unit with an input functon for 3,000 – 4,000 characters in four to five different typefaces.
  - (2) Final target: Additional functions allowing input of Chinese characters together with speech, replacement of *kana* (the Japanese syllabary) by the Chinese characters and vice versa, and meaning comprehension.

- Picture and image in- and out-put system
  - (1) Interim target: A tablet coordinates input device with 5,000 x 5,000 – 10,000 x 10,000 dots.
  - (2) Final target: More advanced intelligent functions based on specifications laid down in the research theme for applied picture and image understanding system.

- Speech in- and out-put system
  - (1) Interim target: Ability to identify 500 – 1,000 words.
  - (2) Final target: More advanced intelligent specifications as laid down in the research theme for applied speech understanding system, including a meaning comprehension ability and capability to party handle natural languages.

- Integrated terminal with a multimedia input and output functions
  All the above functions will be combined on VLSI basis to develop integrated personal computer terminals.

Item group

- **VLSI technology**

Development of architectures making full utilization of VLSI and processing from the component devices to fifth generation computers.

Themes in R & D

- VLSI architecture

Development of architectures to make full utilization of VLSIs characterized by about ten million transistors per chip (as are expected to be available around 1990).

R & D details

- Techniques for constructing new advanced architectures (basic study)
  - VLSI device rule book
  - Design QA system
  - Architecture data base
  - CAD for VLSI architectures

- VLSI architectures
  - Complete 1-chip architectures
  (one million transistors/chip, ten million transistors/chip)
  - Function parts architectures

- VLSI system
  - Function division and connection techniques

Targets and specifications

(1) Interim target: Complete one-chip architectures for one million transistor/chip.

(2) Final target: Complete one-chip architectures for ten million transistors/chip.

| Item group |
|---|
| • **VLSI technology** |
| **Themes in R & D** |
| • Intelligent VLSI-CAD system<br><br>Development of an integrated VLSI-CAD system capable of storing design know-how for effective utilization |
| **R & D details** |
| • Architecture data base<br>• Know-how data base<br>• Inquiry system for VLSI-CAD design<br>• Development of a technology for heuristic design |
| **Targets and specifications** |
| • An application designer should be able to design a masking pattern for a VLSI custom chip with one million transistors/chip within one month (a desired chip must be available within three months). |

| Item group |
|---|

- **Systematization technology**

  Consistent systematization, of devices, architectures, and basic as well as applied software, and development of techniques relating to the cycle comprising system design, development, maintenance, and management.

| Themes in R & D |
|---|

- Intelligent programming system

  Development of a system fetching programs from an algorithm bank (knowledge base) by user requirements, and synthesizing a program which meets requirement specifications by inference. Furthermore, the system must verify, by a process of inference, whether the program generated to meets the requirements optimally.

| R & D details |
|---|

- Modular programming and a verification theory
- A theory to specification description and program synthesize
- A system for program verification and synthesis and a program base
- A system to maintain, imporve, and manage programs
- A consultant system for program designing

| Targets and specifications |
|---|

- System for program verification and synthesis, and program base
  (1) Interim target: Improvement through synthesis and converison of programs for particular fields, minimizing data base retrieval.
  Development of a small-scale program base.
  Generation of a system to verify functional, logic, and data-abstraction programs.
  (2) Final target: Synthesis of large-scale programs for data base management systems, language processors, etc.
  Development of a large-scale program base.
- System to maintain, improve, and manage programs
  (1) Interim target: Generation of a system to comprehend functional and logic programs.
  Equivalence transformation experiment.
  (2) Final target: A system to evaluate program performance, and a system improved through equivalence transformation.
  A system for correcting programs.
- Consultant system for program design
  (1) Interim target: Basic design
  (2) Final target: Question answering in natural languages.
  A system capable of offering consultation in data base management system design, data base application systems, etc.

Item group

- **Systematization technology**

Themes in R & D

- Knowledge base design system

A system with an organically contained basic knowledge base. The base must store the technical data and knowledge necessary to design, develop, and operate a knowledge information processing system, and to support creation of knowledge base systems from the basic knowledge base.

R & D details

- A system to express and use metaknowledge.
- Development of a system to support design and development of a knowledge base.
- Development of a system to support widening a knowledge base.

Targets and specifications

(1) Simple creation of a knowledge base system to offer consultaition to the specialists about problems requiring sophisticated, specialized knowledge.

(2) The knowledge base system must be designed to comprise knowledge in the form of about 20,000 rules.

(3) Partial system verification at the level of metaknowledge. A large-scale knowledge base system must allow. Simple debugging.

(4) The interim target will be to achieve 30% of the final knowledge base system target.

Item group

- **Systematization technology**

Themes in R & D

- Systematization technology for computer architecture

Architecture-related systematization technique to complete a systematized fifth generation computer.

Development of techniques to build up virtual systems and real systems, optimization for system configuration and load balance, designing and developing a large-scale system, and techniques for high reliability.

R & D details

- Techniques to build up virtual system and system configuration
- Optimization techniques for system configuration and load balance
- Design and development techniques for large-scale systems
- Technique for ultra high reliability
- Development of a local network as architecture developing tool

Item group

- **Systematization technology**

Themes in R & D

- Data bases and distributed data base systems

Development of a data base system for the fifth generation computers, technique of integrating and utilizing two or more data base systems, and integration of knowledge base systems.

R & D details

- Research of data semantics and data models
- Development of a flexible structured data base system
- Development of a system to support scheme designs
- Development of a system to support data storage
- Development of a QA system using natural languages
- Development of a distributed data base system
- Development of a metacharacter data base system
- Development of a data base machine

Item group

- **Development supporting technology**

  Various systems will be research and developed to support the development of hardware and software, as well as the system as a whole.

Themes in R & D

- Development support system

  Construction, at an early stage of the project, of VLSI-CAD, personal computers, computer networks, and systems to support development of software/knowledge base.

R & D details

- Computer network
- Support system for software development
- Support system for knowledge base development
- Use of VLSI-CAD as a supporting tool
- Use of personal computers in research and development

## 4.4 Research and development procedure

### 4.4.1 Relationship between the different item groups

The items calling for research and development are classified into seven groups (see 4.3). Research and development activities concerning the project as a whole must organically link the respective items.

The project will be divided into initial, intermediate, and final stages as shown in Figure 4-6. The respective research items will be interrelated and mutually adjusted at the beginning of the intermediate, and final stages.

The following deliveate the principles on which the research items will be interrelated and adjusted:

(1) Development of techniques supporting development will have priority over all other themes.

(2) Basic research will be taken up independently, and positive efforts will be made to use their results in subsequent research and development activities in other related field.

(3) Research on basic software systems will constitute the nucleus of the present project. Its results will be used in two ways, namely:

1) In practice they must contribute to the development of application systems (including a basic application system), and
2) Resulting language specifications must serve for architecture development (primary and secondary).

(4) Research and development of the new architectures will be carried out after selecting some practicable approaches before the final language specifications are completed, and they will be combined together when the results of the studies on the basic software system are obtained (primary and secondary).

(5) Research and development of the basic application system will be based on the results of research on the basic software system, and the results of such research will in turn be used to help imporve the basic software system.

(6) Distributed function architecture and systematization technology will be research in parallel with the other activities. From time to time their results will be incorporated in the supporting systems to reinforce them. Thus, results of research and development will be fully and constantly utilized.

(7) First version of the VLSI technique will be used to design the intermediate machines, and the second version to design the final machines.

Figure 4-6 provides the flow chart for the development of basic software, new advanced architectures, basic application system, and the systematization and supporting technologies.

### 4.4.2 Procedure for research and development of themes related to basic software and architecture

Research and development themes related to basic software are closely related to research and development themes related to architectures. Particularly, the research of problem solving and inference system will also determine the specifications underlying the core language forming the basis of the research and development specifications for the new advanced architectures. Research of a knowledge base management system will help lay down the specifications for the relational algebra machine to support a knowledge base. Likewise, study of an intelligent interface system will indicate specifications of a hardware system to support the system.

Thus, overall research and development will be achieved through interchange of research results between software and hardware.

Research and development themes relating to basic software and advanced architectures and their interrelation will be discussed with reference to Figure 4-7.

Basic theories and systems will be studied in advance for the themes related to basic software. These studies will mainly involve system of expressions including a knowledge representation language, a core language for problem solving based on predicate logic, phoneme identification and a sophisticated speech synthesizing system, a system for parsing and semantic analysis, and systems to abstract the characteristics of picture and image, and to generate and display them. The studies will be carried out by repeating a cycle in which various trial systems will be developed with the help of the support systems and then evaluated. In this way, new perceptions will emerge, leading to the development of commonly usable software based on support systems. Primary specifications for the systems to be developed line with the respective themes will be laid down by the end of the early stages of development. In particular, specifications will be laid down for the core language to precede the first stage language for the development support machine during the research on problem solving and inference systems.

During the intermediate stage, small-scale prototypes will be developed respectively for the problem solving and inference system, the knowledge base management system, and the intelligent interface system. These prototypes will help review the specifications determined in the early stages, delineate the problems, and thereby determine the specifications for the prototype systems to be developed in the final stage.

Results of the research will be utilized during the research on architectures. Also, the support software will serve as an effective aid in the research on the basic application system. At the same time, the theory underlying the fifth generation computer systems, with the core language and the knowledge representation system will be given a concrete shape.

By the end of the interim stage, specifications will be finalized for the basic software systems to support the architecture and the hardware.

During the final stage, simulators will be developed for the target machines on the basis of the support system and the prototype machines developed during the interim stage. The purpose of these simulators will be to develop the software system satisfying the final specifications. During this stage, problems will be brought to light and the target specifications will be reviewed and streamlined.

Another objective will be to establish a systematic basic theory for the fifth generation computer systems.

Finally, the basic softwares will be combined to serve as the core for the fifth generation operating system.

As for the architecture-related themes, a machine based on the initial core language specifications will be developed along with the logic programming and functional machines. The machines to be so developed will serve as a model for the supporting machine to aid research and development on soft- and hard-ware.

A part from machines based on use of the new language, systems will be built up early enough to support development. Such systems will be easy to handle and will include the existing general-purpose machines for use in, say simulation.

To deal with the themes related to the new advanced architectures, important fifth generation computer functions will be research separately. During the initial stage, research will be undertaken on machines serving different functions to build accurate computational models and construct the hardware. In addition, the relation between the programming language and the system architecture will be elucidated. To this end, both the software and the hardware will be used to develop a simulator and collect the basic data required to lay down the primary specifications.

Interrelation between the respective themes will become clear with the progress in their study during the intermediate and final stages. By the time the language specifications are combined, it will be desirable to streamline the relations between the machines and integrate them under a new theme. Such integration may be effected between functional and data flow machines, or between functional and abstract data type support machines. This, however, will presuppose a corresponding combination between the core language and its computation model.

Some of the themes concerning the architecture are described below in greater detail.

Since the data flow machine is to constitute the core of the sophisticated parallel execution mechanism, it will be better to incorporate if as an essential component of the logic programming and the relational algebra machine architectures. However, till date the theoretical basis for the integration to allow this remains incomplete. Thus precise targets must be finalized on the basis of the research results up to the interim stage. Depending on these results, it may be possible first to develop the data flow machine as a functional language machine, and then to examine how it can be connected to the logic programming machine.

An innovative von Neumann machine will recombine tag and dynamic architectures and the associative memories, improving the conventional von Neumann machines for the purpose of VLSI, and constructing new machines. The innovative von Neumann machine will be used in the initial stage as the basis on which to construct the support machines.

Machines related to new advanced architectures will be developed according to specifications laid down in the course of the studies carried out on the corresponding basic software themes. At the beginning of the interim stage, an experimental machine will be developed on the basis of the primary specifications related to the initial basic software themes. Part of the machine will serve as a basic software development tool.

During the early stage of development of the distributed function architectures, a group of supporting machines will be constructed, conducting basic research on distributed function systems like local networks. During the interim stage, a distributed function network will be constructed to include a data base system. Techniques will be developed for distributed OS, network, firmware, and UHM* leading to machines featuring new advanced architectures and groups of soft- and hard-ware modules constituting the foundation for the basic software.

Final targets for themes related to the new advanced architectures will be set up by considering the secondary specifications laid down during basic software researches and development of the respective machines.

Targets should preferably be precised first at the beginning of the interim stage and secondly at the beginning of the final stage.

Items related to the new advanced architectures in the final stages should preferably be split into two: one for the logic programming machine supporting a problem solving and inference machine, and the other for a data base machine having in its core a relational algebra mainly supporting a knowledge base management system.

In addition, results of the study on the individual machines forming part of the new advanced architecture will be used in developing the hardware required to support the intelligent interface system.

In addition, other useful machines may come up also and the possibility of such development should be evaluated at periods when precise targets and laid out (indicated

* UHM: Universal Host Machine

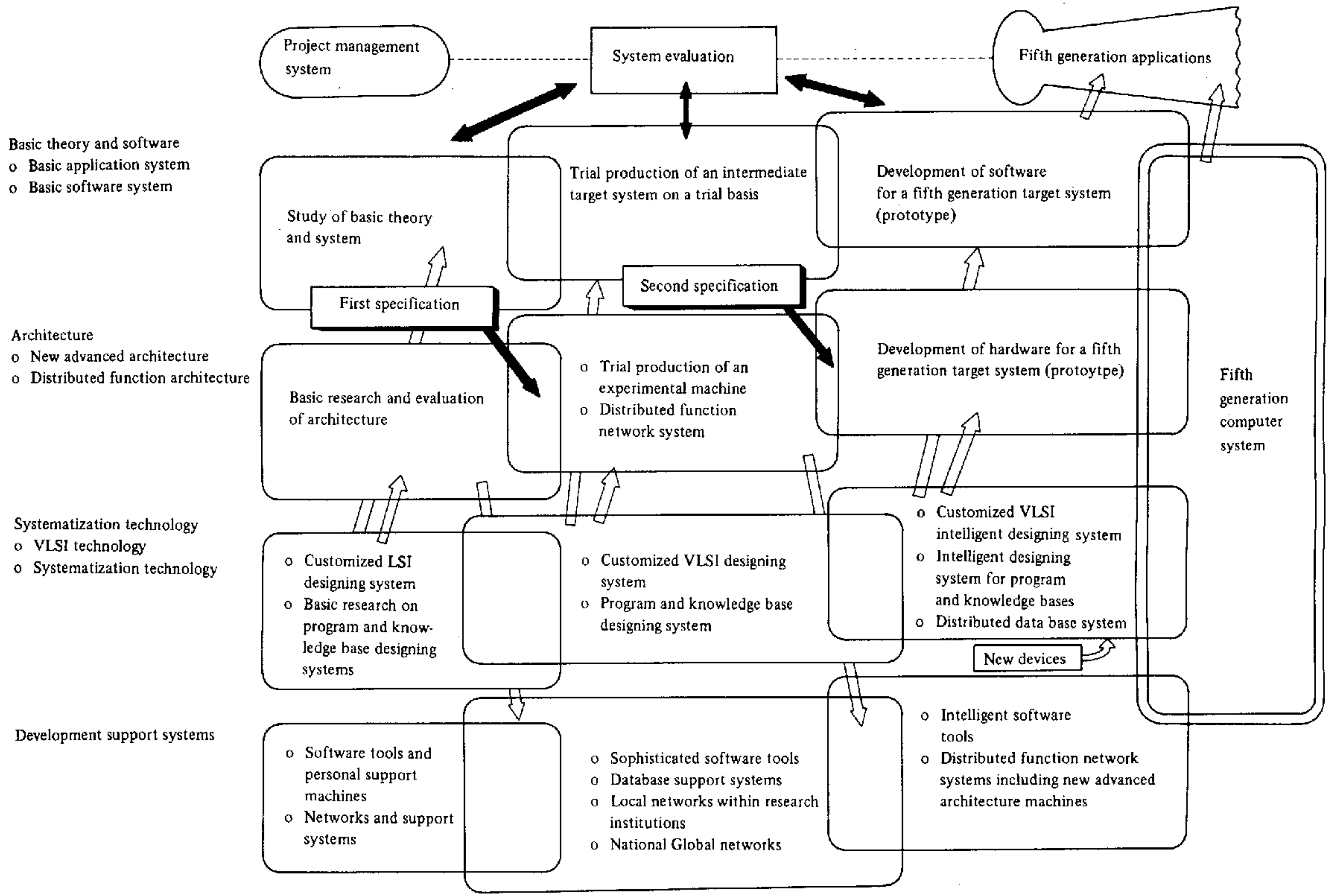



**Fig. 4-6 Concept diagram showing how research and development are to progress**

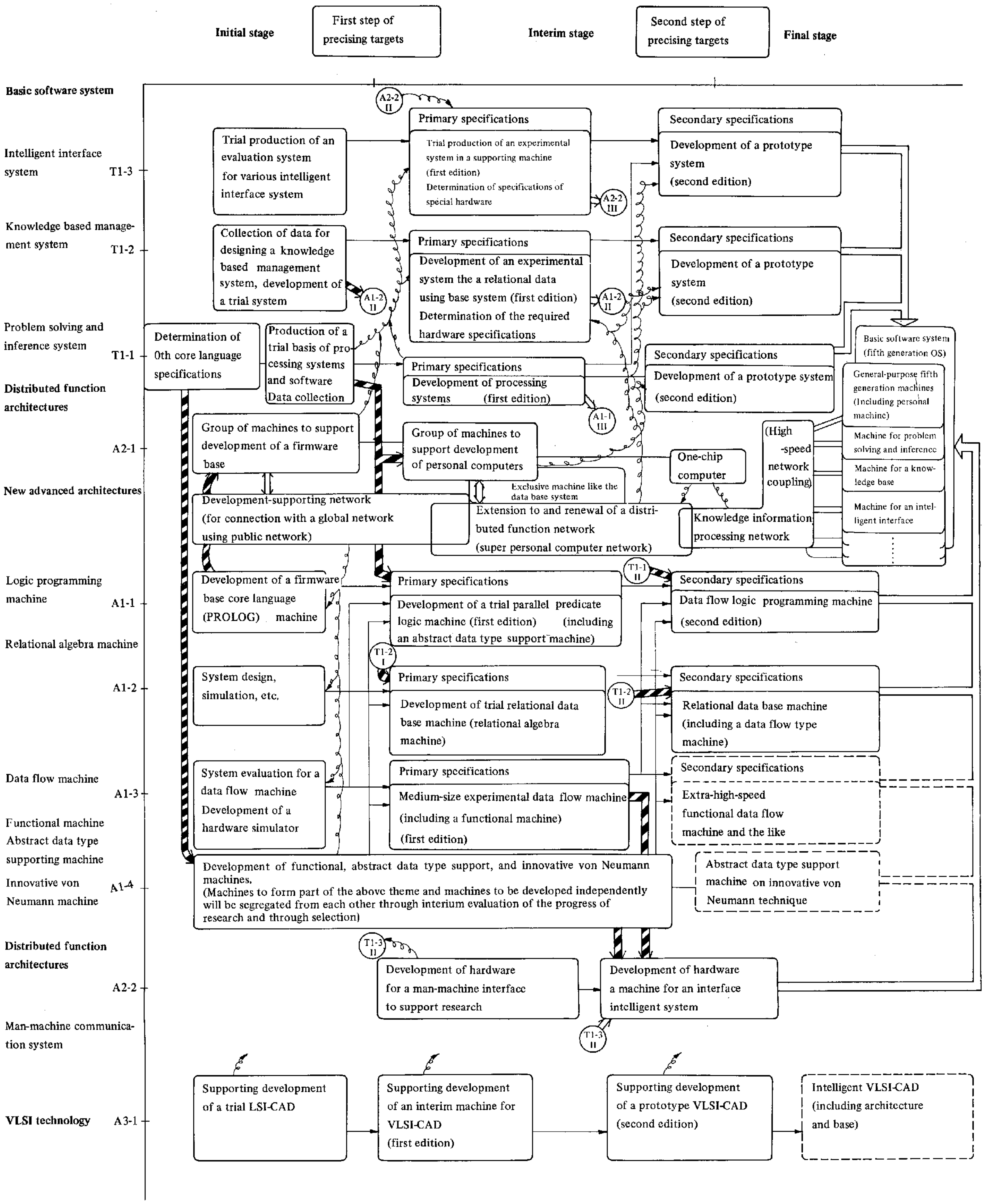



**Fig. 4-7 Steps in research and development of themes related to basic software and architectures**

: Related as a tool supporting development

: Related as a component determining specification

: Possible target (as a result of greater precision)

by dotted frames in Figure 4-7).

At the end of the final stage, themes rleated to, respectively, basic software and architecture will be combined to build up problem solving and inference functions, knowledge based management function, intelligent interface function, etc.

The hardware systems to be developed will comprise all ranges of machines, covering the general-purpose computers which have these functions, including their smaller versions, namely, the personal computers, and computers incorporating particular enhanced functions.

These fifth generation computers will be linked to a distributed function network system to be developed under the themes related to distributed function architectures. This will lead to the development of an overall knowledge information processing. Detailed specifications of software and hardware for the computers must be finalized by fixing precise targets at the beginning of the final stage.

Practical application of VLSI technologies will be indispensable in developing machines in conformity with the foregoing architecture. Finally, the themes related to VLSI will be taken up.

The first object concerned with VLSI technology is to fabricate customized VLSI chips in a short time. To this end, VLSI architectures and a VLSI-CAD system must be researched to support development of an experimented machine and a proto-type system. First, a trial VLSI necessary for the experimented machine must be built up during the interim stage, to be followed by, say, a CAD system to support the development of a prototype fifth generation computer.

The success with which this theme is pursued is of pivotal importance to the fifth generation computer systems. Research and development on this theme must, therefore, be carried out on priority basis with provision for adequate facilities. A simple interface to be connected to the supporting network must be available for the research on architectures, speech processing etc.

Research on this item will finally aim at development of an intelligent CAD system including the related architecture and data bases in the final target aimed at in this study.

### 4.4.3 Procedure for research and development of themes related to basic application systems

Roughly speaking, five items may be associated with the basic application systems, each deserving extensive research. For the best results, they should be taken up both as an interdependent scheme allowing mutual exchange of findings, and independently to give reins to creativity.

Progress of research and development on the respective themes related to basic application systems will be discussed with reference to Figure 4-8.

In the initial stage, a start will be made by studying the themes independently to delineate the basic systems, systematize basic data, and construct experimental systems. The following three conditions must be satisfied for this:

(1) Data must be accumulated by using an experimental research and development supporting system. Standardizing the programming language and a basic data accumulation format.

(2) Study results must be exchanged in view of the common information underlying machine translation, question answering, and speech applications.

(3) Innovations in basic software must be utilized progressively. Also, research on basic software must be closely followed up by a feedback system. With the support of the preparatory basic software system (serving as the foundation for the first specifications), data accuring will be fed back from time to

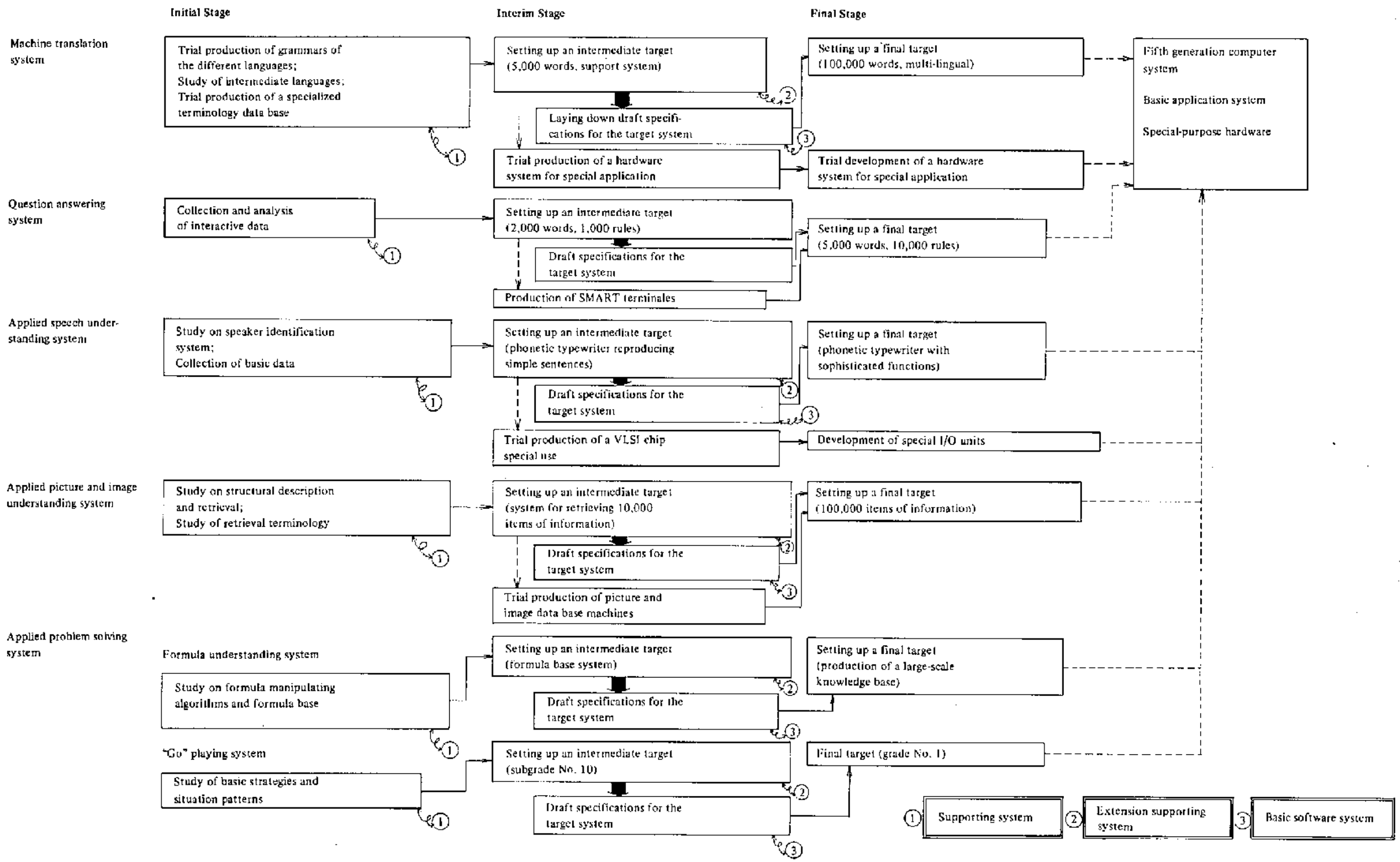



Fig. 4-8 Steps underlaying research on basic application systems related themes

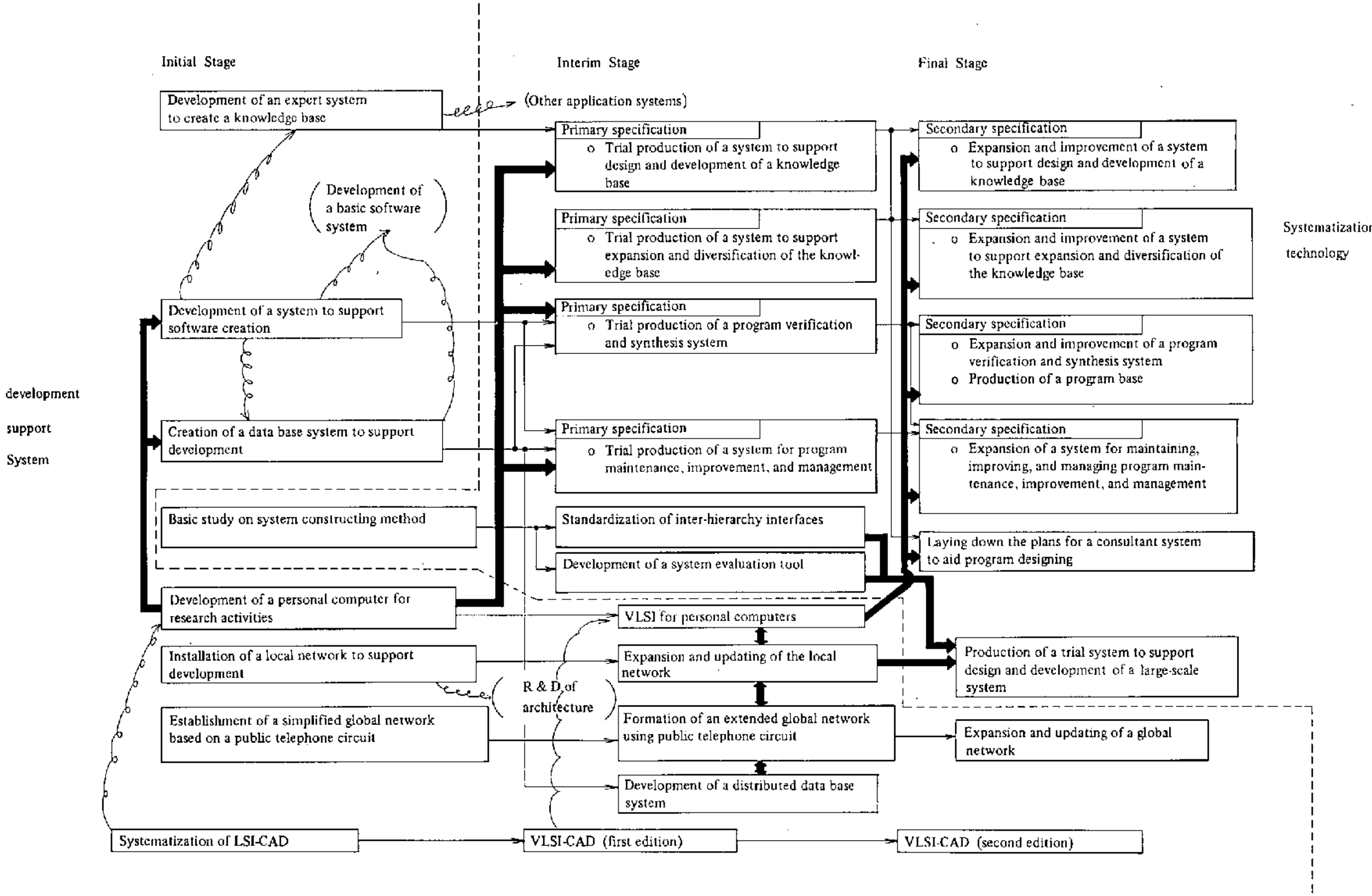



**Fig. 4-9 Steps underlying systematization and creation of system to support development**

time to ascertain the requirements that were not considered earlier.

Precise targets in the interim stage must be worked out on the basis of earlier results. Further progress must be based on a supporting system, which by this time will have been considerably expanded by making use of the advancements made in the basic software system. The intermediate target systems will not only fulfil the respective target specifications, but will, to a certain extent work hand in hand, forming a distributed function systems via a supporting network. For example, systems responding to speech, picture image or questions may be combined to built up an question answering system using speech and figures. Based on the experiences accruing as the intermediate target systems are constructed, final specifications for the basic software systems must laid down and adjusted to ascertain the final target system specifications in detail. Along with this, the special-purpose VLSI chips in the respective systems and hardware comprising, say, the input and output devices must be developed on a trial basis in harmony with research and development activities related to architectures.

Target systems will be developed in the final stage incorporating, as best as possible, the merits of the basic software system. Obviously, the basic software features need not all form part of the target machines. Where adjudged desirable systems will be built up independently.

### 4.4.4 Procedure for research and development of themes related to systematization technology and development support system

Figure 4-9 illustrates the steps underlying research on systematization technology and development support system.

Apart from the basic studies, research on systematization during the early stages will aim mainly at creation of development support system. Thus, research on systematization technology will center on the development support system. Final target in the process of development must be oriented to seeking newer avenues for systematization technology with the fifth generation computer system. Components found usable as tools in the course of development must be incorporated from time to time into systems to support the creation of the fifth generation computers.

Systematization technology includes device, architecture, basic software, and application system techniques, but over here the discussion will center on software. Systematization technology at the level of architecture in taken up under the discussion on distributed function architectures.

Research items related to software are oriented mainly to develop knowledge bases and the systems supporting the entire life cycle of programs. As shown in Figure 4-9, these items comprise:

1) A system to support design and development of a knowledge base;
2) A system to support diversification and expansion of the knowledge base;
3) Consultant system for designing programs;
4) A program verification and synthesis system; and
5) A system for maintain, improve and manage programs.

One of the research items concerning software and hardware systematization includes basic studies on system constructing method, standardization of inter-hierarchy interfaces, development of a system evaluation tool, and development of a system to support design and development of a large-scale system.

Development supporting technologies must be ready at the early project period to aid research on personal computers intended mainly for software studies, a local network to support the software and hardware for the personal computer, and LSI/VLSI-CAD. Preferably, a global network should be built up with the aid of NTT (Nippon Telegraph and Telephone Public Corporation)'s switched network.

The required number of personal computers for research and development must be mass-produced by referring to the results of the initial studies on logic programming machines (forming part of the research theme related to new architectures). The machines must be produced on VLSI basis by the middle of the intermediate stage. VLSI-CAD also must be developed by that time.

## 4.5 Research and development schedule

### 4.5.1 Research and development schedule

Table 4-4 furnishes the research and development schedule (see also 4.4). Section 4.5.2 below details the procedure envisaged for the respective items in research and development activities.

### 4.5.2 Research and devleopment procedure

(1) Initial stage

T1) Basic software system

1) For the knowledge base management system, studies will cover an expression system including a knowledge representation language, and methods of acquisition and application of knowledge, as well as control of distributed knowledge. These studies will emphasize on systematization of the basic theory, and collection of basic data through trial production of basic functions. Primary specifications must be laid down for the knowledge representation language.

2) For the problem solving and inference systems, specifications of the primitive core language must be based on predicate logic, which in its turn will center on PROLOG. This will serve as the language specifications for the logic programming machine forming part of system firmware and supporting research. Basic studies on must be carried out on the problem solving and inference system, combining the coding elements into a programming language and laying down the primary specifications for a core language. High-level problem solving modules must be developed tentatively and studied for use in the knowledge base management system and intelligent interface system. Results of modification and expansion of the core language specifications must be transferred immediately to a supporting personal computer to assist in developing other systems, and at the same time to collect data and analyze whether the specifications are acceptable or not.

3) For the intelligent interface system the approach will be to study of the methods as well as the basic theory encompassing natural languages, speech, picture and image systems. This will include a phoneme identification system an speech synthesiser, a parsing component, a sematic analyser, a picture and image analyzer, and picture and image displaying system. Basic studies must be carried out on knowledge representation and problem solving. To evaluate and improve these systems, the supporting system must be supplemented by tentatively developed systems. In this way the primary specifications will be laid down.

T2) Basic application systems

1) The machine translation system will be based on trial generation of grammars of the different languages, study of interim languages, laying down the guidelines for a basic system for trial production of a sentence and terminology data base, and basic data collection. These jobs will be performed with the help of a supporting system, creating, at the same time, a number of trial systems. This will be accompanied by study and design of suitable system hardware including sophisticated word processing techniques and a terminology data base machine.

2) For the question answering system,

## Table 4-4 Research and development schedule

| | Initial stage | Interim stage | Final stage |
|---|---|---|---|
| Basic application system | (1) Preparing the basic systems (grammars, recognition systems, description systems)<br>(2) Collection and systematization of basic data (conversation data, terminology data base, formula base)<br>(3) Trial production of an experimental system | (1) Setting up interim targets (a system to support translation involving 5,000 words: a specialized question answering system involving 2,000 words and 1,000 rules: speech understanding system handling simple sentences: a system retrieving 10,000 picture and image data items: a algebraic formula manipulation system using formula base: Go-playing system of subgrade 10 level.)<br>(2) Draft specifications for the final target<br>(3) Trial production of hardware for specialized use | (1) Development of final targets (a multi-lingual translation system involving 100,000 words: a question answering system involving 5,000 words and 10,000 rules: a sophisticated phonetic typewriter: a system retrieving 100,000 graphic and picture data items: a formula understanding system with a knowledge base: Go-game playing system at the grade level.)<br>(2) Development of hardware fo for specialized use |
| Basic software system | (1) Determination (initial stage) of specifications for a core language (such as PROLOG) prior to primary specifications<br>(2) Basic study on the respective software items and laying down the primary specifications<br>(3) Trial production of some of the systems and collection of data | (1) Improvement and extension of the core language laying down the final specifications oriented to machine software (secondary specifications)<br>(2) Trial production of a knowledge base management system on a relational data base machine<br>(3) Development a prototype intelligent interface on a development-supporting machine Hardware specifications for specialized use<br>(4) Drafting the new theoretical system | (1) Intermediate target machine simulator<br>(2) Target system specifications<br>(3) Development of a target system<br>(4) Improved processing of the core language through addition of say, knowledge base enquiry function<br>(5) Development of a final system, integrating it with the fifth generation machines<br>(6) Establishment of a new theoretical system |
| New advanced architectures | (1) Evaluation of the proposed machines through simulation, and collection of basic data<br>(2) Trial production of some of the experimental machines<br>(3) Development of the firmware-based PROLOG and LISP machine to support development | (1) Incorporating the results of initial evaluation into machines proposed<br>(2) Trial production of a medium-scale experimental machine (using VLSI of the first edition) Support to trial production of software<br>(3) Extending the scope of machine to support development of a firmware base and application of VLSI for personal use | (1) Finalization of the proposed machines and their specifications<br>(2) Development of a prototype and its combination with the software. Fifth generation prototype machine (use of second edition VLSI) |

| | Initial stage | Interim stage | Final stage |
|---|---|---|---|
| Distributed function architecture | (1) Establishment of a method of designing distributed function architecture design method<br>(2) Basic study on distributed OS and local network techniques<br>(3) Expansion of local network for development supporting | (1) Development of a distributed function network (super-personal computer network) including development-supporting machines<br>(2) Development of distributed OS and related software, and hardware like data base machines<br>(3) Trial production of UHM, supporting its use in personal computers | (1) Expansion to a distributed function network with the fifth generation prototype machine as its core<br>(2) Development of intelligent interface hardware |
| VLSI technology | (1) Support to the new advanced architectures with customized LSI<br>(2) Study and trial production of software for VLSI-CAD and organizing CAD<br>(3) Basic study on architecture for VLSI | (1) Support to the study of new advanced architectures with customized VLSI<br>(2) Development of a VLSI-CAD system and making it available for use<br>(3) Development of an architecture data base for VLSI | (1) Supporting development of a prototype with customized VLSI<br>(2) Expansion to intelligent VLSI-CAD<br>(3) Conversion of VLSI architecture into a knowledge base |
| Systematization technology | (1) Basic studies on the method of system construction (modularization hierarchic organization)<br>(2) Basic studies on knowledge base design<br>(3) Basic studies on life cycle management technique | (1) Studies on expansion and diversification of systems<br>(2) Standardization of interfaces between the hierarchic levels<br>(3) Design and development of a system to support systematization (vertification and synthesis of programs, supporting design and development of knowledge base)<br>(4) Development of a system evaluation tools | (1) Enlargement of a system to support systematization (support to expansion and diversification, design consultant system)<br>(2) Development of a design supporting systems and development for a large system<br>(3) Development of maintenance, improvement and management systems |
| Development supporting technology | (1) Installation of local network to support development and connecting the development-supporting machines<br>(2) Establishment of a simple global network using public telephone line and connecting the development supporting machines<br>(3) Expansion of PROLOG, LISP machines as development supporting tools<br>(4) Organizing LSI-CAD<br>(5) Development of software development support system (language processing, editor, data base)<br>(6) Development of an expert system to build up a knowledge base | (1) Installation of a local network applying the distributed function architecture (expansion and updating of the local network)<br>(2) Expansion of the global network using public telephone lines and installation of a data base station<br>(3) Supporting development with a superpersonal computer using a core language (first edition)<br>(4) Development and utilization of a distributed data base | (1) Supporting systematization by expanding the distributed function system and local network<br>(2) Expansion of the global network |

conversation data must be collected and analyzed, basic specialized data such as specialist's knowledge must be collected, and a system must be built up to handle and use such data. To proceed with this job, use will be made of the supporting system, accompanied by creation of experimental systems. Input and output devices suitable for the systems, such as the SMART terminals, must be designed and fabricated on a trial basis.

3) For the applied speech understanding system, a basic system for, say, speaker identification must be created besides collecting basic data. The supporting system will be used and various experimental systems will be developed. Also, the hardware for specialised use, and input and output devices must be studied and their basic design laid down.

4) For the applied picture and image understanding system a structural programming system, a retrieval system, and a basic system for fundamental research on retrieval language will be taken up. Various experimental systems will be built up and studied with the help of the supporting system. Specialized hardware for, say, picture and image data base machine must be studied and designed.

5) Under the applied problem solving system, the processing algorithm and the methods for knowledge base formation must be studied to build up a mathematical expression understanding system and basic systems like one capable of taking cognizance of fundamental strategies and situation patterns must be developed for Go-game. Various experimental systems will be built up with the help of the supporting system, and basic studies on the input and output devices for specialized application.

A1) New advanced architectures

1) Most of the basic design data must be collected during this period for the machines considered prospective candidate, to constitute the new advanced architecture. This will be done through simulation and trial production of the hardware (hardware simulation). For this purpose, an existing large-size machine must be used.

2) A machine must be developed to aid in basic studies on the software to process intelligence. This machine will be built up using innovative von Neumann technique to support PROLOG and LISP and constitutte the firmware base.

A2) Distributed function architecture

1) Experimental models will be used side by side with simulation to evaluate the methods considered in distributed function system design.

2) Using the local network to aid development, prototype distributed OS and communication control systems will be built up. These will help carrying out basic studies on the distributed function network.

3) A personal computer system will be developed with high resolution display and a picture input functions. Specifications must be laid down for the other components of the distributed function network.

A3) VLSI architectures and CAD

1) Enviornment must be set up for the proudction of customized LSI in the way of aiding, in the way of aiding, development of experimental machines of conforming to the new advanced architecture as well as the development-supporting machines.

2) Simultaneously, software and hardware must be built up to fabricate customized chips and data must be collected besides accumulating the software to develop VLSI-CAD.

3) Basic research must be carried out to develop the architectures for VLSI.

S1) Systematization technology

1) For the intelligent programming system, basic study must be carried out to delineate the guideline for laying down the

specifications, and to build up system for verification, synthesis etc. An experimental system must be established at the same time. Side by side with this, a sophisticated conversational programming system must be developed with the support of the personal computer to provide the methodology for research and development activities. At the same time, a systematic approach must be evolved to collect data for systems to be studied and developed. Primary specifications must be decied in conformity with the basic software system specifications.

2) For the knowledge base design, a basic metaknowledge representation, inferences and verification system must be studied, producing an experimental system for deeper insight into the problems. At the same time, trial systems must be produced with the support of the personal computer for single functions. They will help research and development, and as data is collected side by side. Primary specifications must be laid down in conformity with the basic software system specifications.

S2) Development supporting technology

1) A local network must be laid by using existing technique to support development activities. To this should be connected the PROLOG/LISP machines to support development activities as well as existing machines.

2) A global network must be built up on the public network to connect several research sites.

3) LSI-CAD must be connected to the local network to aid the researchers. By this time, local networks similar to the ETHER and CHAOS network of, respectively, Xerox corporation and MIT will be laid.

For LSI, something like CIF, the standardized pattern describing language of U.S. will be specified.

(2) Interim stage

T1) Basic software system

1) For the knowledge base management system, a prototype will be developed by using the tentatively constructed data base machine utilizing the data collected during the initial stage. This will be used in research on other systems and evaluation data will be collected to prepare a draft for the final specifications. During this period an attempt will be made this with other basic software systems. The framework of a new theoretical system (such as a knowledge theory, a knowledge representation theory, etc.) should be given.

2) For the problem solving and inference system, experience with the supporting system will be utilized to extend the scope of the core language. At the same time, greater precision must be achieved in the link between this system and the meaning and specification coding systems. A trial processing system must be produced from the trial inference machine, and used tentatively to collect evaluation data and prepare the draft final specifications. During this period integration of merits of other basic software systems must be incorporated this system. The framework must be established for a new system of theories.

3) For the intelligent interface system, a prototype must be built up, with the help of the supporting system and the trial machine to assist in research on the basic application system. The final specifications must be drafted by using the experience gained in their use. During this period, attempt may be made to combine this system with other basic software systems. Framework must be built up for a new system of theories (such as a recognition theory, an comprehension theory, or an representation theory). Furthermore, specifications must be laid down for hardware comprizing VLSI chips and input and output devices with enhanced functions.

T2) Basic application system

1) Using the support system, a small-scale translation aid system handling 5,000 words must be produced as part of the interim target. By developing and evaluating such a system, data will be prepared to help study specifications of the basic software system and the supports to the intelligent system, and draft the specifications for the target system. Data base containing the dictionary and the collection of terminologies must be expanded, leading to the formation of an intelligence base. Specialized hardware will be built up tentatively.

2) With the help of the support system, an interim system will be built up to accommodate a limited collection of vocabulary (2000) aux rules (1000) for specialized work. Development and evaluation this system will yield the data needed to decide the specifications for the basic software system and the supports to systematize intelligence. The system must be harmonized with other basic application systems, and detailed specifications must be drafted for the target system. The respective specialized knowledge bases must be expanded, incorporating greater sophistication. These will include system and components related to the fifth generation computer system itself. Specialized hardware including, say, the SMART terminals must be constructed on a trial basis.

3) For the applied speech understanding system, the support system will be used to develop an interim phonetic typewriter handling simple sentences. In the course of development and evaluation of such systems, data will be available for the study of specifications for the basic software system and the supports to systematization of intelligence. Detailed specifications must be drafted for the target system special-purpose hardware must be developed.

4) For the applied picture and image understanding system, an interim target system will be developed with the aid of the supporting system to handle about 10,000 picture and images. Development and evaluation of the system will yield the data needed to work out the specifications for the basic software system and the support to systematization of intelligence. Detailed specifications must be laid down for the target system taking into consideration the problem of harmonizing it with other system. Special-purpose hardware, such as a picture and image data base machines must be constructed on a trial basis.

5) As for the applied problem solving system, the interim target will be to built up a rule-based mathematical expression understanding system, using the support an aid. The interim target will also include productions of a Go-game playing system of about subgrade No. 10 standard. During the development and evaluation of such systems, data will be available for the specifications of the basic software system and the support to intelligence systematization. Also, detailed specifications must be drafted for the target system. Rule bases etc. must be made more sophisticated, and special-purpose hardware must be developed.

A1) New advanced architectures

1) An interim experimental machine must be constructed to comform to the new architecture. It must be based on language specifications laid down during basic software study. For such development, the first version of customized VLSI must be used.

2) During this period, results of studies on machines conforming to the new architectures must be combined, and the number of candidated for the fifth generation computers must be reduced from 6 to 3 machines.

3) The PROLOG/LISP machines developed during the initial stage must be produced on VLSI basis to build up a small-szie personal computer and improve its performance.

4) Part of the machines forming part of the new advanced architectures must be taken

up for the software study.

A2) Distribŭted function architecture

1) Based on the results of initial studies, a distributed function network will be developed. This will expand and update the supporting local network. In addition, the base of the software, namely the distributed OS, must form part of the architecture.

2) The supproting personal computer, data base machine, etc. must be interconnected to form a superpersonal computer network. The global network must be expanded packet switching network, etc.).

3) Continuous research must be carried out on the software for the distributed function network and machine components like the universal host machines.

A3) VLSI architectures and CAD

1) A CAD system for custumized VLSI must be developed to support research on the new and distributed function architectures.

2) Based on the study of VLSI architectures, function modules, circuits, programming languages including mask patterns, and processing systems must be developed. The coded items must be accumulated and systematized to develop the architecture data base for intelligent VLSI-CAD.

S1) Systematization technology

1) For the intelligent programming system, the interim target will be to develop a system for simple verification and synthesis, a small-scale program base, and a basic system for understanding programs. These systems will be combined to expanding the scope of the supporting system functions. Based on their merits, these must be combined with the basic software system to design on integrated basic system and to draft the specifications for the target system. Specifications, verifications and synthesis must be backed up theoretically.

2) For the knowledge base design system, the interim target will be build up a system with a small metaknowledge representation system, an inference and verification system, etc. Through trial and experiments, basic integrated system will be built up in conformity with the basic software system. Metaknowledge representation must be backed up theoretically. At the same time, scope of the must be expanded in the supporting system. Furthermore, experiments must be to collect a large volume of knowledge conducted.

S2) Development supporting technology

1) To exchange results of research on soft- and hard-ware, methods of modularization and must be standardized together with their interfaces.

Also, a data base system must be created to store the results obtained, and management must be systematized.

2) The distributed function and global networks must be used as tools to support development. Peripherals must be organized for the personal computers (PROLOG/LISP machine).

(3) Final stage

T1) Basic software system

1) A simulator must be developed for the target machine, using the interim trial machine or the software support to the knowledge base management system. At the same time, specifications must be improved and made more accurate. The system developed must be incorporated in the target machine to built up the basic application system. Based on the data collected in the process, more sophisticated learning functions must be supplemented for research on an extended system. A new theory must be established.

2) For the target problem solving and inference system, a simulator must be built up, using the interim trial machine or the support to the processing system underlying the target machine. The processing

system must be integrated with the intelligent programming system and incorporated the target machine to develop other systems. Using the data collected, more sophisticated functions will be supplemented for problem solving. At the same time, research must be carried out to improve the performance of the machine. Attempts must be made to establish a new theory.

3) A simulator must be built up to develop the target intelligent interface system. This will include construction of the hardware for special purposes. This must be integrated with the knowledge base management system as well as the problem solving and inference systems so as to constitute part of the target system. Based on the data collected, research must be carried out to provide additional sophisticated functions aiding comprehension. Attempt will be make to found new theories.

T2) Basic application system

1) The target machine translation system will be based on the final machine to handle 100,000 words and several languages. For this efforts will be made to combine this system with the basic software, organizing various sophisticated knowledge bases, special-purpose hardware must be developed and improved.

2) The final question answering system will be built up with the target machine as the base. This will deal with various specialized fields using 5,000 or more words and at least 10,000 rules. The developement must be matched with the basic software system as well as other basic application systems. Knowledge bases must be organized and made more sophisticated besides being increased in number. The target system includes an intelligent utility system.

3) The final speech understanding system will include, say, a phonetic typewriter understanding speech, developed as a sophisticated I/O device for the target machine. The system must be matched to the intelligent interface system.

4) For the applied picture and image understanding system, the final target will be to develop picture and image data base handing 100,000 items of information. An integrated retrieval system must be developed at the same time. These systems must be matched with the intelligent interface system as well as other basic application systems.

5) The applied problem solving system in its final shape will comprise a large-scale formula processing system in the target machine with a knowledge base storing rules and formulas, and Go-game playing system of, say, grade 1 performance.

A1) New advanced architectures

1) Prototype machines constituting the new advanced architectures must be developed according to the new language specifications (final) decided in the course of studies on the basic software. During this period the number of candidate machines for the architecture must be reduced to one or two by referring to the results achieved by the intermediate stage. The machines are to be classified according to their respective scales of performance. Some machines must be developed to conform to the applied software.

2) The second edition of VLSI must be used in the development of such prototype machines.

3) The architecture must be evaluated and at the same time, improved as application systems through combination with the software of the final specifications.

A2) Distributed function architectures

1) The distributed function network developed in the intermediate stage must be expanded. Also a knowledge information processing network must be constructed with the core machines constituting the new advanced architectures.

2) Special-purpose VSLI chips and proces-

sors must be developed to process speech, picture image, and natural languages. The development must accord to the specifications determined during studies on the basic application system.

3) Packaged software groups must be incorporated in the distributed function network to process knowledge information. Also, these must be evaluated and improved.

A3) VLSI architectures and CAD

1) VSLI-CAD must be used to create the second edition of VLSI required for the development of prototypes of new advanced architectures.

2) Development of chips for exclusive use in speech, picture and natural language processing must be supported in the same way.

3) The VLSI architecture data base must be expanded and linked to the CAD system. A prototype must be built up for the intelligent VLSI-CAD system on the basis of the study resulty on knowledge bases. At the same time, algorithm design, a circuit design, pattern design etc. must be integrated.

S1) Systematization technology

1) For the intelligent programming system, a consultant system must be developed by using the target machine to design programs on the basis of questions and answers in a natural language. The system must include say, a large program base and an improved systems with their defects removed. The development must be effected in a sequence that will allow it to be utilized for the development of the basic application systems. Theoretical basis must be secured for specifications, verification, and synthesis.

2) For the knowledge base designing system the target machine must be used to develop an integrated knowledge base design support system. This will include, among others a system to support expansion and diversification, and a system to support program execution. The development will be carried out in a sequence that will help its utilization during the successive stages of basic application system development. The system must be linked with the intelligent programming system to integrate the two as a support to supporting systematization. Attempts must be made to secure the theoretical base underlying metaknowledge verification, etc.

S2) Development supporting technology

1) For better flexibility, the interface specifications must be standardized at different levels to include the modules and I/O channels across the prototype machines.

2) Results achieved during the process of development must be organized and systematized to form a data base.

3) Methods and tools must be developed and made available for system evaluation.

### 4.6 Project organization and execution

#### 4.6.1 Basic principles to be followed in research and development activities

Features characterising the project will be set forth before the procedural details are discussed. Also, organizational details underlying the research activities and the procedure envisaged to execute the project will be discussed on the assumption that the essence of the project will not be sacrificed to facilitate its execution.

From the standpoint of organization and execution, the project features may be enumerated as follows:

(1) The project must play a pioneering role besides being creative in its approach

The purpose of the project does not lie in following or imitating the techniques developed by the other advanced countries. Instead, its aim is to achieve a target that will at once realize the social and technological ideals laid down for the future. One of the distinctive features of the project will be its emphasis on research and experimentation to set up this target.

(2) The project must look far ahead into the future

For the present, the project expects to achieve its target in ten years. At the same time, however, it contemplates a number of intermediate steps leading ultimately to knowledge information processing of a standard comparable to human brain. Corresponding to each to these steps, the project provides room for an interim target. On the other hand, the project looks forward to acting as a link with the future, and, therefore, attaches great importance to securely founded research activities. When contemplating the project, therefore, it is imperative to consider other projects also that may evlve out of it.

(3) The scope of the project must be wide enough to encompass the entire computer industry with all its ramifications, which, by the 1990s, will hopefully emerge as the mainstay of all industrial activities

In view of the natural limitations to human and material resources, the project will be content with a relatively modest but clearly described target during the initial stage of research and development. This, however, will be intended as a means to aid the progress of the project which will be implemented with a view to bearing upon the computers and the industry as a whole. Thus, it will be linked with, say, the studies under way on high-speed numerical computation systems for scientific and technical jobs relating to large-scale industrial research (large-scale projects) or current research to serve the techniques on which the next generation of industries will be established. In short, a wholesome development will be aimed at.

(4) The project must be an internationally oriented one

As mentioned earlier, this project is expected to play a pioneering role. This implies that the project will have a very important influence not only on Japan's own domestic industries but on other countries as well. If successful, the creative aspects of the project will make an important contribution to the international community besides, as it is hoped, elevating Japan's position in the world. As these suggest, the project must be oriented in an international perspective.

A project so distinctive in nature cannot be carried through without taking the following into consideration:

(1) It has already been noted that one of the most important topics for the research activities related to this project will be to set up the right target for it. The target must be an ambitious one in order to stimulate originality in research. At the same time, however, care must be taken not to overdo things by losing sight of the question of economic feasibility. Thus, the target must be finalized by striking a balance between demand and economics.

(2) Even if it is at the cost of ahving to cross numerous hardles, considering the long-term implecations of the project, a target may be set up if it is considered indispensable for the future. However, to lead the project along a sound course, it will be necessary to proceed to this target in bits by achieving more realistic interim targets established securely on economic and demand considerations. The interim targets so set up will include both conservative and ambitious ones, thought being given to the possibility of achieving them in parallel. In particular, during the initial stage of the project, attemps may be made to carry out parallel research on some topics not too far removed from the basic ones. On the other hand, subjects of research should be reorganized or integrated on the basis of strict yearly evaluation.

(3) Procedural aspects of research and the results thereof must be considered apart when

evaluating them. It must be understood that early detection of steps leading to a negative result is no less important than a successful end. Furthermore, evaluation emphasizing on originality is apt to lead to creative results capable of initiating further developments.

(4) The thinking behind the project must not lose its consistency during the long period of execution. To assure this, both the basic project targets and the ideas underlying it must be delineated in clear terms. In addition, strict guiding principles must maintained all throughout as the project is carried through.

(5) Success of a project of this nature depends much on the ability of the researchers. The more a research reflects originality, the greater is the likelihood of its having depended substantially on the researcher's own merits. Thus, it is important to appoint the researches from all walks of life, be it government offices, industries, or the academic world.

(6) Besides maintaining rigid guiding principles throughout the project, opinion of the individuals involved in the project will be honored, and attempts will be made to embody such opinion in planning. In addition, consensus of the researchers participating in the project will be obtained in each step, respecting their individuality and creating an atmosphere where they can exercise their originality. These steps are considered important for smooth running of the project. A feedback system must be established, creating a route for exchange of views through both top-down and bottom-up approaches. The guiding principles on which the execution of the project will rest must conform to the conditions prevailing in Japan.

(7) The fifth generation computers to be developed during this project will ultimately form the core computers in the 1990s. This will necessitate a transition from the machines of the preceding generations. All care must be taken to make this transition smooth and natural.

### 4.6.2 Desirable Research and Execution Principles

Further to the basic principles laid down earlier, all research and development activities in this project should, as far as possible, adopt the following norms:

(1) Consistent guiding principles

This is a project that ventures into untrodden grounds. It also is a project that will take a long time to execute. Accordingly, it must be based on consistent ideas with clearly delineated policies and targets.

The basic ideas underlying the project represent the point whence all the planning for its execution will start. These ideas should be set forth clearly in writing and no change must be made to them unless one clearly detects an error. This measure will prove effective not only if a project leader is replaced, but avoid possible deviations from the original stand as time passes even where the activities are led by one and the same person.

(2) Setting up targets

It is difficult to set up a rigid target for long-term research. On the other hand, as far as possible, the final target should be in full view throughout the project after it has been fixed through a strict selective process. In addition, interim, short-term targets must be set up clearly on concrete grounds, so that they may be referred to as the criteria to evaluate the project by. The final target must be fixed by embodying in it the latest developments in the related fields and amending it, if necessary, during the intermediate stages. A systematic approach is necessary for this, taking cognizance of the changes around and the maintaining up-to-date information on the latest technological developments abroad.

On principle, research will be scheduled by setting up and evaluating targets at three to four years' interval.

(3) Project systematization

This project covers an extensive area as reflected by the diverseness and number of topics taken up for related research as well as by the long period that intercepts its near-basic research stage and the actual development activities. In order to carry the project through, therefore, one must elucidate the relation of the individual research topics or developments to the project as a whole. Positive efforts must be made to modularize the approach and present the entire system clearly to the view.

During the initial stage a number of alternative techniques must be studied in parallel, selecting some and rejecting the others through a process of strict evaluation as progress is made towards the next stage. Each of these must be clearly related to the developments, the system being capable of applying the results attained in a particular stage during the course of the succeeding stage.

(4) Human resources

During its long execution period the project must provide opportunities for effective participation of researchers around the country. During the initial stages cooperation of the university researchers should be sought. For this a network must be established to connect the researchers around the country, equipping it with high-performance personal computers.

An organizational mechanism must be set up at the center to establish mutual contact between the participating researchers so that they cooperate with each other. This will necessitate appointment of really competent persons. For example, only persons with ability to lead must be selected as project leaders, playing a pivotal role in assuring uninterrupted progress of the project along the guidelines laid down.

(5) Role of organizational nucleus

To avoid interruptions to management and control as the project is carried through, the necleus of project organization should be constituted by an impartial body that will represent the government, industries, and the academic circles alike, without bias to any of these. Apart from drawing up plans and supervising the project, this nucleus will be responsible for fundamental technical research and investigations needed for project execution. It will pilot the project and will evaluate and select the techniques to be applied. In addition, it will lay down accurate specifications for research on each individual topic after thorough investigation and analysis. It will entrust actual execution of research to the makers, laboratories, and universities, but will, at the same time, undertake to supervise and evaluate the progress of such research and even offer advice as one of its essential functions.

Some of the activities expected of the organizational nucleus will include the management of components shared during the progress of the project, such as the different data bases, the CAD system, or the hard- and soft-ware tools that will eventually open fresh avenues for further advancement. At the same time, it will offer services in the related fields, and make it possible for the researchers in the different fields to exchange their views.

If required, the organizational nucleus will include not only a control center, but centers for, say, studies and investigations and services as well.

Besides the staff members appointed exclusively for it, the nucleus should include persons on loan from the government, industrial, and academic circles. This is expected to strengthen the bond between the different circles and ensure smooth running of the project as a whole. The group central to the project may comprise a small number of persons, but they should remain with the project along its entire course.

(6) International cooperation

Basically, international cooperation implies mutual effort by the different countries of the world to promote further advancements in information collecting and processing techniques. To this end, the project will, among others, encourage active participation in international conferences, make positve endeavours to attain standardization and sponsor periodic symposiums.

Emphasis must be placed on the importance of PR activities to avert misconceptions about this project in countries abroad and, instead, to stimulate enthusiasm in the different countries about its progress.

For a number of reasons it is difficult to make this an international project. Instead, it is desirable to execute it as a national project with Japan having the liberty to decide its course. It must be noted, however, that the project includes research on mechanical translation, mechanical interpretation etc. which must be carried out hand in hand with the other countries. In these fields, therefore, it will be found expedient to link the projects with similar projects abroad or arrange for joint studies where both the project researchers and researchers foreign enterprises will participate alike.

Specific policies must adopted to allow a constant flow of researchers to different countries, for this will pave the way for participation by outstanding researchers from the other countries during the extended periods of research organized under this project. This will not only orient the project in an international perspective, but will be effective in setting up a progressive and creative research atmosphere. The researchers must invited not only from the advanced countries but from all countries of the world alike, emphasis being placed on talent.

One way to encourage foreign researchers to participate in this project is to entrust research institutions abroad with studies on research subjects themselves. Before such a step is taken, however, the results it may be expected to yield must be considered carefully.

# II OVERVIEW REPORT

WHAT IS REQUIRED OF THE 5TH GENERATION COMPUTER – SOCIAL NEEDS AND ITS IMPACT

Hajime Karatsu

Technology consultant

Matsushita Communication
Industrial Co.,Ltd.
Yokohama, Japan

## Preface

The name "5th generation computer" which is quite attractive will be realized in the year of 1990. I was in charge of the committee to study the requirement as to what kind of function the new machine should possess. However handsome a sports car may be, it can not be driven with full power without the well-constructed highway. It would be easier to drive an ox cart than the motor vehicle on a rough road in the country side.

Higher performance with lower cost of information technology is expected to be utilized in the near future by the development of VLSI, optical fiber, communication satelite and so on. More than a million elements will be put on the silicon tip by VLSI technology. About 200km digital signals can be sent through optical fiber cable directly with no relay.
The project of my study group is:

1) What kind of social needs is expected in the new technology
2) To show the direction of the 5th generation computer
3) What impact will occur when the new machine is installed in the future society

## Social needs

In 1990, when the 5th generation computer will be in use, information technology will be installed and utilized as the key tool in almost all social activities, such as: economics, industries, academies, administration, defence, international relation, education, culture, human life of the people and etc.

Therefore, first of all, we must make clear or disclose the requirement for the new machine. Generally speaking, the way to give the answer to it might be to forecast the society of the next decade.

I gathered and analyzed the result of the forecast of Japan today which was done a decade ago. But I saw that there is few paper which hit the status of today. It is quite natural because the way to forecast used hitherto is to extend the past to the future, but the future is not the extention of the past but a matter to be created newly. Then I abandoned to predict the next decade by utilizing the method which was developed as the so called prediction.

I described the future society desirable for us a decade ahead, which is not predicted but a desired shape or pattern of Japan, and we know the social pattern at present.

Through these analysis, we can find out many kinds of bottlenecks to realize the society desired in the next decade. If we could overcome the bottlenecks one after another, the desired society might be realized. All the bottlenecks to be overcome is listed up as follows:

(These bottlenecks can be replaced with the words "social needs".)

## Bottlenecks to realize the society desired

a) Expectation for improvement in the fields of low productivity

The problem to be solved in Japan is to decrease the stress suffered according to the highly grown industrialization of this country. Japanese industry has received a reputation of high productivity, but it only depends on the field of manufacturing. On the other hand, we still can point out many lagging areas in productivity. This can easily be found out by comparing the price of every kind of articles in shopping and service.

The rising rate of the price of goods, supplied from factories where the productivity is greatly improved, is held down quite low during the last decade. The price of motor vehicles went up only 30%, and in the field of TV, camera and wrist watches it went down more than 30 to 50%.

On the other hand, in the field of service industry, which requires much handwork, it rushed up triple or more during the last decade in this country.

And today, the imbalance of the price came to the level which is not endurable. It is quite foolish when you pay the bill for dinner at the price which can buy 4 electronic calculators.

It is crazy!

This is one of the bottlenecks which Japan is expected to suffer in the near future, and the computer must be utilized to improve the productivity of the field left uncultivated, as follows;

- Office
- Engineering design
- Agriculture and Fishery
- Medicine
- Education
- Public service
- Government

To introduce the 5th generation, computer for these areas might bring the innovative development. In the field of manufacturing, we have much more expectation to realize high level quality and productivity by introducing the intelligent robot and no man factory. This is quite natural, because the competitive power of the industry can be shown by the figure of productivity.

Therefore, the demand to the computer utility from the industrial field will become much more serious, and the level of computer utilization might grasp the fate of the enterprize in several areas of it.

b) Concerned with internationalization of Japan

In 1980, Japan showed the figure of GNP $1,1 trillion which means the share of 10% of the world. Japan, the area of the land being only 0,3% of the world, population being 2,7%, with little natural resources, must live more prosperously and peacefully in the chaos of international relationship, To realize this desire in the future, the only way is to motivate and to activate the ability of each person in his own way.

Progress is brought by the wisdom or new ideas. which is the newly developed combination of the information or knowledge known in the past. After the 2nd World War, Japan gathered the new knowledge and put it into the society, industries, economics, administration, education and so on. Some were successful, and others failed. Japan was like the experimental field of new methodology. And today, it has proved which was true or false.

Therefore. we must endeavor to transfer the result and knowhow of our experience all over the world, and to search the way to get the leadership to live with speciality which is just fit to Japan. For example, shipbuilding, electronics, precision machinary and so on. To hold the priority in these target industries, 5th generation computer must play the role to realize strong competitive power.

c) The shortage of energy and natural resources

One of the most serious problems for the human being is how we can utilize the limited amount of natural resources efficiently and to what extent we can find out the new resources to increase the size of the pie. For us, Japanese, this is a fatal problem and can be the narrowest bottleneck to define the limit of the national economy.

To expand the size of the pie, it is quite clear that newly developed computer should become the necessary tool, to optimize operation of all kinds of energy consuming systems, to process the data to search the natural resources, to do R&D, the new energy. And Also the main frame of Japanese industries is expected to be changed from the type of high energy and resources spending to the knowledge accumulated type.

d) High age and high education (Aged and educated society)

Expected problems of the structure of Japanese society is the increase of aged people according to the low birth rate and high education. 96% of middle school pupils enter high school. 74 years is the average expected life of a just born baby.
This is going on each year and may become the bottleneck of Japanese society in the near future.

In the field of manufacturing plant, labour population is decreasing gradually. The workers hate dirty work in the shop floor. At the same time, social cost for medicine and for welfare are predicted to go up steeply, multiplying each other.

To overcome the crisis from these causes, conditions to introduce many kinds of innovative systems are absolutely necessary: the intelligent robot, no-man factory, life time education system to teach and to cultivate new possibilities for people to adapt to new jobs, to avoid the shock caused from the change of industrial structure, medical care information system for aged people and so on. These are expected to be realized effectively by using the 5th generation computer.

e) Information society and human beings

Japan, with more than 100 million people crowded in a limitted area, realized GNP much higher than U.S.S.R., and the society is operated efficiently, organized with high educated people in a large variety of occupation. Therefore, quite a bit of mismatching between the people and society might bring hazardous effect upon each other, which is similar to the effect that some slight trouble of a car on the highway might cause heavy traffic jamming.

From this consideration, it is quite natural for us to have some expectation for the 5th generation computer, that it can solve the problems to be brought from the mismatching between man and society. For this purpose, it might become quite important for man to hold in hand the way to access directly and to make dialogue with the systems which operate the society, and can move as he wants.

Many kinds of interactive video home terminals connected to the database through telephone line are undertested in every country at present. This might be one solution to soften the stress derived from the highly structured society. At the same time, it might be effective to keep the security of the fragile society, very complex and delicately constructed, against destructive attack.

Japan is quite a rare existence in the world, homogeneous in race, narrow range of income, peaceful with a low crime rate, and little trouble has occurred in walking into the information society. But the little trouble in the past does not promise little accident in the future.

Information society means that our life must depend on the computer more and more. Still today, the computer is a very hard machine which does not allow for us to make a single step of mistake. On the other hand, we, human beings are very soft. So, many troubles can happen between the hard machine and the soft human being.

From this consideration, the 5th generation must be the machine that should work to fit the human being intimately, contrary to the present way when human being has to approach the machine to match it and to follow the rule of it.

What is expected from the 5th Generation's Computers

Higher performance with lower cost will be realized in general, according to the progress of VLSI chip technology, fiber optics and communication satellite. We will be surrounded by all kinds of information utility, well arranged and easy to use.

a ) Existence like air

Everyone may utilize the information unconsciously in the future,dissolved into our society. Today, it belongs to only the specialized field and homogenized society by the result of mass production.

Even today, the fully computerized cash dispenser of the bank is installed into our daily life. We use it as the existence of an old friend. Some one said that cashless society may hardly be realized in Japan, because to use the personal cheque is not popular in this country. On the contrary, today, automatic transfer of every kind of payment through the computer is introduced in Japan as a natural phenomenon.

b ) In the manufacturing plant

Automation in the manufacturing plant goes to operation through 24 hours without worker and/or supervisor, which is a no-man factory, to recover heavy investment. All workers do their jobs only in the day time.

c ) To create consumption

The last dark continent, named by Peter F. Drucker, stayed behind modernization until today. Market channel had to be improved in efficiency and get an important status as to create and cultivate consumption of the people. To keep the production activity stable, it is a necessary condition that the consumption pulls the supply continuously. But in Japan, market channel is very complex and as a result, the cost is very high with low productivity compared with progressed countries.

Therefore foreign enterprise blame our complex channel as the no-tax barrier of the Japanese market. But the information technology has the possibility to upset the status from the bottom. The activity of market channel will change the performance of today, from carrying the products for the consumer to creating new consumption for the consumer by disclosing or adding the value of products.

To realize this new performance of the market channel, the computer will supply the tool for cashless, database and so on.

d) Multi-channel society

Until today, 2nd industry was supported by the principle of massproduction system developed by Henry Ford. The principle allows us to buy high technology goods at a very low price. On the other hand, this principle robbed from us the joy of selecting goods as we like.
The next generation computer will open the door to multiple choice.

Even today, on the assemble line of the auto manufacturer, combination of component of each car is changed according to the various requirement of the consumer. Even in the apparel industry, material is cut to fit the body of the consumer by the computer.

e) Medical field

Huge database system will be realized through the thunder storm of the problem of privacy. Computer is able to get new view or knowledge from the same data.

f) Regression to personal education

In the field of education, personal education can be realized easily by use of computer and may be an answer to solve the problem of drop-out students. Life time education is another problem, to match the aged society. For these purposes, CAI is expected to work well. But, today, the methodology of education is required to solve the social problems brought from aged generation.

g) To the government

It is hard to predict the pattern of our government in the future, but heavy deficit of Japanese government has come to the limit to be maintained. To escape from the law of Parkinson, new generations computer should work well. Concerning defence, dependence on the computer goes on and on.

h) Internationalization and Bargaining power

Development of Communication Satellite and optical fiber technique shrunk the geographic size of the earth. Japanese trading company, the Shosha, can not be realized without new telecommunication technology. The Shosha is quite a symbol of information oriented industry. New technology may give the possibility of information utilization not realized and not known till today. The new bargaining power of Japan might be brought from this unknown utility.

i) Public industries

Electric power, gas, highway, port, railway, so called infrastructure are the basement to define the productivity of social activity. Recently, the construction of infrastructure lagged, suffered by opposing activity from political reason. In some country, plan of nuclear power plant was stopped by opposing activity. Therefore we are enforced to use them most efficiently.

j) Each one

I have said that the new machine may be put into our daily work quite naturally, just as if it were air. From this point of view, technology of today is matured at quite a low level. Computer never operates by a single error. On the other hand, 5th Generation Computer should work correctly following the order written by ambiguous natural language. This is the popularization of the computer in the actual sense.

## The new computer system to be expected

To determine the new target of 5th generation, project must be directed to realize new performance. From the needs come application.

a) The non professional without training can handle the new machine. This must be placed at first position.

b) To realize the machine which can aid the creative work for planning and designing, we already have CAD system but the performance of it is still at low level. New generation of CAD is expected, man machine interface, image processing, flexible programming.

c) Walking into the unknown field, to generate the new concept or knowledge, high speed processing for simulation

d) To treat and extract the new knowledge from a mountain of data, knowledge engineering derived from heuristic programming project of Stanford Univ.

These targets are correspond to those of last year.

a) Non-professional use⟶easy to use system

b) Creative use⟶non deterministic job

c) Knowledge acquisition⟶super speed CPU for simulation

d) To handle knowledge⟶high intelligent system

e) To realize the new performance ⟶giant data base

Before talking about the detailed requirement, I would like to point out a difficulty to analyse and to get the conclusion. To show the requirement of the performance is rather easy, but it is quite hard to say what kind of function is required.

It is quite natural considering the statusquo in which the seeds develop the needs, in the computer field. It means that needs can be shown only after the possibility of technology is announced from the researcher and/or manufacturer of the computer. Through the discussion of jointed working group of each committee, the results are grouped in 5 items as follows.

## 1. Man-Machine Interface

The requirement to man-machine interface is divided into two items (a) input-output device, (b) conversational function.

(a) Many requirements to I/O device are concerned with graphic terminal and voice operation. Working group of CAD/CAE shows significant desire to large scale electronic drafter, traditional drafter may be replaced by it when we can realize the specification as follows, size 1m×1m, resolution 10,000×10,000 dot, price under 1 million yen.

Every working group requires 3 dimensional display, not only to show moving pictures, but also to full 3 dimension images, something like a holography from which our eyes can see the depth of the image.

Thin and small I/O display is also required to carry easily, and 3 dimensional display is required to be used as input device for use of design tool for bulky material.

I/O device for human voice operation is expected to make computer usable by every one. And special requirements are given from every kind of computer utility, supersonic holography input device for submarine robots, wireless handy voice operated I/O for salesmen and so on.

b) Requirement for conversational function is concentrated on two items. One is I/O media which is just fit to the ability of human being as he is. Second is the output function which can respond to the user faithfully, as we can accept I/O media as natural. For instance, symbol, table, picture, natural language, etc. Required conversational function is as follows.

1) To do dialogue effectively with ambiguous question

2) Guided Q and A system

3) Automatic error correcting

4) Learning function to specified people ( abbreviated documentation, automatic refining of output message to recognize unique expression, etc. )

To realize such high grade conversational function by the new machine, computer is required to be equipped with common sense. And through this analysis, the necessity of the theory to infer by common sense will be revealed.

## 2. Function and performance required to information processing

Requirement for function and performance is divided as follows: Processor main memory, secondary memory, channel as component and reliability, modularity as the system.

a) Requirements for processor and memory are divided into the fields of utility. Specialized high speed computer demands the performance of $10^3$~$10^5$ MFLOPS and 1M~20MW (1W =64 bits) main memory and required degree of freedom more than 100,000 points for simulation. For the "on line" data base machine 1ooM~10GB, and so on.

The detailed requirement is directed mainly to the architecture which is described in table B.

### 3. Artificial Intelligence

Requirement for artificial intelligence is divided into 3 items as follows:

a) The image of picture processing and recognition

b) To handle the image, the theme is concentrated mainly on 2 dimensional and 3 dimensional processing for engineering use, drawing drafts, automatic picture reading, editing documents and so on.

c) Voice and natural language recognition. The most keen requirement is the machine translation and conversational ability by natural language.

d) Concerned with this case, needs stay behind the seeds, but in the near future the knowledge base will become one of the most prominent utility for every kind of planning or designing job, engineering, marketing, administrative planning and so on.

Requirements for knowledge base summarized are divided into 3 items as follows:

1) Simulation for optimum design or plan

2) Forecasting, inspection. guidance for special job.

3) Extraction of law from raw material or editing knowledge to earn new perspectives.

### 4. Data base

Requirements for database come from utilization, designing of data base and system planning. Especially, many items to be solved are presented from the study of distributed data base.

1) Requirement for the performance of data base system

In concern with performance, the capacity must go up to $10^2 \sim 10^4$ times and access speed to $10 \sim 10^3$ times, but these figures should be studied more carefully. From the function of data base, flexibility and expandability are quite important to change the system efficiently and easily in order to be up to date.

2) From the user

Multi style data base is required which can handle not only character but also voice, picture, text of any style. Easy handling by human voice, conversational function are needed to be able to operate without special training.

3) To handle Chinese character

From the newspaper, keen requirement is presented for us to handle Chinese characters.

4) Distributed data base

There is no special requirement for distributed database from specified purpose, which might be the result of the fact that, still today, the user does not have the precise concept about the utility and merit of database.

### 5. Communication Network

Requirements for communication network are divided into 3 items: performance, service and security.

1) Requirement for performance

High speed communication service is required for use of multi mode data base including the picture, which may be realized by the progress of wide band transmission technology of optical fiber, band compression and multiplexing. And also improvement of the quality of transmission line is expected by introducing optical fiber cable and DDX network.

2) Requirement for communication service

a. Communication network, which can handle various kinds of signals, data, graphics, voice and so on.

b. International data transmission network service with high performance and high quality.

c. Satellite data link of wireless mobile service.

d. Communication service of various kinds of function is desired for the requirement which is expected to have much more variety of utility brought from the progress of I/O and computer.

3) Security and reliability

To keep the secret of information and data has become a serious problem by the widening of computer utility. At the same time, break down of communication system may bring hazardous result, unexpectable especially at the time of emergency.

Requirement of Theory Group (1)

Table A-1

| Item | Theme | Requirement |
|---|---|---|
| Conversational function | Natural media for I/O | · Symbol, Table, Picture<br>· Natural language<br>· Pattern to handle |
| | Conversational performance | · Ambiguous question<br>· Guided function to confirm<br>· question<br>· Error correcting<br>· Learning for specified people |
| Graphics | Graphic processing | · Drafting supported by machine<br>· Graphic reading<br>· Document including graphics and photos |
| | 3 dimension image processing | · 3 dimension graphics |
| | 2 dimension pattern recognition | · Signature<br>· Finger print |
| Language and voice | Translation by machine | · Multi-language documentation<br>· Simultaneous interpretation |
| | Dialogue with computer | · Voice and/or natural language Q & A function<br>· Programming by natural language |
| | Sentence processing | · Imput by voice and/or natural language |
| Knowledge base | Simulation of modeling | · Optimization of simulation model<br>· Decision making<br>· Layout of the cell |
| | Specific system | · Forecasting<br>· Management strategy<br>· Consensus support |
| | Basic function | · Rule finding from raw materials<br>· Summary function forgetting used data<br>· Discovery, storage and reconstruction of knowledge |

Requirement of Theory Group (2)

Table A-2

| Item | Theme | Requirement |
|---|---|---|
| Software | Programming | · Easy programming<br>· Methodology for large scale software ( Module structure )<br>· Auto-programming by program base |
| | Software system | · Integration of functions ( Co-ordinate CAD and simulation )<br>· Flexible software system |
| Data base | Data base system | · Expandability, Flexibility<br>· Auto designing and correcting of data module |
| | User interface | · Usable by non-professional |
| Communication net work | High performance network | · Intelligent electronic mail<br>· Remote conference |
| Distributed data base | Q & A function | · Efficiency<br>· Virtuality |

Requirement of Architecture Group　　　　Table B

| Item | Theme | Requirement |
|---|---|---|
| I/O device | Graphic terminal | · Electronic drafter<br>· 3D I/O<br>· Portable I/O |
| | Voice operated I/O | · Operatable by non-professional |
| Processor/ Main memory | Super high speed processor | · $10^3 \sim 10^5$ MFLOPS, 1M～20MW |
| | 3D model processor | · 100M～1GB on-line data base machine |
| | Small size array processor | · 7 freedom manipulator |
| | Personal work station | · 2MIPS 0.5～5MB with 100MB/1msec disk |
| 2ry memory | Large disc | · Several 10GB/10msec |
| | Small disc | · 100MB/1msec |
| | DASD | · 100TeraB (character file)<br>50 Million·frame light disk |
| Computer system | Modular computer | · Assembled by user |
| | Reliability, Availability | · Self-recover and maintenance |
| | Distributed function system | · General purpose CPU & simulator jointed through 2ry file |
| Data base system | Performance | · Capacity $10^2 \sim 10^4$/today's<br>· Special $10 \sim 10^5$/today's |
| | Specified data base machine | · For design, figure, pattern, distributed system |
| Communication network | Multi function data base | · 200,000 picture/day |
| | Quality | · Optical fiber, digital network |
| | Distributed data base | · Broadcast network with high performance<br>· High precision clock<br>· Protection for dead lock |

## Study of Social Impact

### 1. Introduction

Information technology, such as computer and telecommunication, prepare the tool to handle information just like the canvas and colors to paint a picture. Therefore, the result or impact of introducing it in our society depends upon the purpose as to what kind of target was selected and how we utilize it.

The society which will be brought by introducing 5th generation computer is hard to predict. But we have numerous things to be expected by realization of the 5th. First of all, the impact of the 5th, when realized, must work along the direction set at present.

In addition to it, it is rather natural that realization of the 5th generation might give many kinds of impacts to society and unexpected directions.

When the computer was first installed in the society, the utility was only the field of calculation. Therefore we were concerned only how it can realize the performance as the computation. At that time, no one imagined invasion of privacy might become a social problem.

In the case of the 5th, what kind of problem will happen and to what extent the effect will spread, the social impact must be analysed and predicted and the assessment must be prepared to avoid the so-called computer disease.

The robot plays an active part into the productionline of the factory. It developed to improve productivity and to free people from dirty work. However, in the U. S. and Europe, hesitation to introduce robot comes from unemployment problem. In Japan, robot is accepted as the most effective tool to overcome the problem of shortage of labor force and to improve the product quality as the result of robotization. It increased steeply the productivity of Japanese industry and international friction of trade occurred as its impact.

As you see here, modern technology scarcely stands independently on its own. Various kinds of impacts occur and change the style of our society. TV was invented for the purpose of sending pictures far away, but it changed the style of entertainment and gave influence to the way of thinking for the people. Such kinds of impact might not be expected by the engineer who developed TV technology.

The study of social impact of the 5th was prepared along the line in this way. The description is arranged at four items: social, industrial, international, and personal. However, these items can not be cut apart independently. But it is hard to write them multi-dimensionary on paper. I tried to describe interrelation of each item to be understood as well as possible not hesitating to overlap the same item in every paragraph.

One of the most important target of the 5th is that the machine comes to man by itself, which means every person can use it without hesitation or allergy. Therefore, the population to utilize the 5th will increase explosively.

History tells us that industrial revolution was derived to replace human muscle to steam engine. In the same way, replacement of human intelligence to machine is expected to be realized in the near future.

Therefore, the shock brought by the 5th generation computer must spread in a very wide range and new sociological description might be necessary to understand its meaning. However, I would like to talk on only 4 limited items in consideration with characteristics of this symposium.

### 2. In the Social System

Social System is the basic infrastructure where we live, work, and enjoy, together with atmosphere , aqua-sphere, biological-sphere. Therefore, to cultivate and to give new possibility to infrastructure is one of the big propositions to be given us to hold in hand the key to open the door to the next era of information society.

The infrastructure must be not only stabilized and full of welfare but also operated smoothly holding close relation with foreign area. All kinds of information and knowledge orientated technology will work effectively for it.

Japan meets problems to refine social system as follows:

1) To open and to widen the gate for citizens to join the activity to realize an optimum society.

2) To apply fairly the social rule or principle such as beneficial pay and pollutor pay.

3) To avoid the conflict originated from public protest.

4) To be free from so-called dirty work.

These are expected to be solved fairly by introducing the 5th generation and the walk for healthy and peaceful society will make great stride.

Otherwise, Japan will lose the economical activity of today suffering by the ailment accepted as the fate of advanced industrial countries.

Until today, the information technology was adapted mainly to the area of professionals. But in the near future, newly developed technology of very low cost and high performance will be supplied for all of us.

Through the new machine, we will be able to access directly to utilize computer power and data base. This will change the traditional interrelation between citizen and society. The only way for us to access the society was through senators, newspaper and demonstration.

The QUBE system which was installed in the city of Columbus, Ohio offered the way for direct vote to make conclusion to social problems. This may show the new tool to disolve social conflict.
The 5th generation can accept every kind of language speech and even picture, and summarize them to be understandable for each person, by the aid of knowledge base.

Therefore the 5th generation can be called the tool to give the ability of all citizens to join the social system more intimately, and as the result of it, the 5th will be effective to soften the social stress which comes from the modernization of environment where we live.

Another merit derived by the 5th is the improvement of the low productivity part of social system: education, medicine, aged people, public service, administration and so on.

In the educational field, CAI will actualize the personal education; each person can learn smartly to match the level of his or her progress.

In the medical field, progress will be made amazingly, remote medical check by telephone line will be accepted as quite natural in the near future; computer graphics may give easy diagnosis, data bank will supply all medical information when we suffer by accident far away from home and so on.

Notorious administrative jobs will be refined by introducing the computer which will be handled by natural language. All of this will help to make a small government.

When refinement of administrative job is materialized, the personel of the government, freed from hazardous routine work, can get enough time to think and to plan our future and will be able to get the reliance of all people.

## 3. Impact for the industry

Impact by the 5th generation will be quite big for 2ry industry and will change the structure of 2ry industries. At the same time, much more impact will be expected to hit 1ry and 3rd industries which is lagging in productivity today.

### A) Primary industry

In the field of primary industry, shortage of labor force and low productivity is expected to become a great problem. In concern with shortage of labor force, the situation is greatly improved recently by introducing mechanization of agriculture and fishery.

But, still today, we can point out a mountain of subjects to be solved in the field of weather forecast, survey of natural resources, recycling of resources, rationalization of sales channel.

To improve productivity in the 1ry industry, other way is required than is successful in 2ry industry, automatic machine, conveyor line, and so on. For fishing and forestry, the technology to survey natural resources and to cultivate them is required to be improved in a higher level, and a more accurate weather forecast is necessary for the recycling of resources and rationalization of marketing channel.

The new technology of computer will change the style of agriculture to food industry and we may be able to get perfect self-supplying food from our main island. The channel from the farm to green grocery will be rationalized effectively to cut the cost by use of the 5th generation.

B) 2ry industry

The top star to be built in the 5th generation computer is of course the 2ry industry and most eager to utilize the fruits of it. Therefore there is no doubt that the capital spent to develop the 5th' will be payed only by the output coming from the 2ry industry. Especially intelligent robots in production line and automation in office are expected to give innovative effect by it. ( c.f. contribution of each system )

C) 3rd industry

As more than a half of the labor population in Japan engage in 3rd industry today, the improvement of productivity of this industry is the most keen demand to maintain growth of Japanese economy. But, at present, productivity of 3rd industry of Japan is still in low level compared with the success of 2ry industry, and inferior states of 3rd industry push down the total figure of the productivity of Japan. This may bring a grave problem in the near future.

The weight of 3rd industry is departing from traditional physical labor, hotel, restaurant, to brain work, lawyer, accountant, consultants and so on. In the field of physical labor. rationalization by machine usage or automation will spread quite rapidly.

Expectation of improvement by use of computer in the 3rd industry, is quite big in the field of accounting, production and sales management, data base service and information processing etc.

a. Penetration into every job to save loss.

Of course, one of the main field of computer utility is for accounting. But until today, it was not popular in the field of small shop or factory prevented by the barrier of high cost and not being easy to operate without special training.

The 5th generation will eliminate the two difficulties of price and skill to the direction for low cost and to be operatable by natural language. Even today, we can buy personal computer for about ¥200,000 or more, performance of which is comparable to the middle scale computer a decade ago.

When VLSI will be in use, the performance of personal computer will go up steeply holding the price at the same level. High performance of VLSI will realize the conversational function by natural language. Therefore, computer power will be used almost everywhere in accounting, inventory, planning, and even in the stand shop on the street.

b. Information service

Service industry is sustained by the hospitality of employees. But progress of information technology is adding a new phase to it. All kinds of reservation systems are popular today, and computer can give consultation service to plan sight seeing and travel by use of conversational on-line terminal.

Seeing the change of these newly born service, the 5th generation will act as the tool to create new added value of this industry and at the same time improve its productivity.

c. Administration

At every time and in every nation, to improve the efficiency of government is pointed out as the problem and a small government is always desired. All kinds of physical work may be improved by introducing machine or automation.

On the other hand, improvement of public service is left untouched from the reason of its speciality. This field is unpredictable, requiring decision case by case according to complex law, regulation and check.

But the 5th generation is expected to improve the efficiency of such brain work, easy to access data base, and some kind of job may be replaced by machine completlely. It might be said that the area enjoyable by the merit of the 5th is public service.

Contribution of each systems

A) CAE system

CAE project is derived from the strong and various needs of all kinds of industries for the purpose of improving the productivity of brain work. All plan is to be placed as the program of rolling wave concept. Expected impact of CAE will be as follows;

1) To make minimum the risk which might occur when we attack a big project for the purpose of survival in the next century.

2) To avoid trade friction with developed countries. By softening trade friction, every country can hold their priority of industry specialized toward each other. For this purpose, we must improve the productivity of brain work in the favorite field.

3) To improve productivity of engineering department. Office automation may improve the productivity of the administrative job of white collar workers. As the result of it, engineering department might become the last black continent of the company. CAE is the necessary tool to cut it.

4) New frontier industry is expected when we get in hand the tool to improve the productivity of brain work. In the 1990s, we may establish new industries such as alternative energy, new oceanological resources. energy saving , new transportation system and so on.

B) DSS (Decision-Support System)

DSS is developed for the purpose as follows;

By the aid of
1) highly refined information
2) to support the thinking process

To get
1) better result for decision
2) shorten the time to decide
3) minimize the cost to get definite conclusion

To use the improved DSS will give us the merit as follows;

1) To improve consistency of decision process. By the DSS, we can utilize all information concerned with the subject given. Therefore, the answer from the system may be able to avoid biased result suffered from lack of adequate information.

2) Efficient decision with the group. In DSS, all people can communicate with each other easily and utilize the same data base and simulation programs with each other. Therefore, decision process of group is expected to become quite efficient by introducing DSS.

3) New industry integrated new information technology. The new communication tool and huge data base give us some idea that many kinds of new industry may appear in the next decade so-called total integrated information industry.

4) To introduce to personal life, DSS must give impact to each person. Still today, the so-called personal computer brings a boom in the market. The next step is to connect it to data base by telephone line and utilization of DSS through personal computer may be popular in the near future.

C) Intelligent robot

In the 1980s, it is said that robots for industrial use will become quite popular. The growth rate of robot utility is expected ten times till '90. But the intelligence ability equipped in these robots is still at quite a low level. Therefore at the end of '80, technical barrier might become clear according to its spreading in every industry.

To overcome the barrier,next generation robot will be required to have higher intelligence and a much more improved senser. By the diffusion of robot in our society, all kinds of dirty and risky work might be done without the human hand. And we can operate the work which was left untouched till today, work requiring heavy pressure, radiation, extreme high or low temperature, and so on. It means that the wall of the world which was untouchable for us will be stretched greatly. And innovation of the manufacturing plant has started already by introducing the robot as you know.

The robot will become a symbol of the machine which enlarged human power. Then we must develop the robot and study the impact by it every time to evaluate it correctly.

D) Office automation

In Japan, almost half of employees of enterprizes are engaged in office jobs, but improvement of productivity in the office is still at a slow pace. ( For example, in the electric machinery industry, total wage for workers per sale in the factory went up only 20% during the last decade. On the other hand, wage for office workers increased 60% per sale.) Office automation is expected to overcome such a ridiculous imbalance between factory and office.

## 4. Internationalization of Japan

An unhonorable reputation for Japan might be contracted to three points as follows:

1) What Japanese think is not understandable

2)Japan does not pay effort to defuse the information of knowhow to be utilized all over the world.

3) The Japanese is too shy to speak up his idea, thus breaking the atomosphere of a meeting.

The 5th generation computer may work as the machine which is easy to speak and hear foreign lanuages, and eliminate such language barriers. And foreign people may understand that the Japanese are not people who come from outside space but common beings as they are, and only barrier of language separate each other.

Therefore, the Japanese government should spend money to realize machine translation for the purpose to play a role in the world to avoid international conflict caused by language barrier. Global data base network will support it.

## 5. In concern with individual people

The 5th generation realized many new functions, conversation by natural language, widely distributed data processing, giant data base, inference by machine, automatic recognition and so on. In concern with individual people, many problems not experienced in the past may occur corresponding to introducing new technology.

We can hardly predict all of them. But, at present, we may point out 3.

1) About privacy
2) To cultivate new social order of habit
3) To adapt the change to new order

Today is called " Information oriented age " which is brought by information technology, processing, storing and sending. But still today, they are thrown to our society as it is, without order.

Expectation for the 5th generation is to give the tool to every individual to access and to utilize information freely and fairly. But at present, information utility is given from only supplier and it cannot be used without some special training.

Therefore, we should pay effort fairly to get in hand information and to understand it correctly. We can hardly assure if the result is useful or not, people might feel suspicion for information which has come from a machine. One may hesitate, another may select it as he wants, and the other turn it down .

This is unfortunate, seeing from the splendid progress of technology. One of the target to the 5th generation is to change completely the relation between man and machine. We can use the machine without a terrible keyboard. Every body can make conversation with a machine without special training. Even when we use natural language unbiguously, the machine will understand the order correctly, and give answer just fit to the question.

Until today, man had to walk toward the machine with effort to fit precisely. On the contrary, tomorrow, the machine itself will come to follow the order issued by man.

## AIMING FOR KNOWLEDGE INFORMATION PROCESSING SYSTEMS

Kazuhiro FUCHI
Electrotechnical Laboratory, Ibaraki, Japan

### 1. INTRODUCTION

In contemplating fifth generation computers, what role should "basic theories" play? If there were already an image, the role would have been to provide the generation with theoretical foundations. However, at the present stage, where that image is still being sought, a more aggressive role may be anticipated. Will it be possible?

When computers were born, there can be no doubt that substantial contributions were made from the theory side. Achievements not only in electronics (called in those days electron tube technology) but also in mathematical logics and neurophysiology were skillfully put to use. With regard to logics, Turing's theories contributed to establishment of an image for universal machines, and McCulloch-Pitts's physiological model was closely associated with the philosophy of logic elements. These were cleverly synthesized by J. von Neumann and others, to form the basis for today's computers.

Since then, however, the development of computers has rather been technological and autonomous. Automata theory, theory of formal languages and others emerged but they have not directly influenced the progress of computers. Rather, computer technology can be said to have created more diverse phenomena than the theories existing then could have predicted.

It was early in the 70's that the phenomena were properly formalized (theorized). This formalization belonged in the field of mathematical theories of programming. This movement is still progressing, and is not as yet complete. It may be described as an intermediate stage in organizing the creations of computer technology. On the other hand, in this an image for new machines (new architecture) seems to be gestating.

Independently, at least on the surface, of mathematical theories of programming, since entering the 70's there have been gropings toward a new machine architecture. Noteworthy amongst these are research on data base machines and data flow machines. The reason for this particular attention at this time is that such machines have started exhibiting progress that synchronizes with the progress in mathematical programming theories. The synchronization, however, is not necessarily clear under the conventional philosophy on architecture, and perhaps because of this factor, has led only to spontaneous proposals. It is felt that under a new concept many of the proposals made so far can be integrally organized.

Many voices are raised in dissatisfaction over present-day computers. One of the complaints which perhaps calls for introspection on the computer side is that today's technology is far from the ideal of being truly "handy" for users.

One of the factors concerned with "handiness" is the interface between man and machine. Natural media for communications from the man side are primarily graphical communications and conversations in natural languages. To realize these communication objectives calls not only for expansion of in- and output media, but also for higher performance capabilities, necessarily, of the system itself.

Meanwhile, another factor involved in "handiness" apart from sophistication of communication media, is that functions inherent to the system need to be upgraded. This may sloganistically be termed integration of "knowledge".

With current computer technology, a majority of solutions for a problem must be "programmed". If information relative to the object areas of the problem, plus preferrably laws governing those areas can be integrated, it will distinctly improve the problem solving capabilities of the computer system. Information of this kind corresponds to "knowledge" on the man side.

Integration of "knowledge" is one way of achieving "handiness". This problem at the same time is bound up with such sophistication of communication media as natural languages. On the man side, too, languages and knowledge are caught up in an inseparable relationship.

Incidentally, the subject of realizing high performance capabilities by way of knowledge and languages was, in the 70's, the main theme of research into so-called "artificial intelligence". Of course, the research has not solved this problem. It is necessary to organize the achievements of research efforts in the past, and to develop them still further. At the same time, here is another material to be added in contemplating fifth generation computers. If basic theories are not limited to "mathematical" theories, this material may be included in generalized basic theories.

Research into artificial intelligence has not been aimed at mere realization of high performance capabilities. Awareness of the mechanisms needed to realize it with necessarily accompanies it. For instance, take the subject of a "programming language" for artificial intelligence. Its finalized form has not been established, but proposals made on the basis of such awareness are rich in suggestions for new machine organization.

Of much interest is the fact that such proposals have started resonating with proposals made separately based on mathematical theories of programming. The imminent emergence of a common image from research efforts thus far promoted in separate fields of activity, seems to portend a new computer image for the future.

Judging from basic research achievements accumulated in the 70's, progress in hardware technologies, and other related situations, the time now seems to be ripe to plan new computers together with a new system of information processing.

If a bold proposal were to be made at this point, it might be summarized as: "realization of knowledge information processing and knowledge information processors (inference machines)". This is to colletively pursue the feasibilities of a new architecture (non-von Neumann), and of the utilization of language information and knowledge information.

Such a proposal at present may frankly be entertainable only as precognition or hypothesis. However, whereas the fourth generation is generally believed to be a continuation of the present generation, in contemplating the forthcoming fifth generation computers, such a discontinuity in ideas also appears to be inevitable. In that planning stage, it is thought that, views based on basic theories will provide the required guidance.

Many stages must be stepped through to realize an ideal, but it now appears to be time for the first step in that direction (for a trial). The currently dwindling optimism for continued progress in present-day technology is believed to signify a certain type of maturity in and plateauing of conventional technologies. It is causing some pessimism concerning the advancement of generations, but at the same time it may signify that the opportunity for the next, new generation is ripe.

## 2. INTERRELATIONS BETWEEN VARIOUS BASIC RESEARCH EFFORTS

The 70's may be viewed as an era in which computer-related technologies each exhibited interesting advances. In the latter part of the decade, interrelations between them became more and more apparent. These can be seen as the advance preparations for the overall development to come in the 80's which will eventually bear fruit in the 90's, and permits anticipation today of a new generation of information processing technology and a new image of the computers which will form its core.

Advances in LSI technology, though perhaps not revolutionary technically, have certainly been outstanding, with their impact expected to be revolutionary in providing the groundwork for a new era. It should be the responsibility of the new era to find out how best to take advantage of such advances.

### 2.1 Trials for New Computer Architecture

Concerning computer systems as a whole, distributed processing and building of networks were developed in the 70's as a new direction. Meanwhile, the principles of the internal structure of computers themselves have not changed much since von Neumann's invention. Besides, various situations kept them somewhat tied down to "standard architectures" such as that of the IBM S/370. Changes in the philosophy of system architecture such as distributed processings have the potential to affect computer architecture itself, but have not manifested themselves as yet.

Study of computer architecture itself has long been going on. Such proposals as associative processing, parallel processing, and varriable structures were made some time ago. Procedure proposals, such as stacking, tagging, and hashing, have also been numerous. Architecture philosphies based on high-level languages have already been proposed, too.

However, even though some procedures (such as stacking) have acutally been utilized, most of these proposals have not as yet borne fruit. Prototype machine based on a from-the-bottom-up philosophy has faced programming and other difficulties, and failed to gain favorable evaluation. Meanwhile, high-level language machine philosophies have failed to prove that they are particularly advantageous when implementation were limited to conventional ones, and since the structure of the language employed itself was predicated over a von Neumann type, the effort has essentially been cyclical.

So far, it rather appears that the strength of von Neumann's system has been reconfirmed. Taking this opportunity, let us take a little time to review some features of von Neumann's system that represent the basic principles for all universal computers in use at the present time.

The basic structure of von Neumann's system is in the combination of uniformly structured memory devices and simply structured arithmetic and control units. Realization of complex and sophisticated functions is left to "programs". A program consists of a group of simply structured instructions, and the program itself is stored in uniformly structured memory devices, with its execution being serial and simple.

The inherent property of von Neumann's system is in realizing universal machines with simple structures. It was in conformity with the electronics technology levels of those days as well as those of up until recently. For instance, if he had started with a machine capable of directly executing lambda calculus, realization of universal computers would have substantially been delayed. Since his days, various extentions have been attempted based on this simple structure.

Both the advantages of von Neumann's system and the complaints against it in some quarters stem from this structure.

The most significant of the latest complaints is the complexity and difficulty in programming. The difficulty is especially amplified with large-scale software (programs). The von Neumann system, despite various improvements and extentions, essentially leaves realization of high-performance functions up to software.

With regard to programming, employment of high-level languages has hitherto been promoted as a solution to these problems. Even here however, compilers and other language processing systems are required, inviting cumbersome software.

Meanwhile, parallel processing and associative functions have been considered for higher performance capabilities and more sophisticated functions. Ironically, however, prototypes based on these philosophies invited even worse difficulties in programming. Array architecture and matrix architecture computers were test used for ultra-high-speed numerical computations. Even viewing these as dedicated machines, the difficulties in programming were overwhelming. Regarding associative functions, technical difficulties in realizing associative memory devices for them have helped aggravate the difficulties.

This situation is about to change. On the one hand, the outstanding LSI advancements have started making realization of complicated hardware feasible. At the same time, in attempting to take full advantage of this, the von Neumann system is beginning to be seen as a bottleneck. On the other hand, new ideas have emerged concerning architecture. And furthermore, concerning programming, too, introspection on its styling has recently been going on.

With regard to architecture for parallel computers, data flow type architecture has lately drawn much attention. The data flow philosophy describes the computing processes along with the flow of data rather than centering around controls. Parallel operations are described naturally along with flow of data, which then are intended to be executed by hardware. However, as it is, this development will end up as dedicated processors. In moving toward a variable architecture, Dennis showed one based on a message-exchanging philosophy. Here can be seen a fusion of parallel processing and variable architecture philosophies. In its architecture, it is a totally different system from von Neumann's.

Of particular interest is the recent emergence of yet another fusion with a philosophy from the software engineering side. Data flow is based on a concept of function. On the other hand, as will be described later, software engineering proposed new-style programming to side-step the difficulties in conventional programming. One of these was Backus's functional programming, which was to link with the data flow machine philosophy.

From the conventional fixed idea concerning difficulties in programming for parallel machines, this was an epoch-making change in situations. Backus's proposal itself, as a non-von Neumann programming system, is not necessarily predicated over the existence of a suitable hardware. However, it links with the hardware architecture represented by data flow machines.

Actually, as a functional programming language, there is a single assignment language (SAL) that is a natural intermediate language, and this SAL links naturally with the data flow language. SAL itself conforms to a language that was proposed as an easy language for verification by software engineering from an entirely different motive than for data flow machines.

SAL, furthermore, resembles what has been derived from research on inference systems (connection graphs). The data flow machines made by Dennis and others were initially considered for use in numerical computations, but at this point a still greater potential for them has been found. It is feasible to extend data flow machines to inference machines.

This, as will be described later, covers logical programming philosophies and formal specification languages, includes subjects on relational data bases (and data base machines based on them), and furthermore exhibits a potential for linkage with the semantics of natural languages. Whether such machines will be proven remains to be seen, but this concept is an extremely attractive one. If such machines were realized, all kinds of subjects could be integrally and universally organized including machine levels.

## 2.2 New Programming Styles

For computer technology, a matter of increasingly greater concern lately is the problem of productivity for software (programs). The share of software in the total cost of computers has been increasing and now represents the majority. This is not only due to an increased number of coding volumes. It is due to the increasingly greater labor required to guarantee program quality.

One requirement is to guarantee the "correctness" of programs and another to develop procedures to produce correct programs. It is a major problem for software engineering to establish a system for (correctly) producing programs in accordance with requirement specifications.

To clearly present "specifications", the problem of specification writing methods (the language for specifications) must be overcome. For this purpose, a higher-level language than programming languages will be required.

Supposing a specification is clearly given, if a correct and efficient program in accordance with it could be automatically synthesized, this would be ideal. However, realization of the ideal is still away in the future. Prior to that, for instance, a procedure will need to be established to build up the program by stages while guaranteeing its correctness. This may be restated as a matter of program manufacturing methods (styles).

Since Dijkstra's proposal for "structured programming", there has been frequent introspection over programming styles.

Given such awareness of the problem, are presently used programming languages appropriate? Introspection over styles is accompanied by introspection over how programming languages should behave. Recently, out of this stream of philosophizing, emerged proposals entitlable: "logical programming". One is a proposal for predicate calculus programming, and another a proposal for functional programming. As a language, the former employes a predicate calculus form, and the latter is based on function logics. These proposals recommend use of such logic forms themselves for programming languages.

These correspond to the two streams in formal specification languages (the function and predicate calculus types), while they also urge introspection over how specification and programming languages should behave. Formal specifications may be regarded as generalized (highly abstract) programs. These abstract programs are made concrete in accordance with system requirements. If this transformation could be effected in the same language (logics), it would provide many advantages. Research efforts are beginning to get under way along this direction. Such a philosophy may possibly link in the future with the "data abstraction" philosophy which is also one of the main streams in software engineering.

"Program transformations" need to be effected while guaranteeing correctness (identity of meanings retained). "Verification" (as a system) is also one of the major subjects concerning software engineering. However, if for the programming language, a language normally used is employed, various difficulties will arise. For this reason, easily verifiable programming languages are being proposed, for instance, LUCID by Ashcroft and others. This is a kind of SAL, described previously.

Functional programming languages and predicate calculus programming differ somewhat in style, but in fact there is a natura lrelationship between them. LUCID and other SALs (and extensions of them) have emerged as intermediate languages in this correspondence.

SAL links with the data flow machine philosophy, as stated previously. This leads to the concept of "inference machines" which are extended data flow machines.

## 2.3 Semantics of Programming Languages

Verification of the correctness of programs may be restated as confirmation of identical "meanings". Therefore, the "semantics" of programming languages needs to be established.

It may be inferred that with "artificial" languages such as programming languages, definition of meanings should be simple. In reality, however, considerable difficulties have been encountered, even historically. The syntax of artificial languages can be expressed as "artificial" (simple), but their meanings do not seem to be artificial. The situation parallels the difficulties in the semantics of natural languages.

From the 60's to the 70's, a number of trials in semantics were conducted. One was operational semantics. In it, an abstract machine was imagined, where the meanings of languages were to be provided by descriptions by an interpreter. This was developed in the late 60's as the Wien method. Using it for verification, however, faced great difficulties.

In the 70's, axiomatic semantics were started by Hoare and others. This was a development of Floyd's theory with respect to flow charts in the latter part of the 60's. In this case, selection of the language to be axiomized became a problem. At the same time, the "correctness" of the axiomatic system and inference rules presented a theoretical problem.

In parallel with the above, denotational semantics was developed by Scott and others. This was intended to accurately and mathematically designate language semantics based on a model theory. Scott's theory may also be considered to support axiomatic and operational semantics. With this as a base, the "mathematical theory of programming" recently started developing.

One of the conclusions drawn from these efforts is that normally used programming languages are difficult to handle through semantic approaches. Is it because the semantics are constructed inadequately? Particularly interesting is the fact that items for introspection concerning programming styles which were intuitively pointed out independently of such theoretical approaches happened to coincide with the difficulties in semantics.

In structured programming, for instance, usage of "GO TO" was considered to be harmful. Similarly, its semantic construction is also difficult to analyse.

These situations corroborate the directions for new programming styles and the new programming languages to support them. As Backus pointed out, the difficulties found in conventional languages stem from the von Neumann type machines that they are dedicated to. The new direction is a changeover to non-von Neumann type languages, which leads to expectations of non-von Neumann type computers.

The semantics of the functional and predicate calculus languages previously described are rather clear cut. These, furthermore, link with non-von Neumann architecture such as data flow machines.

## 2.4 Relational Data Bases

Development of data base systems was one of the core elements during the progress in the 70's of computer technology. How to organize and how to utilize gigantic volumes of data wered the questions. The progress was made by accumulating experience. Along with it, efforts to organize such experience theoretically also went on.

Codd's proposal for relational data bases was made early in the 70's, but is only now about to become a major stream in structuring data bases. This is based on a theory of "relations". As query languages for the data bases predicate formulas (relational calculus) and functional formulas (relational algebra) are proposed. These are mutually interchangeable. They can be regarded as a certain kind of special logics, and through the 70's a great deal of theoretical research effort was made in this area.

As described earlier, proposals for logical forms are being made from the standpoint of programming. The philosophy of relational data bases closely is linked with this. Relations between, and the fusing of the two are becoming very interesting, as the themes of latest research.

Research on "data base machines" based on relational data bases was aggressively carried on in the 70's. None of it, however, seems to have attained a usable level as yet. Though relational data bases are theoretically clear-cut, their prototype machines seem to have been structured through adhoc approaches only.

Data base machines, however, are predicted in many quarters to become the buds which will flower as the new architecture machines of the future. However, research on data base machines during the 80's will have to be substantially stepped up. Research archievements toward this objective that are worthy of note have been emerging lately.

As is predictable from the relations will logical programming already pointed out, these were expected to be associated with data flow machines. In fact, a proposal for data flow type data base machines (relational algebraic machines) has quite recently been made (Y. Tanaka). This is still at a dedicated machine stage, but is promising for future universalization. Its high-order retrievals on data bases are just about "inference" itself. It can also be said to have pointed out the feasibility of inference machines from the standpoint of data base machines.

At present, it is common that data bases and programming languages belong to different systems. This is not a desirable situation. Their unification is a subject for the future, but as described in the foregoing, it appears to be quite feasible. This will then become the starting point for a new software system. Simultaneously, we can look forward to organization of the whole based on a unified viewpoint that includes machine levels.

## 2.5 From Linguistics

Expectations are rising that natural languages will be usable in programming and queries for data bases. However, these are sentiments which demand sophistication of communication forms with computers, and at present not many people appear to feel that the theories of natural languages (linguistics) and computer systems will become essentially associated. Is this true?

Research into the substances of languages surely is the main job of linguistics. At the same time, theoretical linguistics establish a language model. At this point, let us view theoretical linguistics with regard to their frameworks.

In the 60's, tremendous developments were made in linguistics based on the Chomsky theory, known as the "generative-transformational grammar" theory. Its framework designates the basic structures from which, by transformational rules, the surface structures of languages are derived. Its main objective was to explain the syntactic phenomena of languages. The framework of the theory began to be subjected to major changes in the 70's. This was associated with the stage in history where attention to "meanings" was pushed to the front. The generative-transformational theory was then divided into the two major streams, genrative semantics and interpretive semantics.

Meanwhile, early in the 70's, a new linguistic theory (by Montague) emerged. This came from a branch of philosophical linguistics, and is directly aware of the structure of semantics. It extended formal semantics contained in philosophical linguistics and applied them to a fragment of the English language, which may be called a model that unified syntax and semantics for the first time. This theme had been considered a difficult problem since ancient times. Mainstream linguistics had rather circumvented it. The handling of semantics was only partially presented in relation to phenomenal analyses, and no clear presentation was made of the structure of the entire discipline of semantics.

Appraisals by linguists proper vary with regard to the Montague theory. Some have negative opinions. On the other hand, get-togethers have also started. This is especially true with generative semantics, due to the similarity of between their models. Meanwhile interpretive semantics, with which Chomsky himself is involved, is also becoming more closely associated with logics, independently of, but in parallel with, the foregoing.

Montague's theory is a logical linguistic theory. In it, "intensional logic" whose semantics are clear cut are introduced as the base logics, and a procedure is given for transforming sentences in natural languages to the intensional logic. In this framework, a method of applicating formal semantics to natural languages was shown, which had previously been regarded as questionable.

Intensional logic is a kind of modal logic that has been transformed into functional forms. The semantics of modal logics (model theory) were introduced by Kripke. They are intended to define the meanings of logical formulas based on "possible worlds".

How will such moves appear when viewed from the computer technology side? Functional logics are no strangers to the world of computers. The language, LISP, is based on such logics. Functional languages have thus reemerged at the forefront by proposals for functional programming. Intensional logics correspond to an extension of them.

The structure of the Montague theory itself closely resembles LINGOL, a language analysis system created under totally unrelated awareness. Thus, the Montague theory (accidentally) largely overlaps with what has been created from computer technology.

The semantics of programming languages

referred to earlier originally stemmed from formal semantics. One of the interesting aspects is that the timing of the breakthroughs made in both fields (again accidentally?) roughly coincided.

Furthermore, the model theory of intensional logics (Kripke's possible world model) is intuitively well understood as seen from a data-base-like viewpoint.

The range of natural languages desired relative to computers would, contrary to the world of literature, be limited to portions that are logically graspable. Were this limitation to be unnatural, it would be against the original intention behind natural languages, but the range is believed to be sufficiently wide. From this viewpoint, progress in logical linguistics will be highly welcome to the computer side. Together with this progress, as many overlapping phenomena are observed, it is conceivable that such linguistics and the computer technology will link internally very closely with each other. To materialize this relationship would be far from easy, but its potential for the future is believed to be almost unlimited.

### 2.6 Flow of Research for Artificial Intelligence

Research into artificial intelligence dates a long way back. There were also many different approaches to it. The flow of research into artificial intelligence turned a great corner after entering the 70's. The characteristic aspect here, is the awareness of the problem of "languages and knowledge". Such awareness appears to have been considered too highbrow from the on-site computer engineers' side. It is felt that in software engineering, data base philosophies, and the like, there was countervailing awareness against it. Lately, however, these movements have begun to overlap to a large extent, and to become fused together.

The debut of artificial intelligence in the 70's is symbolized by the emergence of Winograd's question-answering system. This had as its object a limited world of toy building blocks, but had a knowledge model in its background, and provided questions and answers in natural languages.

Since then, awareness of central problems in research into artificial intelligence such as how to represent and utilize knowledge has emerged. Through this, exchanges of information between linguistics and psychology became active. At the same time, research on methods for their realization began overlaping with efforts for sophistication of computer utilization forms.

In the latter flow is the problem of programming languages for artificial intelligence. PLANNER and other similar proposals may also be regarded as representing the direction of extension for the programming languages themselves.

PLANNER was tried for an inference system converted to programming languages. In it were included concepts for pattern matching functions, non-deterministic processings, and multiple data bases. On the other hand, by subsequent progress in research on theorem proving systems (inference systems), it has been found feasible to directly reconvert PLANNER and other similar functions to provide an inference system. This is represented by the predicate calculus programming proposal described earlier. PROLOG and others based on this philosophy may be regarded as a logically reorganized PLANNER. This is meaningful also from the viewpoint of semantics (of programming languages). The accurate semantics of the PLANNER are not clear, which was probably one reason why it did not last.

PLANNER itself subsequently progressed to an actor model (PLASMA). This modelizes operation processes based on a philosophy of message exchanges. This scheme can be regarded as being an extended form of the data flow philosophy. These two, however, stem from different motives. In reviewing the research endeavors in the 70's, a number of phenomena have emerged from research efforts which started from independent motives and later linked with each other, of which the above is one example.

Research into representation of knowledge is a major theme in research on artificial intelligence, but with this, too, inter-linkage with research on data bases has become increasingly closer. It has come to be linked with awareness of the problem of data base sophistication. Previously these were promoted by separate research groups, but now there are noticeable efforts to fuse the two together for total control. The task of high-order structuring of data bases is believed also to be deeply interrelated with intensional logic and its possible world model referred to earlier.

Another notable aspect of research into artificial intelligence is that in the latter half of the 70's, application of artificial intelligence techniques actually started taking place. This applied artificial intelligence is called "knowledge engineering". Consultation systems integrating specialized knowledge in medicine, chemistry, and other professional fields have been built and are attracting attention. For knowledge engineering, an image, "knowledge base plus inference engine", is being proposed as the system framework.

Though the present knowledge engineering will be able to realize a section of artificial intelligence, it is rather considered as the basis for future information processing. As already made clear, it is acquiring ever closer linkage with software engineering and other research efforts. Judging from the closeness of the linkage, knowledge engineering will probably be able to absorb these into a unified form in a natural manner.

At this time, we would like to call such an extended form of knowledge engineering "knowledge information processing". This, it is thought, will represent the form of information processing in the 90's.

In research into artificial intelligence, there is awareness of the problem of the elucidation of natural intelligence, which at present has led to proposals for congnition sciences. At the same time, research into artificial intelligence has also borne the role of precipitator for advances in computer technology.

Reviewing the 70's, research efforts into computer technology may be said to have been split into a number of branches or 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 for mutual fusion emerged. This may be regarded as a valuable bud which will blossom in the 80's and an important legacy from the 70's.

## 3. STEPS IN DEVELOPMENT

For the advanced information processing systems of the 1990's, a "knowledge information processing system" is postulated, and steps for its development will be considered.

This is an image for the future, and an anticipated attainment. Would such an attainment be adequate? Some prefer to remain conservative. Further discussion will be needed to solidify the image. Also, based on the hypothesis, in what terms should developmental steps be considered? A number of requirements come to mind.

### 3.1 Functional Images

Functional images for knowledge information processing systems can be viewed in terms of the two facets, man-machine interfacing and problem solving capabilities.

Concerning man-machine interfacing, a function is conceivable which would permit natural media of communication with the computer via a natural language, graphs and the like. In order to input a natural language, as well as a keyboard, written and spoken input will be required. For graphic input, hand-drawn and pictorial input will be desirable. To enable communication using this kind of input, the system will have to extract "meaning" from it.

In addition to understanding the meaning of input messages, the system will need to have the capabilities of answering questions and also, occasionally, of giving advice. Furthermore, the ability to give indirect responses after digesting the intent of the questions and summarized answers will also be desirable. Successively realizing these capabilities will most likely be the way to endow computers with truly intelligent communicative powers.

On the other hand, answering questions and adequately accommodating requests will require internal "problem solving functions". At this point, mutual understanding between man and machine regarding "problems" will be vital. The computer will be required to understand the problems according to man's understanding.

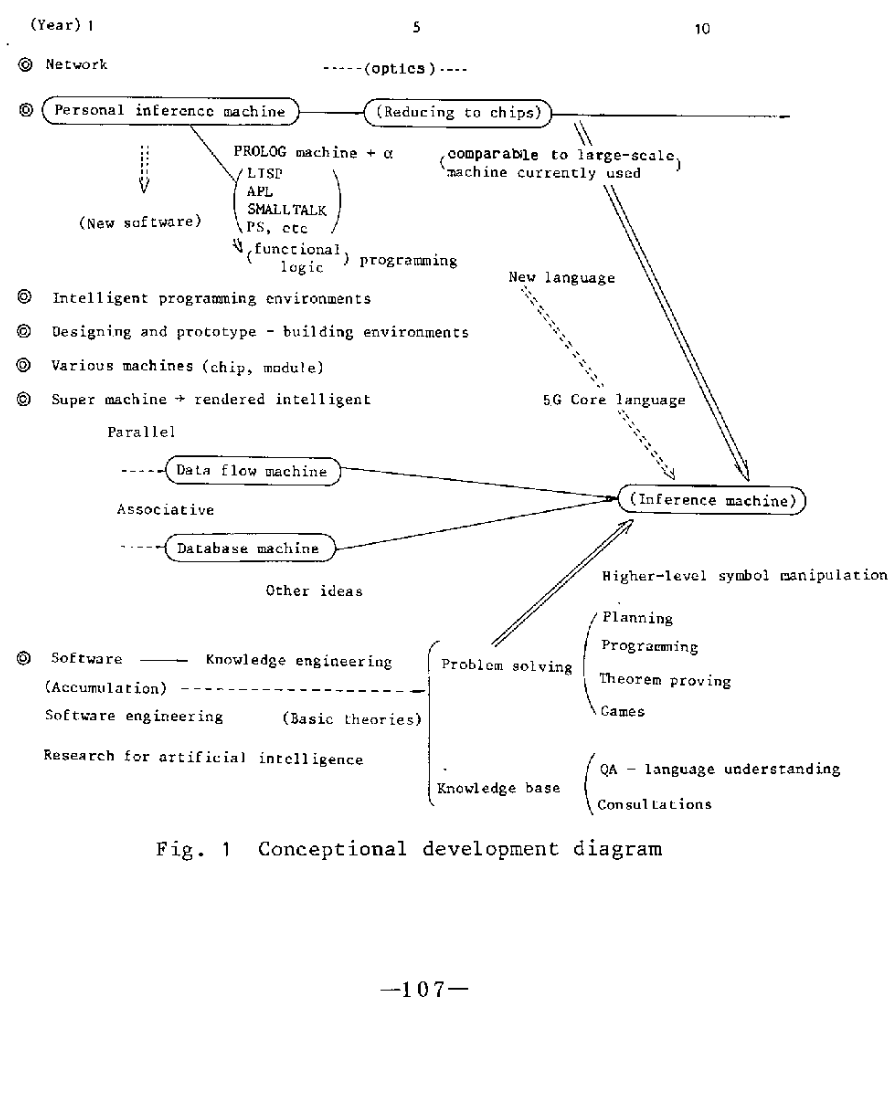



Fig. 1 Conceptional development diagram

This just means the "knowledge" that the computer possesses concerning the problem area. This type of mutual understanding the problem area. This type of mutual understanding (even though at a low level initially) is the basis for co-operation between man and machine. This kind of function will be required of advanced CADs (computer-aided design systems). Such common knowledge will also be the basis for more fluent communication.

Even systems such as described above will have to be instructed by man for high-order problem solving. In decision making, when multiple solution strategies are presented, augmented by examinations and hypothetical approvals by the computer, man's high-order judgement will be effective.

Functions for designating hypotheses and examining and approving will be important, but at the same time, this poses a difficult research theme in connection with how to handle insufficient knowledge.

Also important will be functions for acquiring new information and adding to the memory bank's knowledge on the problem area. This is primarily a matter of integrity, or the problem of fusing the additions with the existing knowledge without contradiction and integrating them. Secondly, there is the question of "inductive inference", or the problem of extracting rules from a set of data. These may be restated as learning problems. There is also the problem of accepting knowledge from man himself. People do not necessarily understand things in an authentically organized form. This resembles the problem of requirement analysis in software engineering.

The above may be regarded as attendant systems for the knowledge information processing system, but can also be viewed as higher-order knowledge (meta-knowledge) systems.

In knowledge information systems, the volume of accumulated software will become vast indeed. For this purpose, too, flexible, extended functions such as learning functions will be required. Regarding the contents for processing, high-order processing will be required for non-numeric (symbolic) data. This is "inference", and will essentially be non-deterministic algorithms. To efficiently execute these will require assistance from hardware, but will also largely depend on the structure and quality of software. Techniques to deal with this are emerging in the field of software engineering. One is the philosophy of data abstraction and the step-by-step programming based on it. This will substantially alleviate the complexity of problems. The second philosophy is the technique of program transformation based on certain rules. This provides a way of transforming the abstract to the concrete. These philosophies, will be integrated in a natural manner into the framework of knowledge information processing.

As to hardware requirements, high-level "symbol processing machines" will be desired that execute inferences at high speed. Included amongst these will be pattern matching functions, non-deterministic processing functions, and accompanying garbage collecting functions (memory control functions). Non-deterministric processing is conceived not only as back-tracking functions but as one possible parallel processing style. Machines of this kind may be termed inference machines.

The second important hardware element will be the data base machine. For better matching with inference systems, relational data base machines are the most promising. Concerning their retrieval functions, relational algebraic machines are conceivable that directly execute relational algebraic operations. These are also a kind of inference machines.

Fifth generation computers can be imagined as an extension of the foregoing as hardware. They will be knowledge information processors, and as systems, knowledge information processing systems.

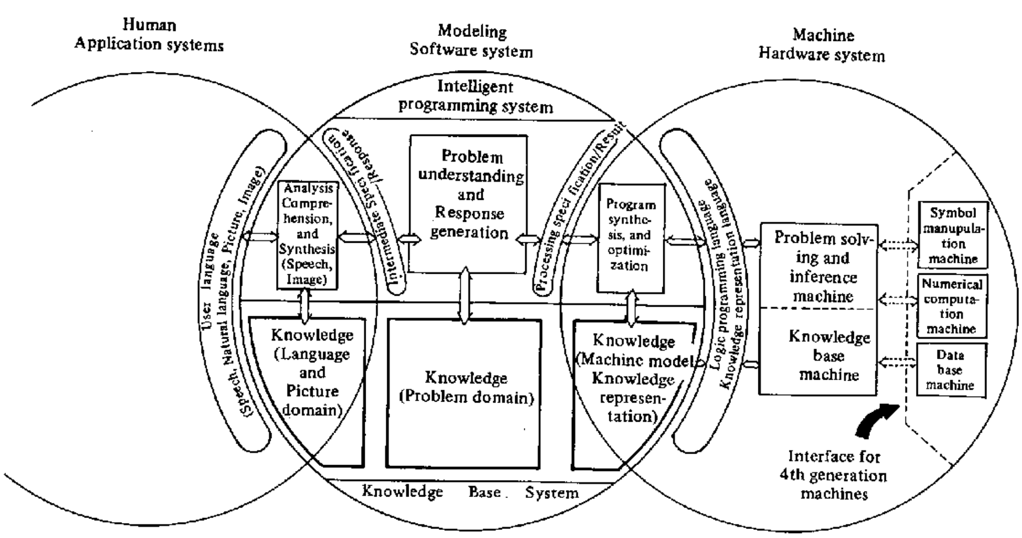



Fig. 2 Conceptual image of the fifth generation computer systems

### 3.2 Inference Systems

Universal knowledge information processors may be considered problem solving machines comprising knowledge bases and inference machines. Inference systems are discussed below in terms of their functions.

The degree of difficulty in problem solving varies widely depending on the nature of the object problem. With regard to the elements affecting the degree of difficulty, those related to logics include completeness and invariance of the set of knowledge. Physically, there is the problem of the volume of knowledge to be handled.

#### (1) Inferences on Data Bases

Normal data bases are theoretically supposed to be invariable in their structure and complete in their knowledge at each respective point in time. Much research work has already been done concerning this, and a technical level has been attained at which retrievals and control can be efficiently performed. Inferences are based on the "closed world assumption", and can be reduced to operations on relational algebra.

Based on the above, various trials on hardware have so far been presented in research into data base machines. Conceivable for their future direction are relational algebraic processors that, as hardware, would execute the relational algebraic operations mentioned above. If these were realized, such operations could be executed at high speeds. A number of processor proposals exist, of which the most promising probably is the one proposed by Tanaka et al., (1980). This is a proposal for a data flow type processor operating in parallel in a pipeline mode. It is an inference machine within limited logics, but it is thought that it would lead to higher sophistications and higher efficiencies for commercial data bases.

#### (2) Inferences in Predicate Calculus

When a set of knowledge is incomplete but static (invariable), it can be expressed by normal (not including temporal variables) first-order predicate calculus.

Inferences in first-order predicate calculus include the well-known resolution method. This possesses simple predicate rules but involves disadvantages in efficiency. Recent achievements including the connection graph created by Sickel (1975) and others where propositional and variable portions are separated, however, constitute efficient procedures for this method.

Also, pertaining to the Horn theory, which is a subset of first-order predicate calculus, PROLOG, a programming language based on it has been proposed, and along with it excellent implementing techniques have been developed permitting its efficient execution. This language is based on non-determinism and pattern matching functions, and has similar functions to PLANNER, but is superior since it is based on logics. Extensions to it also are being studied.

#### (3) Meta-Knowledge and Inference Process Control

The disadvantage in first-order predicate calculus is its inability to handle knowledge relative to the utilization of knowledge (meta-knowledge). In other words, such knowledge generally is information for controlling inference processes themselves, and cannot be expressed in first-order: on the other hand, if utilization of such knowledge were feasible, it would be able to avoid wild goose chases.

Concerning PROLOG, trials are being made to integrate meta-knowledge (Gallaire, 1979), (Bundy, 1979).

#### (4) Truth Maintenance

When the set of knowledge is incomplete and dynamic, it becomes necessary to realize functions of inference based on hypotheses and of revisions of the hypotheses based on new knowledge. This was shown by Doyle (1978) and others to be realizable by problem solving machines consisting of two sub systems: an inference system and a truth maintenance system. The truth maintenance part is the meta-system for the inference part, and can be utilized to control the inference part. Using this philosophy, Doyle developed a back-tracking control system that is theoretically efficient. When an inconsistency is found, the system back-tracks directly to the point where the inconsistency occurred, rather than to the selection point immediately prior to it, as in the case of usual back-tracking functions. As the truth maintenance part is separate, inferences performed so far, except the erroneous portion, all remain valid in subsequent inferences. This provides extensions over previous logics, and can perform exception handling inferences.

This system can also be utilized, when renewing data bases, as a control procedure to constantly maintain consistency in all the data bases. It can be used in by-step learning of knowledge. Systems of this kind possess functions desirable for knowledge information processing, but are as yet at the research stage, and their progress is eagerly awaited.

### 3.3 Intelligent Programming Systems

The theme, intelligent programming, is considered important in pursuit of knowledge information processing technology from two aspects. One is that intelligent programming is indispensable as a tool for developing knowledge information processing systems. With knowledge-based information systems, the software will be on a large scale, and the functions of a high order and complex. Development of these will be accomplished as a kind of evolution. For this purpose, high level, well-structured processors will be desirable, but at the same time, a high-order, development support system will also be required. In programming, the requirement will be for intelligent programming.

The second aspect is that intelligent programming systems themselves will have to be developed as knowledge-based information systems. In this sense, they will represent a specimen application of, and prototypes for, knowledge information systems themselves.

Systems such as the above, accordingly, will come into being in parallel with research on knowledge information processing systems. As a starting point, they will probably materialize on top of the high performance personal PROLOG machines described later on, and be formed into a network through full utilization of conventional technology. Useful software tools developed beforehand will probably be integrated in these systems initially. With this kind of accumulation as its basis, progress will probably be toward knowledge-based processings.

Of past technology, the techniques described below are considered useful:

(1) Interactive processing: Interactive processing developed through TSS will further be developed. Utilization of editors and graphic terminals, and the like.

(2) Execution of structured programming: Structured programming which has taken advantage of conversational processing.

(3) System description languages and specification languages (e.g. the $\iota$ (iota)-system of R. Nakajima et al.).

(4) Graphic languages: There are fewer technical difficulties in using graphic languages than in using natural languages. Furthermore, there are advantages in graphic languages themselves. It will be necessary to reconsider graphic (iconic) programming. Structured displays of programs, and others. Techniques will be required for in- and outputting, editing and controlling graphic data.

(5) Data abstraction

(6) Requirement specifications and specification descriptions: Transformation from formal specifications to programs. Various experiments on informal specifications. Experiments on verification methods.

(7) Algorithm banks: Program libraries will have to be reorganized in a more versatile manner. For this, description in better languages will be required. Such banks will constitute one kind of knowledge base.

(8) Programming in a natural language (Japanese): Intelligent programming systems integrating all the above techniques will provide, on the one hand, opportunities to review and revise existing technology, and on the other, opportunities for new experiments. The systems are themselves tools with which new products can be manufactured. Manufactured products can again work as tools.

So far we have concentrated mainly on programming, but in such systems it should also be feasible to develop intelligent design systems of hardware. While these latter have unique aspects, many aspects remain that are shared with software design systems. Such systems will facilitate design, not only of software, but also of a great variety of chips to enable LSIs to exhibit their full capabilities. These are themselves a major theme for the future. At the same time, they also represent tools indispensable in building knowledge information systems up into a new total organization that includes hardware.

### 3.4 Inference Machines (1): PROLOG Machines and Local Networks

Described below are PROLOG machines and the networks that connect them, which are treated have as a starting point in building up knowledge information processing systems.

It goes without saying that appropriate "symbol processing machines" are desirable in knowledge information processing. The recent development of LISP machines seems to have been aimed toward this same objective.

The reason for our proposal for using PROLOG as the starting point rather than LISP is primarily that PROLOG can be seen as an extension of LISP. Though PROLOG systems currently in use still have some flaws, they are considered capable of integrating all the advantageous functions of LISP systems. They also have pattern matching functions, non-deterministic functions, and other extended functions. They will also be capable of integrating interesting features of other languages than LISP such as, SMALLTALK, PS, and APL. PROLOG seems to be best suited as the starting point in considering a new base languages for knowledge information processing.

PROLOG is a programming language based on logics. Its base is the same as formal specification languages that are also based on logics, thus facilitating transformation of formal specifications to programs. PROLOG also has the same logical base as relational data bases, and is suited as a base for integrating programming and data base query languages. PROLOG is also intrinsically suited as a base for realizing natural language processing and higher level inference functions.

Then, will PROLOG machines be feasible? If they follow the same line as current LISP machines, yes, they are technically feasible even now. These then will represent somewhat extended LISP machines. As PROLOG is a language based on basic inference operations, PROLOG machines could represent the first step toward inference machines.

It should in the near future be technically possible to achieve conversational PROLOG machines, equipped with for instance, 1M 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. A system integrating these in a network would be useful, or even indispensable, for the future development of high-level information processing techniques.

By adding Kanji in- and output functions, these machines will also be able to provide an experimental system of Japanese language machines.

Machines such as the above can technically be considered as the fourth generation. Generally, however, not many would go beyond the fourth generation along this direction.

Pondering over the above systems provides a starting point toward the knowledge information processing systems proposed here, and also a short cut to them. Technologies accumulated so far may provide the route for positive continuation into future technologies.

### 3.5 Inference Machines (2): New Architecture Machines

Research work has progressed on inference systems that lead functionally to knowledge information processing. If seen in the light of these, how will inference processor hardware emerge? In a perspective toward revolutionary architecture, how will it present itself?

Research is about to be promoted which will develop architecture from PROLOG mentioned earlier and connection graph inference systems. To attempt to predict its features, they will be probably something like the extended and developed forms of data flow machines or data base machines that have recently been the subject of intensive research efforts.

Data flow machines are currently regarded mainly as one possible configuration of parallel processors intended for ultra-high-speed numerical computation. That they are considered more promising than past approaches is due to expectations that the type of programming difficulties inherent in conventional parallel processors will be eliminated through their linkage with the functional programming styles of Backus and others.

Meanwhile, thinking about PROLOG and others, and certain restrictions to them and you have data flow architecture. Stated backwards, this means that PROLOG machine architecture could take the form of an extension of data flow machines. Methods for the extension, however, are not clear at present. All the same, such methods are feasible, given that Dennis' data flow machines can be considered as essentially message flow machines, and extended on that basis.

As for data base machines, relational algebraic processors are special inference machines. As stated earlier, proposals have been made to structure these in a data flow style. Though in a special form, these represent data flow type inference machines.

When data flow machines themselves are still at the research stage and even their structure has yet to be established, discussing their extensions may sound like jumping the gun, but various symptoms which point to the feasibility of extensions are being observed.

Non-determinism in inference systems, though efforts must, of course, be made to reduce it by analysis of problems and by adequate algorithms, is essentially unavoidable. With current serial machines, such efforts are accomplished by back-tracking control, which lowers efficiency. This control essentially belongs in the realm of parallel operations. How to achieve parallelism is a problem, and research on data flow machines as parallel processors has become focused on two main streams of architecture research. If inference machines are to come out of extending data flow machines, then research should show the way to achieving inference machines possessing revolutionary architecture.

### 3.6 Systems Applications

In the foregoing sections images have been described and the constituent elements for knowledge information processing, and these will now be reviewed from the viewpoint of systems application.

First, let us take the application field, design systems. Design is generally an intellectual activity, and computer assistance is as yet limited to low levels. Design operations vary with the objective.

Mechanical designs (CAD/CAM), for instance, require in- and output of graphic information. This, however, is a minimum requirement, and to step up computer assistance, it will be necessary to accumulate numerous models relating to ojbect worlds and to utilize them. Such information may be termed "knowledge". Furthermore, to utilize this knowledge, "problem solving" functions intended to achieve specific designs from given specifications will be required. Aside from CAD systems expected to be coming out in the near future, the highly functional CADs of the future will require knowledge-based processing. Here, smooth communications and knowledge utilization will be essential, and "knowledge information processors" such as are described will constitute their basis.

Design objectives are various. In a separate field from mechanical designs, there is, "material design". In order to "invent" a new material with required characteristics, fundamental knowledge (knowledge base) relative to materials is required, to discover the desired characteristics from among a variety of combinations. To aid computers, knowledge-based information processing techniques will also be desirable.

In the meantime, "programming" and "chip design" mentioned above also belong in a design field closely related to computers. For upgrading these, too, knowledge-based processing is anticipated. These are themselves also tools for developing knowledge information processing systems.

As a second application field, let us take "consultation". As an example already in use, there is a system named MYCIN in the medical field.

This system has knowledge on a multitude of new drugs, and makes "suggestions" to doctors for medication according to symptoms. For the suggestions, diagnostic rules need to be integrated. It is a systematized (partial) knowledge that doctors have. The system also has a subsystem to "explain" on what bases it makes suggestions.

To perform such functions as the above, this system has a knowledge base relative to facts and rules, as well as possessing an "inference system" to make it work. The inference system has, of course, been simulated on a current computer, which presents a problem in efficiency and accordingly limits the number of utilizable rules to a level of several hundred. If knowledge information processors as proposed here become a reality, their performance capabilities will be vastly improved.

The philosophy with such as MYCIN is useful not only in medicine but in fields such as education. CAI from some time ago quickly came up against a brick wall, but to develop it into a true CAI, knowledge information processing through conversion of knowledge to rules will be exactly what is needed. A similar situation applies to MIS (management information system) also from some time ago. This proposal was ahead of its time. To perfect it, a high-order knowledge information processing system integrating a variety of economic models and system models should have been prepared beforehand.

Similar comments to the above apply to fifth generation office automation to follow fourth generation office automation (which in the natural course of events will probably come to pass).

In these systems, knowledge and the inference based on it constitute the core, and simultaneously, for the communication form and knowledge representation form, "natural languages" will inevitably be required. This, in our case, will be Japanese.

This relates to awareness of the problem of "Japanese language machines". In the latter case, in addition to surface structure processing of the Japanese language, more advanced processing will be required.

In- and output and processing, centered around character systems such as Kanji and Kana, are presently being energetically developed as "Japanese language information processing". These will spread and become established in the very near future. Beyond such a stage, processing techniques that extend into "meanings" will be called for, for true Japanese language information processing. These are mandatory for natural communications, and simultaneously understanding of meanings will itself take place with knowledge bases and the attendant inference constituting its core.

In the field of natural language processing, there is also the problem of machine translations. Translation systems are again beginning to attract attention. In the background are the needs brought about by advanced internationalization, but at the same time, the attention is motivated by the awareness of the problem of taking advantage of the advances in the last 20 years of computer technology. It is now time to pursue the feasibility of such system. To utilize techniques accumulated over the years is feasible, useful, and meaningful.

At the same time, their limitations are also apparent. For upgrading to a good translation level, translation techniques encompassing processing of meanings will be required. Toward this end, further research in linguistics will be necessary, and at the same time establishing a technological system integrating knowledge information processing as described here will also be needed.

The application fields described so far are hardly exhaustive, but through the examples given it should be discernible that knowledge information processing is exactly the information processing which the new age will require. For its realization,precise images and precise development steps are necessary. Additionally, a great many research tasks will have to be resolved in the process.

## 4. SUMMARY

To construct images for fifth generation computers and a system for the new age information processing technology, one of the necessary conditions will be survey of basic theoretical research. "Basic Theories" here is not used in a narrow sense such as mathematics, but in a wider sense including research into the mathematical theory of programming, artificial intelligence, and pattern information processing.

A mere survey of the fields mentioned will not itself provide definitive image for fifth generation computers, but it is now felt to be time for the survey to at least contribute to the establishment of the image. No conventional definition exists of fifth generation computers: we can only say that they will not be an extension of gradual improvements over current computers, and there is an awareness which expects that the next stage will be a leap forward rather than just an extension. The background for this awareness is primarily dissatisfaction with current computers which may loosely be expressed as that they are "hard to use".

The above applies to contemporary technology as far as end users and the public at large are concerned. The dissatisfaction surely stems from the immaturity of the current technology, but will it disappear if the current system of technology is improved step by step? Or has it been caused by defects in the basic structure of current computers?

Similar dissatisfaction is also emanating from within computer technology. It may be summarized as difficulties in manufacturing large-scale software. Is this latter due to the immaturity of the software engineering? Partially, yes. Elesewhere, introspection over the current architecture computers has also been noticeable.

The first problem to be aware of, then, is:
@ Handy computers (systems).

There are many aspects to handiness. There are a number of ways to describe it from the viewpoint of functions. Needs from the user side include the problem of "Japanese language machines". The current situation is far from the ideal of free usage of the Japanese language. Use of Japanese for data base queries, programming, questions and answers, and suggestions, is still at a stage where research trials have only just commenced. For free usage, handling of "meanings" has to be possible, for which, however, much further research will be necessary.

The Japanese language represents one natural communication form with computers, but from the viewpoint of handiness, graphic communication must also be included in natural communications. For functions to in- and output words, voice (audio) in- and output is also desirable. To achieve these will require highly sophisticated functions. Thus,

@ Sophisticated and numerous functions, and high performance capabilities

represent some of the requirements for handiness.

From the viewpoint of sophisticated and numerous functions, not only will numerical computations be required but also a wider range of "symbol processing" functions that must be made sophisticated and more efficient. These will be needed for the processing of languages and manufacture of software.

The viewpoint that will later on prove to be important for sophisticated functions is:

@ Knowledge information processing.

With current computer technology, most problem solving must be converted to "programs". This relates to lack of handiness in computers. On the one hand, further researches in the mathematical theory of programming will be required. On the other, integrating "knowledge" will become necessary. It will be necessary to substantially upgrade the problem solving capabilities of computers in the direction of integrating information concerning the object areas of problems, plus information on the governing rules there. These types of information correspond to what is called "knowledge".

The above is equivalent to extending the range of mutual understanding between man and computers, and not only will it upgrade the problem solving capabilities of computers, it will also represent capabilities that are mandatory for natural communications, besides being the basis for realizing true "handiness".

Achieving sophisticated functions through languages and knowledge was a main theme in the 70's for research into "artificial intelligence". Numerous research themes remain to be resolved, and here, too, is grist for the mill, in considering fifth generation computers.

Also from the viewpoint of sophisticated functions in the above direction, there emerged demands for a new computer structure (architecture) and suggestions as to what it should be like.

This creates awareness of the problem of:

@ New computer architecture.

This relates to awareness of the problem of:

@ Withdrawal from conventional systems.

Then, a

@ New system of information processing technology

to represent the ultimate goal, having started out from handiness.

With regard to awareness of such problems as the above, the question will be whether

@ Support by basic theories

is feasible.

In this paper, activities in the various fields of basic research into information processing have been reviewd.

Notable is the fact that the courses of such activities, while having been promoted more or less independently, have recently exhibited atendancy to converge.

Judging from this phenomenon, it appears highly probable that during the 80's, basic theories in a wider sense will contribute to new systems of information processing technology, and consequently to new architecture (i.e. fifth generation computers).

Organizing problem awareness directed toward fifth generation computers should result in the designations described above. As a result of reviews conducted from such viewpoints relative to fifth generation computers, images of the following emerged:

@ Knowledge information processing systems, and

@ Knowledge information processors

as their cores.

The latter may coincide with inference machines (inference engines).

Further, in attempting to realize these images, it becomes apparent that

@ Sophisticated environments to facilitate production of systems

are mandatory. This must be considered important not only as the means to attain the goal but also as an element for knowledge information processing systems as progressing systems rather than mere support systems. They themselves will also play the role of prototypes for knowledge information processing. They, furthermore, will provide environments for educating personnel, and the ground in which to nurture knowledge information processing as a well-staffed technology.

Of the themes that will be necessary in the research and development of fifth generation computers (as well as related information processing), summarized below are future research themes classified by the basic research involved:

(1) Research on natural languages including Japanese:

Research on language structure, understanding processes, conversation models, etc.

(2) Research on the representation of knowledge and inference:

Relations between data bases and logics, the handling of large volumes of knowledge, handling of complex (small volumes of) knowledge, and the development of new logics to accompany these.

(3) Research in software engineering:

Research on specification descriptions, verifications, and program transformations. Development and application of semantics. The state of software in the fifth generation.

(4) Research on new computer architecture:

Further development of schemes for data flow machines and data base machines (relational algebraic machines). Study of architecture for inference based machines (inference machines, inference engines). Research on parallel algorithms and architectures.

The various researches described above must be aggressively carried forward. These are intended for development through the 80's and for perfection in the 90's, but at the same time it must be remembered that their basic components were developed during the 70's. While the route to knowledge information processing is an advance to a new age, it can also be viewed as representing the inheritance and development of the legacies of the past from the viewpoint of research efforts. In this sense, the route to knowledge information processing represents a practical philosophy and an inevitable direction for development of information processing technology. The question is rather whether to stand still or proceed, as there are no other paths to choose from.

# FIFTH GENERATION COMPUTER ARCHITECTURE

Hideo AISO
Department of Electrical Engineering
Keio University, Yokohama, Japan

This paper is presented as a discussion of fifth generation computer architectures. It is important to predict computer architectures for computers to be developed in 1990's on the basis of observable technological trends and anticipated applications. However, numerous questions arise and it is difficult to prepare the answers to all questions. The intent of this paper is to suggest reasonable approaches to developments of fifth generation computer architectures and to offer major research subjects to be pursued for future computers.

Intensive discussions from the viewpoints of both associated theories and technologies have led to a conclusion that the objective of the fifth generation computer project is to do research and development of very high-intelligent computer systems oriented toward processing knowledge information, which is referred to as knowledge information processing systems. From a viewpoint of computer architecture, the objective of the research is to look for effective support functions to be implemented at the computer architecture level in realizing knowledge information processing systems.

## 1. INTRODUCTION

Experiences with a wide variety of computer applications for more than thirty years have suggested that a future computer possess at least the following characteristics in general.

(1) It should provide functions capable of organizing appropriate structures adaptable to wide applications.

(2) Various functions or mechanisms to promote the improvement of the software development productivity should be implemented in it.

(3) It should offer to the non-professional users effective means of easy access to itself through very high-intelligent man-machine interfaces.

(4) It should also provide functions offering new approaches to problem solving, that is, human substitute functions to support judgement and decision making.

(5) Functions to facilitate distributed processing and system protection and mechanisms to guarantee high system reliability should be incorporated in it.

(6) Considerations to improve the cost-performance should be emphasized.

It can be said that the objective of the research on fifth generation computer architectures is to look for primitive functions and effective mechanisms to support the realization of the characteristics enumerated above at the computer architecture level. However, it would probably be impossible to realize them satisfactorily in the existing computers without the introduction of drastic innovations in computer architecture. It is true that the fifth generation computers will be designed on the basis of revolutional architectures as will be described later. However, it is important to emphasize that such advanced architectures do not make sense until they are incorporated with evolutional architectures innovated on conventional von Neuman computers, although the explicit distinction between both types of architectue is difficult.

## 2. NEEDS FOR INNOVATION IN COMPUTER ARCHITECTURE

The difficulties in developing software and the large amount of the software property already produced make it very hard to introduce new architectures into the existing computers. As often seen in prior generation computers, if the improvement of cost-performance is the main target in developing a new computers, then it would be a clever way to adopt VLSI technology throughout the system without taking computer architectures into consideration. Nevertheless, many computer scientists bemoan the lack of innovations in recent computer architectures.

The following is a list of a few major reasons.

(1) New applications such as large scale scitific problems requiring plenty of computation time, or image processing dealing with lots of data in real time have been exploited, and thus the considerable improvement of both the capability of very high-speed processing on a large number of data and cost-performance has been solicited.

(2) The existence of the semantic gap, which is a measure of the difference between the concepts in high-level programming langauges and the concepts in the existing computer architectures, has been pointed out. This fact may give an impetus to the software crisis in the 1990's. In order to alleviate this problem, it is definitely necessary to develop new programming language on the basis of a consistent concept, and to provide fundamental functions at the computer architecture level, which drastically narrow the semantic gap in operating-system and high-level language environments.

(3) High-intelligent man-machine interfaces which offer the users more natural interactive processing or more convenient image processing have often been requested in recent sophisticated applications.

(4) Novel applications associated with pattern recognition, semantic data bases, knowledge

bases, or artificial intelligence have claimed to incorporate effective support functions in a computer.

(5) The prior generation computers have been designed on the basis of the deterministic data processing. However, non-deterministic or heuristic data processing techniques have gradually been recognized to be very essential to applications with sophisticated man-machine interfaces. In such applications, for instance, the capability of associative processing on data with ambiguity or inference processing incorporated with the backtracking function will be desirable to be provided in a computer.

(6) It is recognized that ultra-reliable computer systems should be developed employing redundancy techniques or dynamic architectures.

(7) Recent advances in the field of VLSI technology have made possible substantial changes in computer architecture, and thus application-oriented or dedicated systems have easily been developed.

(8) A new style of data processing has been being popular utilizing very high-performance personal computers or intelligent terminals.

The reasons (1)∿(6) come from the needs driven by novel applications and the reasons (7) and (8) from the seeds sowed in technological progresses. Especially, the reasons (1)∿(6) indicate that effective and practical data processing in novel applications will not become a reality without providing appropriate support functions at the computer architecture level and they suggest the necessity to do research on computer architecture.

## 3. PROBLEMS WITH PRIOR GENERATION COMPUTERS

Most computers of prior generations emphasized high-speed operation on large volumes of data. However, as the field of computer applications spreads wider and wider and technologies surrounding computers progress very rapidly, various weak points in their logical capabilities have been pointed out in substance.

The following is a brief list of a few major weaknesses indicated from a viewpoint of users.

(1) While the kind of problems to be solved is being diversified, the computer structure and the implemented functions still remain unchanged or fixed. The existing computers provide only low-level abstract data type programming languages, and thus the users are not allowed to use natural languages through high-intelligence man-machine interfaces. Most computers can not change dynamically so as to have dynamic and optional functions depending on the status of the problems being solved.

(2) The present computers can solve the problems satisfactorily which are described in mathematical forms based on the deterministic logic. In other words, the main function of the computers is aimed at high-speed processing of the numerical problems in which all the cases to be occurred during the program execution have already been clarified exactly. However, problems often appearing in natural environments contain ambiguity, redundancy, or numerous semantic parameters in the data to be handled. In solving problems in, for instance, pattern recognition, decision making, natural languages, or sophisticated knowledge bases, the exact approaches to the problems can not be found out at the beginning of processing, and hence trial and error approaches, which employ elaborated techniques often encountered in the field of artificial intelligence, may usually be required.

(3) The users always have to convert the original problem being solved into a mathematical model that the computer may solve. Furthermore, there are the large semantic gaps between users' concepts and programming language concepts and between programming languages concepts and traditional von Neumann hardware concepts. These semantic gaps lead to not only the difficulty of the problem conversion but also the low productivity of software development. In order to alleviate these semantic gap problems, every efforts should be made for both the improvement of high-level programming languages and innovation in computer architecture.

(4) The emphasis of future computers should be on "logical power", that is, on thinking in a human-like fashion, as well as on speed of processing or volume of data. In order to achieve the qualitative improvement of processing, the computer should be designed so as to provide a knowledge base which stores consistently the knowledge of human experiences and the users may utilize it through an inference mechanism implemented in the computer. It is also expected to have functions which enables the computer itself to dynamically optimize the problem solving algorithm being employed.

## 4. FUNCTIONAL REQUIREMENTS FOR FIFTH GENERATION COMPUTERS

Fifth generation computers will be designed at least to solve the problems described above. This means that they will be suitable for processing knowledge information based on innovative theories and techniques proposed to meet the sophisticated functions which are considered to be required in the 1990's.

Intensive discussions from the viewpoints of both associated theories and technologies have led to a conclusion that the fundamental functions to be incorporated in fifth generation computers resolve themselves into the followings:

(1) problem solving and inference functions aimed at a maximum performance of 100M ∿ 1G LIPS (Logical Inferences Per Second);

(2) knowledge base management function aimed at a performance capable of effecting retrieval of a knowledge base required from inference within several seconds on a data base machine having a capacity of 100 ∿ 1,000 GB;

(3) intelligent interfaces allowing the users to make conversation with a computer through the medium of speech, images, or natural languages.

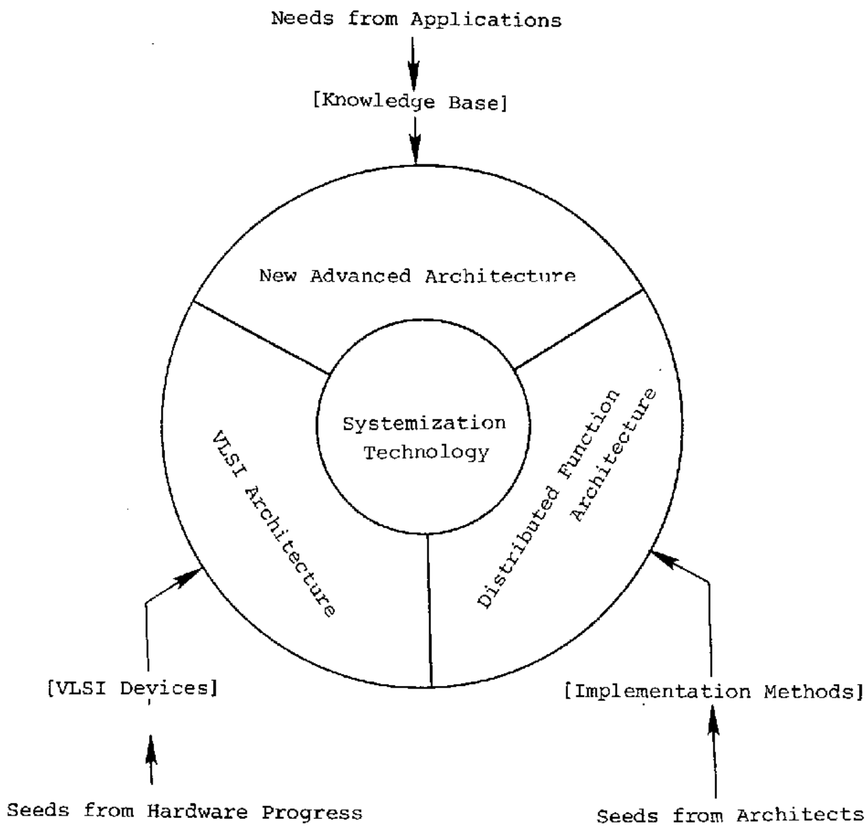



Fig. 1 Technological Factors influenced on Computer Architectures

It is no doubt that such fifth generation computers become the nucleus of a knowledge information processing system when they interface with a human system.

## 5. ARCHITECTURAL IMAGES OF FIFTH GENERATION COMPUTERS

From a viewpoint of computer architecture, the fifth generation computers should be equipped with primitive functions at the computer architecture level, which support to effectively realize knowledge information processing systems. However, as shown Fig. 1, the fifth generation computer architectures will be considered to be strongly influenced mainly by the following technological factors:

(1) needs from novel applications,
(2) seeds from hardware progresses,
(3) seeds from implementation methods, and
(4) systemization technology.

Novel applications will require future computers to have intelligent functions together with a knowledge base system, for which new advanced architectures should be developed as will be described later.

In particular, advances in VLSI technology will make a great contribution to the changes in computer architecture. In other words, fifth generation computers should be designed on the basis of VLSI architectures making full utilization of VLSI technology and producing VLSI chips from the component devices through computers.

From the standpoint of the easy implementation of systems, distributed function architectures will be essential for integrating various dedicated systems consistently. The development of a distributed function architecture assures high efficiency, high reliability, simple use, and easy construction of computer systems, and it also offers adaptabilities to future technological progresses and the implementation of sophisticated functions at the various system levels:

Fifth generation computer systems will be hierarchically organized from various component machines. Therefore, virtualization and integration technologies as well as design and development methodologies for developing large scale, and high-reliable systems should be formatized.

Fig. 2 shows an example of hierarchically organized logical model of the knowledge information processing system. It will be realized in obedience to the following steps:

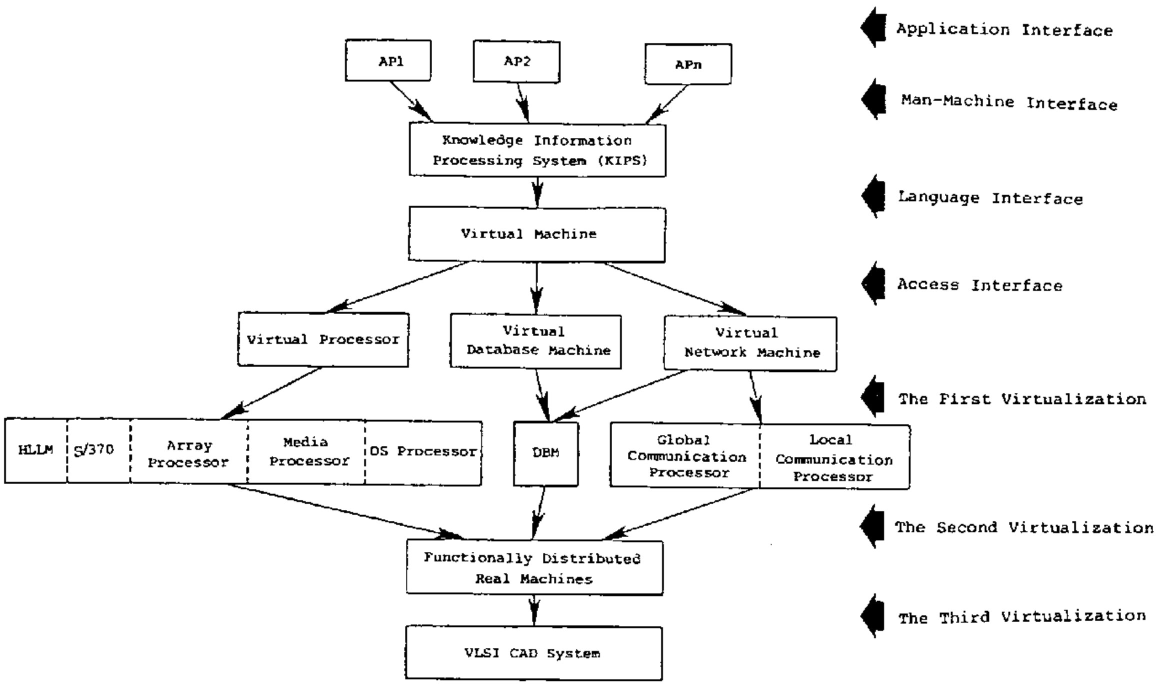



Fig. 2 An example of Hierarchically Organized Logical Model of the Knowledge Information Processing System

(i) System description languages are designed for the system designers at the language interface.
(ii) Images of a virtual machine seen from the users of the languages are proposed.
(iii) The functions of the virtual machine can generally be classified into three subvirtual machines; virtual processor, virtual database machine, and virtual network machine. An instruction set for accessing each of the subvirtual machines is proposed at the access interface.
(iv) Then, architectures for the subvirtual machines incorporating the proposed instruction sets are set up. The architectures may be those related to, for instance, abstract data type, relational algebra type, or data flow type.
(v) Real machines such as high-level language machines, an array processor for scientific use, or a media processor are designed to organize each of the subvirtual machines at the first virtualization level. A real database machine is developed in relation to the virtual database machine. For the virtual network machine, global and local communication processors are developed to deal with long distance communication networks and in-house communication networks respectively. The database machine may provide functions to control distributed database systems utilizing communication networks.
(vi) A distributed function architecture is proposed to integrate the above-mentioned dedicated machines into a single real machine at the second virtualization level.
(vii) All the real machines are implemented on VLSI chips through a VLSI CAD system at the third virtualization level.

Fig. 3 shows an example of the fifth generation computer systems, which is consisting of three levels of computers. Level 1 is called "a super personal compuer" which is a very high-performance VLSI computer such as a PROLOG machine for personal use. When the jobs exceed the ability of the super personal computer, then they may be processed by Level 2 or Level 3 machines. Level 2 machine is called "a service machine" consisting of various machines; user service machines to process user programs, machine service machines to take care of communication networks and distributed databases, and control machines to control the system program and the whole system. Level 3 machine

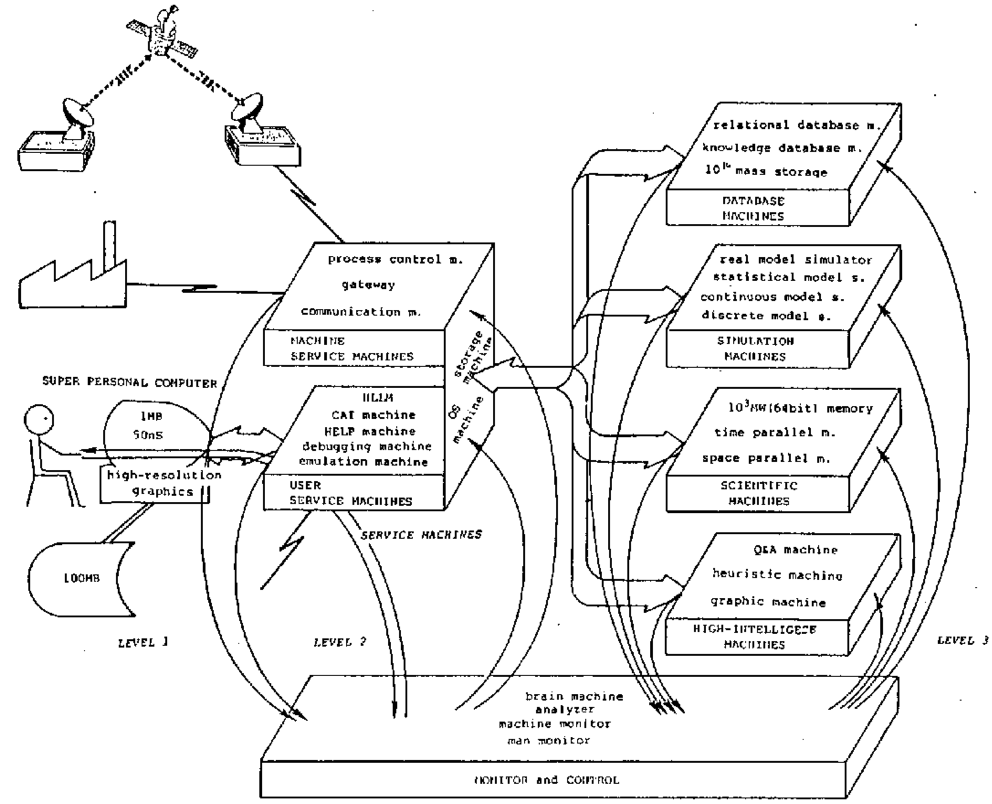



Fig. 3 An Example of the Fifth Generation Computer Systems

is called "a dedicated machine" or "a shared machine". The machines in Level 3 are shared by the users in a virtual mode, some of which may be implemented in Level 1 or Level 2. All machines are connected with each other through tightly coupled lines or loosely coupled communication networks. Useful monitor and control systems for the users, for instance, powerful HELP systems, effective debugging systems, hardware or software monitors, or automatic tuning mechanisms, are distributed throughout the whole system.

## 6. MAJOR RESEARCH SUBJECTS FOR FIFTH GENERATION COMPUTER ARCHITECTURES

In this section, a brief description is given of major research subjects for fifth generation computer architectures, details of which will be described in independent papers.

In order to achieve the goal of the fifth generation computer project, that is, the development of very high-intelligent computer systems oriented toward knowledge information processing, it is recognized that intensive research should be carried out on the realization of the knowledge base system and the associated inference mechanism. They will take a very important role in the nucleus of the knowledge information processing systems and they will also become essential elements for sophisticated software systems in the future. It is true that VLSI technology will give a strong influence on their implementation. With these backgrounds in mind, major research subjects for fifth generation computer architectures are listed as follows:

(1) Fifth generation computer kernel language

Advances in research on software engineering and artificial intelligence have been being offered effective solutions to various problems remaining unsolved in the existing software systems. One of the solutions is a proposal of new programming languages which may be substituted for the conventional von Neumann languages. Logic programming languages and functional programming languages are most promisive examples. They provide the capabilities of performing parallel processing and program verification to improve the software reliability. These programming languages might become an important interface between hardware and software and imply a macro-specification for computer architecture.

In the fifth generation computer project, a very high-level language called "a fifth generation computer kernel language", or "a core language" for short, will be designed so that it will be able to serve as a nucleus of the software systems and a fundamental specification for the computer architecture to be developed. The proposed language will be a type of logic programming language. An extended version of PROLOG is an example of the core language, which will be designed on the basis of a simple inference like a syllogism in logic. It is expected to incorporate the capabilities to specify parallel processing events and to express more advanced functions pertaining to knowledge or meta-inference mechanisms.

(2) Inference machine

The knowledge informaton processing system is considered to consist of three major machines, that is, inference machine, knowledge base machine and intelligent interface machines. As one of the research goals of the inference machine, it is necessary to develop new architectures to effectively support inference mechanisms defined by a fifth generation computer kernel language like PROLOG. Therefor, the inference machine may be seen as a high-level language machine.

To be concrete, architectures for highly parallel processing together with VLSI technology are indispensable to realize powerful machines, which may lead to data flow machines. Architectures for supporting object oriented programming languages and for alleviating the semantic gap problems are also important research subjects. In other words, new architectures for the inference machine will be implemented on a completely innovated von Neumann machine.

(3) Knowledge base machine

Research on architectures for the knowledge base machine will be carried out in the following steps. In the first step, a relational data base machine serving as a memory management system for the inference machine attached to it. In the second step, the extension of the developed relational data base machine to a high-level and high-performance knowledge base machine will be studied.

Advanced architectures for parallel processing, pipelining, data flow processing, set operation structure memory, hierarchical memory control, and routing networks will be very essential research subjects. In particular, the intensive investigation of the relational data base and the data flow machine together with VLSI technology will be the key to the success of the project.

(4) Intelligent interface machines

The inference machine will be integrated with the knowledge base machine to constitute a prototype of the fifth generation computer. It is clear that the fifth generation computers will have to provide various high-intelligent interfaces for the users depending on the medium of man-machine communication. The development of an integrated man-machine system to handle characters, voice, graph, and image will be discussed.

(5) VLSI and system architecture

It is true that VLSI technology has the greatest influence upon the fifth generation computers and thus it is the most important research subject for this project, although numerous problems must be worked out before putting it to practical use.

There are two major research topics on the VLSI chip production. The one is VLSI algorithms which includes the decision on what functions should be implemented and the problem of how to realize these functions on VLSI chips. The other is the support system for the design and the fabrication of VLSI chips, that is, VLSI CAD system. It seems that substantial software efforts should be made in implementing intelligent VLSI CAD systems, and that standardized terminals with intelligent man-machine interfaces are strongly desired to be developed.

Therefore, the research subjects in this field is roughly divided into the following three distinct areas:

(i) development of VLSI CAD systems,

(ii) development of a computer system called "SYSTEM 5G" on which the VLSI CAD systems are expected to run, and

(iii) development of the 5G personal computer, that is, the standard terminal used for interfacing the users with SYSTEM 5G.

SYSTEM 5G will consist of super inference machines and the 5G communication network. The former is organized based on a functional distributed architecture connecting inference machines and relational data base machines through very high-speed local networks. The latter consists of high-speed local networks and conventional global networks. Remote access to SYSTEM 5G will be possible using standard 5G personal computers, that is, "personal logic programming stations" which are very high-intelligent terminals providing an instruction set to support predicate logic and input-output facilities for multi-media of information, for instance, audio or image information.

## 7. CONCLUSION

The background and the functional characteristics of fifth generation computers have been discussed from the standpoint of computer architecture. Although there are numerous research subjects on the fifth generation computers, the preliminary discussion has led to a conclusion that the research efforts for the fifth generation computer project should be devoted to the development of very high-intelligence computer systems suitable for processing knowledge information. As a result, a logic programming language will be designed at the biginning of the project as an interface between hardware and software of the fifth generation computers to be developed. Then, advanced architectures for the associated mechanisms such as inference machine, knowledge base machine or intelligent interface machines, will be investigated in all their aspects. Intelligent VLSI CAD systems and an integrated computer system to be used for designing VLSI chips and computer architectures will also be indispensable research subjects.

ACKNOWLEDGEMENT

The author would like gratefully to acknowledge the encouragement, support, and counsel of Professor Tohru Moto-oka, University of Tokyo. He also wishes to thank Professor Hajime Iizuka, Seikei University and his colleagues, the members of the sub-committee for the fifth generation computer architecture, for their enthusiastic discussions, suggestions and assistance.

# III KNOWLEDGE INFORMATION PROCESSING RESEARCH PLAN

# PROBLEM SOLVING AND INFERENCE MECHANISMS

Koichi FURUKAWA
Computer Science Division,
Electrotechnical Laboratory, Ibaraki, Japan

Reiji NAKAJIMA
Research Institute for Mathematical Sciences,
Kyoto University, Kyoto, Japan

Akinori YONEZAWA
Department of Information Sciences,
Tokyo Institute of Technology, Tokyo, Japan

Shigeki GOTO
Data Communication Development Division,
Yokosuka Electrical Communication Laboratory,
Nippon Telegraph and Telephone Public Corporation,
Yokosuka, Japan

Akio AOYAMA
Language and Application Department,
Software Works, Hitachi Ltd. Yokohama, Japan

The heart of the fifth generation computer in prospect is powerful mechanisms for problem solving and inference. A deduction-oriented language is planned to be designed, which, with its processor, will form the core of the whole computing system. The language is based on predicate logic with the extended features of structuring facilities, meta structures and relational data base interfaces. Parallel computation mechanisms and specialized hardware architectures are extensively investigated to make possible efficient realization of the language features. The project includes an intelligent programming system, a knowledge representation language and system, and a meta inference system to be built on the core.

## 1. INTRODUCTION

It is widely predicted that knowledge information processing will be a major area of computer applications of `90`s, where problem solving and logical inference will play the central computing role.

Therefore, the heart of the fifth generation computer (FGC) is powerful mechanisms for problem solving and inference.

The goal of this paper is to clarify what the problem solving and inference functions the FGC is expected to provide and to sketch out how they should be realized. A special consideration is taken to locate the portion to be hardwared in each stage of the project development.

The mechanisms/functions of problem solving and inference that we have considered range from rather basic concepts such as list processing, pattern matching, chronological backtracking and simple Horn clause deduction, to higher level ones such as knowledge acquisition, inference by analogy, common sense reasoning, inductive inference, hypothetical reasoning and meta knowledge deduction. Furthermore, high speed knowledge information processing based on parallel computation mechanism and specialized hardware have been extensively investigated to enhance efficiency and make the whole system feasible.

The research project for problem solving and inference mechanisms is outlined as follows:

1. The design of the kernel language of the FGC (called FG-kernel language or simply FGKL): According to the current program, there will be three evolution stages for the development of FGKL to follow; the preliminary version for first three years, the first version for the following four years and the second and final version for the last three years.

2. The development of basic software systems on the FGKL including an intelligent programing system, a knowledge representation language, a meta inference

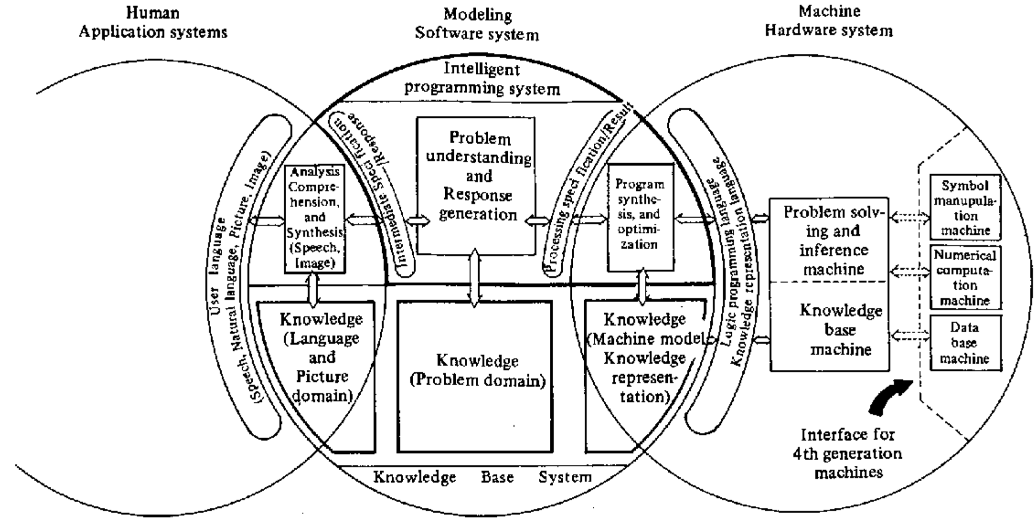



Fig. 1 The portion of problem solving and inference mechanisms in the conceptual diagram of the FGCS.

system and an intelligent man-machine interface system.

3 (ultimate goal of this project). The construction of knowledge information processing system (KIPS) by means of the basic software systems.

Fig. 1. shows the role of problem solving and inference mechanisms in the prospective KIPS.

In the next section, the R & D goals on problem solving and inference mechanisms are introduced. In Section 3, R & D schedule is described. Conclusions are given in Section 4.

## 2. Proposed Research and Development

### 2.1 Outline of the FG-kernel language

The preliminary version of the FG-kernel language (FGKL) is planned to be based on a logic programming language PROLOG (see [Battani 73], [Warren 77]) with three major extended features which are modularization mechanisms, meta structures and relational data base interfaces.

#### 2.1.1 Modularization Mechanisms

One of the most important impacts that we expect in the FGC project is to achieve a substantial progress toward the solution of the software problems. The key issue is to facilitate construction of reliable software by introducing structuring mechanisms for modular programming such as data abstraction and various parameterization.

Several data abstraction and parameterization mechanisms have been formalized and realized in different works (e.g. [Liskov 77], [Nakajima 80a], [Futatsugi 80]). Especially, many-sorted logical approach developed by [Nakajima 80a] is likely to fit into the preliminary version of FGKL. We need more research, however, to incorporate modularization mechanisms into deducion-oriented languages (see [DOGEN 81]). One possible approach is to strengthen unification power (e.g. semantic unification [Kahn 81]).

On the other hand it is highly desired to invent data structures and memory organization to allow side-effect free and still non-copying operations with functional properties. [Furukawa 81a] has recently proposed a flexible memory concept called F-trees which enables both structure sharing, semi-direct accessing and functionality.

### 2.1.2 Meta Structure

Meta level inference capabilities are essential to realize such higher level functions such as knowledge acquisition, inductive inference.

We propose two useful mechanisms for meta level inferences: one is modality and the other is meta control. The modality concept can be considered to provide a basis for various concepts in programming and artificial intelligence such as context switching, multi-world representations and frame structures. The notion of meta control has been investigated widely (e.g. [Gallaire 79], [Georgeff 79]), where the basic idea is to capture control as a set of possible sequences of operations, and the proper selection of a sequence of operations is the main objectives of the study. For instance the FGKL should provide meta control mechanisms to facilitate intelligent backtracking and concurrent execution.

### 2.1.3 Relational Data Base Interfaces

Relational data base models fit greatly the FGKL by virtue of their logical nature. Therefore it is quite natural to accept relational data bases as components of the FGCS.

There are three possible types of interface between FGKL and relational data bases:

(1) parallel execution interfaces;

(2) explicit language interfaces;

(3) higher level language interfaces.

(1) causes challenging issues and will be discussed later (section 2.2). (2) is the most straightforward approach and can be realized under the current state of the art except for the efficiency problems. We intend to adopt relational algebra as the interface language instead of relational calculus-like language, because we will probably have, in future, special purpose hardwares which efficiently execute relational algebra programs [Tanaka 80].

In the approach (3), we plan to realize a deductive question answering relational data base. Knowledge engineering approach will be adopted to realize intelligent interface between a problem solver and the external data base, where a problem solver may be a terminal user or a program on the FGC. Intelligent interfaces will be realized in the form of the meta inference system in KIPS (see section 2.5). The meta inference system has to translate queries in a problem solver's language into data base access procedures (e.g. [Yonezawa 80], [Reiter 77]). Optimization techniques should be worked out in order to achieve reasonable efficiency in the translation procedures [Furukawa 82].

### 2.2 Parallel Execution of FGKL

Logically, FGKL programs are regarded to be and/or search graphs. Therefore, the parallel execution of FGKL PROGRAMS is reduced to that of and/or search graphs. "Or" nodes and "and" nodes can be executed in parallel respectively.

Parallel execution of "or" nodes produces a possibility list whose elements satisfy one of the conditions below the corresponding "or" node.

Parallel execution of "and" nodes is rather difficult since it has to deal with the simultaneous goals. There are three possible ways to execute "and" nodes in parallel; namely,

(1) parallel execution of each "and" literal followed by a join operation to achieve the simultaneous goals;

(2) pipe line execution of a possibility list;

(3) pipe line execution of a string-type argument of a predicate.

In approach (1), logical programs have to be transformed into efficient relational algebra programs. Related works have been done by [Yonezawa 80] and [Furukawa 82].

The second approach deals with parallelism among many alternatives. In order to continue computations for all possible intermediate results in parallel, multi-world storage management is required to keep different computation environment for each possible intermediate result.

The third approach deals with parallelism in finding a single solution and is related to lazy evaluation [Clark 79]. This approach is straightforward and may offer a good exercise to attack in the first stage of the project.

It is well known that data flow mechanism is well suitable for execution of functional programs. Our objective is to work out a corresponding mechanism which is suitable for execution of logical relational programs. The difference between "function" and "relation" is that "function" produces a single answer, whereas "relation" may produce more than one.

Our conjecture is that the execution mechanism suitable for relational languages is the extension of data flow mechanism to be able to handle flow of a set of data as shown in Fig. 2.

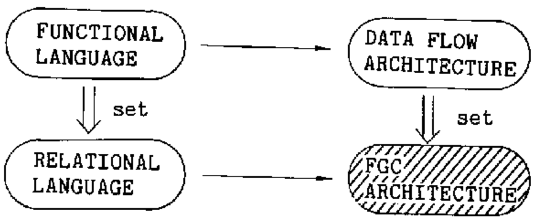



Fig. 2 The relationship between data flow architecture and FGC architecture.

FGKL programs are not logically complete and/or graphs. Often, some instructions have to be executed in a specific order. Therefore, a proper execution strategy should be made in the transformation process from logical programs into physical programs with parallelism.

## 2.3 Intelligent Programming System

One of the main targets of the FGCS project is to resolve the software crisis. In order to achieve the goal, it is essential to establish well founded software production methodology with which large scale reliable programs can be developed with ease. The key technology will likely be modular programming; that is, a way to build programs by combining component modules. It has been already noted that the kernel language will provide constructs for modularization mechanisms, but to enhance the advantage a programming system should provide suitable support. The ultimate goal is that the support system will maintain a data base with modules and knowledge on them to perform intelligent support for modular programming.

Related research themes are stated as follows.

### 2.3.1 The design of a meta language

The kernel language will be provided with a meta language in which a module itself can be described. Information on modules is to be used to facilitate module management in early stage of the project and finally to realize automatic or semi-automatic generation of module hierarchies. A final version of the language will deal with performance properties as well.

### 2.3.2 Development of a modular programming system

An integrated system for software development based on modular programing is aimed at as an intermediate goal of the FGCS project. The system will facilitate creation and modification of modules, maintain consistent relationship between modules, and furnish the user with necessary information on modules and their state of development.

A related work has been done by [Nakajima 80b], which seems a suitable starting point for the initial development of the system.

### 2.3.3 Development of automatic programming system and algorithm bank

The intelligent programming system will support the user to select suitable modules and advise him how to combine them. A knowledge engineering approach is promising in realizing such intelligent behaviors, where automatic programming systems are considered to be one of the major applications in the KIPS in prospect. The automatic programming systems will work with well defined program modules stored in a data base together with thier meta knowledge, which we call algorithm bank. Then the interactive module management system in the initial stage of project will evolved to be able to work intelligently on modules, thier creation, modification and combination.

## 2.4 Knowledge Representation Language

A knowledge representation language is strongly related to the topic of knowledge base mechanisms which will be given in a separate paper of this proceeding. Only implementation ideas will be sketched here. The image of a knowledge representation language that we have in mind can be more or less regarded as a mixture of a production system and a frame-oriented system. Our aim is to implement such a sophisticated language on FGKL. Fig. 3 shows an implementation of a simple production system on PROLOG [Furukawa 81b]. METALOG is another example of a powerful meta system for problem solving written in PROLOG [Dincbus 80]. These studies show that PROLOG is a good candidate for developing a knowledge representation language.

```
produce(X,Y,Z,U) :- prod1(X,Y,Z,U,[]).

prod1(Clist,STM,Rslt,[Rname|Plan],Hist)
 :-recognize(Clist,STM,Rname,Action),
   control_test([Rname|Hist]),
   act(Action,STM,NewSTM),
   prod1(Clist,NewSTM,
            Rslt,Plan,[Rname|Hist]).
prod1(Clist,Rslt,Rslt,[],Hist).
 /*If there are no rules which can apply,
 then the STM at that time is the Result.*/

recognize(Clist,STM,Rname,Action)
 :-prod_rule(Class, Rname : Cond => Action),
   member(Class,Clist),
   hold(Cond,STM).

hold([],STM).
hold([C|CL],STM) :- holdeach(C,STM),!,
                    hold(CL,STM).

holdeach(absent(X),[]).
holdeach(absent(X),[Fact|STM])
   :- not(X = Fact),
      holdeach(absent(X),STM).
holdeach(X = Y,STM) :- X = Y.
holdeach(found(X),STM) :- holdeach(X,STM).
holdeach(X,[]) :- call(X).
holdeach(X,[X|STM]).
holdeach(X,[Y|STM]) :- holdeach(X,STM).

act([],STM,STM).
act([Act|AL],STM,New_STM)
   :- acteach(Act,STM,Int_STM),!,
      act(AL,Int_STM,New_STM).
acteach(delete(X),[],[]).
acteach(delete(X),[X|Y],Y).
acteach(delete(X),[Y|L],[Y|L1])
   :- acteach(delete(X),L,L1).
acteach(insert(X),L,[X|L]).
acteach(replace(X,Y),[],[]).
acteach(replace(X,Y),[X|L],[Y|L]).
acteach(replace(X,Y),[Z|L],[Z|L1])
   :- acteach(replace(X,Y),L,L1).
acteach(Else, STM, STM) :- call(Else).
```

Fig. 3 A simple PRODUCTION SYSTEM written in PROLOG.

However, it is difficult to efficiently implement a frame-oriented system on PROLOG because of its lack of structuring concepts. The proposed extension for structuring mechanisms such as modality and meta control is expected to solve the problem.

## 2.5 Meta Inference System

A meta inference system serves a semantic interpreter between man and machine and also between two different machines. The interpreter must understand natural language and human mental state, and it also has to understand machine language and machine status.

Several interesting works have been done on the use of meta-level knowledge in knowledge engineering systems (e.g. [Davis 77], [Konolige 81a, 81b], [Lenat 78]). We intend to solve several apparently different problems stated below in a single framework of the meta inference system. The problems we consider include

(1) knowledge acquisition,

(2) problem solving control,

(3) belief revision,

(4) conversation control,

(5) program modification,

(6) accessing external data bases,

(7) reorganizing external data bases.

A meta inference system makes use of knowledge about

(a) inference rules,

(b) representation of objects,

(c) representation of functions/predicates and

(d) reasoning strategies

to solve the problems listed above. We owe much to Davis's work [Davis 77] for the above framework of meta knowledge.

A rough sketch of the image of the target meta inference system is shown in Fig. 4. We believe that meta level inference is a key idea to achieve a truely intelligent system.

## 3. Research and Development Schedule

A brief and tentative R & D schedule for FGKL and the intelligent programming system will be stated in this section. Schedules for other three topics mentioned in the preceding section will not be stated separately, but included in that for FGKL.

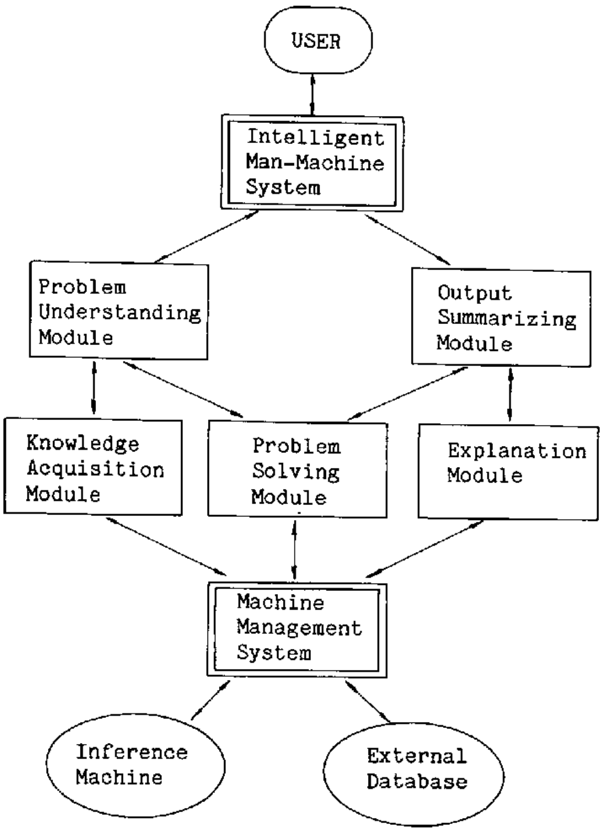



Fig. 4 An image of the target meta inference system.

## 3.1 Research on FG-kernel language

### 3.1.1 Preliminary Version

A preliminary version (the 0th version) of FGKL will be designed and implemented on conventional computers by the end of first one and half years. A firmware based FGKL machine will then be developed (the detail of the machine will be found in [Uchida 81] in this proceeding).

The preliminary version of FGKL is based on PROLOG, with extensions including

(1) modularization facilities,

(2) meta structures, and

(3) relational data base interfaces.

### 3.1.2 First Version

The first version of FGKL will be designed and its software simulater will be implemented during the first stage of the project. One of the goals of this version is a support mechanism for parallel execution.

A new parallel execution mechanism based on breadth first search of and/or graph have to be developed as well as exception handling of forced sequential execution.

The features extended to PROLOG in the preliminary version should also be refined in this stage.

Another important extension to be introduced in this stage is concurrent programming capability. [Futo 81] developed a very high level simulation language which incorporates concurrency control mechanism into a PROLOG-like language.

### 3.1.3 Second Version

The design and implementation of the second and final version of FGKL will be completed by the end of the intermediate stage.

The main feature of the final version is the ability to deal with distributed knowledge bases. Since each knowledge base has its own problem solving capability, a cooperative problem solving will become very important. One of the basic mechanisms to realize the function is message passing introduced by [Hewitt 76]. Since concurrent programming needs inter-process communication, the primitive message passing mechanism will have been developed by the end of the initial stage. The result of the research on knowledge representation language and meta inference system will be utilized in this stage to specify those functions which FGKL is to provide, and to work out means to realize them.

## 3.2 Intelligent Programming System

### 3.2.1 Initial Stage

A practical programming system will be developed to facilitate programming on FGKL. The system includes

(1) syntax-directed editing with automatic coding,

(2) debugging facilities,

(3) module management system, and

(4) program validation facilities.

The basic research theme is the development of a meta language. Program specification is to be written in the language.

The program validation will be another main goal of the intelligent programming system research. A verifier suited for hierarchical and modular programming have to be developed (e.g. [Nakajima 80a]). Extensive research on algorithm bank will be also conducted. A set of verified modules will be accumulated in the algorithm bank, which is utilized by both human programmers and automatic programming systems.

### 3.2.2 Intermediate Stage

The main research theme in this stage is to develop an automatic programming system based on modular programming. A specification/meta language have to be upgraded to deal with both functional and performance properties of programs.

### 3.2.3 Final Stage

A computer aided design system for program construction will be developed during the final stage of the project. Natural language question answering facility will be offered as user interface [Fuchi 81]. The outputs of the program CAD system will be input to the automatic programming system to produce efficient executable programs.

## 4. Conclusion

In this paper, we have described research projects related with problem solving and inference mechanisms together with their R & D schedules.

Especially, we have presented the central ideas about the kernel language FGKL for the fifth generation computer. FGKL is most crucial because it determines the basic characteristics and direction of the entire project. Thus we have discussed problem solving and inference mechanisms rather broadly from list processing to meta knowledge deduction.

We have also placed a strong emphasis on intelligent programming systems as a vehicle to propel the project and as a benchmark to justify the practical feasibility and usefulness of FGKL.

The plan described here needs huge amount of basic research and thus seems rather ambitious and risky. We think, however, that this is the very reason why the fifth generation computer project is a challenge of 90's.

Acknowledgment

Many researchers have participated in the investigation described in this paper (which started about three years ago). Especially, the Basic Theory Working Group greatly contributed to provide the basis of our project. The group, headed by Dr. Fuchi of ETL, included the following members: Professor Ito of Tohoku University, Professor Ohsuga of Tokyo University, Professor Nagao of Kyoto University and Professor Hirose of Waseda University. We would like to express our sincere thanks to all of them.

## REFERENCES

1. [Battani 73] Battani, G. and Meloni, H. "Interpreteur du Langage de Programmation PROLOG", Groupe d'Intelligence Artificielle, Marseille-Luminy, 1973.

2. [Clark 79] Clark, K. L. and McCabe, F. G. "Control Facilities of IC-PROLOG", in Expert Systems in Micro Electronic Age, (ed D. Michie), Edinburgh University Press, 1979.

3. [Davis 77] Davis, R. and Buchanan, B. G. "Meta-level Knowledge : Overview and Applications", Proc. Fifth IJCAI, 920-927, 1977.

4. [Dincbus 80] Dincbus, M. "A Knowledge-based Expert System for Automatic Analysis and Synthesis in CAD", Proc. IFIP Congress 80, 1980.

5. [DOGEN 81a] DOGEN:{Furukawa, K., Nakajima, R., Yonezawa, A.} "Himiko -- A proposal for a Kernel Language for Knowledge Information Processing Systems", in preparation.

6. [DOGEN 81b] DOGEN:{Furukawa, K., Nakajima, R., Yonezawa, A.} "Language Design for High Level Knowledge Information Processing -- Structuring Concepts and Meta-Control", in preparation.

7. [Fuchi 81] Fuchi, K. "Natural Language as a Specification Language for Programs", Lecture Note Series No. 436, Research Institute for Mathematical Sciences, Kyoto University, 1981.

8. [Furukawa 81a] Furukawa, K. and Nitta, K. "F-tree: A flexible memory with functional property", in preparation.

9. [Furukawa 81b] Furukawa, K. "Problem Solving with PROLOG", Proc. 23rd Annual Convention, IPSJ, 1981 (in Japanese).

10. [Furukawa 82] Furukawa, K. "On Intelligent Access to Relational Data Bases", To appear in Computer Science & Technologies, 1982, OHM North-Holland, 1982.

11. [Futatsugi 80] Futatsugi, K. and Okada, K. "Specification Writing as Construction of Hierarchically Structured Clusters of Operators", Proc. IFIP Congress 80, 287-292, 1980.

12. [Futo 81] Futo, I. and Szeredi, J. "T-PROLOG a Very High Level Simulation System -- General Information Manual", SZ. K. I. 1011 Budapest I. Iskola Utca 8, 1981.

13. [Gallaire 79] Gallaire, H. and Lasserre, C. "Controlling Knowledge Deduction in a Declarative Approach", Proc. Sixth IJCAI, 1979.

14. [Georgeff 79] Georgeff, M. P. "A Framework for Control in Production Systems", Proc. Sixth IJCAI, 328-334, 1979.

15. [Goto 80] Goto, S. "DURAL : An Extended PROLOG Language", Lecture Note Series No. 396, Research Institute for Mathematical Sciences, Kyoto University, 1980.

16. [Hansson 79] Hansson, A. and Tarnlund, S. A. "A Natural Programming Calculus", Proc. Sixth IJCAI, 348-355, 1979.

17. [Hewitt 77] Hewitt, C. "Viewing Control Structures as Patterns of Passing Messages", Artificial Intelligence, Vol. 8, No. 3, 1977.

18. [Hogger 81] Hogger, C. J. "Derivation of Logic Programs", JACM, vol. 28, No. 2, (April 1981), 372-392.

19. [Kahn 81] Kahn, K. M. "Uniform -- A Language based upon Unification which unifies (much of) Lisp, Prolog, and Act 1", Proc. Seventh IJCAI, 1981.

20. [Konolige 81a] Konolige, K. "A First-Order Formalization of Knowledge and Action for a Multiagent Planning System", To appear in Machine Intelligence 10, 1981.

21. [Konolige 81b] Konolige, K. "A Metalanguage Representation of Relational Databases for Deductive Question-Answering Systems", Proc. Seventh IJCAI, 496-503, 1981.

22. [Lenat 78] Lenat, D. B. "The Ubiquity of Discovery", Artificial Intelligence, Vol. 9, 257-285, 1978.

23. [Liskov 77] Liskov, B. H. et al. "Abstraction Mechanisms in CLU", ACM, Vol. 20, No. 8, 564-576, 1977.

24. [Nakajima 80a] Nakajima, R. et al. "The Programming System -- A Support System for Hierarchical and Modular Programming", Proc. IFIP Congress 80, 299-304, 1980.

25. [Nakajima 80b] Nakajima, R. et al. "Hierarchical Program Specification and Verification -- a Many-sorted Logical Approach", Acta Informatica 14, 135-155, 1980.

26. [Reiter 77] Reiter, R. "An Approach to Deductive Question-Answering", BBN Report No. 3649, Bolt, Beranek and Newman, Inc., 1977.

27. [Tanaka 80] Tanaka, Y. et al. "Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer", Proc. IFIP Congress 80, 427-432, 1980.

28. [Uchida 81] Uchida, S. et al. "New Architecture for Inference Mechanisms", in this proceeding.

29. [Warren 77] Warren, D. H. D. et al. "PROLOG -- The Language and Its Implementation Compared with LISP", SIGPLAN/SIGART Newsletter, ACM Symposium on AI and Programming Languages, 1977.

30. [Yonezawa 80] Yonezawa, A. "A Method for Synthesis of Data Base Access Programs", Research Report No. C-30, Department of Information Science, Tokyo Institute of Technology, 1980.

KNOWLEDGE BASE MECHANISMS

Motoi SUWA
Information Sciences Division
Electrotechnical Laboratory, Ibaraki, Japan

Koichi FURUKAWA
Computer Science Division
Electrotechnical Laboratory, Ibaraki, Japan

Akifumi MAKINOUCHI
Software Research Laboratory
Fujitsu Laboratories Ltd., Kawasaki, Japan

Tetsuo MIZOGUCHI
Information Systems and Electronics Development Laboratory
Mitsubishi Electric Corp., Kamakura, Japan

Fumio MIZOGUCHI
Department of Industrial Administration
Science University of Tokyo, Noda, Japan

Haruaki YAMASAKI
Distributed Systems Division
Research Laboratory
OKI Electric Industry Co. Ltd., Tokyo, Japan

One of the principal goals of the Fifth Generation Computer System project for the coming decade is to develop a methodology for building knowledge information processing systems which will provide people with intelligent agents. The key notion of the fifth generation computer system is knowledge which is used for problem solving. In this paper we describe our plan of R & D on knowledge base mechanisms. A knowledge representation system is planned to be designed to support knowledge acquisition for the knowledge information processing systems. The system includes a knowledge representation language, a knowledge base editor and a debugger. It is also expected to perform as a kind of meta-inference system. With respect to the large scale knowledge base systems, a knowledge base mechanism based on the relational model is to be studied in the earlier stage of the project. Distributed problem solving is also one of the main issues of this work.

## 1. Introduction

The aim of the Fifth Generation Computer Systems is the development of Knowledge Information Processing Systems (KIPS) based on innovative theories and technologies that can offer the advanced functions expected to be required in the 1990's by overcoming the technical limitations inherent in conventional computers.

During the last ten years a number of attempts have been made to apply the Artificial Intelligence (AI) techniques to building knowledge based systems [Feigenbaum 77]. Some of them have achieved high performance on the problems that are difficult enough to require significant human expertise for their solution. DENDRAL [Buchanan 76] and MYCIN [Shortliffe 78] are of such kinds. This fact has given us a considerable effect to recognize the important role of knowledge which is played in expert systems.

Fig. 1 shows the role of knowledge base mechanisms in the prospective KIPS.

The goal of this work is to develop cooperative knowledge based systems where problems are solved by the cooperation of intelligent agents with distributed knowledge sources. An intelligent agent works as if it is something like an excellent librarian who knows where the relevant knowledge sources exist and how to use it to get necessary information, and who even knows how to solve the problem.

With the future progress of knowledge engineering technology, it can be expected that the size of knowledge bases in

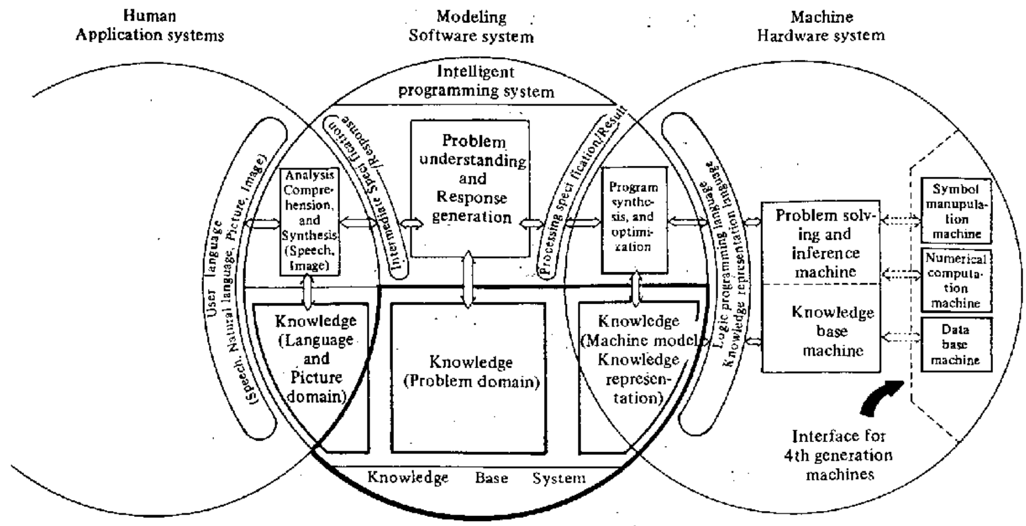



Fig. 1 The portion of knowledge base mechanisms in the conceptual diagram of the FCCS.

practical use will become bigger and bigger. We think the approach aiming at the cooperative knowledge based systems is a solution of the problem how to manage the growing scale of knowledge base in real problems. As the knowledge sources distribute over the knowledgeable agents, inference and problem solving should be executed cooperatively over those distributed knowledge sources.

The research project for knowledge base mechanisms is outlined as follows:

1. The development of a knowledge representation system. The design of knowledge representation language (called Fifth Generation Knowledge Representation Language, FGKRL in short) and support systems for the knowledge base building are planned at the initial stage of the project.

2. Basic research on knowledge acquisition: this system is the key to the cooperative knowledge based system.

3. Basic research on distributed problem solving.

4. Design of the external knowledge bases: the intelligent interface between a central problem solver and external knowledge bases is the key issue of this work. Relational algebra may be an interface language at least in the earlier stages of the project.

In the next section, the R & D goals on knowledge base mechanisms are introduced. In Section 3, R & D schedule is described. Conclusions are given in Section 4.

## 2. Proposed Research and Development

### 2.1 Outline of the System

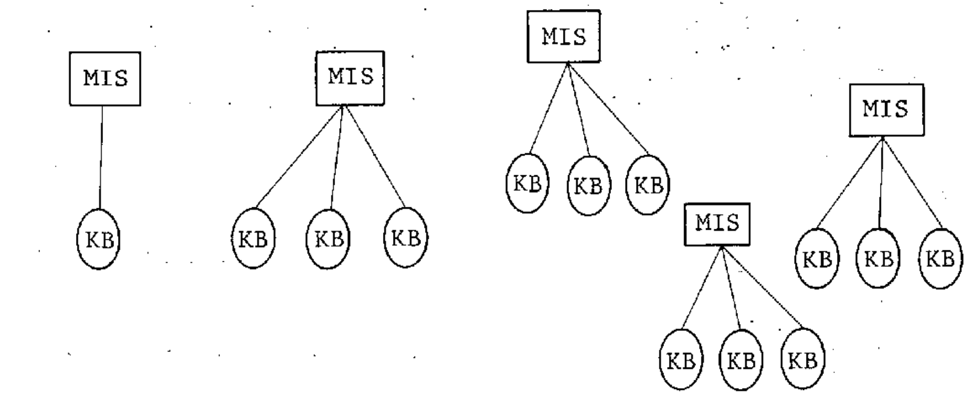
A knowledge information processing system consists mainly of a meta-inference mechanism and knowledge bases. A couple of structural varieties can be imagined ranging from a simple combination of a meta-inference mechanism and a knowledge base to a rather complicated combination of several knowledge information processing systems cooperating each other.

Fig. 2 shows three stages of the developing of the structure. The simplest one is called the single knowledge base management system which is the pilot model we are going to build during the initial stage of the project. The second one, a distributed knowledge base management system, is the intermediate target, which will have been developed by the end of the intermediate stage the project. The last one, a cooperative knowledge base management system, is the ultimate target of the project which is expected to be reached by the end of the project.

### 2.2 Knowledge Representation Language and System

#### 2.2.1 Knowledge Representation Language

Knowledge representation languages are very important tools for building Artificial




(a) Single knowledge base management system.

(b) Distributed knowledge base management system.

(c) Cooperative knowledge base management system.

MIS: Meta-inference Systems
KB: Knowledge Base

Figure 2. Three stages of the development of the knowledge based system.

Intelligence systems. Therefore, researchers in AI have been involved in developing schemes for knowledge representation, in which, for instance, the semantic network, the production systems, and the frame structure are included. These schemes have been derived from the deep observation on "knowledge" to be described. The semantics and deductive theory of those schemes, however, has not been satisfactorily developed.

We intend to implement some of the knowledge representation schemes mentioned above in a preliminary version of FG-kernel language (FGKL), which is based on PROLOG. During the implementation process, we expect that we will be able to capture such important concepts as control and frame structure in terms of logic and thus to design a new knowledge representation language FGKRL in higher order and modal logic frame work.

FOL [Weyhrauch 78] and Omega [Hewitt 80] are forerunners of our knowledge representation language FGKRL. Both of them have a theory based on meta knowledge and also are self descriptive.

Another important theoretical work is the development of non-monotonic logic [McDermott 80] which is intended to give a theoretical foundation of common sense reasoning. An operational system, called Truth Maintenance Systems (TMS), for realizing the above logic has been developed by [Doyle 79]. TMS performs a kind of meta inference about object level inference to find out causes of contradiction.

### 2.2.2 Support Systems for Knowledge Bases

The knowledge information processing system described in this paper is heavily dependent on a large knowledge base of domain-specific information. Many research efforts have been concentrated to find out the proper way to construct such knowledge bases. EMYCIN [van Melle 79] and EXPERT [Weiss 79], for instance, have been proved very useful and helpful to assist system builders in construction of knowledge bases for expert systems.

We believe it is very important to encourage the research and development activities of programming environment for knowledge bases. The followings should be developed extensively in the course of our project;

(1) knowledge base editors,

(2) knowledge base debuggers,

(3) knowledge acquisition support systems,

(4) consistency checkers, and

(5) knowledge base compilers.

Among the support systems mentioned above, we definitely put the strongest stress on the importance of knowledge acquisition problem. The approach to the problem ranges from manual to fully automatic. Our ultimate goal is to develop fully automatic systems which collect domain-specific knowledge directly from computer-understandable sources. At the earlier stage of the project, however, we try to build semi-automatic systems which allow the domain experts to communicate to the representation system and to interact with the knowledge base.

### 2.3 Knowledge Acquisition

We admit that knowledge acquisition research is very young and that results by now are not sufficient for planning a systematic research project on the topic. We, however, try to draw a rough sketch on the project since it ranges the coming ten years.

Knowledge acquisition researches may be grouped into the following three categories:

(1) case studies on acquiring various kinds of knowledge,

(2) the developments of support system for knowledge acquisition based on meta knowledge, and

(3) cognitive science approach based on human mental model.

The aim of approach (1) is to establish such basic technologies as knowledge acquisition by induction, analogy and so on. Although the developments of practical knowledge acquisition systems depend heavily on the results of these basic researches, we cannot afford to make intensive researches on the topics.

In approach (2) and (3), the target is to develop a support system for a domain expert to transfer his knowledge into knowledge bases. The requirements for such systems are:

(a) domain experts need not know the detail of knowledge representation,

(b) the support system must be responsible to keep the knowledge base consistency, and

(c) the support system has properly to guide domain experts to provide valuable knowledge to the system.

Requirements (a) and (b) are concerned with the approach (2), and (c) with (3). Davis developed TEIRESIAS ([Davis 78]), which is a meta system for developing knowledge based system. TEIRESIAS has meta-level knowledge about knowledge representation and utilizes it to acquire new rules from human experts.

We will make our knowledge acquisition research starting from Davis's work and pursue a better formalism which satisfies the requirement (c) as well as (a) and (b). In particular, we intend to develop a TEIRESIAS-like system on the FGKL and study the knowledge acquisition problem through experimental work.

### 2.4 Distributed Problem Solving

As concurrent programming techniques greatly enriched the descriptive power of programs, distributed problem solving, which is problem solving analogue of concurrent programming, is expected to greatly strengthen the problem solving power.

In order to establish distributed problem solving, multiple processes (or agents) will have to work cooperatively by exchanging messages to each other, as processes in concurrent programs do.

The different points of distributed problem solving from concurrent programming are:

(a) actions of each agent are not fully described in advance, and thus need planning, and

(b) each agent has to possess problem solving power.

The difficulty of multiple-agents planning is that a planner has not sufficient knowledge to solve the entire problem by himself and has to make a plan depending on knowledge about problem solving abilities of other agents ([Konolige 80]).

Even there may be a case where a planner does not have any knowledge about other agents and makes task announcement to all other agents expecting that some of them will apply for the task ([Davis 78]).

To realize cooperative problem solving, we need concurrency control of multiple problem solvers. TPROLOG developed by [Futo 81] is a concurrent version of PROLOG and we appreciate it as a good starting point of our research. The language UNIFORM [Kahn 81], in which the unification of most features of ACT1, LISP and PROLOG is proposed, is also a notable forerunner.

The handling of external knowledge base which will be discussed in the following section is a typical example of distributed problem solving. In that example, we need one more important feature besides the above two; namely, understanding of interface language and knowledge representation detail of the external knowledge base.

## 2.5 External Knowledge Bases

### 2.5.1 External Relational Databases

Since many knowledge sources are to be developed separately, it is crucial that inference systems can effectively utilize external knowledge bases in order to realize large scale knowledge-based systems.

One of the important issues for realizing such systems is a good selection of the interface language. We intend to select, at least in the initial stage of the project, relational algebra interface because of the following three reasons:

1. Relational model fits to the FG-kernel language since both of them are based on first order logic.

2. It is possible to make interfaces to existing databases.

3. Relational algebra machines with very high performance are expected to be developed in a few years.

Actually, the development of relational algebra machines are scheduled in the earlier stages of the entire project.

We intend to grade up the interface language to allow deductive question answering capability. Relational database systems will be extended to support such high level functions as accessing through multiple views, intelligent query optimization and self reorganization of internal structure.

To accomplish such extensions, we need to work out on formal descriptions of data representation and data manipulating operations. Those descriptions will take the role of meta knowledge and will be used in planning data access and reorganization ([Mark 80], [King 80], [Konolige 80], [Furukawa 82]).

The entire system is to be formulated as a distributed problem solving system described in 2.4.

### 2.5.2 Image and Text Databases

The high level of the performance of the knowledge information processing systems is supported partly by the intelligent interface systems introduced in [Tanaka 81]. In the intelligent interface systems, image databases and text databases play very important roles to achieve the high performance.

The key issue to the image databases is to provide them with the ability of structuring images which extracts features from given image and then compose the relationship between the features. The symbolic access of the image data will be activated by this technology.

The text database should have the ability of text understanding which can abstract the meaning of texts and structurize the relationship between texts fragments for intelligent retrieval.

## 3 Research and Development Schedule

A brief and tentative R & D schedule for knowledge base mechanisms will be stated in this section.

### 3.1 Knowledge Representation Language and Support Systems

[Initial stage]

A preliminary version of FG-KRL and its support systems are to be designed and implemented in the 0th version of FGKL on conventional computers by the end of the first three year period of the project.

The framework of the FG-KRL is production rules with frame-like structure description mechanism. The language is also to be designed to deal with such kind of raw data as pictures.

The support systems include a knowledge base editor, a knowledge base debugger, a knowledge acquisition supporter, a consistency checker and a knowledge compiler.

[Intermediate stage]

A first version of FG-KRL is to be implemented with the extension of the functions which deal with distributed knowledge bases. In order to realize these functions, meta knowledge and meta inference mechanism are to be extensively studied.

Support systems is to be extended to the first versions of them, respectively. Main extension will be done on the management of meta-knowledge.

Knowledge acquisition system in this version is to be implemented based on the knowledge acquisition formalisms constructed upon meta-level knowledge management mechanism.

[Final stage]

The second and the final version of FG-KRL is to be used to construct distributed problem solving systems which provide users with flexible and friendly information processing system. The programming environment is to be vastly improved compared with the traditional systems.

### 3.2 Large Scale Knowledge Base

[Initial stage]

A preliminary version of the Fifth Generation Large Scale Knowledge Base (FG-LSKB): The interface between an inference mechanism and knowledge bases will be studied to establish the method to access knowledge bases employing relational model. The inference ability is not necessarily expected in the knowledge base itself.

[Intermediate stage]

The first version of FG-LSKB: This version is expected distributed. Some amount of inference ability should be given to knowledge bases.

[Final stage]

The last version of FG-LSKB: The final image of the FG-LSKB is that it provides the knowledge information processing system with the mechanism to carry out the inference with mutual assistance among knowledge bases.

## 4. Concluding Remarks

In this paper, we have described research projects related to the knowledge base mechanisms together with their R & D schedules. Especially, we have put the stress on the importance of knowledge which is used to solve problems.

The central mechanism of the knowledge system includes the knowledge base mechanism based on the relational database, and the knowledge representation systems.

Although the policy of the project is to develop general purpose systems which do not depend on a particular domain of problem, the research activities should be carried out on the basis of application oriented.

The ultimate purpose of developing the Knowledge Information Processing System (KIPS) is to provide people who are only potential users of current information processing systems with smart and friendly facilities in terms that they can easily let the computer do "what" they want to do.

The plan described here needs huge amount of basic researches and thus seems rather ambiguous and risky. We believe, however, that this is the very reason why the fifth generation computer project is a challenge to 90's.


## Acknowledgment

Many researchers have participated in the investigations described in this paper (which started about three years ago). Especially, the Basic Theory Working Group greatly contributed to provide the basis of the project. The group, headed by Dr. Fuchi of ETL, included the following members: Prof. Ito, Tohoku University, Prof. Ohsuga, the University of Tokyo, Prof. Nagao, Kyoto University and Prof. Hirose, Waseda University. We would like to express our heartly thanks to all of them.


## REFERENCES


[Buchanan 69] Buchanan, B., Sutherland, G. and Feigenbaum, E. A., "Heuristic DENDRAL: a Program for Generating Explanatory Hypotheses in Oeganic Chemistry," in Machine Intelligence, Vol. 4 (Meltzer, B. and Michie, D., eds), 1969.

[Davis 78] Davis, R., "Knowledge Acquisition in Rule-Based Systems -- Knowledge about Representations as a Basis for System Construction and Maintenance," in Waterman, D. A. and Hayes-Roth, F. (Eds.), Pattern-directed inference systems, New York: Academic Press, 1978.

[Doyle 79] Doyle, J., "Inductive Reasoning on Recursive Equations," Artificial Intelligence, Vol. 12, No. 3, 1979.

[Feigenbaum 77] Feigenbaum, E. A., "The Art of Artificial Intelligence: 1. Themes and Case Studies of Knowledge Engineering," Proc. of IJCAI-77, 1977.

[Furukawa 81] Furukawa, K. et al., "Problem Solving and Inference Mechanisms," in this proceedings.

[Furukawa 82] Furukawa, K., "On Intelligent Access to Relational Data Bases," to appear in Computer Science & Technologies, Ohm North-Holland, 1982.

[Futo 81] Futo, I. and Szeredi, J., "T-PROLOG a Very High Level Simulation System -- General Information Manual," SZ. K. I. 1011 Budapest I. Iskola Utca 8, 1981.

[Hewitt 80] Hewitt, C. et al., "Knowledge Embedding in the Description Systems Omega," Proc. of First AAAI, 1980.

[Kahn 81] Kahn, K. M., "UNIFORM -- A Language Based upon Unification which Unifies (much of) Lisp, Prolog, and Act 1," Proc. of IJCAI-81, 1981.

[King 80] King, J. J., "Intelligent Retrieval Planning," Proc. of First AAAI, 1980.

[Konolige 81] Konolige, K., "A Metalanguage Representation of Relational Databases for Deductive Question-Answering Systems," Proc. of IJCAI-81.

[Mark 80] Mark, W., "Rule-Based Inference in Large Knowledge Bases," Proc. of First AAAI, 1980.

[McDermott 80] McDermott, D. and Doyle, J., "Circumscription -- A form of Non-Monotonic Reasoning," Artificial Intelligence, Vol.13, No. 1,2, 1980.

[Shortliffe 76] Shortliffe, E. H., "Computer-based Medical Consultations: MYCIN," American Elsevier, New York, 1976.

[Tanaka 81] Tanaka, H. et al., "Intelligent Interface Systems," in this proceedings.

[van Melle 79] van Melle, W, "A Domain-Independent Production-Rule System for Consultation Program," Proceedings of IJCAI-79, 1979.

[Weiss 79] Weiss, S. M. and Kulikowski, C. A., "EXPERT: A System for Developing Consultation Models," Proceedings of IJCAI-79, 1979.

[Weyhrauch 78] Weyhrauch, R., "Prolegomena to a Theory of Formal Reasoning," AI Memo AIM-315, 1978.

# INTELLIGENT MAN-MACHINE INTERFACE

Hozumi TANAKA
Electrotechnical Laboratory

Seibi CHIBA
Nippon Electric Co., Ltd

Masatsugu KIDODE
Toshiba, R&D Center.

Hideyuki TAMURA
Electrotechnical Laboratory

Tsuguo KODERA
Sharp Co. Ltd.

One of the goals of research into intelligent man-machine interfaces is to develop fundamental techniques which will afford flexible interactive facilities for the fifth generation computer systems. The plan for researches into intelligent man-machine interfaces is divided into three fundamental categories: 1) natural language processing, 2) speech processing, and 3) picture and image processing. These researches will also help provide the foundation for the development of such basic application systems as an intelligent question answering system and a machine translation system.

## 1. Introduction

Human beings communicate using a wide variety of forms: natural language, both spoken and written, pictures, images, documents, and the like. It is not easy for current computers to understand these natural input/output forms and to respond to them intelligently, since they are not equipped with intelligent man-machine interfaces.

We can observe a considerable progress in modern computer terminal for front-end users. In many senses, those terminals as man-machine interface are enhancing the usability of computers. For example, a high resolution and/or color CRT is extending the versatility of computer graphics. As another trend, it should be noted that highly interactive and easy-to-use text editting capabilities become essential to current computer systems[15]. In this sense, say, the Xerox 8010 Star workstation may be a typical example of future computer terminals.

However, such developments are still continuing in commercial base. Both from hardware and software sides, many new functions will be built in. The straight-forward extension of current technology could be expected in the forth generation. Thus, we aim at more intelligent interface in our fifth generation computer system (FGCS). As a research and development tool, a state of the art terminal may be used in our FGCS Project. In the final FGCS, however, the intelligent inteface, which facilitates man-machine communications in the natural I/O forms mentioned above, will be constructed.

In the 1970's, a great deal of research efforts was made toward developing intelligent man-machine interfaces, especially in the field of artificial intelligence, computational linguistics, and pattern information processing. Although these research efforts were of limited scope, there have been a few promising results. Using these as a start, we will, over the next ten years, be able to move further ahead in the research and development of intelligent man-machine interfaces, which will be better able to assist both professional and non-professional computer users.

One of the goals of research into intelligent man-machine interfaces is to develop fundamental techniques which will afford flexible interactive facilities for the fifth generation computer systems (FGCS). This research will also help provide the foundations for development of such basic application systems as an intelligent question answering system and a machine translation system.

The plan for researches into intelligent man-machine interface is divided into three fundamental categories: 1) natural language processing, 2) speech processing, and 3) picture and image processing. Generally speaking, an intelligent man-machine interface system will itself be a kind of knowledge information system composed of a front-end processor of various input/output forms, flexible knowledge based systems and problem solving/inference systems. However, we shall use the term "intelligent man-machine interface system" in a narrow sense in this article. The term will be used to denote only a front-end processor for input/output in the forms such as natural language, both spoken and written, pictures and images.

Our research schedule will be divided into three stages: 1) the initial stage (3 years), 2) the intermediate stage (4 years), and 3) the final stage (3 years).

The initial stage is planned to be an experimental stage for creating plans and goals for the intermediate stage in which a small scale pilot model will be demonstrated. A prototype of an intelligent man-machine interface incorporating a knowledge based system, and a problem solving and inference system will be developed in the final stage. These plans will be explained in more detail in the following sections.

## 2. Natural language processing

The larger the number of computer users becomes, the more urgent will become the need for high level conversational language. Natural language is the highest level conversational language and is also the one which non-expert users prefer to use. The development of more advanced techniques for processing natural languages has thus become indispensable, though it is very difficult computational task. This is one of the most important goals in our project on natural language processing.

Still, we would also like to emphasize here some other points in our research which we expect to have a significant effect on the problem of huge amounts of text data and documents, a problem which will become more acute as the volume of such data increase drastically in our offices, schools, and so on in the near future. In Japan, as elsewhere, the recent rapid progress in word processing techniques will no doubt increase the volume of text data and documents that have to be handled by computer to an intractable level. Sooner or later, as the problem of extracting useful information becomes more severe, we will have to turn to computing power in order to process these huge amounts of documents at reasonable speed. Our research on intelligent

| Languages | Japanese, English, .... |
|---|---|
| Vocabulary size | Depends on application:<br>For front-end processing of question answering system; 5,000 - 10,000 words.<br>For text data processing; more than 50,000 words. |
| Number of grammar rules | Less than 2,000. |
| Application domains | Front-end processor of question answering system:<br>Information retrieval, computer aided instruction, medical consultation, decision support system, .... .<br>Text data processing:<br>Scientific literature, technical manuals, newspapers, .... .<br>Machine translation. |
| Processing speed | 0.05 sec. for each sentence (of 25 words) (including both syntactic and semantic processing). |
| Accuracy | 99 % of input sentences parsed correctly in syntactic processing without human intervention (post editing etc.). |

Table 1. Specifications of natural language processing system.

man-machine interface will help to solve this problem.

Another goal of our research is the development of machine translation systems and question answering systems as basic application systems for fifth generation computers. The results obtained from the research on natural language processing is to be utilized for the development of these improved basic application systems and research on natural language processing, and the development of these basic application systems will be carried out side by side. The difference between these two categories of research is that the former is more fundamental than the latter.

## 2.1 Research and development goals

Natural language processing is a very difficult computational task involving numerous problems that will have to be solved one by one. The following is the target specifications of our natural language processing system:

## 2.2 Primitive techniques

The following are the primitive techniques which we have to develop in our research on natural language processing:

1) Determination of a basic vocabulary in each application domain;

2) Creation of a text data base in each application domain;

3) Development of a flexible parser;

4) Development of grammar rules;

5) Morpheme and syntactic analysis;

6) Sentence generation;

7) Semantic analysis;

8) Pragmatic analysis;

9) Natural language processing machine.

Subjects 1) to 6) are the main ones to be taken up in the initial stage. According to our experience, 2,000 words which appear most frequently in ordinary Japanese sentences account for more than 80% of all natural language texts. From this, we are able to infer that a basic vocabulary of less than 5,000 words is sufficient.

In the initial stage, we investigate the parsing techniques based on logic programming developed at Edinburgh University.[34] This approach will be taken, since our nucleus programming language will be a somewhat modified version of Prolog, which is a typical logic programming language. We feel that parsing techniques based on logic programming show promise because they make it easy to implement augmented context-free grammar rules in a natural way. Through a careful study of logic programming based parsing, we hope to be able to elucidate some of the deficiencies of the current version of Prolog, and contribute toward designing a better nucleus language.

Sentence generation will be taken as a basic technique for the machine translation system. This technique will be useful not only for a machine translation system but also for designing the explanation functions which an intelligent question answering system must have.

In the latter stages, the focus of our research shifts to semantic analysis and pragmatic analysis. Semantic analysis is the main research topic of the intermediate stage.[40],[43],[44] We have to decide what representation format[25] will be better for the semantic representations.[1],[2],[3] Case frame notation[11] is one of the promising candidates for semantic representation, but we should not confine ourselves to only one formalism. If we use Montague Grammar[9] for natural language processing, it will be reasonable to adopt logical formulae as semantic representations. Therefore, in the initial stage, we have to evaluate the various kinds of semantic representation foramlisms and choose the best.

Since the theory of pragmatics has not been fully developed, we have to delay our research on pragmatic analysis until the final stage. However, in the intermediate stage, we will try to develop the most promising approach for pragmatic analysis and in this will look into such currently available techniques as the extraction of current theme or focus from a given sentence, detection of focus shift, resolution of abreviations and determination of anaphoric relations.

## 2.3 Natural language processing in basic application systems

The relative role of natural language processing in a question answering system[35],[42] is depicted in the Fig. 1. The portions enclosed by the straight bold lines are encompassed by our research into natural language processing. The dashed lines in

| | | |
|---|---|---|
| Speech input | | |
| | Recognition object | Continuous speech. |
| | Vocabulary size | 50,000 words. |
| | Word recognition rate | 95 %. |
| | Speaker | Multiple speakers with moderate adaptation. |
| | Pronunciation | Accurate and careful. |
| | Processing speed | 3 times real time. |
| Speech output | | |
| | Vocabulary size | 50,000 words. |
| | Speech synthesis | Synthesis by rule. |

Table 2. Specifications of speech processing system.

the figure denote transfer phases between a source language and a target language in a machine translation system.[15] Here we adopt a bilingual approach in the initial stage, and the shift to a multilingual approach in the middle stage. By adding speech processing at the bottom of the figure, we will have a question answering system with voice input. We will also have a simultaneous interpretation system in the case of combining speech processing with a machine translation system as shown in the figure.

## 3. Speech processing

As the purpose of speech processing will be the same as that mentioned in the section 2, we will not repeat it here. Over the last 10 years, the techniques of speech processing have advanced very much in Japan. Speech recognition systems have been in practical use even though they have been based on word recognition technique. Fifth generation computer systems will be endowed with more advanced speech recognition systems. As mentioned in the section 2.3, a speech recognition system will be developed in conjunction with a natural language processing system.[29,37]

### 3.1 Research and development goals

The final goal is to realize a speech processing system[6,8,19,30] which is much more advanced than the present state of the art. The target specifications of the system are shown in Table 2.

### 3.2 Primitive techniques

The following are the primitive techniques which we have to develop in our research into speech processing:

1) Speech analysis;

2) Feature extraction;

3) Phoneme classification;

4) Basic speech synthesis;

5) Database of speech wave;

6) Adaptation or tuning methods for individual speaker differences;

7) Spoken sentence understanding;

8) Japanese speech output;

9) Hardware system for speech understanding;

10) Intelligent speech interface system.

In the initial stage, speech analysis and feature extraction methods and basic speech synthesis methods will be investigated and established.[12,20,36]

In the intermediate stage, the adaptation or tuning methods for individual speaker differences, spoken sentence understanding methods and a hardware system for speech understanding will be

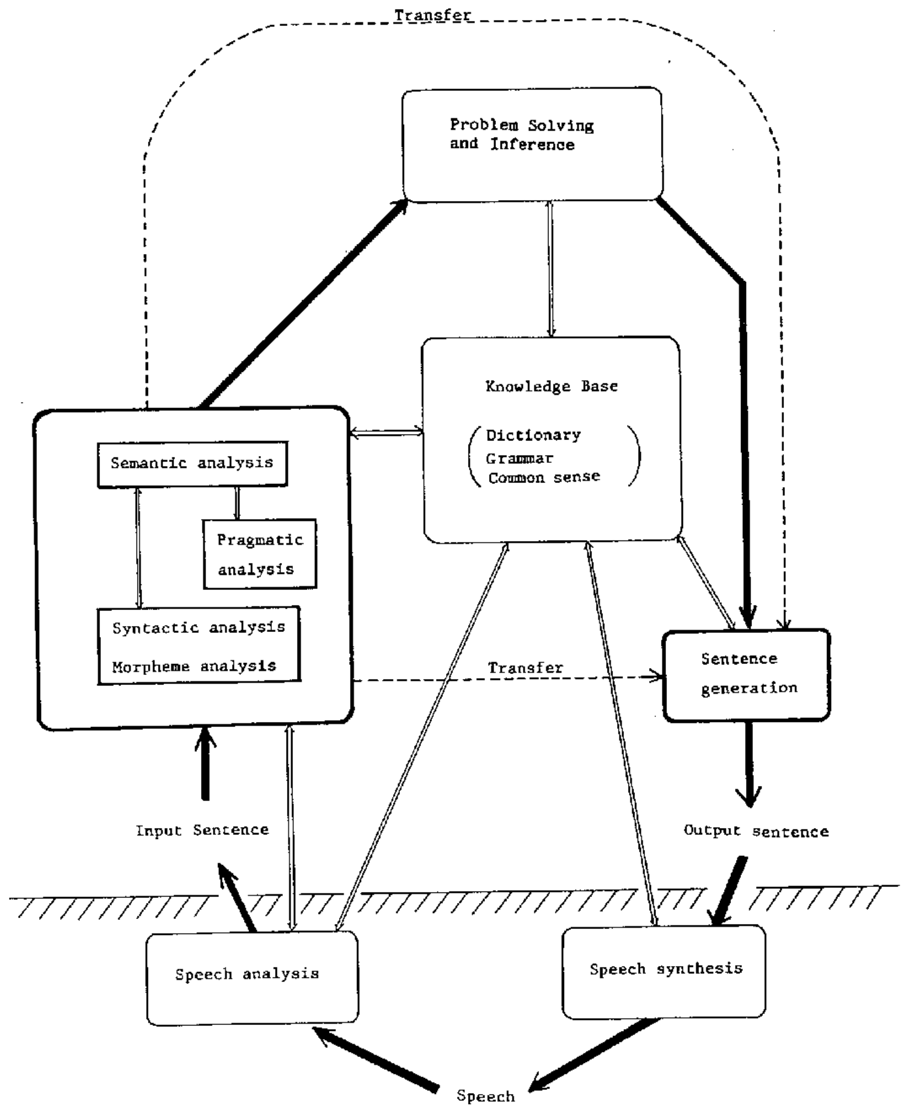



Fig. 1. Concept of natural language and speech processing in basic application sytems.

investigated. The following subgoals are to be attained by the end of this stage:

- Recognizing continuous speech using 10,000 words;

- Processing speech at several tens of times of real time;

- Other conditions are the same as shown in Table 2.

In the final stage, a spoken sentence understanding system, a speech output system and several speech application systems including a voice activated typewriter and a voice dialog system will be developed. The results obtained from the research into natural language processing will be utilized to develop a prototype intelligent speech interface system.

Let us turn next to the problem of how to approach the goals. The speech understanding process can be divided into two processing phases:

1) acoustic level processing starting with speech analysis and ending with phoneme classification;

2) language level processing wherein the meaning of the speech is understood using the results of the acoustic processing.

Acoustic level processing is peculiar to speech. The main problem here is how accurately to transform speech waves into phoneme sequences. For this purpose, research is needed concerning speech analysis methods with fine resolution both in frequency and time, feature extraction methods which extract only those components useful for phoneme classification, and phoneme classification methods which transform feature parameter sequences into phoneme sequences. In this stage, as it is very difficult to determine each phoneme uniquely, so called the phoneme lattice will be used which assigns multiple possible phonem at a single time portion.

In language level processing, the phoneme lattice is processed, and understanding of the meaning of the sentence is undertaken. Although this process has many common features with natural language processing, more complex and highly involved processing methods are required for syntax analysis, semantic analysis and so on. It is because the processing starts from a phoneme lattice including uncertainties rather than from a reliable character string.

The research activities are supported by research tools such as large scale computers and high performance personal computers supporting Lisp and/or Prolog. In the final stage, speech processing algorithms developed are to be implemented on the high speed signal processors and Lisp/Prolog machines developed in this fifth generation computer project.

The role of speech processing in basic application systems is shown in Figure 1.

## 4. Picture and image processing

One of the most important functions in the intelligent man-machine interface is to provide an environment in which a user can handle pictorial and image data as a knowledge source. Various kinds of knowledge are obtained from source data which are represented in two-dimensional forms, that is, as picture and image. Image-oriented knowledge is, generally speaking, composed of structural and spatial features (sketches) which are described in a symbolic form, and/or the image itself in a two dimensional signal form.

Here at the intelligent interface, pictorial and imagery source data are stored, manipulated and retrieved as a knowledge base (image database). Within the frameworks of this image database, image understanding and image-oriented application studies can be performed in efficient ways; for example, line drawing interpretation in CAD/CAM,[46] aerial and satellite image understanding in remote sensing,[4),21),27] chest X-ray image analysis in medical diagnosis,[33),39] and so on.

### 4.1 Research and development goals

Our major goals in this research are as follows:

- to show attractive possibilities for the utilization of two-dimensional signal data (picture and image) as a knowledge source;

- to implement an intelligent image database system in which pictorial and imagery data are input, converted into sketches and/or compressed for editing, retrieved by image examples and similarities, output for displaying and further processing, and so forth;

- to provide an intelligent knowledge acquisition subsystem;

- to demonstrate that this intelligent database can be efficiently used in practical image understanding systems such as handwritten drawing recognition in the VLSI CAD system, and X-ray image analysis in the medical consultation system.

Figure 2 shows a fundamental concept of the intelligent man-machine interface from the viewpoint of image database system.

### 4.2 Primitive techniques

The following techniques may be required to realize an intelligent man-machine interface with the functions above. Each primitive technique is a stand-alone function, which facilitates incremental system development and integration.

1) Intelligent data input: to input pictorial and/or imagery data at high speed with intelligent control.

2) Image-sketch-relation conversion: to extract features (shape, geometric,

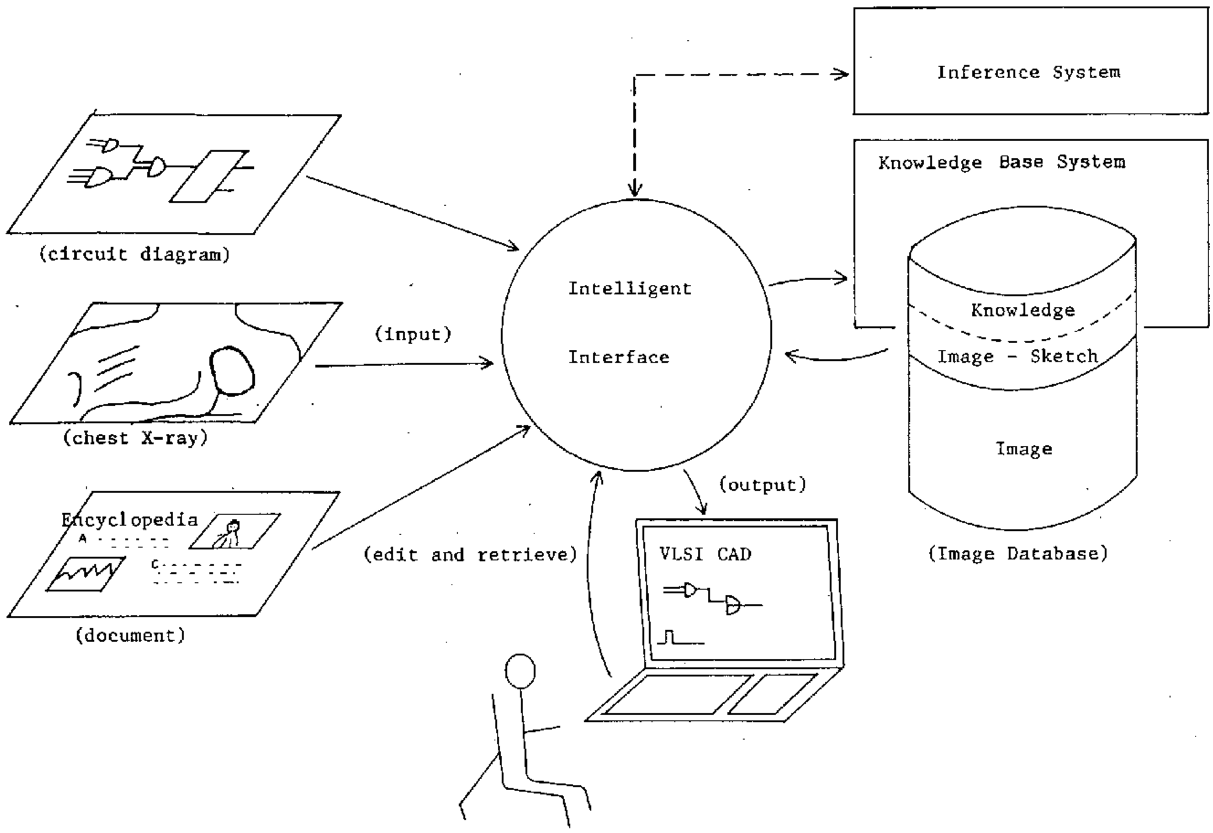



Fig. 2. Concept of pictorial functions at intelligent interface.

gray level, texture, etc.) and structures (spatial relationships) as sketches from images. These sketches are represented in symbolic form and stored in a relational knowledge base.[19),22)]

3) Efficient image storage: to compress the original two-dimensional data to efficient codes if necessary for images which cannot be converted into sketches. Otherwise, these images are stored as they are stored in two-dimensional form into a high speed image database machine.

4) Flexible data manipulation: to manage the above two databases by operations of insertion, deletion, and replacement.[4)] These editing operations may be performed interactively at an easy-to-use console. In addition, special image operators should be prepared for constructed sketches and images. Desired images in the database can be retrieved by giving examples and calculating similarities.[5)] These query examples are provided in a string of symbols and/or by the pictorial form.

5) Intelligent data output: to reconstruct images from sketches and from compressed data in the display console. Here, we need a display mechanism to look at a local area in detail as well as to monitor a global outline of given large image data. In addition, hard copies are to be generated with high-resolution in an attractive color format.

6) High level control: to control the above primitive functions by high level control languages; image representation, description of manipulation procedure. High level protocols to the inference engine and knowledge base system are defined as an intelligent interface.

Figure 3 depicts the integration of the technique mentioned above for pictorial and imagery data management.

### 4.3 Research and development schedule

The above mentioned picture and image processing functions are implemented as

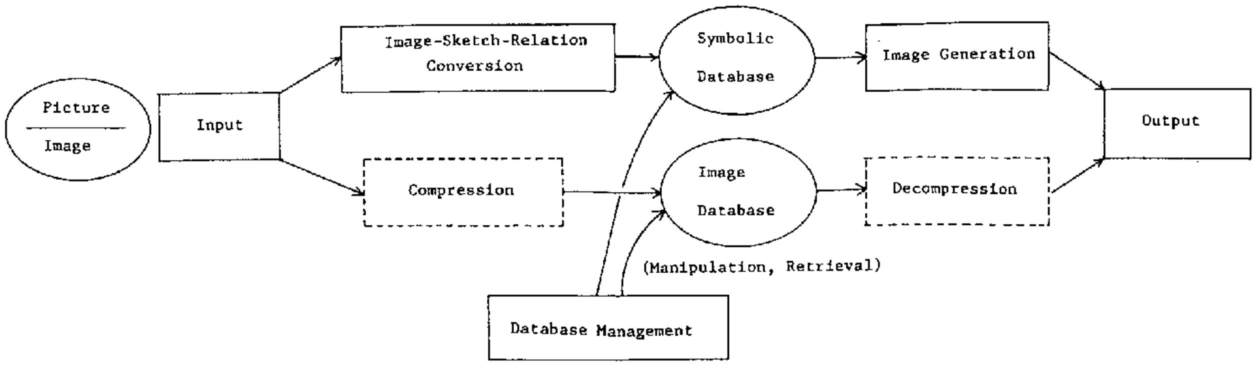



Fig.3 Management of pictorial and imagery database.

system components of intelligent interface. They can be divided into two types: hardware components and software components.

A) Hardware components:

Several feature extracters are to be specially designed to convert two dimensional signal data into sketches at high speed, including required preprocessors. The selection of appropriate preprocessing and feature extracting functions is one of the major research subjects. The machine architecture and design for effectively performing the desired functions with sufficient throughput is related to VLSI development.[32]

The specifications and design of the optimum image database machine may become a subject of much research. The image database machine is required to have quick access time, high speed data transfer and mass storage capacity with rewritable function. Various kinds of pictorial and imagery data are stored in terms of pixel size, bit per pixel, resolution, etc.

Because of advances in image I/O devices, optical character readers and easy-to-use console equipment, we can expect much of the intelligent hardware for the full interface system.

B) Software components:

High level languages are expected to be developed for image manipulation description and procedure representation.[45] The handling of pictorial and imagery data in a program is one of the key issues. Knowledge based procedure language is to be efficiently used in an interactive model.

A query language is prepared for retrieving desired images by giving image examples and calculating their similarities. These high level languages are interfaced with the inference engine and knowledge base system through a high level protocol.

Advancing image processing and pattern recognition techniques are also to be integrated into this intelligent interface.

4.4 Research and development schedule

The research and development schedule is divided into the following three stages corresponding to the initial, the intermediate and the final stage mentioned in the section 1.

1) Experimental stage: mainly research and development into high level languages in an experimental system. Design study on hardware architecture of feature extracters, display generators, and image database engine to be carried forward.

2) Pilot model implementation stage: integration of a small-scale intelligent interface with the associated hardware in conjunction with the inference machine and knowledge base machine.

3) Prototype implementation stage: integration of a full-scale intelligent interface. Application studies are to be carried out to demonstrate such attractive posibilities in the pictorial and imagery knowledge base system, as handwritten drawing recognition for VLSI CAD and chest X-ray analysis for medical consultation system.

## 5. Final image of intelligent interface

In the final stage, all researches mentioned above are to be integrated into a subsystem called Intelligent Interface in FGCS. The system image is shown below:


## Acknowledgements

Many reseachers have contributed to this article. We would like to express our sincere thanks to each member of the Basic Theory Working Group, Dr. Fuchi of ETL as the head, Professor Nagao of Kyoto University, Professor Ohsuga of Tokyo University, Professor Ito of Tohoku University and Professor Hirose of Waseda University.

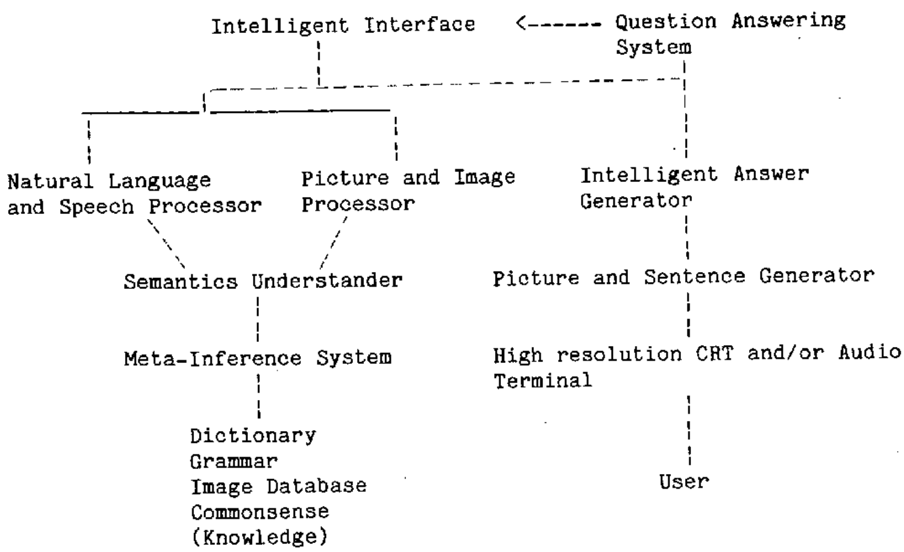




Fig. 4. Intelligent interface system.

REFERENCES


1) Barr, A., and Davidson, J., Representation of Knowledge, Stanford Memo, HPP-80-3, Computer Science Dept., Stanford Univ., (1980).

2) Bobrow, D.G., and Winograd, T., An Overview of KRL, a Knowledge Representation Language. Cognitive Science, 1, 1 (1977), 3-45.

3) Bobrow, D.G., and Collins, A.M. (Eds.), Representation and Understanding, Academic Press, (1975).

4) Bryant, N.A. and Zobrist, A.L., IBIS: A Geographic Information System Based on Digital Image Processing and Image Raster Data Type, IEEE Trans. Geosci. and Electron, GE-15, 3, (1977), 152-159.

5) Chang, N.S. and Fu, K.S., Query-by-Pictorial Example, IEEE Trans. on Software Eng., SE-6, (1980), 6, 519-524.

6) Chiba, S., Watari, M. and Watanabe, T., Speech Recognition System, PIPS-R-No.25, Electrotechnical Laboratory, (Dec. 1980), 157-165, (in Japanese).

7) Davis, R., and King, J., An Overview of Production Systems, Stanford AIM-271, Oct (1975).

8) Dixson, N.R.and Martine, T.B. Eds., Automatic Speech and Speaker Recognition, IEEE Press, (1979).

9) Dowty, D.R. et al. (Eds.), Introduction to Montague Semantics, Reidel, (1981).

10) Fant, G. and Tathan, M.a.a., Auditory Analysis and Perception of Speech, Academic Press, (1975).

11) Fillmore, C.J, The Case for Case, in Buch and Harms (eds.), Universals in Linguistic Theory, Holt, Reinhart and Winston, (1968), 1-210.

12) Flanagan, J.L., Speech Analysis, Synthesis and Perception, 2nd Edition, Springer-Verlag, (1972).

13) Hutchins, W.J., Progress in Documentation-Machine-Aided Translation, Journal of Documentation, 34, 2, June, (1978).

14) Kaplan, H.M., Anatomy and Physiology of Speech, McGrow-Hill, (1971).

15) Kay, M., The Proper Place of Men and Machines in Language Translation, CSL-80-11, Xerox PARC, (Oct. 1980).

16) Kidode, M., Local Parallel Picture Processing and Hardware Implementation, Ph.D. Dissertation, Kyoto Univ., May, (1979).

17) Kunii, T. et al., A Relational Database Schema for Describing Complex Pictures with Color and Texture, Proc. 2nd Int. Join Conf. on Pattern Recognition, (1974), 310-316.

18) Lea, W.A., Trends in Speech Recognition, Prentice-Hall, (1980).

19) Lindblom, B. and Ohman, S. eds., Frontiers of Speech Communication Research, Academic Press, (1979).

20) Markel, J.D. and Gray, A.H., Jr., Lenear Prediction of Speech, Springer-Verlag, (1976).

21) McKeown,Jr., D.M. and Kanade, T., Database Support for Automated Photo Interpretation, Proc. Image Understanding Workshop, April, (1981), 7-13.

22) McKeown, Jr., D.M. and Reddy, D.R., A Hierarchical Symbolic Representation for an Image Database, Proc. Workshop on Picture Data Description and Management, (1977), 40-44.

23) Miller, G.A., and Johnson-Laird, P.N., Language and Perception, Harvard Univ. Press, (1976).

24) Minifie, F.D., Hixon, T.J. and Williams, F., Normal Aspects of Speech, Hearing and Language, Prentice-Hall, (1973).

25) Minsky, M., Framework for Representing Knowledge, in Winston (ed.), The Psychology of Computer Vision, McGrow-Hill, (1975).

26) Nagao, M. (Ed.), Special Issues on Japanese Information Processing, Journal of Information Processing, Information Processing Society of Japan, 20, 10, (1979), (in Japanese).

27) Nagao, M. and Matsuyama, T., A Structural Analysis of Complex Aerial Photographs, Plenum Press, (1980).

28) Newell, A. et al., Speech-Understanding Systems: Final Report of a Study Group, North-Holland, (1973).

29) Newell, A., Production Systems: Models of Control Structures, in Chase (ed.), Visual Information Processing, Academic Press, (1973).

30) Niimi, Y., Speech Recognition, Kyoritsu-Shuppan, (1979), (in Japanese).

31) Norman, D.A. et al. (Eds.), Explorations in Cognition, Freeman and Company, (1975).

32) Nudd, G.R., Image Understanding Architectures, Proc. NCC, AFIPS Press, (1980), 377-390.

33) Onoe, M. et al. (Eds.), Real-Time Medical Image Processing, Plenum Press, (1980).

34) Pereira, F.C.N. and Warren, H.D., Definite Clause Grammars for Language Analysis, Artificial Intelligence, 13, (1980), 231-278.

35) Petrick, S.R., On Natural Language Based Computer Systems, IBM J. Res. Develop., July, (1976), 315-325.

36) Rabiner, L.R. and Schafer, R.W., Digital Processing of Speech Signals, Prentice-Hall, (1978).

37) Reddy, R., Speech Recognition, Academic Press, (1975).

38) Schank, R.C., Conceptual Information Processing, North- Holland/American Elsevier, (1975).

39) Shu, T. et al., Managements of Sketch Information for Chest X-Ray Image Database by Graphical Terminal, Tech. Paper of IECE Professional Group on Pattern Recognition and Learning, PRL, (1981), 80-105, (in Japanese).

40) Tanaka, H., A Semantic Processing System for Japanese Natural Language Understanding, Researches of the Electrotechnical Laboratory, No. 797, July (1979), (in Japanese).

41) Tamura, H., An Image Database Management for Pattern Information Processing Studies, in Chang, S.K. and Fu, K.S. (Eds.), Pictorial Information Systems, Springer-Verlag, (1980),198-227.

42) Waltz, D.L., Natural Language Interfaces, SIGART Newsletter, 61, Feb. (1977).

43) Winograd, T., Understanding Natural Language, Academic Press,(1971)

44) Woods, W.A., Syntax, Semantics, and Speech, in Reddy (Ed.), Speech Recognition, Academic Press, (1975).

45) Workshop on "High Level Language for Image Processing", Windsor, 1979.

46) Yoshida, M. et al., Image Processing System for Handwritten Design Chart, Journal of Information Processing, 22, 4, (1981), 300-306, (in Japanese).

# LOGIC PROGRAMMING AND A DEDICATED HIGH-PERFORMANCE PERSONAL COMPUTER


| | |
|---|---|
| Toshio YOKOI | Electrotechnical Laboratory |
| Shigeki GOTO | Electrical Communication Laboratories, NTT |
| Hiromu HAYASHI | FUJITSU Laboratory. |
| Susumu KUNIFUJI | IIAS-SIS, FUJITSU Ltd. |
| Toshiaki KUROKAWA | TOSHIBA Co. |
| Fumio MOTOYOSHI | Electrotechnical Laboratory |
| Hideyuki NAKASHIMA | University of Tokyo |
| Katsumi NITTA | Electrotechnical Laboratory |
| Taisuke SATO | Electrotechnical Laboratory |
| Tomokatsu SHIRAISHI | SHARP Co. |
| Kenichi UEDA | MATSUSHITA Research Institute |
| Mamoru UMEMURA | NIPPON ELECTRIC Co. |
| Shinji UMEYAMA | Electrotechnical Laboratory |



The role of our group is the preliminary research on FG-Kernel Language (version-0) and a high performance personal computer which are springboards for the research and development of Fifth Generation Computer System, adopting PROLOG as the starting point. Until March 1982, our work on the way will reach enough results. Necessary items have already been surveyed. In this paper, the rough image of FG-Kernel Language(version-0) and a dedicated computer for the language which we suppose is introduced.


## 1 Introduction

The role of our group is the preliminary research on FG-Kernel Language (version-0) and a high performance personal computer which are springboards for the research and development of Fifth Generation Computer System, adopting PROLOG as the starting point. Until March 1982, our work on the way will reach enough results. Necessary items have already been surveyed. In this paper, the rough image of FG-Kernel Language(version-0) and a dedicated computer for the language which we suppose is introduced. But changes and modification may well be hereafter since on their way.

There are many attributes that prescribe a computer system, however, the most important one is what language we accept as the main programming language. For application areas, the basic structure of software system and the frame of computer architecture are all determined by this language. So in this project, we call this main programming language FG-Kernel Language and regard as the most important research theme. The research and development of this language must be carefully pursued on the basis of systematic studies on various aspects such as artificial intelligence(problem solving and knowledge representation), software engineering, examination of various programming language proposed etc..

Our group is setting forward this research by the following approach. At first, as the basis of FG-Kernel Language, we adopt a logic programming language, PROLOG(Edinburgh version[1]). Then we improve and extend it and make it have necessary functions for a general purpose programming language and system. Lisp is the most valuable language and system as a textbook and a standard language in the knowledge information processing research and PROLOG will be extended so as to include valuable features of Lisp.

Our another work is to propose the machine specification of high performance personal computer for the proposed language. As we see in such examples as Lisp machines(Symbolics 3600, MIT machine[2] etc.), DORADO[3], SPICE etc., a high performance personal computer and network system are going to be indispensable tools for the research and development of information processing systems. Furthermore they are coming into the limelight, since they are able to do much for future office automation and home computer.

In this project also, a dedicated high performance personal computer and local network system will be developed and be supplied as a standard tool for research and development at the very early stage.

Language, improvement and extension of PROLOG, in Section 2 and the machine dedicated to it in Section 3, their overviews are respectively introduced as the interim report as under.

## 2. Language

To begin with, the reasons why a logic programming language(PROLOG) is chosen as the kernel of FG-Kernel Language are described.

(1) It is appropriate for programming of knowledge information processing system. List processing, database mechanism similar to relational database, pattern matching(unification) which clearly represents the composition and decomposition of data structure and database search, non-deterministic processing etc. are indispensable processing functions in programming of knowledge information processing systems. PROLOG has all basic parts of these functions, and moreover, is able to be extended to get more high performance functions.

(2) It gives new paradigms of programming. Non-procedural representation scheme, high modularity, a happy blending of computation and database search etc. are new programming paradigms. These paradigms, what is better still, make it much easier to deal with program and programming as formal objects and give great possibilities to realize a program verifier and an automatic programming system.

(3) It succeeds to the results of efforts made by current programming languages. Much has been discussed about the relationship between logic programming language and functional language, and it has become generally appreciated that these languages will play the leading part in future programming. To be concrete, also as to Lisp, the functional language that is most widely put into practical use at present, it is possible to extend PROLOG efficiently to include useful functions of Lisp as a subset. PROLOG has a close relationship to programming languages for artificial intelligence which could not give us too much success. PROLOG can put search mechanism with backtracking control into practical use by using logical formula(Horn clause) as language constructs and improving implementation techniques, though Micro-planner could not. However, the current version of PROLOG does not have every function which programming languages for AI like Micro-planner, Conniver and so on present to us. How to extend PROLOG to include these functions is remained unsolved.

(4) It introduces new computer architectures. FG-Kernel Language will be first implemented on an conventional large scale computer and then on a high performance personal computer which will be mentioned below. According to the research plan, the language will be improved and extended step by step, based on actual experience and various research results. And finally, the language will become a machine language for the target machine of this project. Consequently, the language(version-0) must be such a language as fundamentally has all of the appropriate mechanisms for data flow machine and data base machine architectures supposed as basic architectures of the target machine. PROLOG has a great possibility in this point, too.

From the above reasons, PROLOG is chosen as the kernel of Kernel Language.

Next, main features of improvement and extension of PROLOG now under study are enumerated. As the version-0, we give priority to arrangement of all primitive and necessary functions over invention of high level ones.

(a) Abstract data type(encapsulation)

The usefulness of abstract data type has been well known and recent most new programming languages have adopted it as the basic function. But the current version of PROLOG doesn't have this construct explicitly. So, we have to introduce it in natural way. To introduce every function of abstract data type and to make clear it's function for program specification and program verification are remained as a long term research theme. In this study, we pay attention to its ability on encapsulation and localization of side-effect.

It is to be desired that this extension is made by natural enlargement of functions which PROLOG has now. PROLOG has one internal database. In this database, all clauses(unit and non-unit) are stored. There are predicates which assert and retract these clauses, and the way to cause side-effect is to alter the contents of the database with these predicates. This situation can be interpreted there is only one abstract data type called internal database. Consequently, to make it possible to define a number of abstract data types is to make it possible to create a number of databases, which we call Micro databases. A super predicate represents Micro database is shown in Fig.1 with some syntactic sugar. Clauses in Micro database represent the internal state and procedures manipulating the state of Micro database. Exports-predicate defines predicate names(Horn clause names) permitted to be accessed from the outside. But the creation of Micro database is not mentioned in this paper.

```
<name >[ <clause>

         <clause>
            .
            .
            .
         <clause >

          exports ( ......) ]
```

Fig. 1. Micro Database.

Various advantages are obtained by the introduction of Micro database. For instance,

- Side-effect is localized.

- Structures are introduced into program. If nest structure is permitted among databases, more complicated program structures can be represented.

- Separate compilation becomes available. Clauses which are not exported, are never accessed from the outside. So, it is possible to compile calling sequences(unification) to these clauses.

Various techniques are available for the implementation of Micro database. Depending on the number of clause, the frequency of change etc., an appropriate internal mechanism for Micro database has to be adopted.

(b) Refined higher-order extensions.

PROLOG is a simple and powerful language based on first-order logic. For practical use, however, various higher-order extensions have to be introduced. About what is essential, it is still open to discussion. For example, it is said that higher-order extensions like lambda expression and predicate variable are not much essential and first-order logic has enough ability[4]. As the first step, we try to introduce the most primitive one in this study. In Lisp, for example, the most primitive mechanism for higher-order is that program and data have the same structure, and that quote and eval functions are provided, which control whether some data structures are regarded as program or data. This mechanism is introduced to PROLOG too as a primitive one. Basic data structure of Lisp is list(s-expression). To PROLOG, tuple is regarded as a basic one. Every term, predicate and Horn clause(Fig.2(a)) are able to be internally represented as tuples(Fig.2(b)). At the head of each tuple, the tuple name is placed and the attribute of this name indicates what the tuple represents. And then, for composition and decomposition of tuple, unification is extended and some predicates are introduced.

$p\ (X\ \ f\ (Y\ \ Z))\ :-\ q_1\ (\ .\ .\ .\ .\ )\ ,\ \ .\ .\ .\ ,\ q_n\ (\ .\ .\ .\ .\ )$

(a) Horn clause

$(:-\ |\ (p\ |\ \ X\ \ (f\ |\ \ Y\ \ Z))\ \ (q_1\ \ .\ .\ .\ .)\ .\ .\ .\ (q_n\ |\ .\ .\ .\ .\ .\ .))$

(b) tuple (with name)

Fig. 2. Basic data structure.

The most fundamental construct for control structure of PROLOG is cut-operation. This operation is very powerful, but its effect is very hard to be understood. So, it is compared to goto statement in a conventional language. We have to introduce more structured constructs for control and banish cut-operation as possible as we can, as well as we did on goto statement. For example, the introduction of selection mode for clauses is possible.

(c) Enough preparation of programming tools.

Evaluation with backtracking makes debugging very difficult. This means it is necessary to prepare more powerful tools. Debugger, which traps evaluation by error or break, keeps environment as it is and responds to various users' commands, Tracer, which traces the history of evaluation of specified predicates and variables and displays it in pretty format, Stepper, which evaluates program steps one by one and displays various states by the minute, Editor, which edits clauses with pattern matching etc.. These tools are combined into one total programming system in order to be invoked at any place.

(d) High level data structure.

It is pointed out that such data structures as set and bag which collect elements to satisfy a certain condition(represented by predicates), are important. For this, the most primitive higher-order predicate is provided to PROLOG as well. We introduce these and tuple mentioned in (c) as basic data structures. And we also introduce predicates and extend unification for manipulation of these structures. Moreover, we introduce some higher-order predicates which iteratively perform some operations on each element of these data structures.

(e) Basic functions for Japanese language processing.

All programming languages and systems hitherto have been based on English only. But now, such systems as we can write programs and documents using respective native languages are required. Japanese becomes available for predicate and variable name, adding Kana(the Japanese alphabet) and Chinese characters for daily use to the basic character set. Japanese is also used for messages displayed and comments inserted into a program. Moreover, the alphabets of other foreign languages(French, Germany etc.) are added to the character set too and this system becomes appropriate for the research and development of natural language processing and machine translation.

(f) Useful functions for system description.

Interpreter, compiler, file system, tools for debugging etc., a lot of system programs have to be developed. Kernel part of them will be implemented by micro program. The rest are desirable to be implemented by PROLOG itself. For this purpose, we have to introduce efficient system description functions into it. For example, they are;

- Abstract data type with good efficiency. Compiler is able to transform Micro database introduced in (a) into very efficient object codes under a certain restriction. For example, it transforms a clause in Micro database into such codes as fetch and store directly terms in a predicate which represents its internal state.

- Refined system data structures. Data structures which represent the internal state of system are refined. Basic predicates which access and manipulate them and basic protection mechanism are both provided.

- Constructs for parallel processing. Necessary parallel processing constructs for programs controlling external devices are introduced as simple as possible.

Compared with an ordinary system description language, PROLOG has far high level functions, therefore, it is apt to be thought that it is not appropriate for system description. But, under natural restrictions and degeneration of functions, it is able to guarantee the same efficiency as an ordinary system description language does. For example, they are:

-There is no non-deterministic selection.

-Unification is restricted. A term is a variable or a constant. Furthermore, it is restricted to the parameter binding of an ordinary functional language.

Some of these restrictions are explicitly specified in a program and others are automatically detected by compiler.

(g) The others.

Besides the above, we study the following functions.

-Large scale database, connection with external database (relational database).

-Other search modes different from top-down and depth-first search.

-Improvement of backtracking search mechanism.

## 3. Machine

For Lisp, high performance Lisp machine for a single user has been developed and been put into practical use. In the same way, for the extended PROLOG mentioned above, high performance PROLOG machine has to be developed. The block diagram is shown in Fig.3. and rough specifications of each module are as follows.

- micro-coded CPU(writable control storage and two or three million PROLOG instruction/second processing unit).

- 1 Mega words of 36 bits each of main memory.

- fast-access Winchester type disk in the 200-400 megabyte range.

- 2000 x 2000 BW(or RGB) bit-mapped TV-display.

etc.

It is, on the whole, designed with a compact size and beautiful appearance. Basic architecture is language oriented machine architecture like high level machine language, tag/descriptor and so on. It must be, however, flexible for the future modification.

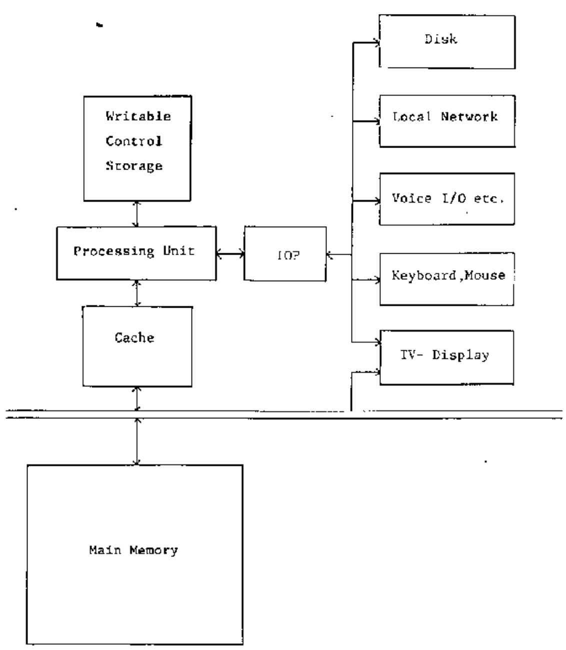



Fig. 3. PROLOG machine block diagram.

## 4. Conclusion

To design language and machine, it is necessary to accumulate experiences such as writing and studying programs on many examples. Available PROLOG systems in Japan for this purpose are listed in Table.1. PROLOG(Edinburgh)[1][5] is the full-scale system and efficient processing is possible with it. PROLOG(Marseille)[6] is the system developed first of all in the world. It is used for the study of internal mechanism of interpreter. PROLOG(IBM)[7] is a compact and standard system. PROLOG/KR[8] is the pilot system in order to research a new programming language through the integration of PROLOG and Lisp. The next new trials are in it, that is to say, list structure for internal representation of every object, plentiful second-order predicates and constructs for control structure, various kinds of tools for debugging etc.. DURAL[9], in spite of the small scale system, has such new mechanisms as relative Horn clause, modal operator etc..

| name | authors | implementation |
|---|---|---|
| PROLOG (Edinburgh Univ.) | F.M. Pereira<br>F.C.N. Pereira<br>D.H.D. Warren<br>L. Byrd | MACRO (etc.)<br>Dec Tops-10<br>-20 |
| PROLOG (Marseille Univ.) | G. Battani<br>H. Meloni | FORTRAN |
| PROLOG (IBM) | J.F. Sowa<br>G. Roberts | VM/CMS |
| PROLOG/KR (Tokyo Univ.) | H. Nakashima | UTILISP |
| DURAL (NTT) | S. Goto<br>H. Okuno | MACLISP<br>Dec Tops-20 |

Table. 1. Logic Programming Systems

REFERENCES


[1] Pereira, L.M., Pereira, F. C. N. and Warren, D. H. D.,"User's Guide to DEC system-10 PROLOG", Dept. of AI, Univ. of Edinburgh (Sept. 1978)

[2] Weinreb, D. and Moon, D.,"Lisp Machine Manual", Second Preliminary Version, MIT (Jan. 1979)

[3] Clark, D. W.,"The Dorado: A High-Performance Personal Computer, Three Papers", CSL-81-1, Xerox PARC (Jan. 1981)

[4] Warren, D. H. D.,"Higher-order Extensions to PROLOG - Are They Needed?", D.A.I. Research Paper No.154, Univ. of Edinburgh (Apr. 1981)

[5] Byrd, L., Pereira, F. C. N. and Warren, D. H. D.,"A Guide to Version-3 of DEC-10 PROLOG", Univ. of Edinburgh (June 1980)

[6] Colmerauer, A., Kanoui, H. and van Caneghen, M.,"Etude et realisation d'un systeme Prolog", Groupe d'Intelligence Artificielle, U. E. R. de Luminy, Universite d'Aix-Marseille II (1979)

[7] Sowa, J. F.,"A PROLOG to PROLOG", IBM System Research Institute (Jan. 1981)

[8] Nakashima, H.,"PROLOG/KR User's Manual", Information Engineering Course, Univ. of Tokyo (Aug. 1981)

[9] Goto, S.,"DURAL : An Extended Prolog Language", Lecture Note Series No.396, Research Institute for Mathematical Sciences, Kyoto Univ. (Sept. 1980)

# IV ARCHITECTURE RESEARCH PLAN

# New Architectures for Inference Mechanisms


Shunichi UCHIDA  Electrotechnical Laboratory

Hidehiko TANAKA  University of Tokyo

Mario TOKORO  Keio University

Kinji TAKEI  Toshiba Corp.

Masakatsu SUGIMOTO  Fujitsu Ltd.

Hiroshi YASUHARA  Oki Electric Industry Co.,Ltd.



This paper describes a research plan of the inference machine which is one of the subprojects involved in the research and development of the fifth generation computer systems. According to the research plan of whole project, there are two other subprojects concerning to the architectures of the fifth generation computer systems. They are called as the knowledge base machine and the intelligent interface machine. In the final stage of the project, the research results of the inference machine, mixed with the research results of two other machines, will constitute a prototype of the fifth generation computer.

Research goal of the inference machine is to develop new architectures to support inference mechanisms and build experimental machines which are powerful enough to fulfill the requirements of many applications in knowledge information processing. The inference mechanisms are given as a new programming language and its computational model, which is called the fifth generation computer kernel language (FGKL). It is based on procedures of mechanical theorem proving in first order predicate calculus which are recently given as logic programming languages like PROLOG. The inference machine may thus be looked upon as a high level language machine. To provide the machine with sufficient power, highly parallel architectures combined with VLSI technology are indispensable. And it is expected that dataflow machines will become the basis of its architecture because their model seems to be close to the model of logic programming.

The plan spans 10 years and is diveded into three stages. They are the initial stage, the intermediate stage and the final stage. In the initial stage, basic research of dataflow machines is planned to be done. And sequential inference machines will be developed as a tool for software development. Architectures for supporting object oriented programming are also studied to implement related functions involved in FGKL. In the intermediate stage, an experimental parallel inference machine is planned to be developed,which is expected to contain about 100 processing elements. In the final stage, all the research results will be combined to constitute a prototype of the fifth generation computer which is expected to contain about 1000 processing elements.


## 1.Introduction

Recent innovation in computer science has motivated us to start the research and development of new computer systems which will be generally used in 1990's. After over two years of investigation, a plan which is called the research and development of the fifth generation computer systems (FGCS) has been developed.

In this project, future computer systems are characterized by their intelligence. It is expected that they can help human workers much better than existing computer systems as not only diligent but also intelligent assistants who can make inference using knowledge bases which contain a variety of knowledge corresponding to such applications as office automation, computer aided design and consultation systems. The fifth generation computer system is thus called as a knowledge information processing system (KIPS).

Accordingly, KIPS is considered to have three major mechanisms. They are inference mechanisms, knowledge base mechanisms and intelligent interface mechanisms.[23][25] In these three, inference mechanisms and knowledge base mechanisms are closely related each other ,however, intelligent interface mechanisms are different from the other two and will contain various functions depending on the medium of man machine communication.

These three mechanisms are implemented both in software and hardware ,and the hardware systems will mainly consist of custom VLSI chips designed by advanced VLSI CAD systems. Each of software and hardware mechanisms and VLSI CAD systems roughly corresponds to the subprojects of this research.

This paper describes the subproject to develop new architectures to implement inference mechanisms. Thus, it is simply called as an inference machine subproject. As the interface between software and hardware is generally given as language specifications in this project, functions of the inference machines are indicated by the language and its computational model which are developed in a problem solving and inference mechanism subproject.

This language is called as the fifth generation computer kernel language(FGKL), which is based on procedures of mechanical theorem proving in the first order predicate calculus, an example of which is PROLOG.[1]-[4] Functions of FGKL are planned to be extended two times and thus three versions,Version 0 to 2, will be made corresponding to each stage of the project. Each version will characterize the research activity of each stage. As FGKL becomes an object language of the inference machine, functions of the inference machine are also extended according to the revision of FGKL.

For the performance of the inference machine, the analysis of the basic applications such as machine translation systems and question answering systems suggests that the machine has to be 100 - 1000 times better than existing computers in power so as to make these applications practical. As the inference machine becomes one of the major components of the fifth generation computer, it should have sufficient processing power and capacity for many practical applications in knowledge information processing.

This requirement implies the necessity of highly parallel architectures and fast and small logic elements. The base of the architectures is expected to be given by the dataflow machine. And it is expected that some software or hardware mechanisms which relate the dataflow model to the logic programming model will be developed during this project.

For the implementation of such complex architectures, VLSI will be indispensable as well as good VLSI CAD systems. It is expected that VLSI will not only change the methodology of computer design but also change the criteria of the cost performance.[6][7] Without such changes, highly parallel architectures like dataflow machines will be impractical.

Accordingly, the research goal of the inference machine is to develop a very high level language machine based on a highly parallel architecture which is mainly implemented by VLSI technology.

This fact means the research goal is a collection of major topics in recent computer science. Thus, it is expected that the project is not only very ambitious but also very productive.

The project spans 10 years and is divided into three stages. They are the initial stage( 3 years ), the intermediate stage( 4 years ) and the final stage( 3 years). In the initial stage, basic mechanisms will be studied and sequential FGKL machines will also be developed as a research tool. In the intermediate stage, an experimental parallel inference machine, containing about 100 processing elements, will be developed.

In the final stage, a prototype of the fifth generation computer will be developed reorganizing all the research results.

In the later sections, background of the research, research goals and activities in the three stages will be described.

## 2. Background of the research and development

Recent research on computer architectures has apparently made much effort to harmonize the improvement of processing power with good programmability. On one hand, many multiprocessor systems have been developed, however, they are not always successful because they lacked high level programming languages.

On the other hand, many high level language machines have developed, however they failed in the improvement of the processing power because adherence to conventional programming languages such as FORTRAN and COBOL prevented them from employing highly parallel architectures.

However the situations is rapidly changing. The characteristics of conventional architectures as well as conventional programming languages based on sequential program execution have deeply been understood through the active research on software engineering and artificial intelligence as well as the research on new architectures in the past decade. And new ideas and concepts are being proposed.[10][11]

For example, relation between some functional languages and parallel architectures is formalized as dataflow machines. Research on relational database machines is trying to combine relational algebra with parallel architectures. Relation between concept of data abstraction and object oriented programming is indicated as well as their architectural mechanisms. And new high level languages such as logic programming languages are presented with its computational models. And many other ideas are proposed. These new ideas and theories are suggesting the departure from conventional architectures represented by existing commercial computers.

Furthermore, the progress of VLSI technology is going to change the methodology of computer design and introduce new criteria for cost performance of hardware systems. Intel's new microprocessor, iAPX 432 has indicated that VLSI can inexpensively implement complex mechanisms to support abstract data type.

This situation will make the research on highly parallel processors like dataflow machines and data base machines very attractive and also practical.

In planning the project, consideration of the situation described above has made us to adopt logic programming languages as the object language of the target computer system and the dataflow machine as the basis of its architecture.

## 3. The final goal and interim goals of the research

### 3.1 The final goal of the project

The project is planned to attain the final goal in 10 years. As this research includes many unknown factors, however, a conceptual diagram of the prototype system to be developed in the final stage is shown in Fig.-1. As shown in this figure, major three components which are developed in three subprojects are expected to be integrated in one machine which we call the prototype of the fifth generation computer system.

The final system is defined as the most powerful machine that can be made in the final stage. However, many types of machines will probably be made actually and some of them may have special augmented functions such as a large capacity of knowledge base.

This final system should provide us with processing speed and memory capacity enough to build practical machine translation systems and question answering systems. Usually the speed and capacity required will depend on the applications, however ,we roughly determined the targets for them. The expected speed of inference is 100 M - 1 GLIPS.( Here, LIPS stands for Logical Inference Per Second and one LIPS means that the inference machine can perform a step of resolution in one second in average, including the unification.) The expected memory capacity including secondary memory is 100-1000GB. Furthermore, such a question answering system that includes more than 5000 vocabularies and more than 10000 rules will be practical.

Principal functions of the inference machine are suggested by the specification of FGKL. FGKL is planned to be extended in stepwise according to the progress of the research, however, the final version will include such functions as follows:

1) modularization mechanisms for object oriented programming;
   - structure memory for context mechanism, - dynamic type checking and capability mechanism,

2) capability of handling Chinese characters and kana letters;
   - programming using Japanese character set,

3) description of parallel processing;
   - augmented model of logic programming,
   - meta level control,

4) description of concurrent processing;
   - control of multiprogramming.

As these functions usually require some hardware supports for efficient implementation, they include common research items of both software and hardware.

### 3.2 Relation between the subprojects in the project

To achieve the final goal described in 3.1, several subprojects should be put into practice in parallel and synchronized, because all the subprojects are mutually related.

The research of the inference machine is closely related to such subprojects as follows:

A) development of the problem solving and inference mechanisms,

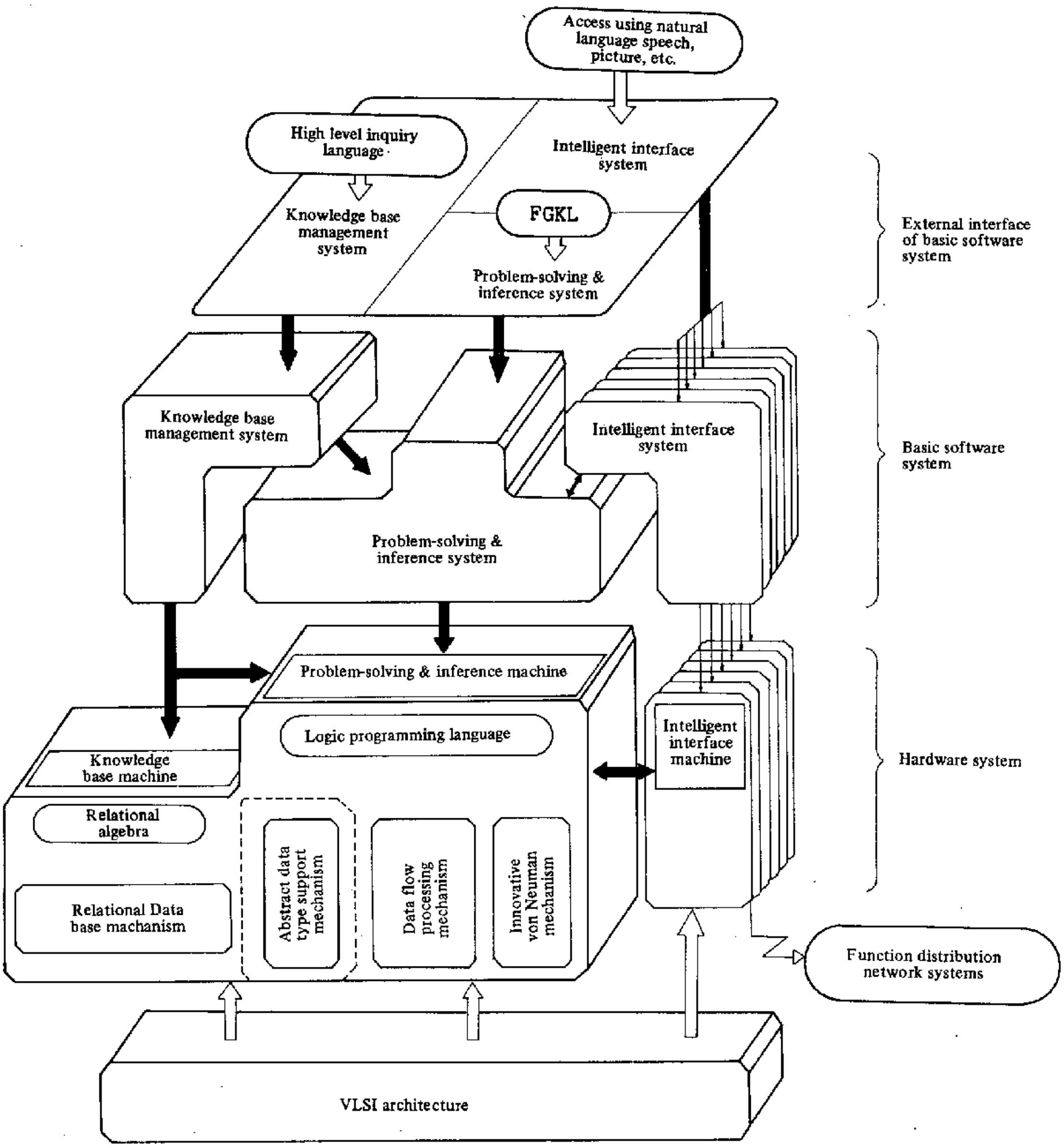



Fig. 1 A conceptual diagram of the Fifth Generation Computer Systems

B) development of the knowledge base machine,

C) development of VLSI technology and VLSI CAD systems.

As one of the purposes of subproject A is to determine the computational model and the language specification of FGKL, it directly relates to the research of the inference machine. This also develops algorithms of the resolution and the unification suitable for parallel architectures.

The subproject B includes the development of an experimental relational data base machine as its first step towards the knowledge base machine. Then, the machine is extended to include such functions as involved in the production systems to support knowledge base systems.

However, such functions are expected to be similar to the ones included in the inference machine. In the final stage, the knowledge base machine and the inference machine are expected to be reorganized into one unified machine. Consequently, the research of the knowledge base machine can be regarded as another approach to the final goal, although these two are not developed together till the end of the intermediate stage.

The subproject C has its own final goal as the intelligent VLSI CAD system implemented on the fifth generation computer system. Furthermore, it plays an important role as a supporting tool of the development of all the hardware systems. Without any good VLSI CAD system, development of even an experimental parallel inference machine in the intermediate stage would be impossible.[7][26]

In addition to the subprojects discussed above, there is another subproject in which special purpose hardware systems are to be developed. They will be connected to the inference machine for supporting high resolution graphics, speech and picture recognition and real time control of various equipments.

As the examples of basic applications,a machine translation system and a question answering system are to be developed. These systems are planned to be used as a kind of test programs to evaluate performance and effectiveness of the experimental machines. The details of these systems are described in another paper. [23]

## 3.3 The research items in the development of the inference machine.

As intermediate steps, several interim goals are planned as follows:

A) development of sequential inference machines which support FGKL Version 0 and are used as the tools for software and hardware research,

B) development of the hardware mechanisms to support data abstraction and object oriented programming,

C) development of a small scale dataflow machine or a dataflow machine simulator to collect the basic data for designing the architecture of the parallel inference machine,

D) development of the parallel inference machine which support FGKL Version 1.

These interim goals are selected so as to determine the research items which should be done in the initial stage. Accordingly, new unknown interim goals will probably emerge at the end of the initial stage. The details of the interim goals are explained below.

### 3.3.1 The sequential inference machines

Several sequential inference machines are developed in the initial stage,which support FGKL ( Version 0 ). FGKL of Version 0 will include the functions of existing PROLOG plus the capability of modularization as well as the functions to handle Japanese characters.

The first purpose of these machines is to support the software research as a powerful and efficient tool for the development of various programs.

The second one is to develop the architectures to support logic programming languages using the conventional hardware technology such as microprogramming, high speed microprocessors, logic in memory. Through the design of these machines, the architects will be familiar with the mechanisms of inference.

The third one is to use these machines for various architectural experiments such as development of firmware and hardware mechanisms to support data abstraction for object oriented programming, experiments to develop interface mechanisms between the inference machine and the relational data base machine. In these experiments, new firmware and hardware modules will be added to these machines.

The details of machine specification have not been determined yet, however, it will include such components as follows:

1) high speed microprocessors controlled by microprograms,

2) tags and descriptors for memory structures,

3) special firmware and hardware modules to support the resolution and the unification such as stack mechanisms and hashing mechanisms,

3) the virtual memory system,

4) a high resolution display, a pointing device, input/output devices for Japanese characters and other peripheral devices, and

5) local network interface.

In the initial stage, several machines

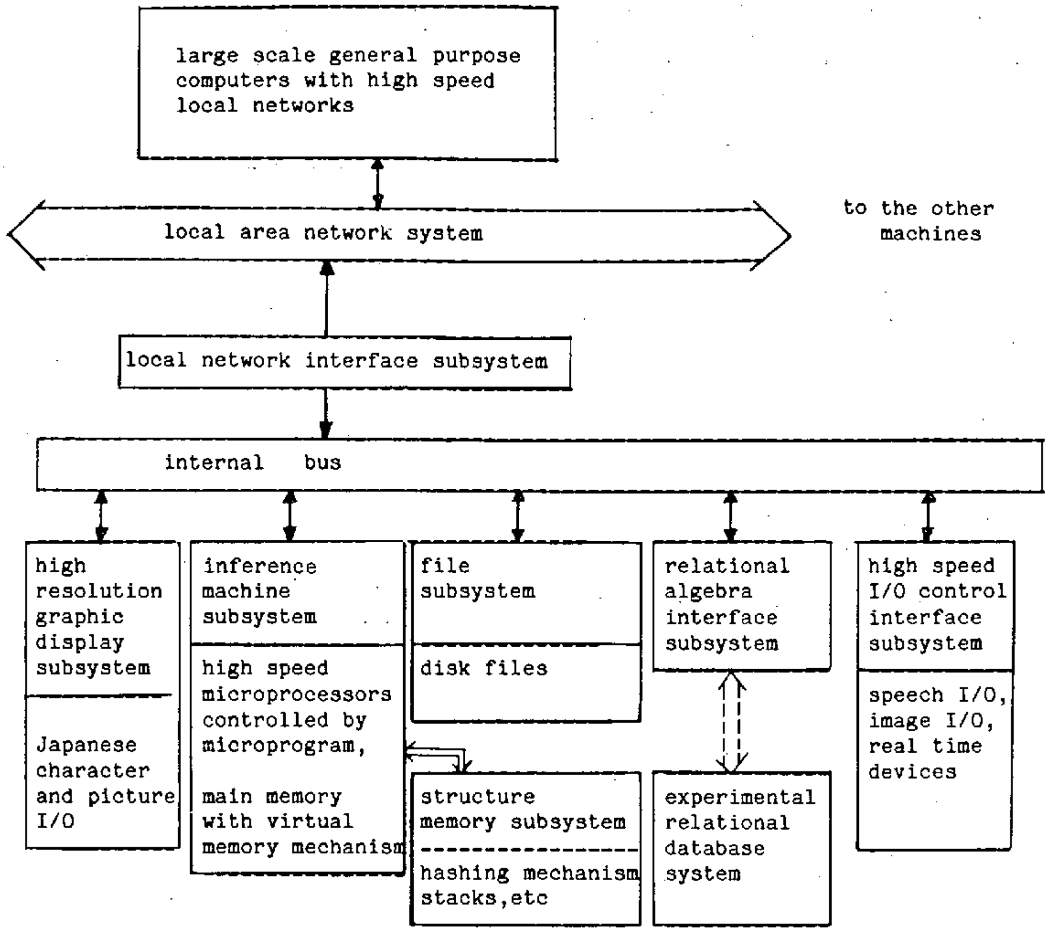



Fig.-2 Configuration of the sequential inference machine

of this type are planned to be developed. After some modifications or extensions are made in the initial stage, custom LSI or VLSI chips will be designed to make the machines smarter.

And in the intermediate stage, the machines on the new model will be given to the researchers to provide them with a good programming environment for their experiments. These machines may be looked upon as a prototype of future personal computer systems.

Accordingly,it is very important to provide the machine with good man machine interfaces which are realized in such machines as the MIT's CDR and the Xerox's ALTO and DORADO. [7][8][9]

In the intermediate stage, some of these machines will probably be connected to a experimental relational data base machine developed in the knowledge base machine subproject so as to support large scale software experiments. Expected configuration of the machine is shown in Fig.2.

The research items involved in this interim goal are summarized as follows:

-efficient algorithms for the resolution and the unification as well as their firmware and hardware mechanisms,

-the structure memory to support the heap mechanisms and the fast garbage collection,

-firmware and hardware mechanisms to support data abstraction,

-the virtual memory system and interface mechanisms to connect to the relational data base machine,

- support of Japanese character input/output for the programming using Japanese character set,

- system description and high level microprogramming languages,

- local networking,and

- development of custom LSI and VLSI chips for the smart personal machine model.

Software systems for these machines are developed in the inference mechanism subproject described in another paper.[22][24]

### 3.3.2 Hardware mechanisms to support data abstraction and object oriented programming

Recent software research has proposed many new concepts and ideas to make reliable and well structured software. Abstract data type, object oriented programming and capability addressing are examples of these concepts.

From the architectural point of view, these three concepts are closely related each other and expected to have considerable effect on future computer architectures.

First step of the architectural research on these items has been done in such projects as the ALTO machine of Xerox in connection with its programming language Small Talk, the new microprocessor iAPX432 of Intel in connection with the new programming language Ada and the proposal of CLU machine of MIT. [12][13]

In this project, it is considered that FGKL will include such new concepts in its language specification and thus, expect some hardware supports.

Accordingly, this research intends to develop firmware and hardware mechanisms to support such concepts for the inference machine. This interim goal includes such research items as follows:

- architectures to support abstract data type, object oriented programming and capability addressing,

- experiments for efficient implementation of the architectures in the sequential inference machine, especially related to its structure memory,

- VLSI implementation of the architectures,

- study on distributed algorithms for process synchronization and resource allocation in connection with parallel execution of FGKL and also with dataflow machines.

In the initial stage, the experiments will be made in software and firmware. The research results will be introduced in the sequential inference machine. In the intermediate stage, they will be used for designing the structure memory of the parallel inference machine.

### 3.3.3 The dataflow machine as the basic architecture of parallel inference machine

One of the most important characteristics of dataflow machines is that they can combine highly parallel architectures and high level programming languages. They can fill the so-called semantic gap, which has made most conventional parallel processors not so successful.

In inference mechanisms, much parallelism may exist, however,the structure of computations is anticipated far irregular than the one included in such problems as matrix calculations.

Dataflow machines are basically applicable to such irregular computations and is expected to be useful to improve the power of the inference machine.

Although there exists the gap between dataflow models and logic programming models, it seems to be possible to find out some mechanisms to fill this gap in the near future. And the fact that dataflow machines are generally suited for VLSI implementation makes them more attractive in this project.

However, there is a variety of discussions about their performance. Thus, the first step of the research will be the estimation of the performance for various machine structures through the simulations and experiments using hardware simulators or small scale dataflow machines.

The purpose of this research is to develop a new parallel architecture to support parallel logic programming language( FGKL, Version 1).

In the initial stage, the research is mainly done to determine the basic structure of the machine as well as the specification of main components such as the processing element, the structure memory and the network module. Then, the architecture will be developed to support FGKL. Thus, this research may be regarded as a bottom up approach to the inference machine.

The research on dataflow machines is being done in many places such as MIT in U.S.A.,CERT( University of Toulouse) in France and Electrical Communication Laboratory of NTT in Japan and a variety of machines have ever been presented.[15]-[19] However, a few machines are intended to be used for symbolic data processing, with which inference mechanisms are mainly organized.

Accordingly, the simulations of the machines for symbolic data processing are especially important to determine basic machine structures.

This research includes such items as follows;

1) development of an experimental dataflow language and a simple programming system,

2) development of software simulators and experimental operating systems;

- scheduling and resource allocation systems,

- system level simulators,

- machine component simulators,

- performance measurement systems, etc,

3) development of a hardware simulator;

- 16 microprogram controlled processing elements, each of which contains functional units and activity memory, and

- a structure memory of 8 -- 16 banks with parallel garbage collector,

- a packet switching network,

4) Study on basic mechanisms;

- basic structure of the machine,

- mapping methods between logical and physical machine,

- processing of structured data and management of structured memory,

- message passing and switching networks,

- VLSI implementation of basic mechanisms,

5) Study on high level mechanism;

- introduction of object oriented architectures,

- mechanisms to support FGKL in dataflow architectures,

- study on parallel algorithms,

- VLSI implementation of high level mechanisms.

The research items involved in 1,2,3,4 will be started from the initial stage, however, the other ones will be done in the intermediate stage. The research results of dataflow machines are planned to be mixed with the other research results of this subproject in the initial stage and taken over to the development of the experimental parallel inference machine in the intermediate stage, which is expected to contain more than 100 processing elements. Furthere details of the research on dataflow machine is described in another paper.[27]

### 3.3.4 The parallel inference machine

This machine is intended to support the parallel execution of a logic programming language ( FGKL, Version 1). In the intermediate stage, an experimental parallel inference machine is planned to be developed. It will include more than 100 processing elements and its basic architecture will mainly be taken from the dataflow machines studied in the initial stage.

The research results in the intermediate stage will be combined with the research results of the other subgoals such as the knowledge base machine. All the research results in the this stage will become the basis of the prototype of fifth generation computer systems.

In the initial stage, the research of this subgoal is mainly theoretical study although some software developments are probably included.

First step of this research is to consider the relation between logic programming models and dataflow models so as to find out the mechanisms to fill the gap between them.

Thus, this research may be regarded as top down approach. Many research items are common to the development of FGKL and its programming system.

Algorithms for parallel execution of logic programming languages have not been studied well enough to present the hardware mechanisms.

One approach is probably to look for the new search strategy of the and-or tree that is suited for parallel processing. There may be the other approaches for this problem.

Consequently, several mechanisms are expected to be developed. Some mechanisms may be implemented as a kind of compilers which translates the programs of FGKL into the dataflow graphs. The other mechanism may be implemented in hardware and included in each processing element of the machine. In this case,each processing element may become a small logic programming machine.

Another important research item is the study on parallel algorithms for basic application programs. Without any good parallel algorithms, the machine would not operate efficiently.

Concerning to this item, analysis of programs is planned to investigate how much parallelism is involved in typical application systems such as a natural language understanding system and a VLSI CAD system. The research results will be indispensable for the research of dataflow machines.

This research includes such research items as follows:

1) study on high level mechanisms;

-computational models for parallel execution of FGKL,

-proof procedures and their algorithms including meta level control,

-relation to functional models (dataflow machine models),

-relation to relational models (relational data base machine models),

-hardware mechanisms to support basic functions of inference mechanisms such as the unification and the resolution,

-parallel algorithms for application programs,

-investigation of parallelism involved in application programs,

2) development of an experimental parallel inference machine;

-64-256 processing elements, each of which contains 8-16 functional units,
-activity memory of 64-256 banks with virtual memory,

-a structured memory of 16-128 MW,

-VLSI implementation.

## 4. The research and development plan

In this subproject, there are several research items as described in 3.3. These research results are assembled into three types of inference machines.

First type is the sequential inference machines which are developed in the initial stage.

Second one is the experimental parallel inference machine which is developed in the intermediate stage.

Third one is the prototype of the fifth generation computer developed in the final stage.

The precise research plan of each stage is determined depending on the progress of research, however,general plan is summarized as follows.

### 4.1 The initial stage

The research done in this stage are characterized as the development of basic component technology and research tools.

First important research item in this stage is the sequential inference machines explained in section 3.3.1,which support FGKL Version 0. These machines are intended to provide the researchers with a good programming environment or a work bench for both software and hardware research. Some of these will include special hardware depending on their applications.

The research on abstract data type and object oriented architectures in this stage is done at first to develop the mechanisms to be added to the sequential inference machines for efficient support of modularization of FGKL Version 0. Theoretical research will also be done for such purpose as to combine object oriented architectures with dataflow machines. Experiments of this research will be mainly made by firmware and software on the sequential inference machines or large general purpose computer systems.

The research on data flow machines in this stage is concentrated on the decision of basic machine structures and functions of each machine component. This will be made through the software and hardware simulations.

The software simulators include simple

programming systems to write various programs in typical applications to collect the information about the computational structures and the amount of hardware resources required.

The hardware simulator which contains about 16 processing elements will be developed to get quantitative information about the behavior of programs such as required processing power and memory capacity.

The research on the parallel inference machine in this stage will mostly be theoretical study,however, it will produce a variety of mechanisms for efficient support of FGKL version 1 and indicate the functions which each machine component should provide.

At the end of this stage, specification of each machine component to be used for the parallel inference machine is expected to be clear enough to design custom LSI or VLSI chips.

### 4.2 The intermediate stage

Main goal of this stage is to develop an experimental parallel inference machine including more than 100 processing elements. And improvement of the sequential inference machine is also planned to provide the researchers with a better tool more widely.

At the beginning of this stage, the details of the plan will be reconsidered depending on the degree of the research progress in the initial stage.

For the sequential inference machines, it is expected that many claims to the hardware system will be made during the initial stage. Accordingly,improvement may be required both in architecture and circuit technology.

Architectural changes will be made to include such new mechanisms as better man-machine interface for speech and picture input/output, new structured memory developed in the research of object oriented architectures. Changes in the circuit technology will be the adoption of custom LSI or VLSI chips and it will make the machine smarter. It is expected that one chip inference machine will also be developed for high level personal computers of general use.

In knowledge base machine subproject,a small scale relational data base machine is planned to be developed in the initial stage. If it is successfully made, it will be connected to some of these machines to perform software experiments more efficiently.

For the development of the experimental parallel inference machine, several research results will be used. One result is the basic machine structure given by the research on data flow machines. Another one is the high level mechanisms proposed by the theoretical research of the parallel inference machine in the initial stage.

Furthermore, the research on object oriented architectures will also contribute to the organization of the structure memory, methods of resource management,and so forth.

For the implementation of this machine, custom LSI or VlSI chips are indispensable because of the complexity of its hardware containing more than 100 processing elements. And the amount of hardware involved in each processing element will exceed existing minicomputers. Accordingly, a good VLSI CAD system should be prepared by the beginning of this stage.

### 4.3 The final stage

In this stage, the research results of the parallel inference machine and the knowledge base machine are planned to be unified into a target machine which we call the prototype of the fifth generation computer.

Although its image is given in 3.1, a rough sketch is as follows. The prototype is expected to include about 1000 processing elements, however, it is a collection of the processors having different functions.

For example, one group of the processors performs operations for the inference such as the unification, and another group performs operations for knowledge base such as relational algebra ,and some other group performs operations for resource management,and so on.

Communication between these groups or among the processors in a group will be made through packet switching network system. Control of the machine is mostly distributed among the groups and the processors which operate based on the data or message driven mechanism. However,some control like the management of data base will be centralized.

VLSI technology will be fully used to implement machine components and it will make the machine as small as existing large scale computers. The machine organization will be based on distributed function architecture, although the function of each component is very high because of VLSI implementation. Accordingly, it provide the machine with highly modular structure which assures flexible configuration as well as high availability.

In more macroscopic view, development of hardware and software will be done using a local network system which is prepared as one of the basic research tools. And the scale of network is planned to be extended in accordance with the progress of the development. Thus, in the final stage, the network system will include many hardware systems and also the large accumulation of software. This will surely be another big product of this project and may be called as the knowledge information processing network system.

## 5. Conclusion

The research and development of the inference machine described in this paper intend to produce a new machine architecture combining three major concepts. They are logic programming, dataflow machine architectures and VLSI technology.

This fact naturally means that it exploits new fields outside the familiar conventional architectures. Furthermore, it makes this project more important that the project will also contribute to the improvement of conventional computer systems which will be still used in the near future.

This plan will be reconsidered and reorganized at the beginning of each stage to cope with many unknown factors involved in research items. However,the interim results as well as the final goal will be fruitful and become the good basis of computer science in the next generation.

## Acknowledgment

This plan was made through the investigation of various fields of computer science. Authors would like to express their gratitude to to all the researchers who contribute to the investigation.

## References

1. David H.D.Warren:IMPLEMENTING PROLOG -compiling predicate logic programs,vol.1,D.A.I. Research Report No.39,Univ. of Edinburgh,May 1977.

2. David H.D.Warren:IMPLEMENTING PROLOG -compiling predicate logic programs,vol.2,D.A.I. Research Report No.40,Univ. of Edinburgh,May 1977.

3. David H.D.Warren:LOGIC PROGRAMMING and COMPILER WRITING, D.A.I. Research Report No.44,Univ. of Edinburgh,Sep. 1977.

4. L.M.Pereira and F.C.N.Pereira and D.H.D.Warren:User's Guide to DECsystem-10 PROLOG,Sept. 1978.

5. J.Allen,et al:VLSI design automation activities at M.I.T., MIT/VLSI Memo No.80-33,Oct.1980.

6. G.J.Sussman,et al;Computer aided evolutionary design for digital integrated systems,MIT/AI Memo No.256,May 1979.

7. C.P.Thacker,et al:Alto:A personal computer,CSL-79-11,PERC Xerox,Aug.1979.

8. A.Goldberg and A.Kay:Teaching SMALLTALK,SSL-77-2,PERC Xerox, Jun.1977.

9. D.W.Clark:The Dorado:A HIgh-Performance Personal Computer,CSL-81-1,PERC Xerox,Jan.1981.

10. P.Wegner(Ed):Research directions in software technology,MIT Press,1979.

11. G.J.Myers:Advances in computer architecture,John Wiley and Sons,1978.

12. A.Snyder:A machine architecture to support an object-oriented language,MIT/LCS/TR-209,MIT,1979.

13. J.Rattner,et al:Ada determines architecture of 32-bit microprocessor, Electronics,p.119-126,Feb 24,1981.

14. J.D.Dennis:The varieties of dataflow computers,Proc.1st Int'l Conf. on Distributed Computer Systems,p.430-439,1979.

15. W.Ackerman:Dataflow languages,Proc.NCC,p.1087-1095,1978.

16. W.Ackerman:A structure memory for dataflow computer,MIT/LCS/TR-186,MIT, 1977.

17. Arvind,et al:An asynchronous programming language and computing machine,Tech.Rep.No.114A,UCI,1979.

18. P.C.Treleaven:Exploiting program concurrency in computing systems,

COMPUTER,p.42-50,Jan.1979.

19. A.Plas,et al:LAU system architecture: A parallel data-driven processor based on single assignment,Proc.Int'l Conf. on parallel processing, p.293-302,1976.

20. R.M.Keller,et al:A loosely-coupled applicative multi-processing system, Proc.NCC,p.861-870,1979.

21. A.L.Davis:A data flow evaluation system based on the concept of recursive locality,Proc.NCC,p.1079-1086,1979.

22. K.Furukawa,et al:Problem solving and inference mechanisms,in this proceedings.

23. H.Tanaka,et al:Intelligent man-machine interface mechanisms,in this proceedings.

24. T.Yokoi,et al:Logic programming and dedicated super personal computer, in this proceedings.

25. M.Amamiya,et ai:New architectures for knowledge base mechanisms,in this proceedings.

26. K.Sakamura,et al:VLSI and system architecture,in this proceedings.

27. H.Tanaka,et al:Preliminary research on data flow machine and data base machine as the basic architecture of fifth generation computer, in this proceedings.

# New Architecture for Knowledge Base Mechanisms


| | |
|---|---|
| Makoto AMAMIYA | Musashino Electrical Communication Laboratory, N.T.T. |
| Katsuya HAKOZAKI | Nippon Electric Co. Ltd. |
| Toshio YOKOI | Electrotechnical Laboratory |
| Akira FUSAOKA | MITSUBISHI Electric Corp. |
| Yuzuru TANAKA | Hokkaido University |



Hardware implementation of knowledge base management mechanisms is a very important problem in order to make fifth generation computer systems perform knowledge information processing facilities with high performance. This paper discusses issues on research and development of a new architecture for knowledge base mechanisms. First, a definition of knowledge base mechanism and its role in the fifth generation computer system are discussed, from the viewpoint of relations to the inference mechanism. Then, research items of a new architecture for the knowledge base mechanisms are described from two viewpoints, relational data base machine architecture and its extension to knowledge base machine, and implementaion of associative access control mechanisms in a knowledge base memory system. Finally, a research and development plan is described. The development is scheduled in three phases. Relational data base machine is developed, based on an advanced von Neumann type machine architecture in the first phase. Highly parallel knowledge base machine subsystem is develcped based on non von Neumann type parallel machine architecture in the second phase. In the third phase, knowledge base machine subsystem and inference machine subsystem are intergated into a fifth generation computer system.


## 1. Introduction

Fifth generation computer systems will be designed as knowledge information processing systems (KIPS) which realize a very high level and flexible man-machine interface, based on a large amount of generalized (common sense) and specialized knowledge data. Furthermore the fifth generation computer systems should be designed to offer high performance and to store a large amount of knowledge data. Therefore, one of the most important research items for fifth generation computer systems is the mechanism for managing such a large amount of knowledge data efficiently.

The knowledge base management mechanisms should be clarified from the software and hardware implementation viewpoint. Especially, it is important for knowledge base management mechanisms to be implemented as a hardware system specialized for storing and retrieving knowledge data items efficiently, since the efficiency of knowledge base mechanisms directly affects the performance of fifth generation computer systems as knowledge information processing systems. Although the design items for the hardware should be specified, theoretically, through the software system design for knowledge base management mechanism, some of those items should be clarified, practically, from the hardware implementation viewpoint.

This paper discusses the issues on the research and development of knowledge base management mechanisms from the hardware implementation viewpoint.

First, the concept and structure of the knowledge base management mechanisms, whose hardware implementation is called a knowledge base machine, is discussed. A definition of the knowledge base machine and its role in fifth generation computer systems are discussed.

Then, issues on knowledge base machine development are discussed from two viewpoints. One viewpoint is that the knowledge base machine is developed based on the relational data base machine by extending its functions to support some of the inference mechanisms. The other viewpoint is that the knowledge base management functions are implemented in such a structure that a knowledge base machine has interface directly to the inference machine, which only controls a reasoning process in problem solving by using information offered by the knowledge base machine.

Finally, a plan for knowledge base machines research and development is described. The research and development is scheduled in three phases. In the first phase, a relational data base machine is developed. The reason why the relational data base machine is developed in the first phase is that it is expected to offer an architectural base to knowledge base machine development. The developed relational data base machine will also be used as a general data base machine system.

A preliminary version of fifth generation computers will be developed in the first phase, based on the advanced architecture of the von Neumann type computer. It will consist of an inference machine component and a relational data base machine component. The inference machine component will be specialized to execute programs written by a core language, which will be designed as some extended version of PROLOG, efficiently. The relational data base machine component will be designed as a memory management component and attached to the inference machine component. This relational data base machine component will consist of several processing modules specialized to operate relational algebra functions.

In the second phase, the knowledge base machine will be designed to realize high level and high performance knowledge base management functions. This knowledge base machine will have a hierarchical memory structure which stores both intentional data (rules) and extensional data (facts). It will have functions to store and retrieve rules and facts through pattern matching operations. Parallel and pipeline processing techniques are considered as key factors for realizing such a high level management for a large amount of knowledge data at a high performance level. The most important problem is how to implement the data flow control concept in knowledge base machines. The experimental system will be constructed of about a hundred knowledge operation modules and about the same number of structure memory banks. The control mechanism for each knowledge operation module and structure memory module is based on a data flow control conept which offers a highly parallel processing effect. All of those modules will be built effectively by making use of VLSI technologies.

In the final phase, the knowledge base machine and the inference machine will be integrated into one system, which is a prototype of fifth generation computer systems. The inference control modules will be highly distributed in knowledge base memory system. Knowledge operations will be executed within the knowledge base memory system. The system may be constructed from about a thousand inference control modules, knowledge operation modules, and associatve memory modules, all of which will be interconnected through several connection networks.

## 2. What Is a Knowledge Base Machine ?

The hardware for fifth generation computer system will consist of three major functional components, from the stand point of their basic software interface:

(a) Problem solving and inference component.

(b) Knowledge base management component.

(c) Intelligent man-machine interface component.

These components are so closely interrelated that it is difficult to separate each function clearly. Especially, in the final version hardware structure, inference component and knowledge base management component will be intertwined with each other. However, it is important to clarify the knowledge base management component functions, which include mechanisms for retrieving and storing knowledge data, and the memory structure suitable for those functions, because the development of fifth generation computer systems will start with the concept that inference control mechanisms and knowledge base management mechanisms are treated separately, both logically and physically, in the analogy for conventional separation of the processing unit and the memory unit.

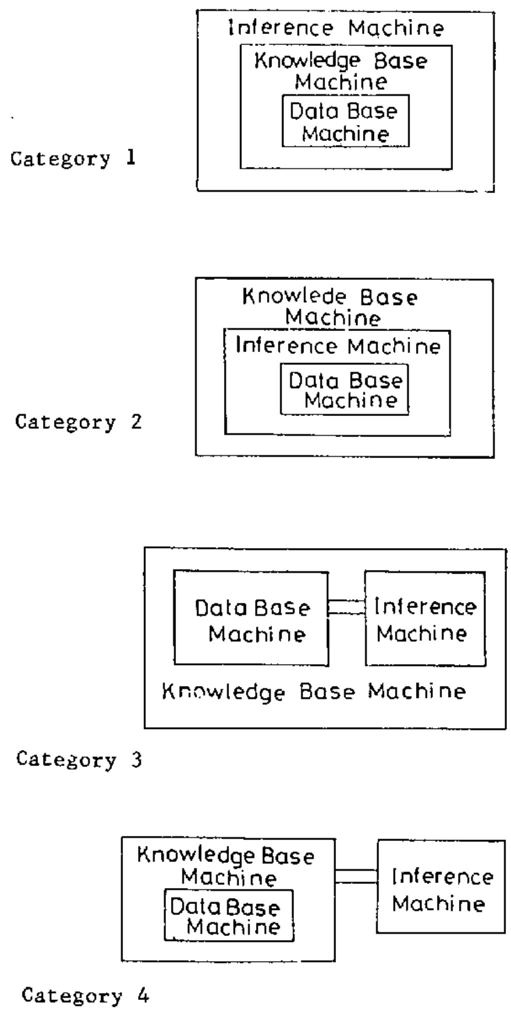



Figure 1.
Fifth generation Computer Hardware Structure

The knowledge base management component and the inference control component are called a knowledge base machine and an inference machine, respectively. The knowledge base machine role is summarized as follows, from the viewpoint of function distribution between the inference machine and the knowledge base machine.

(1) The knowledge base machine holds a large amount of knowledge data, which are structured well for accessing each knowledge item effectively.

(2) When the knowledge base machine receives a demand from the inference machine, it searches and retrieves knowledge items effectively and hands them to the inference machine.

(3) When the knowledge base machine receives knowledge data items from the inference machine, it compiles and integrates them into the knowledge base.

Although this view of the machine structure is acceptable as a logical structure from the analogy of the conventional machine structure, i.e., the relationship between the central processing unit and the memory unit, some fundamental problem lies in the definition of the knowledge base machine and the inference machine, and their relationship. The structure depends on the requirements from the software model of a knowledge base system, such as knowledge representation languages and knowledge data structures. For instance, the structure and the relationship between inference machine and knowledge base machine may vary, depending upon the model on which the realization of software systems is based, among the production system model, the semantic network model, the frame theory model, the actor model, and so on.

Several system structure categories may be considered, some of which are shown in Fig.1. In any case, knowledge base machines may be constructed founded on the relational data base concept and its hardware implementation technique, since the relational model has a sound theoretical foundation, such as relational calculus and relational algebra.

In the Category 1 model the fifth generation computer system is considered as the inference machine itself. The inference machine is constructed on the knowledge base machine by adding higher level inference facilities, such as meta-inference mechanisms. The knowledge base machine has several low level inference facilities, such as associative pattern search and variable bindings. These knowledge base mechanisms are built on the relational data base machine. The Category 2 model conceptualizes the fifth generation computer as a knowledge base machine, in which knowledge base management mechanisms are realized by implementing inference machine facilities, such as variable bindings, syllogism, context generation, consistency checking, etc., on the relational data base machine. The Category 3 model is different from the Category 2 model in regarded to the point that the inference machine and the relational data base machine are built as independent hardware modules. In the Category 4 model, the fifth generation computer system is constructed of the inference machine and the knowledge base machine, each of which is built as an independent hardware module. This model is rather more acceptable than other models, in the sense of the conceptual relationship between the knowledge manipulation control module and the knowledge memory module with associatve access control facilities.

There arises, however, a question about whether the knowledge base management mechanisms can be separated clearly from the inference machine, as in the Category 4 model. This induces the Category 5 model, as shown in Fig.2. In Fig.2, the interrelated component, between the inference machine and the knowledge base machine, is left unspecified as a black box. This component is to be clarified during research on the fifth generation computer system. The major architectural issue is to make this black box component clear, from both the inference machine viewpoint and the knowledge base machine viewpoint.

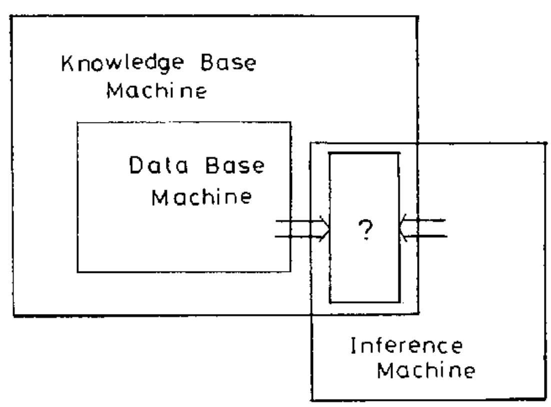



Figure 2.
Interface Component between Knowledge Base Machine and Inference Machine

3. Issues on Knowledge Base Machine

The architectural base of the knowledge base machine is considered from two stand points; the mechanisms which support a data base system, based on the relational data base model effectively, and the mechanisms which support the data base access control co-working with problem solving and inference mechanisms. The former implies an implementation problem of the relational data base machine, which involves a relational algebra operation on a huge amount of data. The latter offers a problem of interface to the inference machine, which is discussed in Section 2.

The implementation of relational algebra machine is a starting point for knowledge base machine development project. Its related issues are to be solved in the next few years. As several researches have been working on the data base machine architecture, main themes to be worked on are to evaluate architectures ever offered as ideas or developed in laboratories, and to select and/or create new ideas from those experiences for data base machine architecture. Based on this consideration, several issues on the implementation of relational algebra machine (or relational data base machine) are summed up in the following.

(a) What performance can be obtained from the relational data base machine, which is constructed on the basis of today's or next few years' device technology, such as LSI devices and memory devices? The performance is measured from the viewpoint of storage capacity, searching or access speed, relational algebra operation speed, facilities for search and operation, and so on.

(b) The developed relational data base machine itself may be used not only as a base system for knowledge base machine development, but also as an independent data base machine system. As the main objective of the machine development is, however, to utilize it as the basic hardware system for experiments on the knowledge base machine system, the machine performance has to be sufficiently efficient to use as a supporting tool for designing and developing the fifth generation computer software system, e.g., knowledge representation language design and knowledge base implementation, basic application software systems implemented on the knowledge base system, and so on.

(c) The limitation and criticism for the relational data base machine which was developed based on today's technology, if any, should be clarified and a means of solving the problem should be suggested from the viewpoint of enlarging the data base capacity and obtaining higher level associative search and access to knowledge base. (This means the way to realize some inference mechanisms in the knowledge base machine.)

(d) As system structure and performance strongly depend on the memory system, it is necessary to specify memory device requirements in order to realize a memory system for the data base machine, based on prediction in regard to advances in device technology.

(e) The major current research topics on computer architecture are distributed processing, parallel processing, pipeline processing, abstract data type support machine concept and tag architecture. The data flow machine concept is expected to be a prominent new concept, which unifies these processing concepts. The way of unifying these new concepts is a very important issue, in order to make the relational data base machine more efficient so as to facilitate knowledge base management. One of the key problems to be solved is how to implement the history sensitive processing, so as to support data base update, since the knowledge base is changed very frequently in the environment of knowledge base management which facilitates knowledge acquisition mechanisms.

(f) A large amount of data should be structured hierarchically so that the required data items can be searched for efficiently. The structure memory construction for such a large data base is also an important problem. Though the abstract data type support machine and tag architecture concept have come from the requirements in the software engineering and language design philosophy, it may affect the construction method used for structure memory in the data base machine design, in which individual relation tables may be clustered hierarchically and a description or directory to each relation may be set into the structure memory.

(g) Another important problem concerns the data base access control algorithm. This problem is one of the main issues in the current research for a distributed data base system and/or a multi-user accessible data base system. Issues concerning this problem are security control, integrity control, concurrency control, and so on. When the relational data base is applied to the knowledge base system, these issues are thought to be very important, since each data item may be updated very frequently in the knowledge base management environment.

From the viewpoint of interface to the inference machine, the main issue is to specify the function and its mechanism and the structure of the interface component which is left unspecified as a black box in

Section 2. In order to clarify the black box, the system image and its structure should be specified for the first time. This specification will be accomplished based on the software system design.

The principal issue is what kind of functions should be supported by the knowledge base machine and what kind of functions should be included in the inference machine. This problem, presented in other words, is: what kind of functions in problem solving (or in knowledge processing) should be defined in the context of the inference mechanism and what kind of functions should be defined in the context of the knowledge base management mechanism. The functions defined in the knowledge base management mechanism are called knwoledge operations. In any way, knowledge operations may be specified by extending relational data base operations to the operations which support frame structure and context mechanism and binding mechanism for quantified variables.

Issues on the knowledge base machine are derived from the assumption that the black box is a unification control part (or general pattern matcher), i.e., the knowledge base machine consists of relational data base machine and unification control part (or general pattern matcher). These issues are listed in the following.

(a) What kind of knowledge data should be stored and managed in the knowledge base machine?

Knowledge data are classified into two categories. One is extensional data, another is intentional data. The extensional data, or called facts, do not have any quantified variables. They represent assertions of individual facts. The intentional data, or called general facts, rules or theorems, involve quantified variables in them. They represent the intentional concept, such as general facts or theorems, which are used in inference processes such as syllogism.

The extensional data may be stored in the relational data base machine. On the other hand, for the intentional data, there arises a problem concerning where the intentional data should be stored; in the relational data base machine, within the inference machine, or in any other new storage module which unifies intentional and extensional data.

(b) Can relational (algebra) operations and inference operations match each other well?

The fundamental concept of the inference is a deduction mechanism, and it is classified into two basic categories. One is a resolution principle based mechanism such as PROLOG, and the other is a production rule based mechanism such as production systems. For both inference categories, pattern search and unification are basic mechanisms. The relational (algebra) operations are basically set operations, including the search for relation tables. In order to join inference operations and relational (algebra) operations, it is very important to design a unified data structure which can represent both the intentional and extensional data economically. The knowledge base machines design strongly depends on the data structure.

(c) How should the memory system for knowledge base be structured and arranged in the hierarchy?

Knowledge data may be classified into several categories, from individual fact data to the most general meta-knowledge data. The access frequency may vary depending on the kinds of knowledge data. For the individual fact data, which is considered as relational data, the access frequency may be low but many data items may be accessed at once. This kind of data may be stored in the memory, called a long term memory. For the meta-knowledge data, on the other hand, the access frequency may be high, but only a few data items are accessed. This kind of data may be stored in the memory called a short term memory.

It is also very important for each data item to be grouped into some context or frame structure, so that the search pattern area is localized and useless pattern search is inhibited. In order to implement this context or frame structure, the structure memory organization is a key problem. The data abstraction concept and its machine architecture (abstract data type support machine) offer a hint to structure memory construction; tag and descriptor mechanism, heap area management mechanism, garbage collection mechanism, and so on. Of course, the memory architecture depends on VLSI technology. Especially, VLSI chip implementation of associative memory is a key factor.

(d) How is the knowledge base machine implemented from the viewpoint of parallel and distributed processing?

The prallel processing and distributed processing should be a basic architectural concept of knowledge base machine implementation, in order to obtain high speed operations for the high level functions required for the knowledge base machine. One of the most promising of the parallel and distributed computer architectures is the data flow control concept.

The knowledge base machine may be organized to such a structure that a number of processing modules, which are specialized to execute the operations in MIMD (multiple instructions multiple data) fashion, are connected with each other through inter-connection networks. In order to realize such a system structure, parallel execution control, pipeline processing control, stream processing control and distributed processing control should be integrated into one system architecture. The data flow control is considered to be most suitable for this purpose, since the data driven or demand driven execution control offers a basis for parallel and distributed control, in principle.

## 4. Research and Development Plan for Knowledge Base Machine

Following the master plan for fifth generation computer system research and development, knowledge base machines will be developed in three phases. The major goal for each phase is as follows. In the first phase, the relational data base machine will be developed as a memory management system for the inference machine, which will be designed as a PROLOG machine or like that. The machine is a multi-processor system of four to eight processing modules which are implemented based on an advanced von Neumann machine architecture. In the second phase, the knowledge base machine subsystem will be developed as a knowledge base management system, which will support high level mechanisms, such as knowledge operations. The machine is highly multiple processors system consisting of about a hundred processing modules and structure memory banks, each of which is implemented based on a non von Neumann parallel machine architecture. In the final phase, the knowledge base machine subsystem and the inference machine subsystem will be combined to organize a fifth generation computer system, in which inference operation modules and knowledge operation modules will be highly distributed in the system.

The development plan and research items in the three phases are described in the following. A research and development schedule is shown in Table 1.

### 4.1 First phase plan

(a) Design goal

The relational data base machine consists of four to eight processors. An example of such a hardware structure is shown in Fig.3. Each processor module, which is specialized to perform the relational algebra operations, is implemented using the conventional firmware or hardware techniques, based on the sequential von Neumann machine concept. Almost all of the operations in relational algebra; i.e., union, intersection, difference, projection, join, division and selection (restriction), are set operations. Those set operations are reduced to primitive operations; search, sort and join.

Some system features are listed in the following.

(1) Each processor module has high speed cache memory, so as to perform set operations at high speed. Each processor may integrate sort and search oriented special hardware devices.

(2) Relation tables are stored in relation memories, which are a silicon disk. Temporary relation tables are stored and manipulated in this relation memory.

(3) A large amount of fact data is stored in back up memories in the form of the relation data. This data are read into relation memories when they are accessed. Each back up memory has selection operation modules in it. One of its implementations may be an intelligent disk with logic-per-track mechanisms.

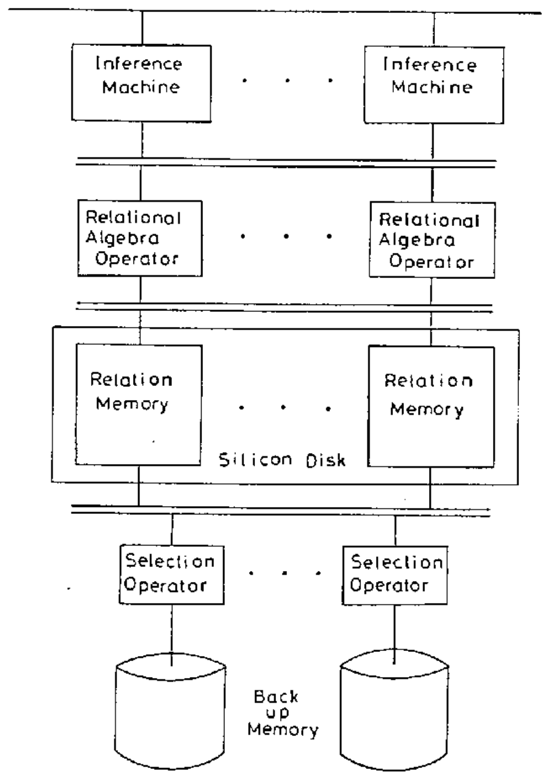



Figure 3.
An Example of System Developed in the First Phase

(b) Performance requirement

(1) Silicon disk: Access time is less than 10 micro seconds. Transfer rate is more than 4 mega bytes per second per bank. Memory size is 100 mega bytes to 400 mega bytes.

(2) Back up memory size is about 10 giga bytes.

(3) Operation speed is less than 0.25 seconds for join of 1 mega bytes by 1 mega bytes, where 1 tuple is assumed to be 100 bytes.

(c) Design items

The major design items are listed in the following. These items are clarified through hardware design and software simulation process.

(1) Hardware module design, specialized to perform relational algebra operations.

(2) Relation memory design: It involves specification of relational data structure, extraction and specification of associative access functions, implementation techniques (e.g., whether hashing hardware implementation or content addressable memory chip implementation), interface to relational algebra operation modules, and so on.

(3) Interface to the inference machine, and interface to data base query languages or software systems.

(4) Design for back up memory and selection operation modules.

(5) Specification for data abstraction mechanism from the viewpoint of structuring the relation tables and back up memory directory control.

(6) Access control strategy design: it involves integrity control and concurrency control related to data sharing and protection against changing of data items.

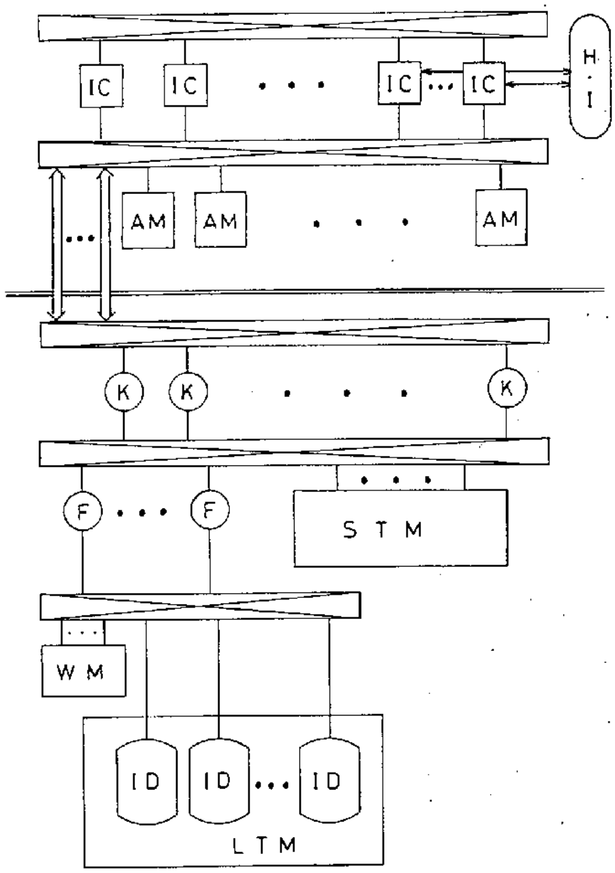
(d) Basic research items

Basic research on elementary techniques for the knowledge base management mechanisms is undertaken in parallel with the development of the relational data base machine. The elementary techniques, which are the basis of the knowledge base machine development in the second phase, are listed in the following. The related problems are examined through hardware experiments and software simulations.

(1) Research on implementaion of relational algebra operations, based on parallel processing concept, such as data flow control, pipeline control, and data stream concept: The way of combining actor model and data flow concept is a hint to the research. The implementaion of history sensitivity mechanism is one of the most important problems, so as to support integrity and concurrency control for data base update.

(2) Research on an implementation of a structure memory suited for storing list structured data or tuple structured data: Data abstraction mechanisms will be a basis for structuring knowledge and/or relation data in the hierarchy. VLSI implementation of content addressable memory is involved in the research.

(3) Relations between relational data base model and inference mechanism, such as PROLOG and production system.




H.I : Human Interface
IC : Inference Controller
AM : Active Memory
K : Knowledge Operator
F : Filtering Operator
WM : Working Memory
STM : Short Term Memory
LTM : Long Term Memory
ID : Intelliger + Disk

Figure 4.
An Example of System
Developed in the Second Phase

(4) Specification of knowledge operations through examinations of knowledge representation model: The knowledge representation models to be examined are frame structure, context mechanism, KRL (knowledge representation language), production system, semantic network, and so on.

(5) Research on packet communication mechanisms and interconnection network structures: The implementation of interconnection network is a key problem for realizing highly parallel systems.

4.2 Second phase plan

(a) Design goal

Highly parallel knowledge base machine subsystem is constructed of about a hundred processing modules and memory modules, which facilitate a high level associative access control. An example of the system is shown in Fig.4. The architecture is based on a non von Neumann machine concept, such as data flow control, stream processing, pipeline processing, and so on. The knowledge base machine subsystem is integrated to fifth generation computer systems joined with the inference machine subsystem, which is also implemented based on a non von Neumann parallel machine architecture, such as data flow machines. The developed system is used as a support tool for knowledge base management software development.

(b) Performance requirement

(1) Knowledge operation module: Performance is about 4 to 8 MIPS (million instructions per second). The number of modules is more than 100.

(2) Short term memory: Structure memory with associative access control mechanisms. Access time is 0.5 to 1 micro seconds. Memory size is more than 4 mega bytes per bank. The banks are more than 100.

(3) Long term memory: Intelligent disk with logic-per-track mechanisms. Transfer rate is more than 100 mega bytes per disk. Memory size is more than 100 giga bytes.

(4) Interconnection network: Packet communication and pipeline staging. Transfer rate is more than 2 mega bytes per second per port. Network has more than 100 input and output ports.

(c) Design items

(1) Implementation of knowledge base management mechanisms, from the viewpoint of data flow concept: It involves a unified design of stream processing and data flow control mechanisms.

(2) Short term memory system design: It involves designing a structure memory, which may be implemented using VLSI chips in the content addressable memory.

(3) Knowledge operation primitives design: It involves integration of relational algebra operations and inferential data access operations, and specification of interface to the inference machine subsystem.

(4) Design of data filtering operations, executed on the intelligent disk.

(5) Network system design: It realizes a flexible and high performance connection structure through a packet switching mechanism.

(d) Basic research items

Basic research on the prototype design or advanced architecture for fifth generation computer systems should be undertaken in parallel with the knowledge base machine subsystem development. Pertinent research items are listed in the following.

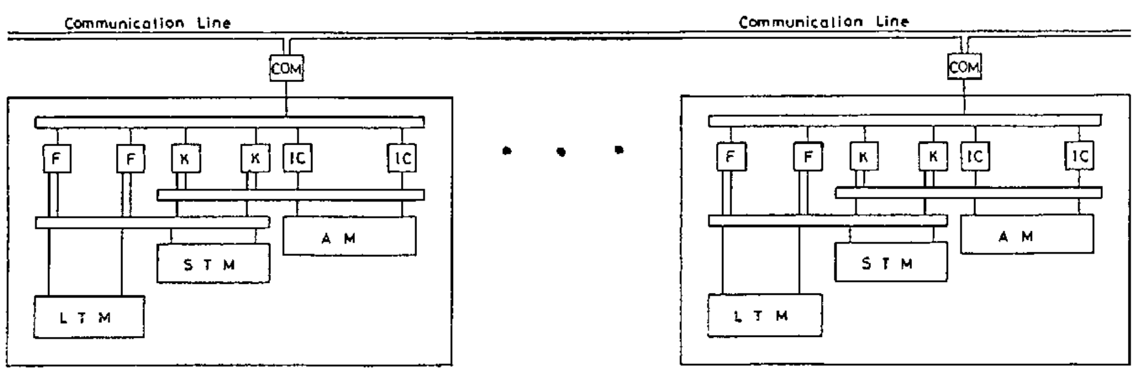



Figure 5.
An Example of System
Developed in the Third Phase

(1) A simulation of kowledge base management models, which are made based on a parallel processing concept. The simulation is performed on the hardware system developed in the first phase. A preliminary model of the system, on which the second phase development of knowledge base machine subsystem is based, is refined through the simulation.

(2) Several decentralized processing models, such as actor model and semantic network model, are investigated from the hardware implementation viewpoint, assuming VLSIed architectures.

(3) A highly distributed knowledge base management and/or inference system is investigated, assuming VLSIed data flow processors. The system may be a realization of actor machine, autonomous processor network system, hardwired semantic network system, or such like.

### 4.3 Final phase plan

In a prototype for fifth generation computer systems, knowledge base mechanisms and inference mechanisms may not be separated clearly, but may be integrated into one system. Therefore, the knowledge base subsystem developed in the second phase is refined and re-designed from the viewpoint of realizing such a system structure. Elementary techniques, which are clarified in the second phase research, will be used as a basis for the development of a prototype of fifth generation computer systems. A fifth generation computer system will consist of a thousand modules, which are connected to each other through communication networks. Individual modules, which are realized by VLSI chip, are solely used for knowledge operations, deduction operations, meta-inference operations, knowledge base structuring control, associative memory access control, and so on.

Knowledge information processing systems are realized with network connection of fifth generation computers, each of which realizes some expert knowledge base system by software implementation. An example of such a system structure is shown in Fig.5.

Research themes in the final phase, are summarized as:

(1) To refine the system developed in the second phase, so as to integrate inference mechanisms and knowledge base mechanisms into one system.

(2) To extend system size so that the system is constructed of a thousand function modules and associative memory modules.

(3) To implement the system by making use of advanced VLSI technology.

## 5. Conclusions

This paper discussed a research and development plan for knowledge base machines. First, a definition of knowledge base mechanism and its role in fifth generation computer systems was discussed. Then, issues on new architecture for knowledge base mechanisms were discussed from two stand points: The relational data base machine architecture and its extension to knowledge base machine, and the point of implementation for associative access control mechanisms in a knowledge base memory system with interface to an inference control module. Finally, a research and development plan was discussed. The development is pursued in three phases. In the first phase, relational data base machines, which consist of several specialized sequential processors, are developed. In the second phase, knowledge base machine subsystems, which are composed of about a hundred knowledge operation modules and structure memory banks, are developed. Each module is designed based on non von Neumann type parallel machine architecture, such as data flow control concept. In the third phase, a prototype of the fifth generation computer systems, in which knowledge base mechanisms and inference mechanisms are integrated into one system, is developed.

Definition of knowledge base mechanisms and its role in fifth generation computer systems are very difficult problem, since requirements for knowledge base management functions are not yet clarified in knowledge information processing systems. Therefore, some of the functions and mechanisms, which are dogmatically specified from the view point of knowledge base management in this paper, may overlap with those of inference machines.

The preliminary study on knowledge base machines just started in June of this year. As only a few months have passed since the beginning of the study, this paper is written under insufficient investigation into the definition of knowledge base mechanisms. Some of the issues described in this paper may be changed through further investigation, or during the research and development of knowledge base machine. Especially, it should be noted that specification of knowledge base management functions and their implementation on the knowledge base machine strongly depend on the requirements from the software design for knowledge base management system.

Tbale 1. Knowledge Base Machine Development Schedule

| | First phase | Second phase | Third phase |
|---|---|---|---|
| | Sequential Type Relational Data Base Machine (RDBM) | Parallel Type Experimantal Knowledge Base Machine | Prototype of fifth generation computer |
| Development | • Specialized hardware of relational algebra operation (firmware, several units)<br>• Silicon disk (100 - 400 MB, several banks)<br>• Back up memory (10 GB) | • Data flow control<br>• Relational/Knowledge operation oriented processor<br>• Network connection (100 by 100 ports)<br>• Structure memory (associative access, 100 banks)<br>Intelligent back up memory (100 GB) | • Integration of inference machine and knowledge base machine<br>• Highly distributed system (1000 operation modules and logic-in-memory modules)<br>• VLSIed function module<br>• Message/Data stream processing<br>• Actor machine model<br>• Semantic network structure |
| Basic Hardware Experiment | • Relational algebra operation element (parallel mechanism)<br>• Knowledge operation element (parallel mechanism)<br>• Data flow control mechanism<br>• Stream processing mechanism<br>• Structure memory (logic-in-memory mechanism, list/tuple structure)<br>• Network (tree, array, omega)<br>• Hierarchical relation structure and memory organization | • Integration of relational algebra operation and knowledge operation<br>• Actor machine and distributed data flow processors<br>• Highly distributed functions<br>• Hierarchical memory sructure (associative access mechanism, highly distributed memory structure)<br>• Hardware of semantic network<br>• VLSIed architecture | |
| Software Simulation | • Specification for knowledge operations (Large-scale computer)<br>• Software implementaion on {Lisp machine + RDBM, PROLOG machine + RDBM}<br>• Production system<br>Semantic network model<br>Actor model<br>Frame structure<br>KRL<br>• Access control strategy (security, integrity) | • Specification for highly distributed system structure | |

VLSI AND SYSTEM ARCHITECTURE — THE DEVELOPMENT OF SYSTEM 5G


| | |
|---|---|
| Ken SAKAMURA | University of Tokyo |
| Akira SEKINO | Nippon Electric Co.,Ltd. |
| Toshihiko KODAKA | Hitachi, Ltd. |
| Takao UEHARA | Fujitsu Laboratories Ltd. |
| Hideo AISO | Keio University |



A research and development proposal is presented for VLSI CAD systems that are indispensable for the construction of the Fifth Generation Computer and for a hardware environment called "SYSTEM 5G" on which the VLSI CAD systems run. Proposed CAD systems use a hierarchically organized design language in which we can design anything from basic architectures of VLSI to VLSI mask patterns in a uniform manner. The CAD systems will eventually become intelligent CAD systems that acquire design know-hows and perform automatic design of VLSI chips when the characteristic requirements of VLSI chip is given.

SYSTEM 5G will consist of superinference machines and the 5G communication network. The Superinference machine will be built based on a functionally distributed architecture connecting inference machines and relational data base machines via a high-speed local network. The transfer rate of the local network will be 100 Mbps at the first stage of the project and will be improved to 1 Gbps. Remote access to the superinference machine will be possible through the 5G communication network.

Access to SYSTEM 5G will use the 5G network architecture protocol. The users will access the SYSTEM 5G using standardized 5G personal computers, "5G personal logic programming stations", very high intelligent terminals providing an instruction set that supports predicate logic and input/output facilities for audio and graphical information.

In the following, details of the research and development plan and major research topics on SYSTEM 5G are discussed.


## [1] Overview

The Fifth Generation Computers aim at "Knowledge Information Processing" and they are different from conventional computers in terms of the structure and the theory on which they stand. In its construction, it is, therefore, important to tailor every system components including electronic parts consciously from the beginning so that they are suitable to the Fifth G.C.S. There are various candidates for the basic electronic components. However,our decision on this is using silicon-based devices. This decision is based upon the research in the past and the prediction on the state of technology in 1990s, when the Fifth Generation Computer is expected to go into operation. The method to construct silicon-based devices, namely VLSI with more than 10 million transistors, becomes the key to the success of the Fifth Generation Computer once our choice of silicon is made.

There are two major research topics on producing VLSI chips. One of the topics is VLSI algorithms. This research topic includes the decision on what functions should be implemented on VLSI chips and the problem of how to realize these functions on VLSI chips. The other research topic centers around the support system for the design and the fabrication of VLSI chips, that is, VLSI CAD systems. Besides these two topics, there are other research problems we must face. We must make the VLSI CAD systems easy to use and standardize them because these VLSI CAD systems will be used by many researchers associated with the development project. It seems that substantial software efforts should be made in implementing these systems and that standardized terminals with intelligent man-machine interfaces are strongly desired to be developed.

VLSI CAD systems usually require much computing power. Even the highest speed computer of today will find the shortage of computing power to support VLSI CAD systems. However, at least a VLSI CAD system is needed from the beginning of this project, and it must be able to run on a new more powerful computer whenever such a computer becomes available. Needless to say, these VLSI CAD systems will be required to run on the Fifth Generation Computers when they become available in 10 years from now. We must prepare the hardware environments to satisfy these requirements. Development of such a hardware environment necessitates research on highly flexible and evolutional system architecture that can adapt to the change in the hardware environment. We call the system with such flexible architecture "SYSTEM 5G".

We keenly feel that a new development methodology is required for the Fifth Generation Computers which differ from conventional computers in various technological points. We also recognize that it takes relatively long times to develop such a system. However, we want to start using it soon for the development of the VLSI CAD systems. How can we satisfy our needs? In facts, the Fifth Generation Computers will be the computers that can meet our criteria! How can we solve this dilemma? We believe SYSTEM 5G can give a solution to this problem, as will be described in this paper.

## [2] Background and Requirements

The research theme is roughly divided into three distinct areas. They are:

<1> Construction of VLSI CAD systems,

<2> Development of SYSTEM 5G on which the VLSI CAD systems are expected to run, and

<3> Development of the 5G personal computer, that is, the standard terminal used for interfacing users with SYSTEM 5G.

### [2.1] VLSI CAD systems.

In the first stage of constructing VLSI CAD systems, it is clear that we must make every effort in the development of software satisfying the requirements described below. The research efforts, of course, should be based on the fundamental research on VLSI algorithms.

(1) The CAD systems can be used to design customized VLSI chips with more than 1 million transistors. They must be able to describe the specification of electronic circuits, logic circuits, devices, fabrication processes, mask layout routing, wiring, timing and testing procedures. Furthermore, it must perform the simulation of the circuits at a reasonable speed. It is also required to generate layout patterns of the mask.

(2) The CAD systems must be able to perform pattern check of the mask layouts generated. This layout pattern, once verified, must be fed into the mask generator without human intervention. Furthermore, test patterns must be generated automatically when a mask is generated and these test patterns must be fed into the VLSI testing equipments.

(3) The CAD systems must provide uniform and standardized user interface which must be easy to the uses. The ease of use means that a university graduate whose major is computer science or related subjects can master the use of the system in less than half a year training.

(4) The CAD systems must run on a Knowledge Information Processing System which will become available in 10 years from now.

(5) The CAD systems must be put into use by the end of the first stage of the project. Item from (1) to (4) above must be satisfied in 10 years. The CAD systems must run on SYSTEM 5G and the personal computers.

The present CAD systems in Japan and abroad does not satisfy the requirements listed above. No commercial CAD system offer the capability of the descriptions from the architecture level through the mask layout generation level. Few CAD systems are made with growth in mind. The research of CAD systems that can run on a Knowledge Information Processing system has just begun. There seems to be no CAD system that has been built with strong emphasis on the ease of use and the simplicity of the training requirements.

[2.2] SYSTEM 5G

In general, SYSTEM 5G will be used in the following way:

(1) Users sit in front of personal logic programming stations.

(2) The personal logic programming stations are directly connected to an in-house local network.

(3) Users use the personal logic programming stations as a terminals to access superinference machines.

We want to expand the system by implementing itself on a network architecture The use of network architecture is expected to bring forth the following advantages: We can offer a variety of services by using distributing functions on a network. The efficiency of the processing may be improved by using special-purpose machines on the network High serviceability and extensibility of the system are also attained. The system based on a network architecture can be expanded by incorporating new machines into the network.

The favorable features of using a network architecture described above are to be pursued by existing computer networks. We feel, however, that the conventional network architectures such as SNA developed for von Neumann machines are not satisfactory for the Fifth Generation Computer System, because the Fifth Generation Computer System includes, computers of completely new architectures. We, therefore, wish to establish a new network architecture suitable for move advanced computers. The system will be constructed so that a variety of remote computers will be give an image of one virtual machine to each of the users. Virtualization of the system and ease of use obtained through well-designed high-level man-machine interfaces are the two most important objectives of the research on SYSTEM 5G.

Hence, research must be focused on the personal logic programming station, the super-inference machine and the network architecture. The first research topic, that is, the personal logic programming station will be discussed in [2.3]. The latter two topics will be explained in the following.

[2.2.1] Superinference Machine

The superinference machine will normally be installed in a building. It will be a computer complex that incorporates the fruits of the project, specifically inference machines, relational database machines and network controllers, all connected on a 1 Gbps high-speed local network. These computers will be the state of the art products of the year 1990.

The superinference machine will provide the list processing capability that is expected to be 1,000 times as fast as that of a conventional 4 MIPS general purpose machine. It will be equipped with 1,000 GB data base. It is estimated that it takes ten years to build such a machine. It is our belief that we should start doing research on network architecture which may offer the basis of the system growth.

It is inevitable that computers connected to the local network to be developed at the beginning will be general-purpose computers of conventional architectures. However, these computers will be gradually replaced by new computers of novel architectures and will disappear completely in ten years. The required characteristics of the superinference machine are summarized in the following.

(1) The main objective of SYSTEM 5G is to run the VLSI CAD systems ( described in <1>). However, SYSTEM 5G also has to support the development of software for the 5th Generation Computers, facilitate the information exchange among researchers, and help manage the project research activities.

(2) SYSTEM 5G must allow various application-oriented computers to be easily built into the system. This necessitates that SYSTEM 5G must be of an architecture of distributed function architecture.

(3) SYSTEM 5G must be an extensible production system. Hence, the system must have the reliability to allow serious production-runs. The system will be put into production use in three years after the project begins.

(4) SYSTEM 5G must be easy to use, because many researchers will use it. This requires that 5G personal computers be connected to it. Documents, manuals and training systems must be provided so that an individual with fundamental knowledge of computers can use it after one week of study.

(5) SYSTEM 5G must be accessible from anywhere in Japan through DDX services of NTT. It must have communi‿ ‘ion links with international networks to fac‿ ‿ate the information exchange with researchers living abroad. Data securities on these networks must be maintained properly.

Fundamental research to satisfy the above requirements seems to have been already done, but we can hardly say that there is a practical and extensible system that allows fully distributed function architecture; the distributed network architecture including high-level inference machines has never been studied before. Running SYSTEM 5G for production purposes requires the use of various research results of today's computer science and the realization of the SYSTEM 5G will have a great influence on the fundamentals of tomorrow's computer science.

[2.2.2] 5G Network Architecture

The 5G network architecture includes local networks of personal logic programming stations, global networks maintained by NTT etc., as well as SYSTEM 5G high-speed local networks described above. The requirements for the 5G network architecture are summarized, as follows.

(i) A local network architecture of personal logic programming stations must be first established. This architecture must have adequate considerations on connecting with inference machines later.

(ii) The established network must connect with the global DDX network serviced by NTT.

(iii) The global network must connect with world-wide global networks.

(iv) Users normally access resources within the local network. But they should be able to access, without any difficulty, resources beyond the global network. Access to the global network will be made via a gateway processor. The superinference machine must be connected with the global network.

Besides personal logic programming stations and the superinference machine, the following devices will be connected with the 5G network. They are telephones, facsimiles, sensors of various types, optical character readers, intelligent copiers, etc.. Hence, communications in the form of voice, graphic images, digitized images will be possible. High level of security and reliability must be attained.

[2.3] Personal Logic Programming Station

The requirements for the personal logic programming station are summarized as follows:

(1) A personal logic programming station is basically a personal inference machine. Hence, LISP and PROLOG, for instance, must be available as the basic languages. The size of the station must be that of a TSS station residing on a desk because the personal logic programming station is for personal use at researchers' sites. The personal logic programming station will serve as the standard terminal in SYSTEM 5G.

(2) Although there would be several models of the personal logic programming station, at least, two models for VLSI CAD systems and for software development must be developed as soon as possible.

(3) Ease of use must be sought as much as possible. Hence the personal logic programming station must provide the processing capabilities of voice, graphics and digitized images.

(4) The personal logic programming stations must be installed widely once the project begins. Hence, its physical size must be reasonable. The use of VLSI seems indispensable. The number of the personal logic programming stations to be used by researchers will be several hundred.
There has been no personal computer that satisfies the requirements listed above. Therefore, a completely new processor including useful man-machine interfaces should be designed.

[3] Research Outline

The three research activities should proceed simultaneously. In the following, we describe how the requirements listed in the previous section will be satisfied by various research activities as the time passes.

[3.1] VLSI CAD system
Four steps must be followed in designing VLSI chips. That is;

<1> specification of VLSI functions or the description of its architecture,

<2> functional, logical and electronic design,

<3> layout pattern generation and,

<4> test pattern generation.

In designing VLSI chips, decomposition of the design should often be done because it is very hard to handle a chip with tens of thousand of gates at once. Close relationships among <1>, <2>, <3> and <4> are frequently observed. Therefore, a hierarchical design principle should be taken into account. Unified description and automatic data handling via a database throughout the hierarchical design steps are essential for smooth design procedures. Such a database should store know-hows of the design and is desired to work with a design consultation system. This CAD system is an application of the knowledge information processing technique.

In practice, it has an inseparable interface to the equipments at the fabrication processes. In Japan today, the interface between the layout patterns generated and fabrication equipments is different to different companies. This fact causes a big problem. When a user wants to make a customized VLSI chip and has finished the layout of the mask by himself. He still has long way to reach the goal since he has to learn the particular style of interfaces for the fabrication process he has chosen. Hence, it is very important to design the VLSI chips for the Fifth Generation Computer in a standardized way.

(i) Standard VLSI Design Language at the initial stage

The initial stage means the first five years of the project. The desire to use a standard design language as soon as possible forces us to adopt the best language currently available and then improve it as the project proceeds. We have been investigating what language should be chosen and have not decided on any language yet.

One of the languages that attracts our attention is the Hierarchical Specification Language (abbreviated as HSL hereinafter) that has been used at the Musashino Electrical Communication Laboratory, (Nippon Telegraph and Telephone Public Corporation) since 1980. The outlook of HSL is shown in Figure 1. The HSL system has a database and several compilers for design simulation are provided in it. It is possible to generate test patterns and perform checks on design descriptions. With these design procedures spanning over several levels of VLSI, it is possible for us to input the description of a logic circuit with 4,000 gates in 220 man-hours. It is also said to have the capability of completing the layout design of 32-bit VLSI processor with 12,000 gates in one month.

We would like to start using a language as powerful as HSL when the project commences. We refer to this language as 5G-HSL in the rest of this document for convenience.

Maintaining a standard language means supplying various documents including one for education. We call this educational manual "5G VLSI Design Rule Book". This manual must be completed at the early stage of the project. It must be written in a self-taught way so that a university graduate whose major is a computer-related subject is expected to read this manual to obtain a first-hand experience in three months. At the beginning, 5G-HSL is expected to run on a large-scale general purpose machine in SYSTEM 5G.

(ii) 5G Project-Authorized Fabrication Plant

In the future, the compiled version of 5G-HSL will be sent to fabrication plants via a 5G global network. We should authorize the fabrication plants that can accept the compiled code of 5G HSL as the fabrication process description. Therefore we must draft the qualification standard that must be met by an authorized fabrication plants. We do not feel that the 5G project development promotion center (an interim name) should have a fabrication plant of its own. It is desirable that authorization is open to any companies so that foreign fabrication plants can qualify for the authorization offered through the 5G project.

(iii) Construction of Knowledge Data Base for VLSI Design

It is necessary to construct a database which gathers the know-hows of VLSI design. This data base is used in the intelligent VLSI CAD system which will run on the Knowledge Information Processing System ( KIPS ), a major theme of the research in the later phase of the project. The data base will, at the beginning, be used for a Question and Answer System for human designers. But it should be used through CAD systems in the later stage of the project. Hence, a smooth transition of the usage style from human interactions to dialogue through CAD systems must be established when the database is constructed. Furthermore, more and more know-hows of VLSI design will be collected as the project proceeds. The data base must be constructed to allow for later growth.

[3.2] SYSTEM 5G

SYSTEM 5G will be built on the 5G network architecture, a high-level network architecture for the SYSTEM 5G environment. In ten years of the project, SYSTEM 5G will probably become a very large knowledge information network that incorporates local and global networks connecting numerous personal logic programming stations and superinference machines. SYSTEM 5G, as a hardware environment, will host the VLSI CAD systems, especially the 5G-HSL system. We will briefly describe SYSTEM 5G from a viewpoint of VLSI CAD systems and software development.

Let us first look at the construction of the superinference machine. The basic design philosophy of SYSTEM 5G is as follows: It must always supply the users with the most powerful computing power for the VLSI CAD systems. Therefore, the network architecture is expected to be most suited for the implementation of SYSTEM 5G. The network architecture in our mind is a combination of high-speed (100 Mbps) optical fiber local networks and low-speed (10 Mbps) conventional local networks. The optical fiber local network may be composed of a pair of duplicated paths to guarantee high reliability. Machines for VLSI CAD and software development will be connected to the high-speed local networks and 5G personal computers to the low-speed local networks.

It is necessary to install the whole superinference machines in one place. Hence, the construction of the system center in which 5G SYSTEM is to be accommodated should begin as early as possible. The superinference machine should be accessible from the outside of the center via DDX offered by NTT. Users will access SYSTEM 5G using personal logic programming stations connected to the network. Hence, communication protocols for these networks and communication control programs for both the superinference machines and personal logic programming stations must be developed. The 5G network architecture will have been established when these are developed. It is expected that it will take at least three years to develop the superinference machine. Hence, the following construction scenario is proposed:

(i) Initial Machines for VLSI CAD/Software Development

We must plan to use one of the fastest conventional general-purpose machine for initial VLSI CAD machines, because we have no other choice if we want to use it as soon as the project starts. 40 MIPS seems to be the minimum requirement for its processing power at the first stage of the project*.

---

* This requirement was derived by considering the following three hypothetical jobs as the expected workload:

Circuit simulator program: A program which needs the execution of 10G instructions simulates the dynamic physical behavior of 2,000 transistors on a VLSI chip over the time interval of 2 microseconds.

Automatic placement and routing program: A program for automatic placement and routing of devices on a VLSI chip. This program needs the execution of 30G instructions.

A simulator for logic circuits with 500,000 gates: This program needs the execution of 20G instructions.

It may safely be said that these programs need the execution of 20G instructions on the average. Suppose we wish to process such a 20G instruction jobs in 15 minutes CPU time and 1 hour elapsed time.

Then the required processing speed of the computer is calculated as follows:

20G/(15min*60sec)=22*10**6=22M[1/sec]

Hence the speed must exceed 20 MIPS which covers only production runs. Debugging and editing will also require the same order of the processing capability. Therefore, the processing speed of more than 40 MIPS will be required.

As to the machine for software development, we also have to use currently available machines. But more emphasis is laid on the ease of use than on the processing speed. The machine must run Interlisp, Maclisp and Edingburough PROLOG as soon as it is delivered. The machine must provide a TSS OS that has proven to be well-accepted in the market. The architecture of the machine for software development must be popular in the world to ease the information exchange with foreign research organizations.

(ii) SYSTEM 5G Network Architecture

The fundamental design of 5G network architecture and a Superinference machine must be finished by the end of the 3rd year. As we have stated, the completion of Superinference machine construction would take 10 years, because it is inevitable that its component machines at the beginning will be the traditional von Neumann computers. So our initial activity will aim at the development of fundamental software necessary for SYSTEM 5G.

Initial research topics for this software development includes the development of a distributed operating system to control the network and the virtualization technique to enable us to use a personal logic programming station as a virtual terminal of the Superinference machine.

Incidentally, building construction for the SYSTEM 5G computer center "The 5G Computer Development Support Center (tentative name)" should ,of course, be finished by the time the SYSTEM 5G is completed. The machines that will be used at the beginning will be the ones mentioned in (i).

(iii) Research for SYSTEM 5G at the later stages

At the later stages of the development, incorporation of inference machines and knowledge base machines into the SYSTEM 5G will become the major research activity. It is recommended that the data transfer rate of the high-speed local network be increased to 1 Gbps.

[3.3] 5G personal logic programming station

The 5G personal logic programming station is a terminal for the Fifth Generation Computer and is a personal inference machine. It is a superpersonal computer equipped with high-level man-machine interfaces. There would be several models tailored to specific applications. One point that may attract our attention is that these personal computers must be supplied on a commercial base throughout the whole project. The number of these personal computers used in the project will be several hundred. These personal computers will be delivered to researchers and be connected to the SYSTEM 5G local network on DDX.

(i) Model 1

It will take three to five years after the start of the project to develop Model 1 of the personal logic programming station. Model 1 will be used in SYSTEM 5G as a machine for VLSI CAD systems. PROLOG must be available and this characterizes the personal logic programming station Model 1. The processor will be based on an enhanced von Neumann architecture and be realized employing firmware techniques. The graphics capability is very important because the personal logic programming station Model 1 will be used as a VLSI CAD terminal. Customerized IC chips for voice recognition, pattern recognition, display controller and central processing will be designed and using 5G-HSL.

(ii) Model 2

Model 2 of the personal logic programming station will be used as a terminal station for software development. Model 2 will also be used for information exchange among researchers and as a terminal for accessing databases. The central processor will be similar to that of Model 1 and will execute PROLOG. Model 2 will have less advanced specifications than Model 1, because Model 2 aims at the wider use than Model 1. Model 2 is also expected to run the software developed in the initial stage of the project in an emulator mode.

The ratio of the number of Model 1 personal logic programming station over that of Model 2 will be 1:5. It is desirable that we develop Model 1 first then begin to produce Model 2 personal logic programming stations in large quantities. We want to retain the functional compatibility between Model 1 and Model 2. The difference lies in their performance. We will not make a compromise in achieving the required performance of the Model 1 personal logic programming station, since Model 1 must serve perfectly as a VLSI CAD terminal. The processor may not be realized on a single VLSI chip to achieve the high-speed processing capability. In designing Model 2 the cost-effectiveness will be pursued. This means that the number of VLSI chips required for the processor should be reduced by sacrificing its performance.

[4] Development Plan

In this section, research activities to be performed at the early stage of the project are described. The outcome of the research activities must satisfy the requirements outlined in Section [2].

[4.1] VLSI CAD

(i) Development of 5G-HSL System

The 5G-HSL system will be realized on the conventional general-purpose machines determined by the project. The development of the 5G-HSL system must produce the following documents:

<1> The external specification of 5G-HSL

<2> The internal specification of 5G-HSL

<3> The requirement specification of the 5G-HSL Terminal

<4> The requirement specification of the au-

thorized fabrication plant

Document <3> describes the necessary requirements of the capability that the terminals should provide, for instance, input and output of graphical information, requested by the VLSI CAD system in 5G-HSL. These requirements may be the same as those for the 5G personal computer. The 5G personal computer will be designed according to the specification for the 5G-HSL terminals. Document <4> describes the qualification standard that must be met by the authorized fabrication plants in the 5G project.

<5> 5G-HSL Program Source Listing and Explanatory Manual

<6> 5G-HSL System Verification Manual

5G-HSL is expected to be gradually modified for the improvement of its performance. Version 1 must be running at the end of the 1st year of the project, and version 2 will be at the end of the 3rd year.

(ii) Development of the 5G-HSL Education System

Much more emphasis is being placed on the education system in constructing the 5G-HSL system. The reason comes from the fact that the huge growth of the VLSI designers is anticipated in the future, and the education system must meet the demand.

<1> 5G VLSI Design Rule Book

This also serves as an introduction to the 5G-HSL system. "Self-taught" style is desirable.

<2> 5G VLSI Design CAI system

This is a computerized version of <1>. It is required that <1> be finished in one and a half years and <2> be finished in two years from the beginning of the project.

(iii) Fundamental Research for Intelligent CAD System

The development of intelligent CAD systems should be coordinated with the research on the inference machine and intelligent data base. Full-fledged research on intelligent CAD systems will be done in the later stage of the project. However, basic research should commence from the beginning. One of the very important topics in the basic research is how to collect and store the design rules and the design know-how, into an intelligent database system.

<1> Development of Methodology for the Design of VLSI Architecture

Design rules and algorithms suited for VLSI will be established through basic research for building intelligent CAD systems. We must find a way to store these rules and algorithms into a knowledge base.

<2> Development of Design Consultation System

A design consultation system which stores design know-hows and communicates with human designers in a Q & A way must be developed. Emphasis at the early stage of the research is placed on the establishment of a knowledge base. The construction of a knowledge base for the VLSI CAD system will be carried out through the first hand experience that helps build up the methodology. The collection and cataloging of the design know-hows should begin as early as possible.

<1> and <2> should be finished in the first half of the project and they will be used in

the research activities at the later stage. The intelligent VLSI CAD system, which is the final objective of the research, may look like the one shown in figure 3. It incorporates a knowledge base management system and an inference machine. The knowledge base mangement system activates the knowledge base machine and stores the design rules, know-hows and device information into the knowledge base. The inference machine is the core of the intelligent CAD system, supporting the automatic generation of mask layout pattern once the users feed a high-level description of the VLSI, such as the ISP ( Instruction Set Processor ) description, into the CAD system.

[4.2] Development of SYSTEM 5G

[4.2.1] Development of SYSTEM 5G Hardware

The development of the hardware for the Superinference machine is divided into two phases, that is, the version 1 hardware developed at the first stage of the project and the version 2 hardware at the later stage of the project.

<1> Version 1 Hardware

A sketch of the version 1 hardware is shown in figure 4. Main research subjects concerning version 1 hardware are discussed below.

(a) High-Speed Optical Fiber Local Network

Our attention in the first stage of the project will be paid manily to the connection of different conventional computers on local computer networks. Therefore, the transfer rate of 100 Mbps will be required for the local networks at the beginning. The transfer rate of 3 MB/sec will be sufficient to handle the traffic between a processor and a message switch. In order to maintain the high reliability a dual optical fiber systems might be considered. When a message switch malfunctions, the system must still operate satisfactorily after detaching the faulty message switch from the system.

(b) Connection of Different Computers on a Network

Figure 4 shows a case when the VLSI CAD machine is made up of two processors. There is a dual path to the input/output devices such as disks, MSS, or MT shared by the both processors to enhance the reliability. The systems for software development are apparently different from the VLSI CAD system. Hence, it is necessary to establish a technique to connect processors of different architectures on a network. It is important to dedicate our research efforts to the connection of the inference machines and the relational database machines using the technique developed. All the machines will be integrated on the 5G Local Network at the final stage of the research.

(c) Gateway Processor and Low-Speed Network

Gateway processors will be used to interface the low-speed and the high-speed local networks which have different transfer rates and protocols.

(d) Communication Processor and Global Network

The 5G network architecture will use the DDX network offered by NTT as one of the physical networks. The DCNA protocol will be used basically, but we feel the particular characteristics of the knowledge information network, that is, the use of numerous personal logic programming stations and a powerful superinference machine, will necessitate the inclusion of specialized features into the protocol. Communication facilities for audio and visual information through various devices should also be provided.

<2> Version 2 Hardware

At the later stage of the project, the version 1 hardware will be enhanced by the introduction of the inference machine, knowledge data base machine, super computers for scientific calculation and numerous high-performance 5G personal logic programming stations. The transfer rate of the high-speed local network will be increased. However, a very high-speed local network will have to be provided for the database machine that deals with various files containing audio and visual information and for the super computer for high-speed scientific calculation as shown in Figure 5.

The system as a whole will have become a giant complex of component systems and local networks. It will be implemented through the fruitful research results on distributed control methods.

Many research problems have to be solved before the 5G high-speed computer network described above is realized. Table 1 is a brief summary of the research activities for the high-speed local network. This table does not include research activities in device and material technologies, in which many research problems to be studied are involved as well.

The software for SYSTEM 5G is dealt with in detail in the next section.

The overall plan of the software development is to run the version 1 software in the 3rd year and then version 2 software in the 5th or 6th year of the project. The version 1 software aims at the support of the SYSTEM 5G hardware for research and development. The version 2 software is to support a distributed system and is to serve as a prototype of the future software. After the version 2 software comes into use, the research will move to the construction of the version 3 software which supports future computers such as the inference machine or the knowledge database machine. To give a reader a better glimpse of SYSTEM 5G, expected features of the SYSTEM 5G software and the research objectives are explained below.

[4.2.2] SYSTEM 5G Software

The SYSTEM 5G software must run on the SYSTEM 5G hardware and support the favorable features described in the previous sections. To satisfy these requirements, the development of the SYSTEM 5G software must be carried out, paying careful attention to the following system characteristics.

<1> System Configuration

* Functional distribution on the system should be accomplished as much as possible.

* System control should be decentralized.

* The locality of the processing and the data access should be maintained as much as possible.

<2> Distribution of Functions

There exist at least following three levels of functional distribution in SYSTEM 5G.

* Functional distribution between personal computers and the central system, i.e., the Superinference Machine.

* Functional distribution among specialized high-performance machines in the central system, e.g., inference machines, knowledge base machines, etc.

* Functional distribution among concurrent components within each high-performance machine.

<3> Distributed Control

* The system control should be equally distributed over the operating systems of all computers. This necessitates construction of a decentralized operating system.

* The distributed database capability should be provided.

* The master-slave relationship among operating systems, if any, should be avoided as much as possible.

<4> Locality of Processing and Data Access

* More than 90% of the users' processing activities and data access should be performed on their personal computers.

* Each specialized high-performance machine of the central system similarly performs more than 90% of data processing and data access within itself. In principle, files used by each machine of the central system should reside within each machine.

* Access to the central system from personal computers should be normally provided through a communication processor for DDX or a gateway processor for a local in-house network.

* The high-speed local network for the central system will be normally used for transferring jobs between specialized high-performance machines.

<5> Virtualization of the system

* The system should give an image of one virtualized computer when accessed from a personal computer.

* A clearly recognizable hierarchical structure must be established for the system so that machines and computers may be added or removed from the system in a modular fashion.

* Communication protocols and interfaces must be established between layers of the hierarchically structured SYSTEM 5G.

Research topics concerning the SYSTEM 5G software include (1) the development of a hierarchically organized logical model of the system, (2) the development of a decentralized control methodology, (3) the development of a distributed data base management system, and (4) the evaluation of the system and its performance by modeling. These topics need basic research before the software development is started. Some more details of these research topics are discussed below.

(a) Development of a Hierarchically Organized Logical Model of the System

A hierarchically organized logical model of the Knowledge Information Processing System (KIPS), as viewed from the personal computers, must be set up. This model should include high-level layers such as users' application layer. An example of the model of KIPS is shown in Figure 6. Interface between each adjacent layers in the hierarchal model must be determined so that we can establish various communication protocols. We should probably consider several existing protocols, including OSI (Open Systems Interconnection) standard model discussed at ISO, as a possible starting point of this research.

(b) Development of Decentralized Control Methodology

Elimination of the global control information of the network operating system must be carefully studied so that we may construct a totally master-free network. The study of decentralized control includes synchronization, concurrent processing, sharing, exclusion, deadlock avoidance, etc. The result of this study may turn out very useful to the study of distributed internal control of specialized high-speed machines of the control system.

(c) Development of a Distributed Data Base Management System

The distributed data base management system of the SYSTEM 5G must handle various forms of data, such as numbers, characters, figures, images, etc., in a uniform manner. The problem of coexisting data base management systems of different architecture should also be studied. The study of distributed data base includes data distribution, directory distribution, synchronization, sharing and exclusion, intelligent access, performance considerations and so on.

(d) System Evaluation by Modeling

Various design decisions and the resulting system performance must be evaluated at various levels. The system evaluation should include the study of appropriateness of functional distribution over computers, the study of a balanced system configuration specifying the number of specialized central high-performance computers and personal computers of each type, and the study of the throughput and response time characteristics of the entire system. Modeling techniques should include analytical models such as queueing models, and simulation models of various kinds.

[4.3] Personal Logic Programming Station

The research and development plan for the personal logic programming station Model 1 and Model 2 is as follows.

(i) Development of Inference Engine

We call the central processor of the personal logic programming station "the inference engine". The inference engine must be designed so that it can be made on VLSI chips, because it will be used for the personal logic programming station for personal use. The architecture of the inference engine at the beginning may be of an enhanced von Neumann architecture. This architecture must support functional programming and object-oriented languages which handle objects of varied structures effectively.

Not much emphasis is placed on the parallel processing capability. The cost performance of the design is pursued with the use of microprogramming techniques. The required performance at the lisp processing capability is set 10 to 100 times as fast as that of a 4 MIPS machine of today.

(ii) Model 1

We have stated that Model 1 will be used as a VLSI CAD terminal. The required features of Model 1 is described below.

(1) The display screen is of raster scan type and its resolution must be more than 2000 by 2000.

(2) High-level editing capabilities, such as screen-oriented editors must be supplied.

(3) The station must be equipped with input devices to allow image processing. A digitizer tablet with the resolution of 0.1 mm or finer must be supplied.

The station must have devices which can recognize hand-drawn figures and sentences consisting of hand-drawn Kanji characters, Katakanas, Hiraganas and alphabetical characters in either on-line or off-line modes.

The Kanji character set must have at least 3,000 characters. The rate of correct recognition on Model 1 must be higher than 95%.

(4) Inputing of voice must be essential. When the user speaks the sentences to be inputed phrase by phrase, Model 1 is expected to be capable of recognizing them in a real-time manner. It must handle more than 10,000 different words. Its rate of correct recognition must exceed 95%. Voice output is also required.

(5) The capacity of the main memory and the secondary storage should be more than 1 MB and 100 MB respectively.

The inference engine will be used as CPU to satisfy the requirements listed above. Software to interface with the 5G network must also be developed.

(iii) Model 2

The core of Model 2 will not be different from that of Model 1, because Model 2 uses the same inference machine as one used for Model 1. However the features (3) and (4) described above can be omitted. Model 2 will require more control memory and user program areas than Model 1, because Model 2 will have to execute high level languages other than PROLOG or LISP and there would be demands to run the software developed at the first stage of the project in an emulation mode.

[5] Summary

We have presented the R & D plan for SYSTEM 5G, which provides a network architecture involving superinference machines and many personal logic programming stations, and also described its major application to VLSI CAD systems. SYSTEM 5G has an evolutionary architecture. Namely, at the beginning it will be used as a support system for the whole project, and it will grow up into a knowledge information network in 10 years after the project begins. Many research problems in the fundamental fields of computer science must be solved in developing SYSTEM 5G. The solutions to these problems are believed to make significant contributions to the advancement of computer science. In this sense, the development of SYSTEM 5G poses a challenging tasks to computer science researchers. The superinference machine will become the most powerful logical programming machine with inference capability in the world of the 1990's. It will be used successfully not only for the VLSI CAD systems but also for natural language understanding systems, translation systems, medical consultation systems, etc.. The use of personal logic programming stations by many researchers will also drastically change the today's view on programming. We are now just going to take the first step to this attractive and challenging field of research.

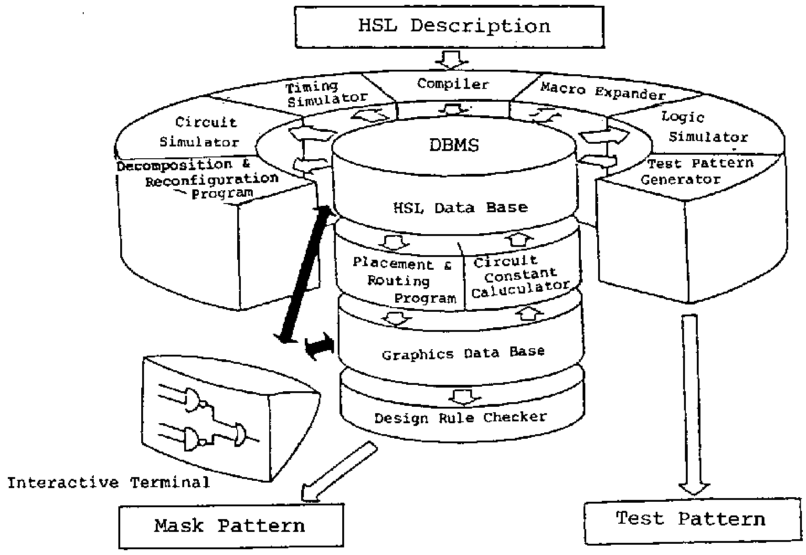



Figure 1 — The structure of LSI DA system *

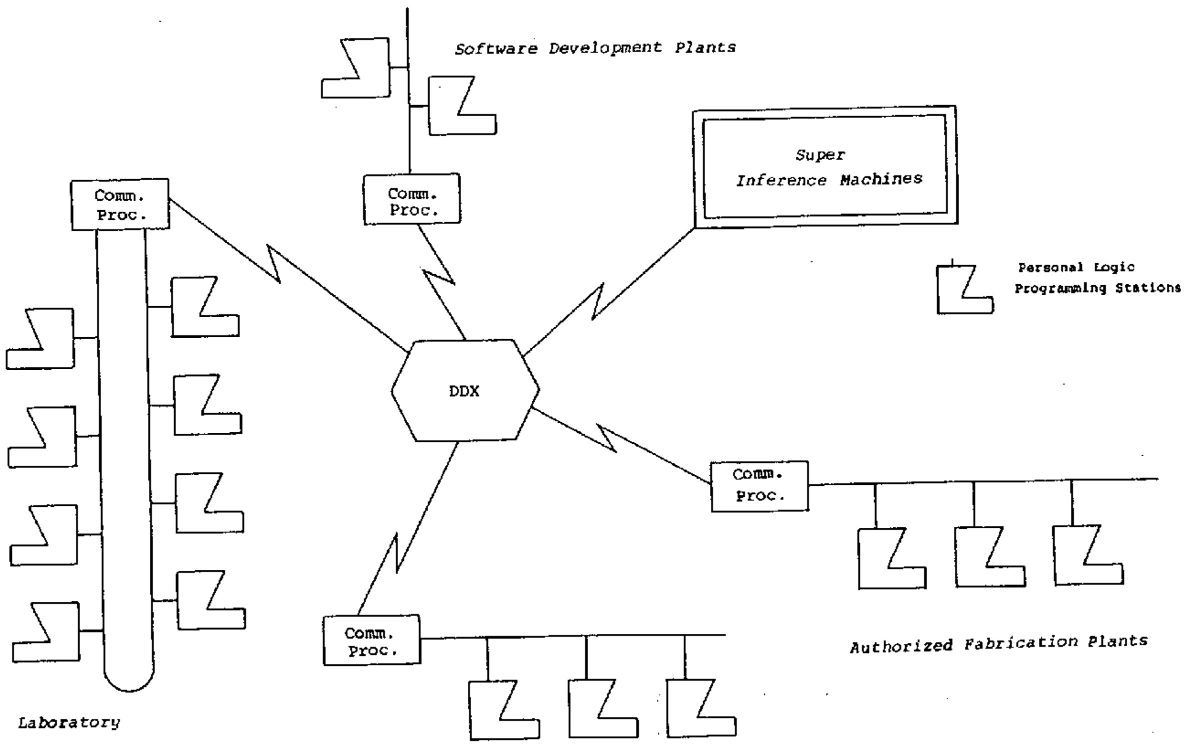



Figure 2 — Overview of system 5G

* from "Custom VLSI Design System"
by Tsuneta SUDO
and Yoshi SUGIYAMA (N.T.T.)

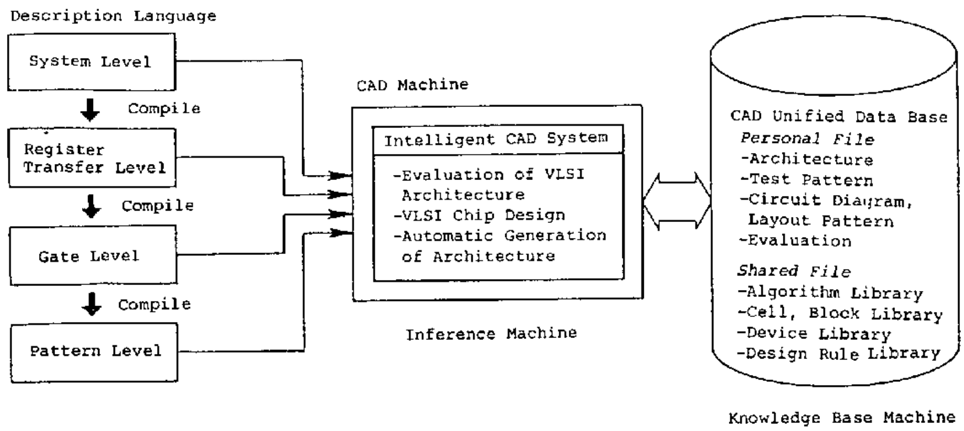



Figure 3 — An overview of intelligent CAD system

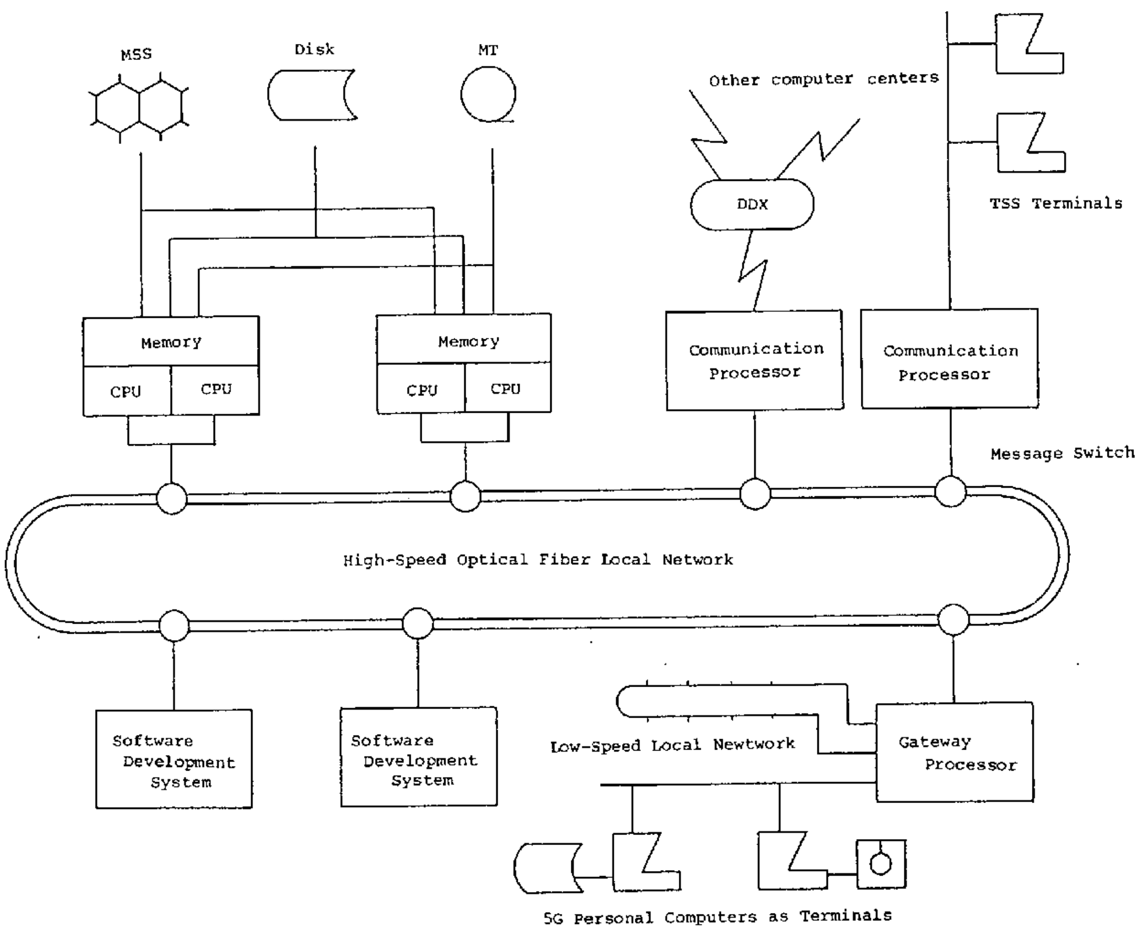



Figure 4 — *Super Inference Machine* —
Configuration in the 3rd year of the project

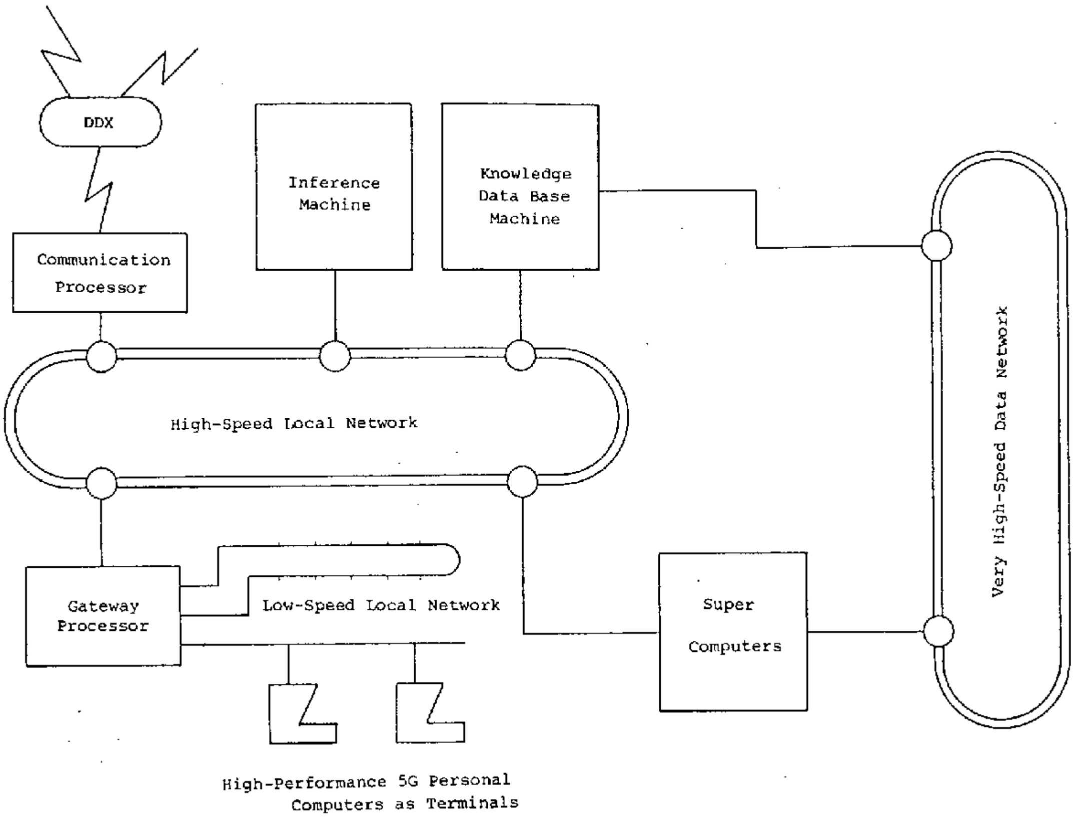
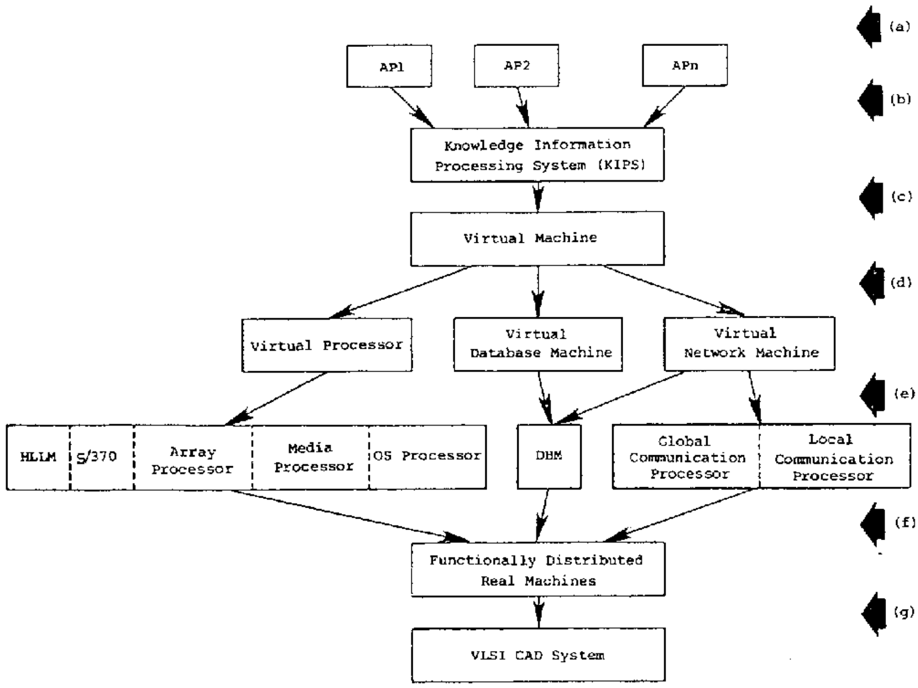



Figure 5 — *Super Inference Machine* —
Configuration at the final stage of the project




(a) Man-Machine Interface

(b) Application Interface

(c) Language Interface -------- DBMS; Programming System; Programming Language

(d) Access Interface --------- Virtual Instruction Set (PROLOG, etc.); Relational Database (QBE, etc.); Virtual Network (QSI, etc.)

(e) The First Virtualization --- Abstract Data Type (Object-oriented, etc.); Relational Algebra; Data Flow; Parallel Processing

(f) The Second Virtualization --- System Monitor & Control; Virtualization of Hardware Resources; Virtualization of Software Resources; Virtualization for Performance and Reliability

(g) The Third Virtualization --- Tag/Capability; Associative Memory; Cryptography; Resource Address Translation; VLSI Algorithm

Figure 6 — An example of hierachically organized logical model of the knowledge information processing system

Table 1 — A list of the research activity to construct the high-speed local network of super inference machine

| Theme | Development in the 3rd year | Outcome after the 10 years | Contents of the Fundamental Research |
|---|---|---|---|
| 1. Transfer capability of the fast ring network | 100 Mbps transfer rate | More than 1 Gbps transfer rate | The improvement of the transfer rate<br>The assurance of a rapid response and reasonable thruput time under heavy load |
| 2. Switching capability of Network Processor Interface | 3 MB/S transfer rate | 30MB/S transfer rate | The establishment of very high-speed interface for CH/Link<br>Allowing other fast devices to be connected to the network and widening application field |
| 3. Reliability | Doubly-structured ring<br>Isolation of faulty node | A completely dual file/processor system that has 100% availavility | The detection of faults and the automatic replacement of faulty devices.<br>On-line diagnosis<br>The control of dynamical reconfiguration |
| 4. Physical protocol | Establishment of protocol based on HDLC | World-wide acceptance of the protocol | The flexibility to allow the growth of components and the virtualization<br>Compatibility with the logical protocol |
| 5. mixed use of machines of different architecture | Physical connection must be possible.<br>Code/data conversion utility programs are supplied. | Architectural differences will be hidden by the use of high level languages | Establishment of standard character code and data representation |
| 6. File transfer between processors | File transfer is only done with neighboring processors.<br>Share disks will diminish the necessity of file transfer | Shared file that uses system very high-speed network hosted by a data base machine | Establishment of standard procedures to locate and reserve files in the distributed system<br>Development of job schedule algorithm in the distributed system |
| 7. Network topology | Simple ring structure (Physically dual system) | A complex network architecture that uses links of different types for different purposes | Algorithm to select the optimal sending path based on local decisions |
| 8. Security | Physical separation of the central system from general users | Cryptograph | Development of capability systems |
| 9. Message switch | Special products | Mass-produced low-cost products | Miniaturization, high-reliability and flexibility must be attained by VLSI |

THE PRELIMINARY REESARCH OF DATA FLOW MACHINE AND DATA BASE MACHINE

AS THE BASIC ARCHITECTURE OF FIFTH GENERATION COMPUTER SYSTEMS


| | | |
|---|---|---|
| Hidehiko | TANAKA | University of Tokyo |
| Makoto | AMAMIYA | Musashino Electrical Communication Laboratory, NTT |
| Yuzuru | TANAKA | Hokkaido University |
| Yoshihiko | KADOWAKI | HITACHI Ltd. |
| Masahiro | YAMAMOTO | Nippon Electric Co. Ltd. |
| Toshio | SHIMADA | Electrotechnical Laboratory |
| Yukio | SOHMA | FUJITSU Ltd. |
| Makoto | TAKIZAWA | Japan Information Processing Development Center |
| Noriyoshi | ITO | OKI Electric Co. Ltd. |
| Akikazu | TAKEUCHI | MITSUBISHI Electric Corp. |
| Masaru | KITSUREGAWA | University of Tokyo |
| Atsuhiro | GOTO | University of Tokyo |



The summary of the preliminary research of data flow machines and database machines, which started in July 1981 and will end in February 1982, is shown in this paper. The objective of this research is to analize the study items of both machines and to show a detailed research plan which will be put into action from April 1982. At the same time, we hope that we can show some concrete architecture of both machines that can be the major components of basic hardware structure of the knowledge information processing system that is the final target of this FGCS project. The requirement specification of both experimental machines that should be developed during the first three years of the FGCS project is touched as well.


## 1. Introduction

The main objective of the FGCS is to realize the knowledge information processing systems(KIPS). The major components of KIPS are the inference engine and the knowledge base system. Although inference operation can be implemented on conventional von Neumann machines of sequential type, some excellent parallel execution mechanism is required to carry out inference operations of practical size with high speed. The data flow mechanism is one of the most attractive and feasible solutions for this problem. On the other hand, KIPS requires knowledge information of fairly large size. To realize a knowledge base system with such high performance, we need some sophisticated hardware support. The data base machine can be the powerfull means for it. Accordingly, we can conclude that the data flow machine and data base machine are the most promising candidates for the basic architecture of the KIPS.

The FGCS project will run for coming 10 years which can be divided roughly into 3 stages. The research of the data flow machine and data base machine follows the 3 stage-plan. Though we are going to develop at the final stage, practical machines in such sence as usefull for KIPS applications, this preliminary research is one of a preceding research that is going to supply some fundamental data for the coming research project of the first stage. At present, we are working to clarify the research themes to be investigated, and allocating them to the 3 stages. Then, we will select the technological alternatives and show some preliminary design specifications for the data flow machine and the data base machine.

The same kind of research planning work is running simultaneously at the architecture and software working groups. However, their work is to make up the overall plan of 10 years and to make clear the role of data flow machine and data base machine in the environment of the KIPS. Our research group concentrates our effort on the plan of the first stage, and is going to develop some hardware architectures which supports efficient execution of data flow operations or data base operations. How to apply these architectures to KIPS is not within our scope, but within theirs.

This paper shows the role of data flow machine and data base machines in some detail, which we expect for KIPS at this time. Following the discussion, the research plan and some reguirement specifications of both machines are explained with research themes.

## 2. The role of data flow machine and data base machine in the FGCS environment

As 'inference' is the fundamental operation in KIPS, the corresponding hardware should be of high level to shorten the semantic gap between hardware and software. 'Try and error' and non-deterministic operations are characteristic of inference machines. Accordingly, some powerfull speed-up mechanism through parallel hardware is indispensable for them. As the data flow concept is an excellent methodology to control parallel operations, we can imagine a inference machine of such architecture that the components are functionally distributed and whole system is controlled through data flow concept.

The nucleus of description system in KIPS is composed of the problem solving description language and the knowledge representation language. The problem solving description language can be designed in the logic programming style and/or functional style. As predicate calculus is a high level descriptive language which permits non-deterministic programming, the logic programming is suitable for some knowledge based processing. However, parallel processing is needed to execute with high speed the program written in logic programming languages because the execution of the language is rather complicated. It is shown that the programs written in logic programming languages are closely related to the data flow graph which is one of the machine languages of data flow machine. With functional programming, it is estimated to be very easy to build the program logic, because the program has no history sensitivity. Functional programming can be the execution mechanism for logic programming and relational algebra. On the other hand, the data flow machine is a natural execution mechanism of logic programming. Therefore, data flow machines can be the basis of inference machines.

Knowledge can be considered to be general data of unfixed form and of complex structure which are described by the knowledge representation language. The knowledge base system is required to be highly general so as to handle any kind of knowledge data. The leading candidate for the basic structure of this system is the data base system which supports relational model. The main component of data base system is the data base machine which supports relational algebra. Another language interface of relational data base is the relational calculus, which is closely related to logic programming and can be mapped easily to the corresponding statements of relational algebra. Although the knowledge representation itself is a very important problem to be investigated with the progress of FGCS project, the other distinct characteristic of knowledge base is that the volume of knowledge data can be very large. Knowledge is composed of individual facts, data which represent some procedures, and so on. So, the volume will become very large, and it can be a very difficult problem to maintain the storage efficiency and processing efficiency to a high level. One of the method to meet this situation is to separate the individual facts in knowledge base and to store them in the data base machine. Existing data bases can be also utilized through the knowledge base system in such organization. Accordingly, data base machine can be one of the important components of knowledge base system.

As mentioned above, data flow machine and data base machine are the basic architectures of the FGCS. These machines are closely related each other, as data flow concept can be used to realize the data base machine, and on the other hand, the structure memory which is a major component of data flow machine can be built using the data base machine, but we are sure that both will play the important two distinct roles of the KIPS as the inference machine and knowledge base system.

## 3. The research schedule of this year

Up to this time, the research work of this year is scheduled to take the following steps:

(1) Evaluation of existing data flow machines and data base machines.
(2) Extraction and categorization of research themes.
(3) Allocation of these themes among 3 stages of the project with the expected performance and required system size.
(4) Analysis of the themes allocated to the first stage.
(5) Requirement specification of data flow machine and data base machine of the first stage.
(6) Design of gross architectures of both machines by selecting a few candidates.
(7) Division of machines into several components and their conceptual design.

(8) Estimation of the performance of both machines.
(9) Clarification of mutual relation between data flow machine and data base machine.
(10) Software system design for the evaluation and test of the machines.
(11) Estimation of the manpower and the amount of money which will be needed to develope the experimental machines.

As for data flow machines, feasibility study is the main theme of the first stage of the project. So, what kinds of research theme should be listed up for it is considered at the step (2) through (5). We are going to design the first experimental data flow machine which supports functional programming. For data base machine, we have already many experimental machines in the world. Therefore the step (1) should be stressed to design the next generation data base machines, and the special characteristics which is required particularly for the KIPS applications should be clarified. We are going to design the first experimental data base machine which supports relational algebra as the basic operations.

## 4. Research of a data flow machine

### 4.1 Study items of data flow machines

Followings are the study items of data flow machines, that should be investigated prior to the real design. The items are grouped largely into 3 parts, hardware related items, software related items and system development related items.

[a] Hardware related items

(1) System configuration

First of all, the general structure of data flow machine should be clarified. What kinds of components should be identified and how they should be interconnected are the primary subjects. Up to this time, we identified five major components: activity memory, activity controller, processing elements, structure memory and interconnecting networks. The activity memory holds the data flow graph and the data which flow on the graph. In order to provide a very large memory space for this activity memory, we think that it will be necessary to have some 'virtual memory' mechanism and some 'cache memory' which holds the 'working set' of the data flow graph. The activity controller detects the fired activities, makes up the operations which are ready to be executed and sends them to processing elements. Processing elements receive the operations from the activity controlller, process them by accessing the structure memory and return the results to the activity memory. The structure memory holds the structure data and executes the basic operations for the structures. It shold be general enough to support all kinds of structures. As many numbers of processing elements access to the structure memory simultaneously, the memory should have very large bandwidth through dividing it into many memory banks. To make the structure sharing possible, some reference count mechanism or copying feature should be supported. The garbage controller will be one of the important components to realize the structure memory, as well.

We identified 4 kinds of interconnecting metworks. First is the arbitration network to connect the activity memory to processing elements. Second is the distribution network to connect the processing elements to the activity memory. Third is the mutual interconnecting network of processing elements. Fourth is the internal network of each processing element which will contain several operation units. These networks should be designed to make the data passing very rapid among many activity memories and structure memory. Fig.1 shows an example of a configuration of data flow machine.

(2) Control mechanism

There are two ways of activity control principle. One is the data-driven system and the other is the demand-driven system(Fig.2). To execute some compiled code of data flow languages, the former will be used. When some interpreter is used to interprete a program, the latter is used usually. At the first stage of the program interpretation using demand-driven control, requests of a operation(say 'A') is issued. At the second stage, some result data are returned from operation 'A' to the previous operation(say 'B'). As operation 'B' is drived by this resulted data, this stage of execution can be said to be data-driven.

Besides the principal method above, another efficient control mechanism such as firable node detection, lazy evaluation, processor allocation, and token labelling are required. The detection mechanism of fired activities is very important to make the throughput very high. There are several alternatives such as associative memory method, hashing method and so on. We can introduce the concept of lazy evaluation to improve the parallelism by permitting the partial start of execution of functions before getting all the operands needed. The allocation mechanism of firable activities to processing elements is also important to keep the utilization level of processing elements high. One method is to use the unit of procedure-code as the allocation unit for each processing element. Accordingly, the tree structure of procedure call reflects on the loading pattern of program code. To control iterative, recursive and pipeline operations, tokens which flow on the data flow graph should be identified uniquely by some identifier (colored token). When tokens are dynamically generated, this token labelling can be serious overhead and may need a large name space for the identifire. Some hardware support will be necessary to cope with it.

(3) Basic operations

The basic operation set should be clarified as the machine language of data flow machine. It will contain system control instructions such as start/stop, structured data manipulation instructions, interative , recursive or conditional control instructions, and arithmetic and logical instructions. We think it had better to include a few instructions that are unification oriented for the purpose of logic programming. The representation format of activity is also a design item. Fig.3 shows two types of formats. a) shows a mixed format of program code and token data. b) shows a separate format in which token data is stored separately from program code. We can use the unit of procedure-processing as the unit of data-driven control. In this case, the execution of each procedure is left free, so that it can run also in sequential style of the conventional von Neumann machines and the procedure can be regarded as a basic operation.

All the items listed above is concerned with the internal structure of processing elements.

(4) Input/Output mechanism

One of the main feature of data flow machine is its fitness for the functional programming. Each operation can be described independently of other operations, and interacts each other only through the availability of the operand data. Input and Output operations may break this functionality. The concept of these operations had better be handled smartly preserving this functionality. For example, one means is to use the concept of 'stream'. Though file is a very important concept in conventional machines, it has never been treated in the research of data flow machines. Some hardware mechanism such as channel that handles the stream should also be considered for the processing elements, structure memory and activity memory.

(5) Implementation technology

The structure of data flow machines should have the full expandability in terms of its performance and capacity. As the machine of the stage II and III will be implemented by VLSI technology, the modulality of the structure should be considered from the beginning of its research. System reliability is also an indispensable item. Retry mechanism for data flow operations, error isolation mechanism, reallocation of activity and so on are the major study items for this theme.

[B] Software related items

(1) High level languages

To develop a data flow machine, we need some data flow oriented high level languages. These languages will be functional type, that has a great merit of making the programming easy in the sense that operations have no history sensitivity. But, this is not all that we should consider for the characteristics of programming languages. How the programming languages that is designed from the programmers point of view can be compiled into very efficient codes in terms of data flow graph is the other important theme which should be solved in future. This can be the key point that decides whether data flow machines can be general purpose systems of wide use or not. Some basic language constructs such as if-then-else, repeat-until and while-do have been investigated. But, other items such as scope rule of variables, separate compilation, task generation, data type, data abstraction and so on are left uninvestigated.

(2) Operating systems

What kinds of operating systems should be designed for data flow machine is entirely unknown at present. Whether each function of operating system implemented on the present conventional machine should be implemented on the operating system for data flow machine or not, and how it can be implemented if it should be, will be clarified as the research progresses. Especially, some history sensitive processing will be required for file handling and data base utilization. Nondeterministic processing will also be requested, because nondeterminicity is essential for resource management that is a major function of operating system. What are the corresponding concepts to job, job abortion, process, interruption, multiprogramming, program status word, and so on ? How can all status information of the machine be dumped out at the check points for the recovery ? What makes the problem complicated seems to be the fact that many processing elements of the data flow machine run asynchronously. But, we think this problem is not so difficult essentially. For example, in order to stop the machine at some point, it is all to be done that the allocation controller of activity (even if there are many controllers, they can stop independently) is forced to stop.

(3) Activity allocation

There are two problems regarding activity allocation. One is the activity allocation to memory. The other is to processing elements. As we will have to divide the activity memory into many units primarily for reasons of throughput improvement, we will face the problem of allocation of data flow subgraph to each unit. This allocation strategy has influence upon the quantity of inter-module communication and the utilization factor of each processing element. Accordingly, this should be done so as to maximize the processing efficiency. A very simple algorithm is to handle each procedure as the allocation unit and allocate procedures called by a procedure to the near modules in the sense of communication overhead. This allocation is performed at the time of program loading, so it is static. On the other hand, allocation of activities to processing elements is dynamic and should be scheduled with the intra/inter communication overhead taken into account.

(4) Parallel algorithm

Data flow machine can realize the maximum parallelism expressed in the program. However, it cannot do beyond the maximum. This maximum depends on, not only the description style of the program, but the parallel algorithm itself. The former is related to the development of programming languages. The latter to the characteristics of the problem which is going to be described in some language. Inference applications are guessed to have very large parallelism. However, this precise analysis is left to the future research.

[C] Items for system development

(1) System description and simulation languages

To fix all design parameters and evaluate the performance of the experimental system, it is necessary to describe the system formally and simulate it with various parameters. Accordingly, we need some system description languages and simulation languages, which are suitable to describe asynchronous parallel operations.

(2) Debug and maintenance facility

We need some debugging aid to ease the debugging of the complicated asynchronous multiprocessor, such as hardware monitor and software conventions. Powerful service processor will also be needed to get the status of the experimental machine. The system maintenance method should be re-examined to match with the new architecture as well.

(3) Connection to other machines

At the first stage of the development, some host machine will be needed to supplement the function of the experimental data flow machine. Software development, compilation, linking, loading and so on will be executed on the host machine. So, input/output interfaces of the machine should be general to support these facility.

(4) Matching to conventional machines

When the computer system shifts from the conventional von Neumann machine to the new data flow machine, program mobility can be a very important factor to keep alive the enormous amount of software property. We will be requested to develop the practical way of this shifting.

4.2 Allocation of the items to 3 stages

As the research of data flow

machine is a system development, we cannot ignore any items listed above. However, all items cannot be solved simultaneously. We assigned to each stage the items which should be put stress to be investegated, as follows.

(1) first stage
The objective of the first stage is to show the basic feasibility of data flow machine and give the basic structure for it. All the items except the ones allocated to stage II and III should be investegated parallel. As for system configuration, the basic structure should be investegated at stage I, but the extensions such as virtual memory support is postponed to the stage II and later. Regarding control mechanism, we are going to develop hardware support mechanism for firable node detection, lazy evaluation, processor allocation, token labeling and so on. As for input/output mechanism, only the fundamental input/output interface is implemented on the machine and the full function will be developed at the stage II. Accordingly, the file concept will not be implemented during the stage I. The implementation technology of first stage is based on the present technology, though the modularity of system configuration should be considered for the later development. The high level language which will be used for stage I machine is based on the languages which had been proposed for data flow machine and functional programming. Only the minimum function will be implemented for the requirements of operating system. As for parallel algorithm, ever-continuing research efforts is needed. Software simulation will be needed to fix parameters at stage I, so we will develop some system description and simulation languages that are very good to describe asynchronous behaviour. The item of matching to conventional machines will not be necessary to be taken into account at the stage I.

(2) Second stage
The objective of the second stage is to test the implementation technology of VLSI and to develop the practical technologies of data flow machines. The hardware will be implemented by VLSI. Secondary memory concept will be realized in very natural way. The reliability issue and operating system design should also be investigated. As for high level languages, a few new languages will be developed for general purpose and for the inference processing. Personal data flow machine will be developed at stage II as well. On the stage II machine, various experiments will be done to test the effects of parallel execution of inference applications.

(3) Third stage
The real operating system will be implemented on the third stage machine. This machine will not only realize the connection between data base system and data flow machine, but be implemented with large scale to show the feasibility of large parallelism. This machine will be the basis of the KIPS.

4.3 Requirement specification of the stage I data flow machine

Followings are the requirement specification of stage I machine, that is set up at present.

- Universal functional unit with microprogram control (writable control storage).
- Hardware support for inference operations such as unification.
- Non-numerical processing oriented operation set such as character handling and unification for inference opration.
- 4 functional units for a processing element with an activity allocator.
- 16 processing elements per a system 4 sets of switching networks ( 16 x 16 packet switch).
- Activity memory of 16 banks, each of which capacity is 1 MB (16 MB in total).
- 4 M words of structure memory, with 10 ~ 16 bytes for a word that corresponds to an elementary cell for general structures.
- Parallel garbage collector for each structure memory bank.
- Data flow oriented high level language with the compiler.
- One processing element acts as an input/output controller.

5. Research of a data base machine

5.1 Study items of data base machine

(1) Specification of functions in data base machine and its host interface
We can design two types of data base machines. One is a stand-alone system that offers data base services to users who are connected locally or through communication lines. The

other is a host attached functional unit that is specialized in handling the data base system functions. Former can be thought to be made of two units, a host function handling unit and a data base function handling unit. This latter unit is the same as the host attached functional unit conceptually. We call this unit a data base machine. The data base machine should be equipped with a well defined interface so that the functions allocated to this machine are well isolated from the host and fully general to be used for any kind of applications. What kinds of functions of data base management system should be assigned to data base machine ? We can distinguish several functions such as data manipulation, data compression and encoding, secondary memory management, mass memory control, query analysis and optimization, integrity control, concurrency control, security control, and recovery control. As we have a plan to use this machine as the basis of knowledge base system, this functional separation is one of the important research items.

(2) Processing algorithm

The relational model that is based on a fine theoretical background is so general that any kinds of data can be expressed by the model. Accordingly, the data base machine that supports relational model is a good target for our objectives. Whether the navigational processing can be supported or not is a study item to be clarified in future, because other data model support may be required. Though it is also a study item to define a set of premitive operations, the first good candidate is the set oriented operations such as ones defined in relational algebra (union, intersection, defference, Cartesian product, selection, projection, join and division). The processing algorithm of these operations is a study item as query processing. However, so far as update operations are concerned, no good set operations have been proposed. When we want to update a shared database with real time, we are forced to face a very diffcult problem of concurrency control. We need some smart locking algorithm for it to preserve data base integrity. At the implementation of these operations, it will be necessary to manage the intermediate result relations that are produced under the execution of operations.

(3) Hardware architecture

Data base processing algorithms are realized by the corporation among software, firmware and hardware. Division of functions among these is an important disign item which trades performance and flexibility with cost. Hardware architecture is a mean to realize the parallelism of processing. We can introduce parallelism in the three forms. First is the parallelism of using many processing elements and memory modules that are connected each other through connecting circuits such as packet switches. Second is the internal parallelism of a processing element that can be made of many functional units such as sorter. Third is the internal parallelism of a memory module made of many memory cells and several functional units such as search modules. First one is closely related to the concurrent multi-user support and the total system throughput. Second one is the main factor to limit the speed of a single primitive operation, and is realized by parallel processing and pipeline processing among many units. Third one acts substantially to reduce the data volume that flows within the network of processing elements. The configuration of the hardware system and the control mechanism is an major study item of data base machine.

(4) Large capacity handling feature

Up to this time, there proposed many data base machine architectures. But, almost all of them don't support large capacity of data that exceeds several hudreds mega bytes. As the coming data base system will be of very large size, it is important to have a large capacity handling mechanism. Examples are the pagenation that supports working set concept for data base, and the staging network that loads necessary relations to working memory modules from mass memory such as mass storage system.

(5) Auxiliary functions

Multiuser support function that permits simultaneous execution of many user queries and/or updates is essential to improve the system throuput sufficiently enough to be used as the knowledge base system for inference machines. Though conventional data base systems can handle the formatted data only, it may be required to handle unformatted data for the data base system to be used as the knowledge base. Some encoding mechanism may be included in the architecture. Security and integrity support functions are also important study items. As the mechanism to support these functions has not always been clarified perfectly, it will be required to investigate the functions of data base management system and to establish the methodology prior to the design of an experimental system of full scale (the machine of the stage II). These functions may require some hardware support mechanism.

(6) Distributed data base facility

The future data base systems will be connected each other by communication lines and make up a total integrated system. Accordingly, data base systems will not be stand-alone, but be mutually related. This distributed data base facility will be very important in future. The main function of the distributed data base is to integrate the many data base systems and to make up a large virtual data system that can be used as if it were a single system from users point of view. To achieve this objective, we can identify many difficult problems such as data model homogenization, realization of data-location-transparency, query processing that extends over many locations, concurrency control of update operations, commitment control against failures, and so on.

(7) Implementation technology

As the data base machine will be implemented by VLSI devices, the hardware structure had better be modular. For memory devices, we have RAM, magnetic bubble memory, CCD, magnetic disk, magnetic tape and optical disk at present. How to combine these devices and to make up memory hierarchy for data base processing is one of the design points. Device technology such as GaAs, HEMT and Josephson Junction device will also be related to this design.

(8) RASIS

As the system size grows, the problem of reliability becomes severer because of its great influence to the users in case of system failure. This is the same for data base machine. Because of its large capacity, it may require great overhead to collect the check point information and the update journal records. Moreover, the consistency maintenance among data will saddle us with very difficult problems especially for distributed data base systems. We need a smart methodology that enables us to limit the region of maintenance within a few systems for RASIS(Reliability, Availability, Serviceability, Integrity and Security).

5.2 Allocation of the items to 3 stages

We can divide the development steps of data base machines into 3 stages. At present, we allocate the each item listed above to the 3 stages as follows.

(1) First stage

The major objective of the first stage is to show the feasibility of a relational data base machine and to give the fundamental hardware architecture for it. We will take the relational algebra as the host interface, design the memory modules which are equipped internally with search mechanism and clarify the architecture of processing elements which contain some special functional units such as sorter. We should investigate the mechanism of large capacity handling feature through pagenation or staging from some mass memory to the working memory that corresponds to the secondary memory such as magnetic disk. We think that the working memory will be made of magnetic bubble memory or MOS RAM in place of magnetic disk. RAS support is also a distinguishable study item that should be treated all over the three stages. Though the distributed data base support is very important, its research will be executed independently of the data base machine development at this stage, because the reseach is essentially a software development. Multi-user support feature will also be considered and have influence upon the architecture design. Together with the development of an experimental machine, the research of data base system itself should be done by implementing many kinds of auxiliary functions on conventional computers to develop the methodology to maintain integrity and security at the multi-user environment. Fig.4 shows a general configuration of a data base machine.

(2) Second stage

A machine of the second stage will be designed to show the high performance of a large scale data base machine and be made using VLSI chips. A number of memory modules and processing elements will be connected through very high speed connecting networks. Handling mechanisms of unformatted data will be designed and implemented on the machine. Various experiments will be done on this machine to test the technology of implementing knowledge base on the data base machine. The architecture of the machine will reflect the result of the software research of the first stage with regard to the data base management functions.

(3) Third stage

At stage III, the technology to realize high performance data base machine and to represent knowledge data on the data base machine will be joined to design a knowledge base oriented data base machine. The distributed data base technology will also be implemented for the stand-alone data base machine and grow as the distributed knowledge base system technology. At the time, the integrity maintenance technology will play an important role in consistency support of knowledge data. The machine of the stage III will be used to make up the knowledge information processing system as the fundamental mechanism of knowledge base system.

## 5.3 Requirement specification of the stage I data base machine

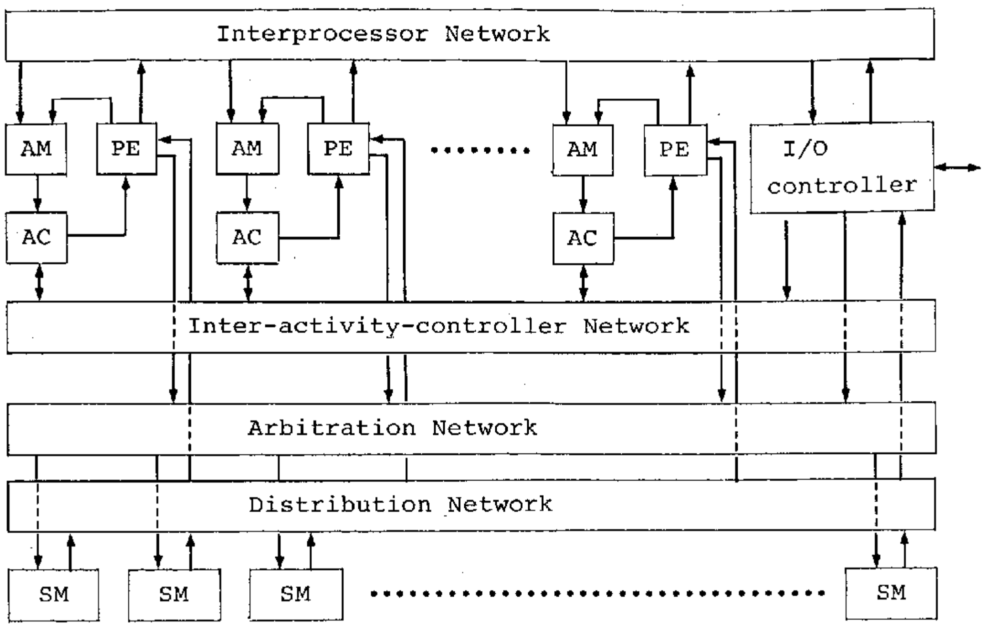
Followings are the requirement specification of the first experimental data base machine, that is imagined at present.

- Relational algebra language support as the host interface.
- Memory modules (working memory), each of which size is 4 MB and the number of modules is 32 (128 MB in total), with search and format transformation mechanism included.
- Processing elements: each element has a local memory of 256kB and special hardwares for set operations. 16 elements in total.
- 32X16 switching network: high speed packet switching network
- Mass memory staging mechanism that supports 20 GB of data with the data transfer rate of 50 MB/sec.
- Performance: 10 times faster than the processing on the conventional large scale machine.

## 6. Conclusion

The preliminary research of data flow machine and data base machine started in July of this year. As it passed only one and a half months since the beginning (this paper is written in August), the content of this paper is only the preliminary results of this 'preliminary research'. Accordingly, it includes some personal view regarding the schedule of the 10 years plan and the specification of the first experimental machines. It may be changed as the research progresses, and will be fixed at the end of this fiscal year (February 1982).

In this paper, we identified the study items of the data flow machine and the data base machine, and showed an example of the 10 years schedule in terms of the items. During the first three years, an experimental data flow machine will be developed so as to show the feasibility and the basic structure. Regarding the data base machine, an experimental system will also be developed to investigate the data operation mechanisms that allow great parallelism by expanding the set operations on many functional units. We hope that this preliminary research will give an explicit and concrete plan that will lead to a promising result.




AM : activity memory module
AC : activity controller
PE : processing element
SM : structure memory module

Fig.1 A configuration of a data flow machine

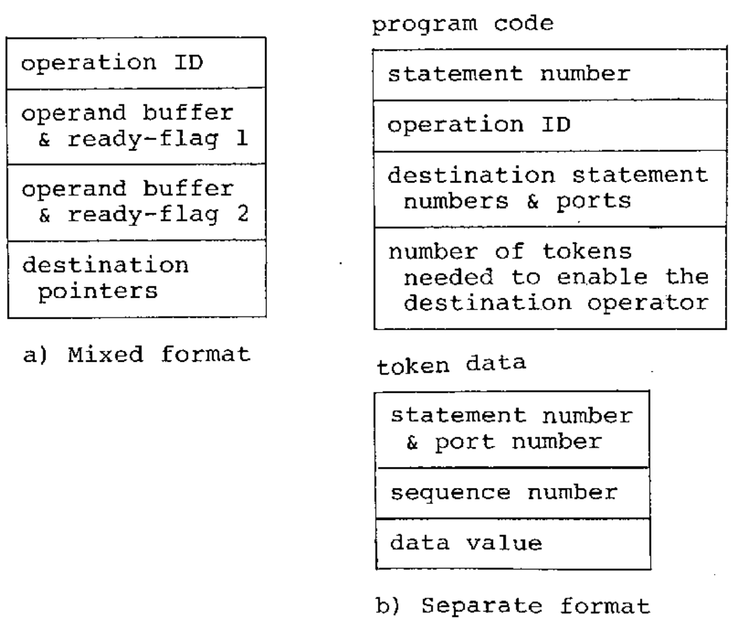



Fig.3 An example of the program and data format

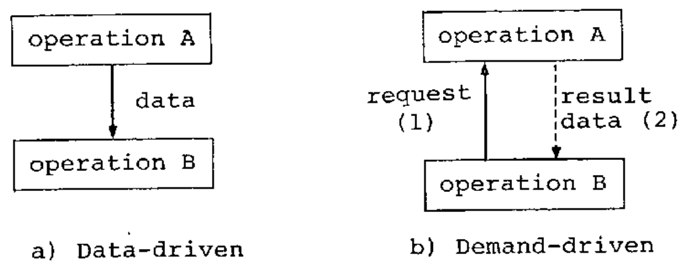
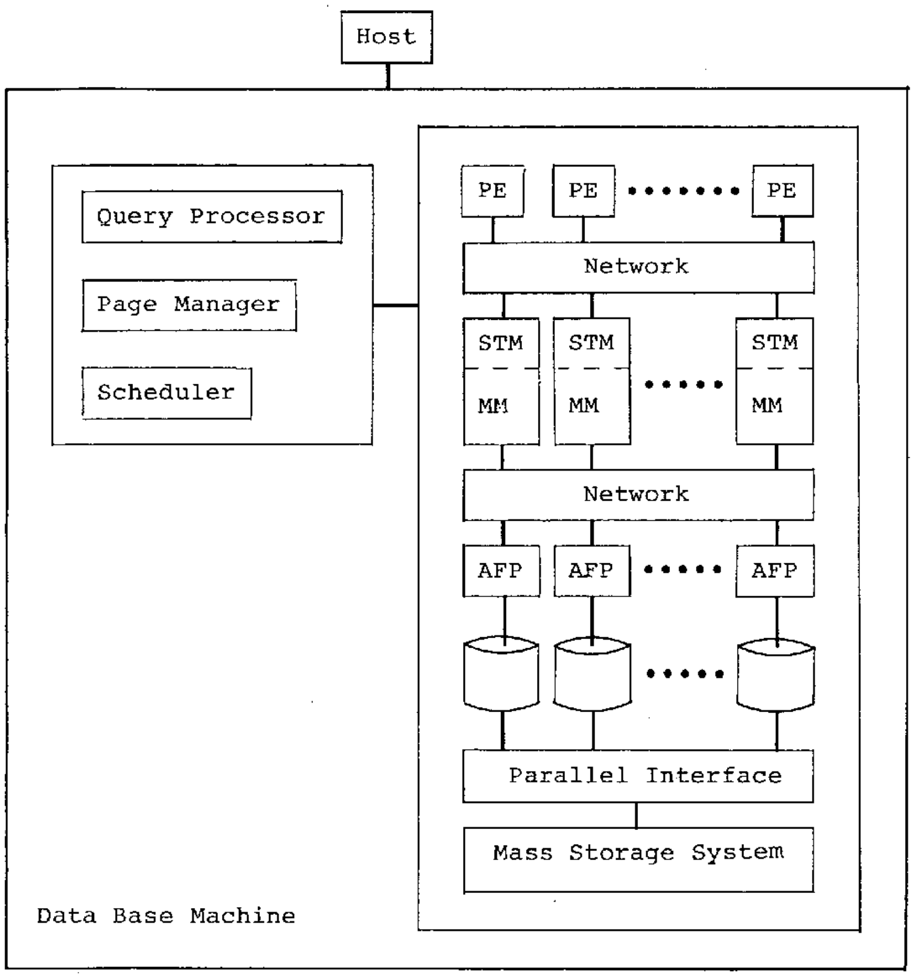



Fig.2 The 2 ways of activity control principle




PE : processing element
STM : search & format transformation mechanism
MM : memory module
AFP : associative file processor

Fig.4 A configuration of a data base machine

# V INVITED LECTURES
## —Knowledge Information Processing—

# INNOVATION AND SYMBOL MANIPULATION IN FIFTH GENERATION COMPUTER SYSTEMS

Edward A. Feigenbaum
Computer Science Department
Stanford University
Stanford, California, USA 94305

Fifth Generation Computer Systems, if successfully developed, will be excellent vehicles for Expert Systems applications. Fundamental is a software methodology known as Knowledge Engineering. Knowledge, not inference, is the key to high levels of performance of Expert Systems. A considerable variety of Expert Systems applications, most having great potential payoff, are already being worked on. Scientific innovations in knowledge acquisition will be required. Innovations leading to an efficient industrial technology for Knowledge Engineering will also be necessary. Managers must exhibit imagination, and the willingness to take technological risks to realize the enormous potential benefit from Fifth Generation systems.

What is the nature of a so-called "fifth generation computer system", and what are the technical and "social" innovations necessary to achieve such systems? In this paper, I will present briefly my views that:

a. That no contraint evident in the "hardware" world can limit the success of Fifth Generation systems.

b. That the critical problems of Fifth Generation systems are software problems.

c. That there are some critical scientific developments necessary to realize the software goals of Fifth Generation systems.
Solutions can be forthcoming but they will require significant levels of innovation.

d. That, equally important, there are considerations of training, skill, and work patterns of software experts for which innovation will be essential if Fifth Generation systems are to mature.

e. Finally, that there must be a definite commitment to innovation in the entire research and development process leading to the Fifth Generation computer system, because we do not yet know how to build useful systems of this type. In other words, there is a major research component to the Fifth Generation research-and-development process.

The material given in the remainder of this printed version of my talk will summarize and recall the major points of my oral presentation. The text is identically that of the visual presentation materials shown during the talk.

**1. For Fifth Generation Computer Systems:**

- Software, not hardware, is the issue
- Knowledge, not logic, is the issue
- For software and knowledge acquisition, some major scientific innovations will be needed
- Innovations in the training and technical management of software engineers will also be needed

**2. Fifth Generation Computer Systems:**

- Will be primarily symbolic manipulation systems
- Will be knowledge processors with arithmetic capabilities
- Will meet major commercial demand of the period 1990 to 2000 for personal and professional Expert Systems

**3. Hardware is not the issue. Software is.**

- Hardware engineers are concept-limited, not component-limited

- They ask: "Now that we can put essentially anything on a chip, what ideas are worth putting on a chip?"
- Hardware constraints pose no fundamental problems for Fifth Generation Systems. Necessary developments will arrive almost "automatically"", on time
- Much of today's software will appear on the chip as hardware.
- Other software will appear as instructions in fast microinstruction memory.
- The software ideas of today are the seeds of the big ideas for the Fifth Generation computer systems.

**4. Software Engineers Are the Critical Resource, Not Hardware Engineers**

- This is a commonplace view among most computer scientists
- It is radical view for most planners and managers
- The 1970s were the years of great hardware ideas
- the 1980s are the years of transition
- the 1990s will be the years of great software ideas
- These software ideas will completely transform the concept of "computing"

**The Critical Resource Among**

**Software Engineers:**

KNOWLEDGE ENGINEERS

5. What is Knowledge Engineering?

- It is the applied end of Artificial Intelligence, the science whose goal is to program intellectual functions.
- Knowledge Engineering focuses on building Expert Systems
- Expert Systems are programs that achieve high levels of performance on work that demands expert-level competence

**6. The Basic Idea of Expert Systems: Putting Knowledge to Work**

- The non-mathematical knowledge used for most of the world's problems
- Knowledge, not numbers!

**7. Expert System Application Areas In 1981**

- Medical diagnosis and therapy
- Equipment failure diagnosis
- Computer configuration
- Chemical data interpretation and structure elucidation
- Experiment planning
- Speech and image understanding
- Signal interpretation for medicine, oil field services, military needs
- Mineral exploration
- Military threat assessment and targeting
- Crisis management
- Advising about computer system use
- Training/teaching
- VLSI design and layout
- Air traffic control

**8. Knowledge Base is Not a Data Base**

Knowledge consists of:

- facts
- assumptions and beliefs
- heuristic rules

**9. For Expert Systems, Logic Is Not the Issue. Knowledge Is.**

- An Expert System, of course, needs an inference procedure.
- The power of an Expert System comes mostly from its knowledge, not its inference procedure.
- Even simple inference procedures are powerful enough.
- Simple inference procedures have the virtue of clarity and transparency to the user. This builds user confidence in system's reasoning.
- The big issues concern knowledge–how to acquire it from experts or nature, and how to represent it.

## 10. The Basic Components of an Expert System

- A problem statement with a goal
- A knowledge base
- A selector that identifies relevant knowledge
- A strategy that determines which knowledge to apply in which order
- An inference procedure that applies the chosen knowledge
- A working memory for storing parts of the solution as they develop
- A friendly interface that helps the user

## 11. An Expert System: Tip of an Iceberg of Software

**A typical Expert System consists of:**

- Expert knowledge specific to the area
- A Consultation subsystem
- A knowledge acquisition subsystem
- An explanation subsystem
- Knowledge base management facilities
- An inference subsystem

## 12. Software Tools for Building Expert Systems

The use of an appropriate software tool can cut the time and cost of building an Expert System by a factor of ten.

Some current tools are:

- EMYCIN (Essential MYCIN)
- ROSIE (Rule-oriented system for implementing expertise)
- UNITS EDITOR
- AGE
- EXPERT
- KL-ONE (Knowledge Language- one)
- RLL (Representation Language Language)

## 13. Market Opportunities For Expert Systems

- Home
- Professional
- Technician
- Military
- Circuit Design
- Signal Interpretation
- Intelligent Agents
- Office automation
- Intelligent computer-aided instruction
- Investment and finance
- Process control (managing systems)
- Software production and maintenance

## 14. Knowledge Engineering in Perspective

- The power is in the knowledge.
- Acquiring, codifying, and refining expert knowledge is the heart of the matter.
- Knowledge + Inference yields performance.
- Some excellent software tools exist that drastically reduce costs and speed development.
- Managers must grow to harness this new technology. They must develop new concepts of what is possible, and new concepts of how systems are built.
- The Expert Systems area is THE area of maximum growth potential for computer applications.
- Fifth Generation Computer Systems, as conceived in the Preliminary Planning Reports, will have a major impact.

## 15. Building Resources: Software Engineers and Knowledge Engineers

- More university-level training is desirable.
- Working in software and knowledge engineering should be treated as an activity of high prestige among engineers.
- Industry-University cooperation: in California's Silicon Valley, we have very fruitful collaborations between Stanford University computer scientists and engineers and the high-technology industries.

## 16. The Major Area of Scientific Innovation Needed for Fifth Generation Computer Systems

- Because knowledge is the key to expert performance, knowledge acquisition methods are of the greatest importance.
- Systematic, machine-aided knowledge acquisition methods are not well developed.
- Without better methods, widespread application of Knowledge Engineering will be impractical, and the effort to design Fifth Generation Computer Systems will be largely wasted.
- Research should focus on such problems as automatic theory formation, truth-maintenance procedures, interactive transfer of expertise, reasoning by analogy, and the understanding of technical text.

### 17. Innovation in Methodology

- Personal, high-speed, large-memory workstations for software engineers
- Excellent "programmer environments" for accelerating programming
- Build-and-improve incremental strategy
- "Throwaway" code

### 18. Innovation in Industrial Technology for Fifth Generation Systems

- Knowledge Engineering has a stable and effective laboratory technology.
- There does not exist an industrial technology and methodology for large-scale, efficient application of the laboratory technology.
- Must be invented in the 1980s if Fifth Generation Systems are to be really useful.

### 19. Innovation in Fifth Generation Systems

- By taking some scientific and technical risks.
- By rewarding those who take prudent risks, even if their ideas fail.

The path to the Fifth Generation Systems must contain many risks, and some failures, but there is no other way.

## RESEARCH AND DEVELOPMENT

## STRATEGY FOR

## FIFTH GENERATION SYSTEMS

Significant international collaboration will be possible and useful in the development of Fifth Generation systems. Japanese computer technologists have only recently become active in the building of advanced software methodologies such as Knowledge Engineering. In other countries, there is more than 20 years of experience, insight, and accumulated software and experetise. Because the Fifth Generation project is so difficult and innovative, it will require as much "intellectual capital" as can be brought together for the purpose. A wise managerial strategy would arrange to bring together such scarce and valuable resources with the Japanese vision and engineering expertise that conceived the Fifth Generation systems. A truly international effort will be necessary, and a major international economic and social benefit will result.

REFERENCES


[1] Barr, A. and Feigenbaum, E.A., eds., *Handbook of Artificial Intelligence,* Vol. 1 (published) and Vols. 2 and 3 (in press), Wm. Kaufmann Inc., Los Altos, California, USA, 1981.

[2] Duda, R. and Gaschnig, J., "Knowledge Based Expert Systems Come of Age", *BYTE* magazine, September, 1981.

[3] Feigenbaum, E., "The Art of Artificial Intelligence–Themes and Case Studies of Knowledge Engineering", *Proceedings of the International Joint Conference on Artificial Intelligence, Vol. 2,* 1977.

[4] Yasaki,E., "AI Comes of Age", *DATAMATION,* October, 1980.

[5] Yasaki,E., "AI: More than a Science", *DATAMATION,* September, 1981.

[6] "Computers that outperform the experts", *The ECONOMIST,* September 26, 1981, p. 91.

LOGICAL PROGRAM SYNTHESIS

Wolfgang BIBEL
Technische Universität
München, Germany

Starting from an analysis of the reasons for the development of more flexible, more intelligent computer systems, some of their basic features are elaborated. Regarding programming as a paradigm for general problem solving, the paper focusses on the various aspects of a future program synthesis system. This discussion is combined with the description of the conceptual structure of a LOgical Program Synthesis system, called LOPS, where such aspects are emerging. Because of the prominent role in LOPS, particular attention is devoted to the theorem proving method underlying its deductive component. In conclusion, an assessment of our expectations is given, and some of the future technical requirements in computer architecture are mentioned.

## 1. The need for more flexible systems

Imagine that some person has written a book containing plenty of logical notations. It were typed now using one of the most advanced text processing systems available on the market. There are such systems with two printwheels working with marvellous accuracy and at a phantastic speed. Nevertheless the person would experience considerable disappointment since logical symbols, such as ∀, usually are not available on standard printwheels.

Of course, ∀ may obviously be composed by V and − ; the system in mind, however, is not flexible enough to realize such a composed symbol. More generally speaking, the controlling part of the system, the software, is unable to exhaust the potentials of its mechanical part, the hardware. Our point is that this kind of disproportion may be experienced with many systems in our world, not only computer systems.

For instance, there is now a train connecting Paris and Lyon at an average speed of 260 km/h, covering this distance of approximately 450 km in less than 2 hours. But say, your destination is St.Symphorien-s-Coise, a little village near Lyon at a distance of 45 km. Then you should be prepared, that you will need much more time for this last small portion of your journey than for the first, 10 times longer one. Again the weak part is the software in form of inflexible timetables, fares, and routes.

What we have in mind are systems which meet such needs in a flexible way. For example, a really smart text processing system, given the specification of ∀ as the composition of V and − , should be able to add such an extra feature to its performance without being reprogrammed by humans. Similarly, we believe that the performance of transportation systems could be improved considerably with intelligent control systems providing a flexible coordination of the varying needs.

These are just two examples demonstrating the need for more flexible systems. Let us mention a few more in order to convince the reader that this need is in fact ubiquitous. For instance, is it a very satisfactory way of life to spend several hours a day with typing texts written or spoken by others, as many secretaries do? Systems serving for the same purpose might be realized in the near future. Moreover there are many jobs that are even much worse under human aspects, and similarly might be taken over by systems.

Switching to a completely different subject, is there not a wide-spread feeling that the economic problems have become too complex to be manageable by the group of leading politicians? Intelligent systems might support the experts in hammering out the details of a more continuous economic strategy which is specified by the politicians in a descriptive way, that is, by defining political goals rather than incoherent procedural measures of detail.

Not only the leading politician, but also the average individual is faced with a world of increasing complexity. It is really incredible what an average citizen of an industrial country in principle is expected to know about law, education, business, government, administration, etcetera. The principle of "égalité" in this situation could be taken seriously only by developing smart information systems with public access which respond to the description of individual situations with appropriate information gathered by the system in an active way.

Last but not least we mention the potential impact, intelligent systems might have for science. For instance, it might be more realistic to involve a smart system in the development of a universal physical theory from the vast amount of data already gathered in billions of experiments rather than spending even more money for even bigger accelerators.

2. The fear of intelligent systems

Today, many people oppose the idea that there is a real need for more sophisticated technology. Rather they are in favour of an alternative style of life with less or "soft" technology in relatively small self-supporting social groups. Here, of course, is not the place to analyze and assess this movement. But it is interesting to note a relation with the theme of the previous section.

Does not part of our disdain for technology originate from the disproportion between the powerful machinery and its imprudent use? The development of more flexible computer systems might in fact further a better balance in this respect.

Moreover, it is certainly not helpful to keep the eyes closed in view of such facts as the rapidly growing world population, the existence of cities with millions of inhabitants, the preference of people for a good quality of life, man's desire for amplification of knowledge, but also the limitation in world resources such as clean water, air, and soil, energy and so on. The development of more intelligent systems might be a chance to bring all these facts to something which is a little closer to harmony than the actual prospects suggest.

Like any human tool, such systems also bear the potential for terrible misuse such as for despotic power or military actions. Keeping an eye on these dangers from the very beginning certainly will be a necessity. On the other side, however, they seem to offer an opportunity for a more liberal, more democratic, more ecological world.

3. Basic features

The prevailing attitude towards the development and use of computer systems still is the one inherited from the early computer days. It regards the use of such systems as a function of the available machinery. As a consequence, the users for instance have to learn to express themselves in relatively unnatural languages.

We believe that the adequate attitude today has to regard the machinery as a function of the needs which is quite a different perspective leading to different, that is, better results, even if the needs can be met only in a partial way. In the first section we have given examples of the kind where systems might be of a real help. In a first approach we may now ask what basic features must be offered by systems solving the respective tasks.

Incidentally, these tasks may be roughly divided into the following two categories. There are the *transformational* tasks which require some code to be transformed into some other code according to a fixed functional transformation scheme. Here, code is meant in a rather general sense, so that transformational tasks include the nasty jobs typically be carried out at an assembly line. The secretary's typing mentioned in section 1, of course, is of that category as well. Secondly, there are the *problem solving* tasks which call for functional solutions of defined problems. Typically, we think of problem solving tasks which are too complicated to be mastered appropriately by humans. For instance, a public transporttion control system would be faced with tasks of this category.

Whatever category we have in mind, it is clear from the examples that appropriate systems would have to deal with natural human utterances, that is with natural languages, perhaps even gestures, drawings or the like. Also it is desirable that these utterances may be expressed in media which are natural for man. Undoubtedly, speaking and writing are the most natural forms of utterances. Systems with these features could be used literally by all people with ease, perhaps an important condition for bringing forth better solutions for the world's problems. Note that even any "artificial natural language" (McCarthy), that is, the restricted part of natural language which can be processed by the actual technology, is preferable to any known programming language.

Programming is a special case of problem solving. For this, most current programming languages force the programmer into an algorithmic way of thinking. Surely, we do have the ability for such thinking. In some cases, it even seems to be the most natural approach to solving a problem since we do have a memory, a "feeling" for certain basic processes. In more complex problems, however, human problem solving follows a different path which is characterized by the additional involvement of a descriptive way of thinking. For the kind of problems, we have in mind, this approach therefore seems to have much more importance which implies that we need systems which support a descriptive problem analysis.

This requirement includes the possibility of referring to (pieces of) an algorithmic solution, already found, in corresponding descriptive terms while communicating with the system. It also includes the necessity observed in the examples of section 1, for allowing changes in (parts of) the problem descriptions thus providing the quality of flexibility.

In the remaining sections of this paper we will focus our interest particularly on some of the technical aspects of this feature of descriptive programming. It seems that programming to a high degree is representative for problem solving in general. Therefore we believe that the issues discussed in the following deal with one of the basic components of any intelligent system.

We conclude the general part of this paper by stating three selected general requirements for the potential use of the envisaged systems. From the very beginning, we should encounter a learning component for self-improvement of such systems. The system should be used in supporting its further development. Given the present state of the art in the respective field, the technical effect will probably be marginal in the beginning. But this provision may be crucial for providing kind of an evolutionary flexibility.

As we said before, these systems may enable man to let machines perform transformational tasks whenever this is preferable, and may enhance man's capability for problem solving. But I personally hate to think of such machines as "social partners", substituting humans in their individual, social, psychological, or creative roles.

Finally, it is hoped that eventually we all may share the potential advantages of such machines not only an élite of people or nations.

## 4. The role of logic in programming

As we said before, it should be possible to communicate with future systems in (artificial) natural language. The idea of such systems, however, is one thing, their development something different. With respect to the latter it is certainly a good idea, to treat separable problems independently. Natural language understanding and program construction are such two fairly independent problem areas in this connection. In order to separate their treatment, we are looking for languages substituting natural language in programming which come close to natural language with respect to their expressive power, and at the same time avoid its specific problems due to its vagueness, its redundancy, etcetera.

The logical languages serve for this purpose in an ideal way, since they combine rich expressive power and relative naturalness on one side with conciseness on the other. Their present role might thus be compared with that of assembler languages in the past, as the appropriate intermediate level of language between the high level language, which eventually will serve for the user, and the machine level which is inappropriate for the development of problem solutions. Similarly, as the use of assembler languages has declined, once efficient compilers for traditional high level languages were available, the use of logic will gradually be abandoned as soon as natural language understanding systems will perform in a satisfactory way.

In the family of logical languages full first order logic (fol) appears as the suitable candidate to begin with because of its intermediate position. On the one hand, its expressiveness is even too rich in many cases for which reason, for instance, the people in the field of logic programming prefer the use of *Horn clause form* a restricted part of fol. On the other hand, there are cases where it seems to be necessary to use richer logical languages such as higher-order logic, modal logic, and others. In the present situation and for the purpose of program synthesis, we prefer fol since clausal form conceals much of the natural structure and, in particular, the important role of quantifiers, while a possible extension to more general logics for certain cases may later be realized on the basis of the techniques achieved on the first-order level.

In contrast to any conventional programming language, logic in particular has the unique property to allow both, algorithmic and descriptive expressions, thus supporting both forms of human thinking mentioned in the previous section. People, who are only superficially familiar with fol, often overlook its algorithmic capability, present in the term language of fol. For instance, they might miss in fol the possibility to use an expression like

```
for i=1,...,n do x←x+1 ; z←x ;
```

Though it is true that standard logical textbooks do no introduce such expressions into fol, there would certainly nothing be wrong with such an extension. In this sense, the previous expression would simply be treated in fol exactly as the more standard first-order expression

$$z = \text{plus-one}^n(x)$$

The reason, why nobody has designed such an extended fol, lies in the fact that after some practice with the standard term language in fol people usually even prefer it to the equivalent way known from algorithmic programming languages, since it supports the *functional* form of algorithmic thinking which is presently en vogue also among the traditional programmers. Anyway, we take it as a fact that fol may be easily extended to include familiar notations not present in its standard form; yet, in its essence it would remain fol. Therefore we continue to speak of fol, even if we have in mind such a more comfortable version.

5. On problem specification

Here is certainly not the place to give any introduction neither to fol nor to its use for representing natural statements in a formalized way (see [Ko], [Ni], a.o.). Only a simple example may show the way of using fol as a specification language. Say, we want to specify the *maximum problem*, that is the problem of determining the maximal element m of any non-empty set S. In fol, this might read

$$\forall S\ \exists m\ (S \neq \emptyset \rightarrow m \in S \wedge S \leq m)$$

corresponding to the natural sentence: For any (set) S, which is not empty, we claim the existence of an m (to be determined) which is an element of S and is greater or equal than any element in S.

For better communication, let us refer to S as an *input variable* (which technically plays the role of a constant input), and to m as an *output variable*. The formula part, $S \neq \emptyset$, is called the *input condition*, and $m \in S \wedge S \leq m$ the *output condition*, consisting of the two *clauses* $m \in S$ and $S \leq m$ which here are simply *literals*.

In this special example, the whole specification consists of a single (conjunctive) part. In more complex problems there will be a number of such *conjunctive parts* (also called *clauses*), which may or may not share common variables, function symbols, and predicates. The specification of systems such as operating systems or nation-wide traffic control systems may even require thousands of such clauses. Such figures frighten people so much that they argue it might be easier to program the whole system right-away with traditional methods rather than to provide first such an immense specification. Arguments of that kind go wrong, however, simply because they are based on misleading conceptions concerning the nature of such specifications in their details and the potential role of a supporting program synthesis system.

For instance, the concept of specification does not exclude the case where parts of the specifying formula are given directly in algorithmic or functional form whenever this is adequate. As we said in the previous section, fol covers this case equally well. One of the problems with traditional software production is that the *whole* problem has to be brought into algorithmic form before it can be tested on the machine. In our approach, the "testing" phase may accompany the design phase from the very beginning, because descriptive features are taken into account. In fact, what we have in mind here with "testing" is much more than running executable code with given input. Even without any synthesizing capabilities, the following kind of support is possible in a future system.

Pieces of the specification, designed by the user, may automatically be combined with previously provided pieces in a logically correct way. Thus the idea of typing in a single huge logical formula is another one of such misconceptions mentioned above. In doing so, the machine may keep track of all occurring symbols, variables, constants, functions, predicates. Thus it may check, and eventually may tell the user, the (syntactic) extent to which, say, functions are already defined by the specification, provided so far, or by standard usage, stored in a knowledge base. This way "structured programming" could be realized with the comfort that the machine traces the chains of definitions through the various levels of specification, rather than the user.

It is amazing, how much valuable further information is contained in the pure syntactic string of symbols of a specification. Its appropriate use may ease the search for a solution, and in fact may lead to a better solution. To illustrate this point, let us consider the following trivial problem.

$$\forall a,b\ \exists x,y\ (x=a+1 \wedge y=b+1)$$

Logically, it is equivalent with

$$\forall a\ \exists x\ (x=a+1) \ \wedge\ \forall b\ \exists y\ (y=b+1)$$

thus separating the problem into two completely independent ones which thus may be executed in parallel (or in any sequence). What appears to be so trivial here, may amount to a real problem for the human programmer when it is to be carried out within a huge specification with the goal of obtaining (partially) independent pieces as small as possible, exactly the kind of problems where machines perform so much better in comparison with humans. In addition to applying logical rules of this kind, concerning the scope of quantities, the propositional structure, and the like, we may also think of the application of other rewrite rules for occurring functions, of the insertion of solutions for known subproblems, etcetera, the information for which is to be stored in advance in a supporting knowledge base.

We thus imagine, that the machine itself in this indicated way contributes to the specification of a problem, and we believe that this part already covers a lot of what is called programming. Although much has yet to be done in this direction, the conceptually more demanding part seems to be that discussed in the following section.

## 6. LOPS

Let us now assume that the problem specification is given. The remaining task consists of synthesizing it to a completely functional form which then may be compiled into hopefully efficient code. Note, however, that this assumption is made for reason of presentation only since in a future system the specification phase will overlap this synthesizing phase as briefly discussed at the end of section 8 below.

The problems arising in the automatic synthesis of algorithms from input-output specifications have been studied in a project, carried out by K.M. Hörnig, A. Müller, and the present author. In this project we are building such an automatic synthesis system, called LOPS for LOgical Program Synthesis. Currently it is able to synthesize only relatively trivial algorithms; but the conceptual approach is broad enough that in the near future the synthesis of more complicated algorithms may be expected.

In accordance with our restricted research goal, the support of the specification phase by LOPS is very limited. It only provides the comfort that the user may simply list the input and output variables, and the clauses of the input and output condition, which pieces are internally combined to a logical formula such as the one for the maximum problem in section 4.

The basic conceptual idea of LOPS centers around the well-known theoretical insight which may be illustrated by the following conceptual equation:

computing = proving.

For instance, recall the maximum problem from section 4. Obviously, a *constructive* proof of the formula presented there, for a given set S would also provide a value for m, the maximal element. This insight immediately leads to the idea to let a theorem prover compute the desired output by proving the specifying formula.

In fact, this idea has been realized in the form of PROLOG [CKC]. It required the specification presented in a very special form, however, so that the natural specifying formula for the maximum problem above, for instance, cannot efficiently be executed by a PROLOG interpreter. Hence this idea, with traditional, basically first-order theorem provers in mind, is too simple for the more general case. But even for the cases suitable for PROLOG there remains the important fact, that each execution of a PROLOG program may require considerable search for obtaining the result. This is certainly alright for the use as a research tool or for simple tasks given to a personal computer. For systems such as an operating system we definitly would expect a more efficient performance.

Therefore in LOPS we have tried to separate the proving part, to be done once possibly requiring a lot of search, from the executing part, to be efficiently performed arbitrarily many times with no more search involved. The global effect of LOPS for the case of the maximum problem may be roughly described in the following way. If

$$\forall S\ \exists m\ MAX(S,m)$$

abbreviates the specifying formula from section 4, then LOPS eventually generates a first-order term max(S) satisfying the following three properties.

(i) max(S) does not contain any variables other than S.

(ii) Any constant or function symbol in max(S) may be compiled into efficient code in a traditional way.

(iii) $\forall S\ \forall m\ (max(S)=m \rightarrow MAX(S,m))$ is a valid statement provided the specifying formula itself is valid.

Here valid more precisely means valid in any mathematical theory rich enough to include, say, elementary set theory with orderings.

It is clear that the goal of program synthesis has been achieved with the generation of such a term in a very satisfactory way, since for any execution with given S a compiler according to (ii) may produce the code which efficiently computes m.

The generation of such a term, like max(S), for arbitrary specifications is the demanding task which LOPS is designed to carry out. Of course, it will always be successful for a restricted number of specifications only. But the point is that its design is without any principal restrictions. Rather the actual restrictions are determined by the actual capabilities of its supporting components such as its knowledge base, its theorem prover, etcetera, which possibly may be improved in an evolutionary way. These components will be described further below.

This term generation in fact is carried out in a number of steps, so that actually LOPS generates a number of intermediate relations $MAX_i(S,m)$, $i=o,\dots,n$, where

$$MAX_o(S,m) \equiv MAX(S,m) \quad \text{and}$$

$$MAX_n(S,m) \equiv max(S)=m ,$$

and for which

$$\forall S \forall m \ (MAX_{i+1}(S,m) \rightarrow MAX_i(S,m))$$

is valid. Obviously, (iii) is then a logical consequence. Each step is triggered by a certain strategy of a rather general nature. These strategies have been described in detail in [B1], and will not be repeated here. We only want to indicate the kind of their nature.

For instance, in the maximum problem it is required to determine m such that both $m \in S$ and $S \leq m$ are satisfied. Obviously, for given S, the first condition alone may be satisfied immediately in a computationally efficient way, simply by selecting any element from S. With this possibility in mind, the problem may be reduced to determining m such that $S \leq m$ is satisfied only, based on the assumption that $m \in S$ is fulfilled already. Logically this means that an equivalent relation $MAX_1(S,m)$ is generated. Even this reduction step is not trivial, since in larger specification formulas combinatorially there may be many different such reductions. LOPS explores both, syntactic properties of the respective formula parts and semantical knowledge stored in a knowledge base to reduce the amount of required search for finding the adequate reduction. The details are determined by two strategies, called GUESS and DOMAIN [B1], realized by respective components in LOPS.

There are several more such strategies or components which all have in common that they take into account very carefully all information relevant for their application which may be found in the syntactic formula structure supported by domain-dependent knowledge. Further, each of these strategies is oriented towards achieving a certain subgoal characterized by certain criteria, where the resultant sequence of subgoals is guaranteed eventually to lead to the global goal of the synthesis.

This goal-oriented, strategical behaviour distinguishes LOPS from earlier deductive approaches based on theorem provers. On the other side, these subgoals are logical formulas; hence it would not be appropriate to regard LOPS as a transformational system studied in more traditional projects. The transformation from one subgoal to the next consists in a search for a deduction leading from one to the other, retaining validity as required for satisfying the condition (iii) above, while such transformational system roughly spoken, substitute certain program constructs by others, according to certain transformational rules.

## 7. The conceptual structure of LOPS

In [BH], we have provided some details on the structure of LOPS as a programming system which is implemented in UTLISP on a CYBER 175. For the purpose of the present paper we are more interested in its conceptual structure as a program synthesis system, since we believe that this structure is of relevance also for more general (or other special) problem solving systems. It is illustrated in figure 1.

The whole system is driven by the central control unit, CCU, which follows the strategies mentioned in the previous section.

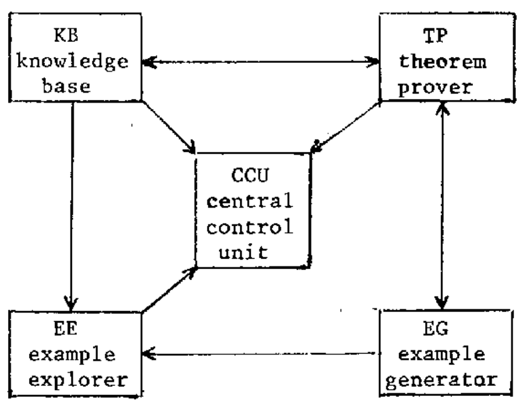



Figure 1. The conceptual structure of LOPS

The flow of control is not shown in the picture, since the arrows rather illustrate the flow of relevant information during the synthesis process.

The role of the knowledge base, KB, for CCU has already been mentioned in the last two sections. For instance, the knowledge, that $m \in S$ may be satisfied simply by selecting m from S, would have to be provided by KB, as we have mentioned there. KB would have to contain knowledge on the meaning of standard symbols such as $\in$, on basic functions, data structures, or previously synthesized algorithms which can be realized efficiently with the available compilers, further on rewrite rules, facts or theorems in the problem area, etcetera. The problem of acquiring, structuring, and storing all that knowledge for efficient use is certainly one of the central actual problems in building intelligent systems. It is mainly here where the evolutionary flexibility has to be located, called for in section 3. Also it is conceivable to envisage exchangeable expert knowledge bases for particular problem areas rather than a universal KB.

While the concept of a knowledge base certainly is familiar in this field, the use of an example generator, EG, and an example explorer, EE, is relatively new, although quite natural. Every human programmer studies small examples of relevance for his problem during the development of an algorithm, so why not a program synthesis system.

For example, in the synthesis of the maximum algorithm one might be faced - as LOPS does - with the case characterized by the clauses $m,m' \varepsilon S$, $S \leq m$, and $m' \neq m$ . Obviously, $m' \neq m$ here may equivalently be substituted by the stronger condition $m'<m$ . In this trivial example one might think of several possibilities of how the system might notice this. But for more complex cases we believe that the only feasible way is the generation and subsequent exploration of (small) examples. For instance, let $S=\{1,2,3\}$ with the natural ordering $\leq$ . Then $m=3$, and $m' \varepsilon \{1,2\}$ will satisfy these clauses, that is we have a model for them. For both cases, $m=1$ and $m'=2$, we, or the system, will notice $m'<m$ . The question, whether this is true in general for the actual situation, can be solved by the theorem prover, TP, which then allows the substitution just mentioned. The potential usefulness of knowledge from KB for this process, that is for both EE and TP should be obvious.

We have just met one of the possible applications of TP. But deductive reasoning, of course, is required all-over in the synthesis process. In this connection we should notice the well-known trade-off between knowledge and reasoning: the less we know the more reasoning is needed, and the weaker our capability for reasoning is the more we have to memorize. Therefore TP plays an important role also in keeping KB at a reasonable size.

For this importance of TP in this system, like in others, we will devote it an extra section. There we will briefly explain the fact that a TP in principle may be used as an EG which is the reason for the arrow pointing from TP to EG. Since the potential support of EG and of KB for TP is pretty obvious, we have thus touched on all the components and arrows in figure 1.

8. The connection method for TP

Traditional theorem provers are based on some logical calculus for fol, in particular on the resolution rule. A common feature of all these calculi is their enormous redundancy.

For instance, consider the obviously valid statement

$$\forall x\ (MANx \rightarrow MTLx) \wedge MANsocrates \rightarrow MTLsocrates$$

where MTL abbreviates MORTAL. A resolution proof for this theorem is shown in figure 2. As we see the formula has been transformed into a set of three clauses from which the empty clause $\emptyset$ is derived in two steps. Now, consider the situation which is given after having performed the first step. In

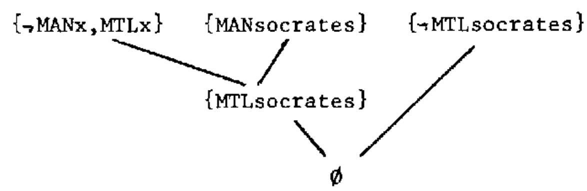



Figure 2. A resolution proof

general, it is *not* the case that the newly derived clause {MTLsocrates} replaces the two *parent* clauses {¬MANx,MTLx} and {MANsocrates}. Rather the search space now consists of all previous clauses plus the newly derived one. Hence in this example, the search space now consists of 4 rather than 3 clauses. Of course, this effect is marginal here, because the formula is so trivial. In complex problems, however, this effect may (and often does) add up to the generation of tens of thousands of clauses.

Unfortunately, these new clauses contain mostly redundant information. For instance there is nothing new in the clause {MTLsocrates} except for the new combination. This is quite different in the *connection method*, developed mainly by the present author [B4]. A proof in it, corresponding to the one in figure 2, looks as follows.

$$\forall x\ (MANx \rightarrow MTLx) \wedge MANsocrates \rightarrow MTLsocrates$$

Apparently, it consists of the given formula with two added structural elements, called *connections*. W.r.t. this formula these two connections satisfy a certain criterion which establishes the proof. The point is that locating such connections and testing for the criterion may efficiently be done by systematically exploring the structure of the formula. In particular, this process never has to take into consideration any part of the formula other than the given ones, which apparently eliminates the redundancy illustrated above without introducing any new one.

It eliminates two further disadvantages of resolution. As can be seen, the connection proof above retains the natural structure of the given theorem which is true in general. At any time of the process it is therefore possible that the user may easily interact and possibly guide the search for appropriate connections. Because of the thousands of potentially generated clauses, the idea of an interactive resolution prover is certainly an illusion.

Secondly, the resolution rule is a locally oriented rule providing no global guidance of the search, which again is the reason for substantial redundancies in resolution theorem provers. In contrast to that the proof criterion for the connection method is a global one, and thus has led to a global search procedure which, in comparison with resolution, eliminates such redundancy. In fact, we have shown in [B3] that any popular refinement of resolution may be simulated by the connection method with equal or (often) less steps. This together with the representational features mentioned above provides it with a promising advantage over traditional theorem proving methods.

A more detailed overview, describing to some extent the technique of the process and further features of the connection method, may be found in [B2] while a full treatment is contained in a forthcoming book [B4], which combines the results of a number of publications of the present author.

As we already mentioned in the previous section, TP may be used also in the generation of examples (or counterexemples). For instance, let us assume that we left out the condition MANsocrates from our present example leaving the formula $\forall x$ (MANx→MTLx) → MTLsocrates. Apparently, this is not a valid statement, that is, there are counterexamples for which it has the truth-value *false*, or, in other words, there are examples for which the negation of this formula has the value *true*. These examples may be extracted from an attempted but unsuccessful proof as we demonstrate now for the present case.

Assume the connection method in its attempt to locate connections starts with the literal MTLsocrates as the formula's main claim. Guided by the formula's pattern this literal is connected with MTLx which entails the implicit substitution of socrates for x. We thus have the following situation.

$$\forall x\ (\text{MANx}\rightarrow\overset{\frown}{\text{MTLx}})\ \rightarrow\ \text{MTLsocrates}$$

The proof criterion is not fulfilled at this point since there is an unsolved subgoal, which is MANx, with nothing else left over in the formula. Thus the proof fails.

Though, a counterexample may now be extracted from this situation by considering all literals on the agenda ("in the active path" according to the technical terminology) which are solved or unsolved subgoals. Here, these are MTLsocrates, and MANx, that is actually MANsocrates taking into account the previous substitution. Theory now implies that we obtain a counterexample by considering a single object, say Socrates, for the single constant socrates, and assigning the truth-value *false* to all these literals. In other words, the single object Socrates in our counterexample is not mortal and is no man. Obviously, the formula is in fact *false* under this interpretation, since its hypothesis trivially is *true* while its assertion is *false*.

Of course, in general this process is not as trivial as here; nevertheless, in principle, it is a generally applicable method for generating examples, which has been described in more details in [BH] together with another more "heuristic" approach.

As an aside, if we imagine that the missing hypothesis MANsocrates above has been left out by mistake then the unsolved subgoal MANx in fact would provide a strong hint to this oversight. This is a very important feature for the application to program synthesis since there in practice we always have to encounter incomplete specifications. That our synthesis approach also in other respects is particularly suited for dealing with such incomplete specifications has been demonstrated in [GG].

With these remarks we also refer to the point mentioned at the beginning of section 6, since they show that even in the course of the synthesizing process the problem specification may be subject to further extensions, thus mingling the specification phase with the synthesis phase.

## 9. State of the art and expectations

Given the fact that only about two man-years could be spent so far for the realization of LOPS, it is clear that the implementation itself is in a relatively rudimentary state. This, for instance, entails the fact that often the user has to play the roles of the supporting components KB, TP, EG, and EE. Incidentally, this user interaction should actually be regarded as one of the features of LOPS remaining to some extent also in the future, since the ease of such interaction is particularly supported by its logical design.

In concrete terms, CCU (developed by Hörnig) runs perfectly for simple examples such as the maximum problem. Work is in progress to cover algorithms of the level of Kruskal's spanning tree algorithm or the well-known linear string pattern matching algorithms (a challenging research area for humans in the seventies). Such a level, which is by far no more trivial, has not been reached by any of the known program synthesis systems (see [BGK] for a spectrum reflecting the general state of the art). That this level may in fact be reached by LOPS with relative ease has been demonstrated by a hand-simulation for these problems contained in [B1].

There are first running versions of TP (developed by Müller on a small computer from the Z80 family) and EG (Hörnig); but they are subject to several restrictions so that much further work has to be invested into them. In particular, expert theorem provers for enhanced performance in specialized areas so far are not available at all.

Knowledge base research is not in the focus of our current interest; therefore KB contains just enough information for running a few examples of demonstration. Finally, EE is still very much of an idea rather than a concrete component.

Based on these facts and on our experience so far, our expectations are of a mixed kind. On the one hand, we strongly believe that program synthesis systems, with a performance competing with that of humans, in fact can be built with the techniques currently available. Similarly for problem solving areas other than programming. On the other hand, there is no doubt about the immense efforts (simply in figures of man-years) which would have to be invested for achieving this goal.

We think that it is currently not possible to predict whether it will be feasible in the near future to implement a first generation of such systems which then already supports the implementation of the second one, and so on. Such evolutionary techniques perhaps could substantially reduce the amount of required work, for which reason we regard their exploration as particularly important for the coming years.

## 10. Computer architecture requirements

Having in mind goals of the kind discussed in this paper bears certainly also concrete consequences for the required programming environment such as the programming language, the hardware, even the whole computer architecture.

Among the existing programming languages, LISP has certainly proved as the most appropriate host language for symbolic computation. Its flexibility might be extended into the following two directions. Although numerical processing in LISP has improved remarkably in newer systems, it certainly could be further enhanced to meet the standard set by FORTRAN compilers. This way, LISP could become a realistic competitor with PASCAL or ADA, the latters offering little for our needs. In the opposite direction, we would like to see LISP embedded into a restricted logic environment such as PROLOG [RS] in order to allow a minimum of descriptive features.

The traditional machines all were built in view of numerical computation, and thus lack the features adequate for efficient processing of such an extended LISP. The newly developed LISP machines provide an important contribution into that direction, which hopefully is followed by others.

Important features of such machines in our context would be efficient handling of large data-bases, powerful associative capabilities, parallel capabilities for coping with the indeterministic features of logic, table mechanisms for look-up of more complex structures such as theorems or problem specifications, built-in algebraic manipulations and transformations for fast evaluation of recursive functions. For instance, some of these features are taken care of in [GI].

We even speculate that hardware technology might support the speed-up of running complex algorithms in a way exemplified for the special case of theorem proving as follows. As we mentioned in section 8, theorem proving by the connection method may be viewed as exploring the structure of the given formula along certain paths through it. The process very much reminds of a current finding its way through a complex circuit to reach a well-defined exit. Could one not realize this analogy by micro-programming the formula into an actual circuit to be tested at a high-speed? If this would be possible for theorem proving, then why not for other problem solving areas?

Acknowledgements. Many of the ideas expressed in this paper have been influenced by numerous discussions with my colleagues K.M.Hörnig and A. Müller. In particular, Hörnig has provided a substantial contribution to the contents of section 7. I express my thanks to them, but also to the Deutsche Forschungsgemeinschaft for supporting the LOPS project, to Dr. Marx from the Bundesministerium für Forschung und Technologie and P. Bauer from the Deutsche Botschaft in Japan for their recommendatory support, and, last but not least, to the Japan Information Processing Development Center for the invitation to this presentation.

References.

[B1] Bibel, W., Syntax-directed, semantics-supported program synthesis. Artif. Intell. *14* (1980) 243-261.

[B2] Bibel, W., Matings in matrices. GWAI-81 -German Worksh. on Artif.Intell.- (Siekmann, J., ed.), Informatik-Fachberichte, *47*, Springer Verlag (1981) 171-187.

[B3] Bibel, W., A comparative study of several proof procedures. Artif. Intell. (1981), to appear.

[B4] Bibel, W., Automated theorem proving. Vieweg Verlag, Wiesbaden (1982), forthcoming.

[BH] Bibel, W., Hörnig, K.M., LOPS - A system based on a strategical approach to program synthesis. In [BGK], to appear.

[BGK] Biermann, A., Guiho, G., Kodratoff, I. (eds.), Automatic program construction techniques. MacMillan P.C., forthcoming.

[CKC] Colmerauer, A., Kanoni, H., van Caneghem, Last steps toward an ultimate PROLOG. Proc. 7th IJCAI, Vancouver (1981) 947 f.

[GI] Goto, E., (et al.), FLATS , A machine for numerical, symbolic and associative computing. Proc. 6th IJCAI, Tokyo (1979) 1058-1066.

[Gr] Green, C., A summary of the PSI program synthesis system, 5th IJCAI, Cambridge (1977), 380.

[GG] Guiho, G., Gresse, C., Program synthesis from incomplete specifications. Proc. 5th Conf. Automated Deduction (Bibel, W., Kowalski, R., eds.), Springer, LNCS *87* (1980) 53-62.

[Ko] Kowalski, R., Logic for problem solving. North Holland P.C., New York (1979).

[Ni] Nilsson, N., Principles of artificial intelligence. Tioga, Palo Alto (1979).

[RS] Robinson, J.A., Silbert, E.E., Loglisp - An alternative to Prolog. Report Syracuse University (1981).

Typescript: S. Tänzer

THE SCOPE OF SYMBOLIC COMPUTATION

Gilles KAHN
INRIA
Rocquencourt, France

ABSTRACT

This paper argues that the applications of symbolic computation will extend far beyond their current scope. By an analysis of several successful research projects, we try to identify future research areas and their implications on computer architecture.

## I - INTRODUCTION

It is a rare occasion where a scientist has to report not on some specific advances or technical results, but on his outlook on the future. In this paper I will try to explain a point of view that has gradually emerged in the course of the last seven years of research at INRIA. This presentation reflects my personal views and biases, but in its broad directions, I believe that my point of view is shared with many of my colleagues at INRIA and also in other research institutions. The topics of research that we have worked on at INRIA cover a wide spectrum : Semantics of programming languages, theorem proving, analysis of algorithms, programming language design, foundations of software engineering, semantics of parallelism, interactive tools for program design. More recently, we have also moved in the direction of design of VLSI circuits and computer preparation of technical documents.

A unifying point of view links all these seemingly unrelated research topics : the need to reason about programs -or programming languages, or circuits- as formal objects, and consequently the need to develop computer tools to assist in resasoning about such objects.

Speaking still in very general terms, we wish to consider a program -or a programming language or a circuit - as a structured collection of formulae in some well-defined algebraic formalism. In fact, for objects like circuits, it may even be necessary to use several related formalisms to represent the same object. If this view is adopted, then the computer tools needed appear clearly to revolve around first symbolic computation and second theorem proving. In a fashion , development of programs as a branch of symbolic computation is akin, everything being equal, to the Cartesian strategy of reducing geometry to algebra.

In this paper, I want to examine the scope of symbolic computation. To do this, I will review some of the more interesting systems in existence today. Next, I will try to explain some methods and problems in this area. Last, looking at the future, I will venture some conjectures about other areas where symbolic computation might expand and the impact this sort of computation has on computer architecture.

## II - MACSYMA

Symbolic computation, for many people, is synonymous with Calculus. Several systems and languages for performing symbolic computations in Analysis have been developped and used in the past 20 years. Formac was based on Fortran, and PL1-Formac on PL/I. REDUCE was developped to solve problems in Physics, but has been used for many other applications. Macsyma, developped at MIT, qualifies best as "system" of symbolic computation. First because it is an interactive system that can be mastered by a working mathematician very quickly. Second because it has been used and developped by so many people in the last 10 years that its coverage of mathematics is quite extensive. Arbitrary length arithmetic, differentiation, integration, power and Taylor series, calculus with matrices,block matrices and tensors are provided. Non linear systems of equations ordinary differential equations and partial differential equation may be solved using the basic Macsyma system.

The implementation of Macsyma is surprisingly efficient. There are two ways for an user to extend the Macsyma system. The easiest one is to write programs in the Macsyma language. This language has an Algol-like flavor but includes lists as a primitive data object. For many applications, this language will be perfectly adequate. Libraries of Macsyma programs may be built and invoked by the user manually or automatically. In this way, a community of Macsyma users develops and the system becomes, indirectly, more powerful. Macsyma is entirely written in MacLISP, a dialect of LISP developped at MIT. It is possible to write Macsyma extensions in LISP directly and to incorporate them into the basic system. This solution is more efficient but demands experts who understand the data structures used internally by Macsyma. Note that the ability to manipulate arbitrarily large integers is integrated in MacLISP itself.

Macsyma has been traditionally used for theoretical physics. But it seems that its use is also spreading very quickly in the area of Applied Mathematics. Not only does Macsyma allow to achieve some results faster, but it allows to derive routinely some results that would'nt even be looked after, in view of the complication of the calculations.

## III - SYNTAX SYSTEMS

One of the earliest achievements of computer science was to understand that the description of automata, and of formal languages could be algebraicized. In other words that the syntactic description of programming languages could be entirely characterized by a set of equations. In the past 20 years, many systems have been developped that analyse such systems of equations and produce, after a considerable amount of computation, an extremely efficient automaton. Furthermore, this extensive analysis gives way, in several systems, to a remarkable error recovery scheme. The restrictions on the kind of grammars treated by these systems have slowly been removed and while early precedence schemes where quite inadequate, from a pragmatic point of view the LALR1 constraint imposed by several current industrial parser-generators is acceptable.

Most researchers would probably not discuss parser generators in the context of symbolic computation . But in fact, using a meta parser is quite similar to using a formal system : Here data are syntax equations. When one modifies these equations, some processing has to take place to check the consistency of the new set of equations and in case of success, a process of compilation of these equations takes place. The result of this compilation may result in a failure, if it was impossible to construct an automaton with some special property. In case of success, an automaton that will permit parsing the sentence in the described language will be produced.

As computing becomes more and more interactive, many applications require the design of a "language" that allow the user to interact with the computation. This problem is still now solved piecemeal in most industrial contexts. In fact it can be given a quite uniformous answer across a whole computing system. In systems of the next 10 years, all input to computers should go through a single, unique, parser, driven by various tables that have been mechanically derived : In this context, it is likely that parsing algorithms will find their way to the hardware of our computers/terminals.

## IV - PROGRAM MANIPULATION SYSTEMS

The idea of considering programs as formulae in some large algebra -to factor out problems of syntactic appearance- originated in the early sixties. All of modern theory of computation is based on this fundamental premise. Slowly however, this idea is being embodied in practical systems. The MENTOR program manipulation system, developped at INRIA in the past 7 years, was one of the systems earliest in claiming that this idea had practical value. MENTOR, like Macsyma, is a system that allows the manipulation of formulae. But in MENTOR, these formulae represent programs. They are built from operators taken in a completely different algebra, that of the constructs of a programming language. In general, the formulae obtained in this way are very large, typically several thousands of lines when printed out in full. A programming language, MENTOL, allows interactive programming of transformations and computations on programs. One obtains in this way the facilities of a traditional extensible editor, with considerably greater safety (only some legal transformations are allowed). Program manipulation is easy because programs are so to speak preparsed. Large libraries of MENTOL programs have been built that facilitate the most usual program manipulations and perform expensive computations. The computations may be aimed at increasing the confidence in programs or at streamlining them. Progressively, more and more of the computations that usually take place inside a compiler are brought out to the surface and made available interactively to the user. MENTOR has been exercised most on Pascal programs. It has grown to be fairly widely used as a tool for Pascal programming in France. In a recent development, we have been successful at parametrizing with the object programming language. That is, to have a MENTOR system working on the Langage $L$, we must specify equations for the concrete syntax of $L$,. the abstract syntax of $L$, the correspondance between concrete and abstract syntax of $L$ and the canonical unparsing of $L$ programs. A unifying formalism, called METAL has been devised for the purpose of writing these language specifications. Of course, METAL specifications themselves are manipulated under MENTOR. As an example, the complete ADA language has been specified in METAL, allowing us to manipulate ADA programs under MENTOR.

From a METAL description of a programming language -or of any well defined formalism in fact- we derive, after performing consistency checks on the definition, tables (or code) for a unique parsing algorithm, tree generating algorithm, tree unparsing algorithm. The uniformity of these codes makes them good candidates for implementation in hardware.

The objects manipulated by MENTOR are not only formulae but rather annotated formulae. That is, it is possible to attach attributes to sub-formulae. These attributes are themselves formulae in another algebra. Of course these attributes may be recursively annotaded by other formulae ect ... . The structure, that originated form software engineering needs, is conceptually quite simple and can also be found in quite a few theoretical studies. MENTOR maintains the integrity of the structure and, switches smoothly between parsers (when input is needed), unparsers (when output is needed) and algebras (where modifications are performed).

MENTOR is being used for a great variety of applications : program development and maintenance, transport, dataflow analysis, semi-automated documentation. A recent application uses it for "Software Physics" measurements. The applications of the new multi-algebra facility seem to be even more numerous : language translation, computed annotations, automatic complexity analysis, test bed for language design.

To extend the power of MENTOR, the strategies are those used in MACSYMA. First the users may define new MENTOL programs, organize them into libraries and share them. Second, it is possible to write directly Pascal programs that access the MENTOR objetcts a collection of primitives. In this way, the user does not need to know the details of the data structures used in the implementation of MENTOR, and still gain a large factor in efficiency.

## V - SEMANTIC DEFINITION SYSTEMS

MENTOR allows an equational definition of the surface aspects of a programming language. Then it provides the user with a system to manipulate (realistic) programs. A system like SIS, designed and implemented by P. Mosses at Oxford and Aarhus University is much more ambitious. SIS allows the specification of the entire semantics of a programming language in terms of so-called "Semantic equations". The idea that the semantics of a programming language would be described by equations was put forward in the sixties but bloomed only after a proper theoretical framework was found to discuss the meaning of these equations. Many languages have now been formally defined using denotational semantics. This is the case for a large part of Ada, in work that was carried out at INRIA, in close cooperation with the language's design team. To define the semantics of a programming language, one defines the meaning of each construct of the language in terms of the meaning of its components. For a language of the size of Ada, the resulting set of equations is huge . Some ways have to be found to structure such a large collection of equations.

The SIS system is (in principle) capable of making use of this collection of equations to execute programs in the source language. In other works, even the task of moving from a language definition to a language implementation seems to have been mechanized. As a system however, SIS is only a wonderful proof of existence. It is handicapped by extreme inefficiencies -in areas where it is known how to eliminate them- and it is extremely batch oriented. However, it is clear that there are many opportunities for improvements, by orders of magnitude. For example, with respect to what is usually called static semantics -roughly the type-checking of programs- it is clear that a completely mechanical derivation from a formal definition is achievable (and with quite efficient results) in the next few years. It is worth noting that DSL, the formalism used to write semantic equations, shares many basic mechanisms with MENTOL even though both were developped quite independently. DSL is more elegant, of course, as it was not designed with terseness of expression as an objective.

## VI - PROOF CHECKING SYSTEMS

MENTOR allows the representation of programs as formulae, SIS allows the definition of the semantics of programming languages by formulae. Both approaches are implicit preconditions for the applicability of LCF. LCF, a system developped in the last 10 years by R. Milner and his colleagues at Stanford University and in Edinburgh allows to manipulate proofs of programs as formulae. LCF comprises in fact two parts : First a meta language called ML, allowing the interactive construction and manipulation of proofs, storage and retrieval of pragments of proofs, of earlier axiomatic definitions etc. Second, PPλ, a logical formalism inspired by Scott's theory of computation. This formalism particularly adequate for proving properties of recursively defined functions. Thanks to ML, a user can program proof strategies and therefore program the proof-checking at a fairly high-level. A powerful simplification algorithm is wired in LCF so that many trivial proof-steps are performed automatically. In a sense, LCF is very much like Macsyma, but works for logical formulae. Quite a few operations can be done mechanically, but in the end, the responsability for organizing a proof resides with the user. Useful strategies can of course be programmed in ML and then stored in libraries to be shared by a community of users.

While the interaction languages of MENTOR and MACSYMA leave a lot to be desired, ML is extremely well designed. Its applicability should extend much beyond its use in LCF. One of its most original features is a clever type inferring algorithm which may show the way of the future for interactive languages.

Many complex proofs of programs have been carried out in LCF, in particular proofs of correctness of small compilers. In spite of all the artillery provided by ML, these proofs are still fairly tedious, I am told. This difficulty may also be due to two facts : (a) the LCF system is very demanding in terms of machine resources, (b) the community of daily users is relatively small.

## VII - THEORY COMPILERS

The dream of automatic theorem proving is still today very remote. An early paper of 1970 by Knuth and Bendix has introduced the idea that some equational theories could be "compiled". A given presentation of an equational theory could be transformed, after a significant amount of computation into another presentation where proofs may be performed by mere use of the axioms as rewrite rules. In other words, this transformation allows from then on to perform computations rather that logical inferences. G. Huet and J.M. Hullot have built a system, FORMEL, that performs such compilations in a fairly general setting. While the compilation process itself involves user interaction (since it does not necessarily terminate), once a canonical term rewriting system has been obtained, proofs become very efficient. G. Huet and J.M. Hullot have shown that the scope of applicability of such theories is far from negligible. For example, many algorithms for solving the word problem in finitely presented algebras are special instances of this general compilation process.

Ideally of course, it should be possible to invoke FORMEL from within another system, like LCF, to be used as a particular strategy.

## VIII - OTHER SYSTEMS

In the sketchy presentation above, I have ignored several extremely important systems, for several reasons :

(a) Some systems are still at very early stages in their development. This is the case for LUCIFER, an interactive symbolic system for VLSI design and for PERLUETTE, a semantic meta-compiler relying on an algebraic compiler definition.

(b) I am not very familiar with some systems. AFFIRM, created by D. Musser, seems to be similar to the systems that I have mentioned in its general structure.
The Boyer-Moore system follows a somewhat different strategy : deduction is replaced by computation after the algorithm has been proved correct (mechanically).
FOL , created by R. Weyrauch at Stanford, has purposes similar to LCF but around first order logic.

(c) Some systems are based on the premise that a fairly powerful general purpose theorem prover is available behind the scene. This is the case, for example, of the Stanford Pascal Verifier. In fact, one purpose of my presentation is to show that symbolic computation is not synonymous with theorem proving. The systems that I have described perform very time consuming computations that do not include blind searches.

Even in the area of theorem proving, the most striking successes have been obtained when restricting the discourse to a particular mathematical theory, non-standard analysis for example.

## IX - METHODS OF SYMBOLIC COMPUTATION

The systems that I have described share one obvious common property : they have been developped by a small group of researchers over a period of many years. FORMEL is the youngest system and three years have already been needed before it acquires a community of users. The second obvious characteristics is that these systems grow in two ways : first by cumulating the user's efforts into libraries, second by writing extensions for reasons of efficiency, when the need becomes clear. Too little attention has been given, in general, to the structuration of user libraries. Ideas about organizing hierarchies of theories, as proposed by Burstall and Guoguen in CLEAR should probably gain a growing acceptance. The programming languages in which these libraries of programs are written differ significantly from system to system, but clearly it need not be so. The user of a system is very quickly at ease in another, a sign that the basic concepts are identical. The ease with which one writes extensions to a system depends on how transparent the architecture of a system is to a user. Systems like MACSYMA, MENTOR or FORMEL, where efficiency is a prime objective, are modular and easy to extend. They allow a fairly naive user to extend the system through an "abstract interface".

Efficiency problems in symbolic computation have traditionally been connected with lack of memory. In this respect Multics, with its enormous virtual memory, is the symbolic programmer's paradise. But even microcomputers, nowadays, have a large address space and virtual memory management, so that the problem is becoming less acute. However the needs for memory, in symbolic computation,are enormous. Garbage collection is still very much needed, in spite of some recent proposals and it becomes more and more costly and disruptive as the actual memory grows. Unless, as it is done in some special purpose machines, it may be made to run in parallel.

In terms of CPU consumption, symbolic computation systems are also very greedy. The are several ways in which this situation will be alleviated in the future. First of all, most machines have not been designed with this application in mind. As it was not too clear which would be the preferred language for symbolic computation, it was difficult to design a good instruction set. Today, the agreement seems to be quite general around the use of LISP. (MENTOR is written in Pascal and SIS in BCPL, so LISP is not absolutely necessary). Very efficient interpreters have been built by the VLISP designers (in Vincennes University) and MacLISP has an excellent compiler. As a result, machines specifically designed for LISP programming have made their way to the market and should florish.

Computer performance can also be improved if the computation can be spread among several CPU's. In the area of symbolic computation, comparatively few parallel algorithms have been designed. The idea of a unique store is of course a bottleneck. But if enough memory is available, sharing of data in memory can often be replaced by broadcasting data, alleviating also the task of a global garbage collector. Imaginative architectures are looked for, and I am not convinced that they will necessarily follow a data-flow scheme.

Once parsing has been eliminated as a technical problem, most problems in symbolic computation follow a unique paradigm : a tree traversal permits the construction of one or several clever data structures that will help in the next computation. These data structures are often attached to the original formula via hash-links that trade space for computer time. Output is obtained by interpreting the "result" data structure. For convenience, output is often expressed as a formula itself. This is the case in particular for multi-pass algorithms. The multi-pass algorithms offer an opportunity for parallel computation again, if synchronization is feasible via tree structured buffers.

## X - NEW APPLICATION AREAS

The ideas of symbolic computation will spread over new fields in the next 10 years. Let me mention briefly some of them :

- VLSI Design. Clearly, designing huge circuits poses a challenge not only to physicists and experts in electronics, but to people in charge of controlling their logical adequacy bu reasoning. From expert draftsmen, the VLSI CAD systems will become symbolic computation systems in charge of the thousands of checks that a design has to go through. Structuring designs is imperative as otherwise these checks will simply become computationally unfeasible.

- Document preparation systems. Even the best text formatting systems lack in ability to structure documents, and to manipulate them according to their structure. Technical documents, where text, graphs, pictures, drawings etc... may be unixed, where extensive internal crossreferencing occurs, where modifications may occur frequently and must be recorded are not unlike program text. These documents must be manipulated according to their logical rather than their physical structure. Familiarity with the Macsyma plotting facility shows how convenient it is to use formal input for graphics. In a recent Ph.D. Thesis, C. wan WYK has shown a clever way of indicating how pictures should be located in text : an equation (or several equations) is provided by the user so that, together with the equations describing the picture itself,a single solution will result. Integration of programs with their documentation has been a lingering problem for years. If a single meta-formalism can account for both programs and documentation, we may have a solution in sight.

- Computer Aided Instruction. The paradigms used in the construction of courses are very close to those used in program development. Techniques transposed from compiler technology will clearly be used in CAI.

REFERENCES


- Ballantyne A.M., Bledsoe W. W. "Automatic Proofs of Theorems in Analysis Using Non-Standard Techniques", JACM Vol 24 N° 3, July 1977.
- Boyer R., Moore J "A Computational Logic" Academic Press, 1980.
- Burstall R.M., Goguen J.A. "An informal introduction to specifications using CLEAR", December 1980.
- Burstall R.M., MacQueen D. B., Sanella D. T. "HOPE : an experimental applicative language" Procs. of 1980, LISP Conference, Standford Univ., Palo Alto, August 1980.
- Donzeau-Gouge V., Huet G., Kahn G., Lang B. "Programming environments based on structure editors : the MENTOR experience", INRIA, Report N° 26, July 1980.
- Donzeau-Gouge V., Kahn G., Lang B. "A complete machine checked definition of a simple programming language using denotational semantics", IRIA, Report N° 330, October 1978, Rocquencourt, France.
- Gordon M., Milner R., Wadsworth C. "Edinburgh LCF" Lectures Notes in Computer Science Vol. 78, Springer Verlag, 1980.
- Hullot J.M. "A catalogue of canonical term rewriting systems", Technical Report CSL-113, SRI International, Menlo Park, April 1980.
- Huet G., Oppen D.C. "Equations and Rewrite Rules : a Survey" Technical Report CSL-111, SRI International, January 1980.
- Mathlab Group "The MACSYMA Reference Manual" Laboratory for Computer Science, MIT, December 1977.
- Mélèse B. "Manipulation de Programmes Pascal au niveau des Concepts du Langage", Thèse de 3° Cycle, University de Paris-Sud, Juin 1980.
- Mosses P. "SIS : a compiler-generator system using denotational semantics" DAIMI, University of Aarhus, 1978.
- Mosses P. "SIS : Semantics Implementation System" Reference Manual and User Guide, DAIMI MD-3, University of Aarhus, August 1979.
- Musser D.R. "Abstract Data Type Specification in the AFFIRM System" Proceedings of the Specification of Reliable Software Conference, Boston, April 1979.
- U.S. Department of Defense "Formal Definition of the ADA Programming Language" INRIA, Nov. 1980.

# VI INVITED LECTURES

## —Architecture—

A COGNITIVE ARCHITECTURE FOR COMPUTER VISION

Bruce H. MCCORMICK, Ernest KENT and Charles R. DYER
Integrated Systems Laboratory
University of Illinois at Chicago Circle
Chicago, Illinois 60681 USA

Our objective is to design a visual analyzer for real-time scene analysis of dynamic imagery. For this purpose we employ a cognitive architecture derived from concepts of the central nervous system. Output of the visual analyzer will be high-level real-time perceptual data suitable either for the guidance of robot action or for the symbolic description of the visual environment. This output resides in a distributed relational database modeling the hypercolumnar architecture of the visual cortex.

The authors conjecture that the principles of cognitive computation described here can be generalized to other sensory and effector information-processing tasks.

## I. INTRODUCTION

### A. Cognitive Architecture

These are times of great expectations in neuroscience. In recent years the columnar structure of the neocortex has been repeatedly confirmed by experiment. Increasingly the hypercolumn is regarded as the modular biological microprocessor underlying perceptual and cognitive computation. Once invented by nature columnar structure has evolved rapidly: in Homo sapiens it has expanded 156-fold over its size in the prosimian brain, scaled for relative body size. As far as we know, this basic modular architecture is used for all sensory cortical information processing -- visual, somatosensory and auditory. Motor cortex is similarly organized. Human neocortex has approximately one million of these hypercolumns. Even higher level intellectual processing seems to have evolved from the same highly parallel structure of hypercolumns that characterizes sensory processing, apparently without fundamental change of organizational principle or architecture. The ascendency of this new parallel architecture for cognitive computation, using the hypercolumn as its basic processing element, began about ten million years ago -- a mere eyeblink in biological time since the first prosimians clinging to their twigs stared at one another. Since then it seems that evolution has embarked upon a blitzkreig campaign to impose the dominance of intellect on the world scene.

In many ways the von Neumann computer, now shrunk and encapsulated in a microcomputer, has shown a similar explosive development. It reminds one of nature's invention of the cell, a revolution that created an entirely new biological world, and in time altered the world environment as well. But these silicon and metal microcomputers are still "growing" as isolated, scattered "cells" on the computational landscape. They await the next evolutionary leap. Now these microcomputers must colonize in self-governing hierarchies structured to create cognitive computation. In truth, cognitive computation is at the dawn of its civilization, awaiting the invention of an appropriate architecture to build its first cities. It is our conviction that the design of a machine visual system paralleling known biological visual system architecture affords the best opportunity to initiate the inevitable evolution of cognitive machines.

### B. Unique Aspects of the Proposed Visual Analyzer

Unique aspects of the proposed visual analyzer are [McCormick 1981]:

Events in Time Are the Units of Analysis. The element of visual analysis is the current action occurring over time, rather than a hypothetical frozen instant of a visual field.

Real-Time Analysis of Time-Varying Imagery Is a Design Goal. To meet the needs of advanced applications, the visual analyzer must be able to process dynamic imagery in real time. This is a design goal from the start.

Biologically Based Design Is Compatible with Advances in VLSI Electronics and Neuroscience. The design uses VLSI technology to express neural architecture as currently understood. Further convergent developments in these areas may be incorporated into the system at a later time.

The Biological Model Can Be Extended from

Perceptual to Cognitive Machines. If we can model columnar processing adequately and can mimic its methods for communicating with neighboring and remote columns, then it may prove to be the case that our architectural work is in some fundamental sense accomplished -- that this class of machine has the full potential to emulate high levels of cognition. Much of our sense of excitement about the proposed project stems from this biological paradigm.

Translated to specific domain of the visual analyzer, it may turn out that we need not design an endless progression of processing elements for each successive level of visual analysis, but rather that a common "hypercolumnar" processing element may suffice for later as well as earlier stages of the cognitive machine design.

The Biological Model Can Be Realized in VLSI Hardware. A three-dimensional array of functional units offers adequate computational power for real-time analysis of events in four dimensions (three-space plus time). Computational processes in the system may interact along either temporal or spatial dimensions of the input data.

## C. Significance

Achieving these goals will have both practical and theoretical benefits. Man-made visual systems of the type described here may be employed not only in intelligent industrial and domestic robots, but also in such robotics applications as visual guidance of high-speed flight of aircraft over unfamiliar terrain, undersea exploration, maintenance of nuclear reactors and service operations in other hazardous and inaccessible environments, high-level monitoring of remote sensing data, and indeed any application of intelligent machines to real-time image processing in three-dimensional time-varying environments.

# II. RATIONALE FOR THE PROPOSED ARCHITECTURE

Underlying the proposed design strategy are two assumptions which we explicitly adopt:

1. Functional modeling of selected aspects of biological image processing is a promising and largely unexplored approach for further development of machine intelligence.

2. Parallel processing computers of the sort described below offer the machinery most likely to realize a real-time visual analyzer successfully.

In the light of previous work in biologically oriented modeling, and the existence of other possible design approaches, these assumptions warrant comment. In this section we shall interconnect the issues of biology and machine design, to describe the design of the visual analyzer and its biological rationale from the perspective of of principles which we perceive to be embodied in the results of a half-billion years of visual experimentation. The order of topics adopted in Section IB will be used to describe the unique aspects of the visual analyzer.

## A. Events in Time are the Units of Analysis

The natural unit for visual information processing, in the machine as in the animal, is the current action, not the current instant. We might capture this distinction by the analogy that in the real world we look at a movie, not at sequences of individual static pictures. The direct perception of reality as an ongoing process is fundamental to our ability to interact with it. Even if the human observer is artificially constrained to look at a single static picture, we know from observation of his eye movements that he decomposes the scene into a rapid time-sequential montage of subscenes, with some subimages repeatedly and redundantly reexamined: he imposes the movie organization on the static picture. Our everyday visual "image" of the world is just such a montage, sequentially built up from views through the tunnel of clear foveal vision. Thus, the ability to analyze time-sequences of images into coherent perceptual scene-models is fundamental to our perception of events in both space and time.

For the purposes of computer vision, we find it useful to generalize the concept of a movie to include: time-sequential images (robotics); spatial sampling of an extended high-resolution image (cartography); multispectral imagery (remote sensing of the environment); and consecutive serial sections of a three-dimensional object (computer tomography, spatial reconstruction in microscopy). We must also point out that, in reality, except for the engineering subdiscipline of multispectral image analysis, there is today only a very limited corpus of experimentation in any of these areas.

Furthermore, we find essential that the visual analyzer provide both image management and image computation. Utilizing this distinction we have attempted to generalize from prior experience with spatial data management and multispectral image analysis to a projected time-varying image-flow architecture for computer vision.

B. Real-time Analysis of Time-varying Imagery is a Design Goal

Advanced computer architecture is still wedded to habits of thought which have evolved from the analysis of static imagery. While such architecture might be employed to analyze dynamic imagery, it cannot cope with it in real time, and thus loses the principal advantage of such analysis. For this reason we place great emphasis on learning from biological models of vision, as biological vision developed explicitly for real-time processing of dynamic imagery as a guide to navigation. From dynamic biological models, structural correspondences can inspire appropriate computer architecture.

Before characterizing what a biological visual system does, it may be useful first to discuss what such a visual system need not do. Consider a chipmunk running across a field. Its visual system certainly does not need to resolve and name each individual blade of grass in its visual field, perhaps some 10-20 times per second. By overloading its visual system, this inessential processing would inhibit attention to the potential presence of a hawk circling overhead. It would suffice for the animal's visual system to discern grass as a textural element, crudely segmented through consecutive frames. A similar argument applies equally well in the case of a human pilot in a low-flying aircraft or a machine-guided low flying missile. To optimize its use of time such systems must perform urgent analyses early at low levels of sophistication, and derive sophisticated analyses later as each image flows through the system. Simultaneously the visual analyzer performs a new set of low-level analyses on the next instance of imagery as it enters the system. Clearly a visual system with this capacity — an action-oriented architecture — would have an evolutionary advantage.

Lampreys and other cyclostomes which first appeared in Ordovician times three hundred million years ago, already had well-developed eyes and visual brain centers. Insects took a different but remarkably parallel evolutionary course. Animals of different classes look upon the same natural scene, but "see" and respond differently. The lamprey and the house fly share few common visual interests; the eyes and visual pathways of these two animals reflect their disparate concerns.

By analogy, we have no reason to anticipate that the rapid industrial evolution of vision-equipped robots and missiles will give rise to one universal architecture for computer vision. Rather, the design of individual computer vision systems must be predicated upon the actions and responses required in their circumscribed visual environments, and diverse evolutionary paths can be expected. The architecture of our device should lend itself to flexible implementation.

C. Biologically Based Design Is Compatible with Advances in VLSI Electronics and Neuroscience

The attempt to produce machines that perform cognitive functions by emulating biological systems is not new, and its motivation is quite straightforward. The brain is the only example we have of a working system which accomplishes the desired ends. Because the basic elements of the brain perform functions similar to those of elements of electronic systems, it is tempting to suppose that many operating principles of the brain will transfer easily to electronic systems.

<u>1. Inappropriate Levels of Modeling.</u> Despite these seemingly plausible arguments, attempts to base practical devices on biological models have not been highly successful. A principal reason appears to be that designers have focused on aspects of biological function unsuited to the available technological base. For example, there has been no shortage of designs for "electronic neurons," despite the total lack of an adequate technology for producing and interconnecting such devices in the quantities needed to simulate brain processes. Attempts at simulating this level of biological functioning also ignore the lack of understanding of detailed brain operation on which to base algorithms for these devices.

At the other end of the scale, there have been very many attempts to model the gross functional aspects of brain operations on traditional serial computers. While such experiments in artificial intelligence have produced many interesting insights, they have not been able to generate practical programs for real-time perception on a useful scale. It seems clear that they are inherently limited by the size and speed of the machinery on which they run, and that no foreseeable improvement in this technology will enable them to compete in real time with a biological system whose architecture is adapted to the efficient solution of problems requiring large numbers of simultaneous operations. In this case as well, information is lacking about the physiological details of the processes being modeled. It is an unfortunate truism that theories of cognitive psychology owe more to the field of artificial intelligence than vice versa. While such programs may attempt to simulate the overall functions of biological systems on the large scale, they are unable to draw upon any existing body of information concerning the actual underlying procedures of large-scale biological systems. They are thus denied any real advantage of having an existing system to emulate.

2. Proposed Level of Modeling. We do not believe that these problems in previous research affect the validity of the biological modeling approach. Rather, such problems seem to reflect difficulties in selecting levels of approach appropriate for existing technology or existing biological knowledge. Both the design approach detailed here and the functional level of the processes targeted for emulation have been carefully chosen to avoid these pitfalls.

The visual analyzer proposed here would model certain lower-level properties of the biological visual system in order to serve as an input preprocessor for advanced visual systems. At this preprocesser level, existing physiological models offer sufficient detail for substantial guidance. In addition, a technological base adequate to realize the system in VLSI hardware either exists or is under development. The functional level represented by these stages of visual input processing is sufficiently advanced to make the development of such a device attractive, while not being so specialized as to preclude its subsequent use as a building block in generalized cognitive machinery.

At the level of image processing represented by the proposed device, numerous theoretical models of biological visual operations exist which provide detailed insight into a variety of important and fundamental perceptual processes. For example, detailed theoretical models describe perceptual information "property channels" in the visual system which extract information on luminosity, color, texture, motion, depth, location and other elementary properties. Other models specify mechanisms for boundary detection and scene segmentation, form analysis, the deletion of redundant information, and the creation of location-free statements of scene content for subsequent use in categorization and naming.

D. The Biological Model Can Be Extended From Perceptual to Cognitive Machines

We shall define the output properties of this system to fit the input requirements expected for higher-level generalized systems. In this way, we will be able to draw upon existing knowledge of human cognitive/perceptual processes in defining the proposed system, even in the absence of specific knowledge of the underlying physiology of these higher processes. A particular advantage of the proposed approach is that, by designing the output to conform to the kinds and organization of data employed in human higher perceptual processes, further research on simulating higher processes will be facilitated. "Bootstrapping" the development of successively more advanced systems based on biological models thus becomes possible.

E. The Biological Model Can Be Realized in VLSI Hardware

We view the real-time visual analyzer as being a three-dimensional stack of two-dimensional processing ARRAY ELEMENTS interconnected in the third dimension by intervening INTERCONNECTION ELEMENTS (see Figure 1). In physical structure, the real-time image processor would resemble the multiple elements of a compound lens or a club sandwich, with alternating array and interconnection elements.

The image data from a particular "moment" of image-time will be contained in, and processed by, an array element at each stage of its analysis, and then passed to the next array element in sequence. Meanwhile, new image data is entering the device continually from the front, so that successive moments of partially analyzed image data flow through the device in a pipelined fashion. Thus, image-time is represented linearly in the third dimension of the device (perpendicular to the elements), while the visual scene at any given moment of image-time is represented by data residing on the two-dimensional surface of an array element.

1. Realization of Array Elements. Each array element is composed of a mosaic of VLSI chips, much as the high-resolution CCD array for the NASA space telescope is built from a mosaic of CCD chips (see Figure 2). We anticipate that processing arrays at different levels of the device will be functionally quite distinct, and will parallel their biological analogs at different functional levels of biological visual systems.

In a number of ways, the constraints on the design of these proposed processing arrays are quite unlike those of other existing image processors, in several ways:

a. Each array element consists of a regular rectangular or hexagonal 2-dimensional array of processing elements (PEs). The PEs provide distributed feature extraction and database storage for several different image properties, with some PEs specialized for one set, and others for another.

b. The array elements are computationally bound to complete their processing of a time-segment of the image in a typical frame time of 33 msec, for use with commercial United States/Japanese television equipment. Nonetheless individual PEs may interpolate the changes in time-varying imagery over several frame intervals.

c. Since dynamic imagery flows through the system, moving successively from one processor array to the next, it becomes necessary to devote special attention to the interconnection between processing elements. The competition between the time required for interprocessor communication and silicon real estate that must be expended on data buses constrains the space-time trade-offs for implementing a processing element and the level of available data distribution to neighboring PEs, both within the same array and between adjacent arrays. We consider that an algorithmic study of these trade-offs is of fundamental importance, with direct bearing on the appropriate design of semiconductor array circuitry and the third-dimension interconnection elements.

Judging from physiological evidence, the visual analyzer will require at least three levels of array processing, and probably will require at least five levels to reach higher levels of abstraction.

2. Interconnection Elements. One potential interconnection technology available today would use optical linkages between the integrated circuit array elements. In this model the connection elements can be viewed as generalized fiber-optic bundles allowing the fan-in and fan-out of signals as image information is passed from one array element to the next.

For devices intended for commercial robotics, we are attracted to an optical/fiber-optics technology in which the individual processing arrays are readily replaceable, much like individual lens elements of a compound optical system. This would also permit easy customization of systems for varied tasks. That the interconnection elements are inert fiber-optic networks embedded in a supporting matrix is also attractive, as these components of the design would be hardy and should require virtually no maintenance. Interconnection elements also serve to conduct heat from the adjacent array elements.

These remarks are intended only to illustrate the intrinsic feasibility of implementing the visual analyzer in a dynamic 3-dimensional architecture. We are not prematurely wedded to any one interconnection technology.

F. Summary

Biological vision is fundamentally parallel in its organization. As a result, we may employ biological models as paradigms of useful processes in image recognition, and as demonstrably successful examples of such processes realized in a parallel processing environment. The major task of the proposed program is determining efficient methods for implementing these paradigms in electronics-based as opposed to neuronally-based architecture. The level of biological modeling to be employed in the detailed design of the processing array elements will be intermediate between that of a "neuron-by-neuron" model of the visual system, and that of a mere "black-box" functional description of cortical operation.

A three-dimensional architecture with space-time interactive processing such as that described above most closely addresses the requirements of a system designed to emulate biological processing. This choice of computer architecture closely parallels the biological system in that it too employs a multi-stream pipelining architecture with strong interaction in the longitudinal (temporal) dimension as well as transverse (spatial) interactions.

## III. RELEVANT CONCEPTS FROM BIOLOGICAL VISION

We have proposed that the best model for simulating physiological vision mechanisms will be a layered series of array elements composed of vertically and laterally interacting processing elements, as described in Section IIE. Such arrays have a very close functional and even topological resemblance to processing stages in biological visual systems. Most processing in the visual system takes place in a series of stages, or levels, corresponding to the various "mainline" structures of the visual pathway. These main structures include the retina, the lateral geniculate nucleus of the thalamus, and the primary, secondary and infero-temporal areas of the visual cortex (Figure 3). At each of the levels, the output of a preceding stage enters a cell layer on parallel lines of axons which ramify to form connections with other cells in the neighborhood. The clusters of cells receiving this input, which function as a local circuit, operate on the input in ways determined both by several local properties of the cellular circuitry and by the state of various other inputs. Preserving the topological homologue of the input data, these cells then send their own output through a bundle of parallel axons to the next processing level. Local properties that affect processing within a cell layer include the local "program" determined by the connections of the local neuronal circuits, and the way in which the data is input. (Data may be input to the cell through either excitatory or inhibitory connections.) Other sources of input which affect local processing include the outputs of neighboring processing units in the level, and feedback and control information from successor stages in the process, as well as control information originating in nonvisual parts of the brain.

Information is carried through the biological visual system on ascending axonal lines which encode the features, or properties, of the image on the retina. At low levels of the system, these are only primitive properties such as luminance, color, and motion and other temporal information [Enroth-Cugell, 1966; Goodwin, 1975; Emerson, 1977]. As the data ascends through the system, however, new derived properties are added at each stage as processing proceeds. Thus, derived properties such as "depth" and "edge" may be added in parallel with the more elementary properties. Combinations of these such as "edge of color" are also derived and added to the assemblage of ascending property submodalities. Property submodalities, encoding for a specific local attribute or "property" of the visual scene, are thus conceived as vertically organized parallel information channels distributed throughout the breadth of the system.

A higher-order spatial organization clusters spatially adjacent channels of common ascending property submodality into functional units whose internal distribution of activity collectively encodes the character of excitation of this submodality over a small retinal area. These assemblages, called "columns," are described in the next section.

A. Role of Columnar Structures in Biological Vision

A general feature of biological vision which is broadly relevant to much of its operations is its division into columnar functional units. Each columnar unit samples and encodes a "confidence estimate" that the value of a property sampled over a small part of the area of the visual field falls within a prescribed range. Neighboring columnar units "see" overlapping neighboring portions of the visual field. In general any pixel of the retinal image falls within a rosette formed from the overlapping receptive field of six neighboring columnar units. This rosette-like organization creates a second type of parallelism in the underlying design of the physiological machine. That is, not only are visual properties processed in parallel columns, but the property channels themselves are replicated in multiple parallel columns, each of which processes property information about only a small part of the visual field.

At subcortical (i.e., retinal and geniculate) levels, the cell assemblies homologous to the cortical columns retain registry with them via the "retinotopic" scheme of projection in the interconnecting fiber tracts. In the primary visual cortex, the columnar architecture is many cell layers deep, and various functional distinctions are found at differing depths. Beyond the primary visual cortex, retinotopic projection schemes project the basic columnar organization onto secondary cortical areas, but in this case the projection branches to several areas, each of which continues the columnar architecture.

Beyond the secondary cortical regions the retinotopic mapping of the columnar architecture abruptly terminates. Instead, higher-level cortical visual areas receive inputs to each of their local circuits from many of the columnar regions of the preceding secondary areas, completely abolishing the previous point-for-point spatial representation of visual properties.

Finally, a secondary visual circuit bypasses the primary visual cortex. It appears to carry location but not content information to the higher levels -- apparently for use in windowing or attention operations as discussed below.

A particularly elegant and logical neuronal structure is seen in the primary visual cortex, where the ascending columns are clustered together to form a new higher-level construct called the "hypercolumn" [Hubel, 1977]. Each hypercolumn in effect represents a complete analysis of a small area of the visual field, examining the full range and distribution of values for multiple properties. The principal subtypes of column identified in the primary visual cortex are orientation columns and ocular dominance columns.

Orientation columns are slab-like structures approximately 50 microns across, extending vertically through the striate cortex. Cells in these columns preferentially respond to specific orientations of geometric features such as lines and edges. Neighboring cells examined in the same axial transversal through the column respond to essentially identical preferred receptive field orientations, even though centered at different adjacent positions [Richards, 1977]. However, along lines tangential to the outer cortical surface, one finds that preferred receptive field orientation angle of these cells transforms smoothly, turning through 180 degrees of feature orientation in about 1 mm.

Ocular dominance columns are columnar slabs within which one or the other of the binocular visual receptive fields of the cortical cells predominate in determining the cell's response. There is a regular alternation of dominance columns for the two eyes. The ocular dominance columns are about 0.5 mm wide, so that a full traverse of a set of orientation columns through 180 degrees of feature orientation takes about as much space as the width of a pair (right/left) of ocular dominance columns. It is thought that the ocular dominance organization is perpendicular to that of the orientation organization with regard to the long and short axes of the slabs, so that a 1 mm square area of cortical surface constitutes a repetitive organization sufficient for dealing with the full range of orientation and binocular data within a limited area of the visual field. Such a complete array of orientation and ocular dominance columns is referred to as a hypercolumn. The overall retinotopic organization of the cortical surface projects on these arrays to result in a hypercolumnar processing unit for each small area of the visual field.

At higher stages in the visual system, for example in the infero-temporal cortex, the functional receptive fields of individual neurons appear to cover a much wider (and dynamically variable) region of the visual field. This corresponds to the fact that at these levels the retinotopic mapping has been replaced by a more global one. The extent to which hypercolumnar structure is preserved at this level is as yet unknown.

### B. Dataflow Programming in Biological Vision

As we have described the system, it is evident that approximately through the level of the secondary visual cortex, the analysis of the image resembles a topography-preserving imaging system, in which the collection of data lines flowing out of the "top end" of each retinotopically mapped hypercolumnar system encodes a multidimensional property analysis of a small area of the scene. In man, the integrity of the system through the primary visual cortex suffices for the "raw experience" of patterned vision, but not for understanding of the scene. This functional level we shall refer to as "intermediate-level" vision.

In considering neuronal data processing, several points are worth noting. First, in the physiological system, the representation of data has both digital and analog aspects. The nature of the quantity being represented is encoded in a "place code," that is, the type of data being transmitted is encoded by the line on which it appears. Thus the "quality" component of the output of a level is defined by which lines are currently active. At the same time, in biological vision, each line also carries "confidence estimate" information quantitatively encoded as impulse frequency, and decoded by the receiving neuron through integration over a limited time period. (An exception occurs at early stages of the retina where this analog information may be passed on electronically without impulse encoding and decoding). At low levels of visual processing, this quantitative signal represents elementary quantities such as luminance intensity, while at higher levels it is basically a measure of goodness of fit between the cell's optimal multidimensional set of inputs and the actual input. Typically information is carried through the system by a channel consisting of a population of such lines [Erickson, 1974]. Subsequent processing operates on the distribution function defining the degree of activity for each channel line.

Second, the system resembles a multi-pipeline processor in which all levels are simultaneously acting on the sequentially transformed image data as it arrives and flows through the system. However, in the physiological system transitions from one level to the next are not clocked (nor are information transfers between neighbors within a layer). In effect, the system behaves as if fully buffered, and data are continually updated as analysis proceeds.

Third, information from lateral neighbors or from feedback and feedforward may have either of two actions. It may modify the basic input information by adding to or subtracting from the analog quantity being decoded by the cell body. It may also, through a process of presynaptic inhibition, enable or inhibit particular input lines, either from local neighbors or other cell layers. The former action corresponds to traditional analog feedback and feedforward operations, while the latter is more akin to modification of the program of the local processing unit.

### C. Higher Levels of the Biological Vision

System

"High-level" vision is used to denote the operations performed, principally in later stages of the system, which result in recognition, naming, and description of visual objects or events [Gibson, 1979; G. Johansson, 1975].

The receptive fields of neurons in these later stages are not bound to small areas of the field as are those of earlier stages, and are dynamically variable in size and location across large portions of the field. It is as if the cells of these areas responsible for high-level recognition functions could examine the map of properties, presented by the lower levels of the system, through variably sized and located windows. This windowing process is directed by a portion of the "attention" process in nonvisual portions of the brain that are processing the current goal structure of the system. The windowing operations also receive guidance from a parallel "location channel" which bypasses the primary visual cortex. (This suggests that the location information inherent in the retinotopic representation may only be utilized by these systems to determine relative spatial relations of properties for scene segmentation and identifying objects.) The output of these high-level stages can be used for matching to category "templates" in memory, for updating the cognitive map that contains the system's world model, and for guidance of immediate interaction with the environment. Their outputs apparently also descend into lower levels of the system to "tune" it in accord with the current requirements of the high-level processes.

We hasten to add that we do not intend the scope of the present paper to include high-level visual functions. At this time, neither the physiological nor the image processing fields are sufficiently developed for higher-level analysis. Rather, we study these processes in order to determine their probable input requirements, which will direct the choice of output data structures in our low- and intermediate-level visual analyzer. We also study higher-level functions to determine the best ways to model certain of their features necessary to the operation of our device. Viewed in this context, our visual analyzer generates a relatively low level of symbolic description of the visual image, which is both useful in its own right and a necessary first step in the study of ways to implement these higher levels and to generalize the architecture to other sensory modalities [Gibson, 1966].

In this section we have followed the flow of data from the retina to the high-level visual functions. In the next section we shall outline the proposed cognitive computer architecture for modeling and simulating this process. For expository reasons, it will be convenient to begin from the top and work back down again to the receptors.

## IV. RELEVANT CONCEPTS FROM MACHINE VISION

In this section we describe the major features of our approach to implementing a real-time visual analyzer for processing time-varying imagery using principles of structure and function implicit in the biological visual system, as described in Section III. It will be most profitable to begin with a description of the way in which data will be represented at the higher level. Later sections will then discuss aspects of the machine visual system's world model including ways of implementing and employing the world model, ways in which it is formed from the functional hypercolumns, means of implementing the underlying columnar operations, and means of supplying them with input.

### A. The Artificial World Model as a Relational Database

Functionally, the "cognitive map" often employed to represent the human cognitive database has close affinities with distributed relational database schemas of information storage. Relational databases will therefore provide a convenient and natural formalism for dealing with the symbolic image codifications produced by our proposed visual analyzer. The relational database receiving this processed image data will store information which is analogous to the "world model" of biological cognitive processes, using mechanisms described in greater detail in Section V. This machine world model will be dynamically updated in real time out of various sequential views over time.

Of course, this machine world model will be much impoverished compared to our own. We do not yet know how to extract from the image data the high-level relations which fill our world models with such items as "furniture," "chairs," or even "that chair," let alone "phyla," "political parties," or "relational databases" [Marr, 1979; Marr & Nishihara, 1978]. On the other hand, the machine world model contained in our relational database will not be impoverished by most other standards. It should compare favorably with that of a lamprey, for example. It will contain all of the low-level relations which codify the content of immediate visual experience [Marr, 1976], and it will contain them for several temporal slices of the recent past, queued to form a structure which encodes the time dimension as well as three-dimensional space [Dreschler & Nagel, 1981; Marr & Ullman, 1981; Ullman, 1981; Lee, 1980].

This level of world modeling is entirely adequate for a great many real-time applications which are too complex for existing image-processing devices, for example, high-speed maneuvering around obstacles. The data generated by the visual analyzer consists of n-tuples which are stored in tables of the relational database. After a sufficient number of "glances" at the environment, this database will contain a collection of relations representing the machine world model at approximately the level of processing accomplished by the primary visual cortex. That is, relations will describe spatio-temporal boundaries of multi-property areas sufficient for patterned vision [Horn, 1981; Thompson, 1980; Marr & Poggio, 1979], but not yet explicitly encoding object segmentation, and certainly not names and categories. As in the case of the biological primary cortex, these higher-level relations are "hidden" in the data. The extent to which higher-level vision can be embraced in the same relational scheme is unknown at present, though as we show later an entire spatial data management system can be built on the substratum of a relational database.

B. Attention Mechanisms

Ideally, we would like to have an "inferotemporal cortex machine" which examines the low-level world model with an attention-driven spotlight or window, forms high-level relations and enters these into additional high-level tables of the world model. More prosaically we choose to begin by providing the kinds of predetermined responses that are analogous to the reflex-level attention-driven processes in biological organisms. Database optimization techniques can specify predefined important relations, and active alerting strategies can give further assistance [Michalski, 1980; Chang, 1980; Chang & Ke, 1979; Chang & Chang, 1978]. These advanced database management techniques normally improve processing speed. Contemporary database retrieval schemes also seem adequate to ensure that the database need not appear as a bottleneck to real-time operation. This low level of world modeling may be adequate for limited sorts of machine recognition.

Another way to examine selected low-level relations in such a database system would be for humans to take the role of the "smart" attention function, as in remote sensing applications. This type of attention mechanism is greatly enhanced by the spatial data management technology described below for advanced and flexible property-oriented displays generated from the database.

The attention function, whether human-driven or machine-driven, will not only scan the contents of the database, but also to direct the acquisition of the data that fills it. Figure 4 illustrates this process. The sampling frame quantizes the space-time scene into successive time slices, T0 through T8. At any given instant the visual analyzer can only deal with limited aspects of the current imagery, but these may be in a sampling window located anywhere in the frame. Additionally, they may be concentrated in a small region by zooming in on small sections of the scene, or more coarsely spread by sampling larger areas. Thus, the window of heightened attention is of adjustable size. Figure 4 charts the path of this adjustable window in space and time as it follows a space-time event on the sampling frame under the guidance of the attention function. This segment of an "attention geodesic" defines the segment of space-time that occupys the current contents of the low-level world model.

C. Columnar Structure as a Distributed Database System

To understand how the relational database should be constructed, we can examine the cortical hypercolumnar structure as a conceptual formalism as well as a physical structure. We hypothesize that a central function of the hypercolumns is to facilitate multi-channel interactions and their use in the description of dynamic imagery. We can view each hypercolumn as serving to define a relation in the technical sense of a relational database [Codd, 1970; Date 1977], where a relation is defined by ascribing to it an n-tuple of concurrent property submodalities. We may further surmise that each column can be activated only by a selected range of the property submodality values, much as in the simple selectors of variable-valued logic [Michalski, 1980]. Each column provides a "confidence estimate" that its submodality has been observed; collectively these confidence estimates constitute the n-tuple stored in the relation for each instant of time (For orientation columns this n-tuple provides exactly the data used in the edge detection scheme of [Marr & Hilreth, 1980]). A hypercolumn therefore contains relational data. It codifies a complex visual relationship among multiple property submodalities and facilitates subsequent visual analysis (e.g., edge detection.)

The hypercolumn's identity codes yet another relation: that which exists between the n-tuple of interacting properties and the spatial location of its receptive field. All these items may be directly stored in a distributed relational database, and in principle need not be ordered by the spatial mapping. Thus, the retinotopic arrangement of the hypercolumns is redundant from the standpoint of data description, but minimizes communication pathways to interacting neighbors.

Seen in this fashion, the actual hypercolumnar structure of the nervous system is only one possible physical realization that might have evolved to encode the spatially distributed database of momentary visual experience. We hypothesize that the hypercolumnar structure makes the best of the physical constraints imposed by connectivity, and we expect that in the physical realization of our visual analyzer similar considerations will produce similar results. Logically, however, we may treat the hypercolumns as defining the relations central to machine visual process, whether or not our device in fact has physically adjacent hypercolumns.

It should be kept clear, however, that the correspondence between relations and hypercolumns is not exact. In particular, a relation is passive in the sense that it only stores information (i.e. n-tuples) in a relational table; it does not act upon its own contents. In contrast, a hypercolumn is a processing element, that contains both internal buffer storage and a functional processor which operates on n-tuples stored in its internal table of n-tuples. Thus while a relation is merely a static description, hypercolumns can adaptively accept inputs and produce output channel encodings based upon the current state of their relational tables.

We may consider hypercolumns as physical groupings of columns for similar submodalities but adjacent value ranges such as adjacent orientations, L/R eye dominance, etc. We view the visual systems as transmitting successive channel-encoded properties to a wide variety of hypercolumns inducing occasional temporary hypercolumnar activations -- subject always to antagonistic/synergistic lateral inhibition and attention tuning.

Given this formalism, we may think of the relational database of our world model as given by the adaptively-defined "logical picture" represented in the relational tables of the retinotopically distributed hypercolumns.

While this understanding of the form of the machine world model is helpful, it might seem to be still a formidable task to develop actual machinery to transform data smoothly from the visual domain to the relational database domain and conversely, particularly if an external attention mechanism will control the processing and convey its desires in powerful and concise high-level statements. Such control implies ordering of the data into appropriate hierarchies. In fact, such a device is already well advanced, and under development for superficially different purposes: spatial data management, and interactive map display for military applications. This work has clarified mechanisms for bringing image data and symbolic data smoothly together -- admittedly by interaction with the user, who takes the role of the "attention mechanism".

## V. VISUALLY EXPLORING THE DATABASE

The operator must be able to visualize the data in the distributed database of the visual analyzer, and thereby reconstruct on human terms the visual environment seen by the analyzer. This facility of the analyzer to allow visual exploration of the machine database is needed (1) to evaluate the operation of the visual analyzer during its development and (2) to guide and interact with its operation.

This reconstruction of the visual environment of the analyzer, however, is not a simple matter, particularly as we must provide the operator with an appropriate graphical interactive display. Our own visual environment is experienced as through maps of varying resolution which project properties on the spatial surface of our perceived visual world. At each instant foveal vision provides but one small snapshot, or "map" of the more comprehensive "map set" of our perception of the visual world about us.

This human process has direct analogy with the way data is represented in our visual analyzer; here again imagery is encoded as spatial maps of properties. Each map, in turn, resides in the distributed relational database. However, it resides in a "data space" which is structured differently, and so can not be directly visualized. The problem then is how to transform data from relational data space into visual space.

We require a system that will interactively map the relational data onto visual space. This machinery must perform an attention-directed search of the relational database by property, or by combinations of properties, and then map the relevant data onto visual space. For display purposes the operator will direct the attention of the mapping mechanism to the appropriate relations (n-tuples of properties) to be retrieved and mapped. To further define this problem, we must specify the way in which the data space will be constructed and ordered in our relational database. Here we can draw guidance from an existing spatial data management system, described below.

A. Spatial Data Management

In the following paragraphs, we briefly outline on-going design for a map-oriented decision support system called DIMAP, developed for electronic display of maps [Chang, 1977]. DIMAP uses a distributed relational database system (RAIN), which provides a convenient language and a set of well-studied techniques which are especially appropriate for structured image representation and image data file management [Dreizen, 1980].

A map is a collection of features overlayed on one another. A terrain map, for example, may show elevation contours, roads, vegetation, bodies of water, and towns. These five features could be plotted separately, perhaps on clear plastic sheets. When the sheets are overlayed, a complete map is obtained. This type of representation is also attractive in other applications, such as medical imaging. For example, the layered strata of the retina: the vasculature, pigment distributions, and localized pathologies (e.g., photocoagulation spots, exudates, etc.) (Figure 5) can be superimposed to give a depth-stratified description of the ocular fundus.

In conventional maps, on paper, 'what you see is what you get'. The map reader cannot tune out certain features (e.g., vasculature) in order to concentrate on others (e.g., visual pigment distribution). There is no such constraint in DIMAP. At any time the user may view a single feature or multiple features in a particular region.

With DIMAP the user views maps through a window on a color monitor. It is possible to view a large map by panning the window in any direction over the map using a joystick. Panning proceeds in starts and stops; smooth panning is possible over an area nine times the area covered by a single window.

By pushing a joystick down, the user zooms (vertically) in for a more detailed view of an entity. The current map is replaced by a map of the entity. Pulling the joystick up causes an outward zoom and loss of detail apparent in the current map X. Map X is replaced by another map in which map X is represented by a simple graphical symbol.

In DIMAP the physical image and its symbolic (called logical) description can be stored separately: the physical images are stored in an image store, whereas the logical descriptions are stored in the graphically oriented relational database system (Figure 7). The immediate benefit, of course, is that the physical images can be stored on an optical videodisc player, while the highly developed database systems technology can be used to manipulate the logical description of a picture dataset.

Since DIMAP makes use of a relational database system, the user may ask questions about entities appearing in a map. All attributes known to DIMAP about a particular entity are obtained simply by pointing at the entity and striking a button. The user may then ask questions about those attributes, e.g., the numerical values of associated properties. This mapping of a relational database onto visual space in a property-oriented fashion is the function we require.

B. Visual Space vs. Data Space

The most important element in the system is the state-descriptive model of the visual environment. In artificial intelligence such models are commonly represented as a set of LISP expressions, each representing an object and an associated property list. Our approach is similar except that we represent objects as tuples in relations in a distributed relational database.

A map is not stored in the database in the way it appears in the window to the user. Rather, maps are generated by various processes that transform database information into a visual form. The overall process of transforming data for display is called visualization. Clearly, there is a close correspondence between information in the database and information on the display screen. In fact, it is the same information, represented in two different ways. From the implementer's viewpoint, this close correspondence can be confusing. What is meant by 'the map', for example? Does it refer to the image created by the visualization process, or to the alphanumeric data that is to be visualized?

To avoid confusion we agree on separate sets of terms, one for the data side of things, one for the visual side. For visual space we choose the natural terms. "Map," for example means what you think it means. For data space we choose terms distinct from the visual space terms, but which suggest visual space as far as possible. Visual space terms and their data space counterparts are listed below, and illustrated in Figure 6.

| VISUAL SPACE | | DATA SPACE |
|---|---|---|
| Map Set | <--------> | D-Map Set |
| Map | <--------> | D-Map |
| Frame | <--------> | D-Frame |
| Overlay | <--------> | Relation |
| Entity | <--------> | Tuple |

C. The Visualization Process

In DIMAP the graphic display terminal is refreshed out of the image memory of an image and graphics processor. Pictures are stored in the image memory as 2-dimensional pixel arrays of 8-bit values. Each pixel, or picture element, can be used to code for gray level or color at one point in the display.

In the simple case, the map fits entirely within one frame buffer in the image memory. The database for such a map consists of just one frame dF. The frame buffer may be loaded from dF, and the map may be viewed in its entirety.

In the general case, the digitized map is sufficiently large that it does fit within a single viewing window. This means that pixels in the map greatly outnumber points on the display screen. (An obvious parallel exists here between a human's/machine's perceptual space in its world model and the field of view imaged at any one time on the viewing window -- at the fovea/focal plane array or display monitor.) Then it becomes necessary to partition the information associated with the map. In the worst case, the map is very large and the database consists of a large number of d-frames dF1, ..., dFn. Since the user can view only one frame at a time it will often be necessary to move the window from one frame to another. Moving the window horizontally over a map is called panning. The panning problem, from the database point of view, is to load the frame buffers from the proper d-frames as the user moves the window over the map [Chang, 1979]. It should appear to the user that he is peering down through a window that can be smoothly moved laterally in any direction.

As described above, a d-frame is the minimal set of relations from which a single frame buffer may be loaded. Each relation in a d-frame corresponds to a group of entities of the same class; their visual interpretations are similar. The visual interpretations of tuples in the relation OPTIC-DISK, for example, are alike insofar as they all depict those attributes of optic disks routinely described in the ophthalmology literature. With each d-frame relation we therefore associate a graphics program that can draw a stereotypic entity characteristic of the relation. The graphical depiction of a tuple is called an icon.

Support for visualization, i.e. the process through which relations are given a visual interpretation, is the central task of DIMAP's display system. The problem is how to associate a d-frame with a frame buffer so that visualization may proceed as quickly as possible, leaving an accurate visual impression with the user. A partial solution to this problem is given in [Chang et al., 1979].

D. Selective Attention and Adaptive Tuning Mechanism

The formal hierarchically structured translation between a distributed relational database and its corresponding map set has been described above. We now address the necessary corollary of providing a mechanism for selective pictorial information retrieval.

Through our relational database system we can provide a generalized zoom capability to retrieve picture objects based upon their logical attributes [Chang, Lin & Walser, 1979]. The concept of horizontal zoom (H-zoom) is illustrated in Figure 8. A zoom graph is first displayed on the color monitor, with a vertical axis corresponding to various picture objects in a picture file, and a horizontal axis corresponding to a user-supplied selection index (which is obtained either by direct computation or by table lookup applied to each image of the picture file). For example, one selection index could be the degree of similarity of a picture object with a given reference picture object. The zoom line can then be moved to set a threshold for selection of picture objects for display. If the zoom line is moved to the left (less restrictive selection criteria), more picture objects will be selected. Thus, we have a comprehensive, or wide angle, view of the picture file. If the zoom line is moved to the right (more restrictive selection criteria), fewer picture objects will be selected, meaning a more selective, or telephoto, view of the picture file. This type of zoom is called horizontal zoom, because we are zooming in on subsets of picture objects belonging to a picture file. The traditional vertical zoom (V-zoom), on the other hand, provides close-up or wide-angle views of a single picture. Once the zoom line is set, the view line can be moved to select a picture object for display. The corresponding picture then appears in the display window.

Figure 8 illustrates picture retrieval by successive horizontal zooms. By moving the view line from one position to another and striking a function key, all picture objects between these limits having a selection index above the zoom threshold will be selected. The zoom line can again be used to further reduce the picture object set, perhaps using a different (user-supplied) selection index. In this manner a reduced picture file can then be constructed.

Finally the view line can be set in the automatic mode, and pictures will appear in the viewing window in rapid succession. If these pictures are stored as successive frames ordered chronologically, a movie is produced.

In summary, the selective attention and adaptive tuning mechanism of the horizontal and vertical zoom provide a selective visual information retrieval scheme that has been simulated in the DIMAP project and is well grounded on information retrieval methodologies applied to this spatial data management problem.

VI. OVERVIEW OF THE VISUAL ANALYZER

The visual analyzer proposed here will accept real-time optical images of actively varying scenes, and will provide as output a relational database codifying multiple properties or attributes of the scene as analyzed in time and space. These properties will be presented at the output in a form adapted to optimal exploitation by attention-directed cognitive mechanisms or by human operators using interactive graphical display.

The architecture for achieving this space-time analysis in real time employs important concepts drawn from the architecture of biological vision systems. Among these are the following:

Parallel Computation. The spatial extent of the scene is divided into small areas which are analyzed in parallel.

Image-Flow Processing. A multiple-stream pipeline processor architecture permits time-varying image data to flow continually into the visual analyzer, occupying stages vacated by preceding image data as it is moved along to subsequent processing levels.

Three-Dimensional Architecture. The combination of spatially parallel and temporally pipelined operations is made possible by a three-dimensional design realized as a stack of alternating array processor elements and interconnection elements. This permits local interaction of image data in space as well as time.

Relational Database. The hypercolumnar architecture of the visual cortex is functionally modeled as a distributed relational database in which the relations of the property submodalities operating over the spatial and temporal dimensions of the image, are both computationally derived and stored. The architecture minimizes intercommunication expense and automatically preserves registration of property analysis with the spatial and temporal events they describe.

Attention-Directed Processing. Processing resources may be actively redeployed along an "attention geodesic" defined by the operator or by higher-level cognitive mechanisms.

Applications for the visual analyzer, either alone or as as preprocessor for advanced cognitive machines, will involve real-time analysis of complex time-varying scenes. These would include robotics applications in home, industry, resources exploration, and the management of operations in inhospitable environments.

Finally we foresee a necessary symbiosis between two disparate groups: scientists interested in the mechanisms of perception and cognition, and computer engineers primarily interested in highly parallel structures for machine visual systems. Each group will become increasingly dependent upon the other for the mutual advance of their respective disciplines -- in much the manner that the community of high-energy physicists is critically dependent upon the accelerator engineers, and vice versa.

APPENDIX: VLSI DESIGN CONSTRAINTS

A. The Input Level

The visual analyzer must accommodate a variety of imaging techniques and transducers customized to particular environments (low light levels, infrared, x-ray, telescopic, etc.). For example, the first array element can be a simple CCD imaging array. Alternatively, a remote television video signal could be sent to the device and buffered in the first array element. Both approaches have advantages. One benefit of the visual analyzer structure that we have proposed is that 'focal plane' array elements can be readily replaced by others with different characteristics or even different sensory modalities, without necessarily changing the remainder of the visual analyzer -- a significant advantage in using this modular architecture.

To facilitate presentation of essential concepts, we have shown the device in Figure 1 as having a single "retina" or input. However two side-by-side image areas are required to generate retinal disparity signals for depth channels (unless, of course, structured light is used, as in some industrial robotic applications). These in turn generate vergence and accommodation movements to define the fixation point in depth.

B. Number of Processing Elements Required

A rule of thumb of the image processing community has been that it takes on the order of 1000 instructions/pixel to process an image through to the intermediate level of vision. These instructions, of course, are on a contemporary serially organized (von Neumann) computer. On this basis we can place a lower bound on the number of processing elements required to exhaustively process commercial television imagery in real-time.

The number of picture elements per second to be processed is:

no. of pixels/sec =

(480 X 640 pixels/frame)
X 30 frames/sec.

= $9.2 \times 10^6$ pixels/sec.

Allowing for line returns and interframe time cuts into the time available for video digitization. Undoubtedly similar limits on processing rate would constrain the operation of the visual analyzer. Accordingly we will assume

pixel sampling rate = 12.3 MHz (max).

We will assume 1000 instructions/pixel and a conservative 3 usec/instruction to derive

no. of processing elements =

($12.3 \times 10^6$ pixels/sec)
X $10^3$ instr/pixel)
X ( $3 \times 10^{-6}$ sec/instr)

= 36,900 PE

This estimate is quite conservative. It assumes a slow instruction time of 3 usec, whereas increased microminaturization may improve this situation. Nonetheless the contemporary estimates of instructions/pixel are for long wordlength machines. The architecture of the visual analyzer, which is designed to alleviate needless address computation and to exploit an estimated five-fold temporal/spatial redundancy in television imagery -- should compensate for this difference.

C. Total Area of Silicon Required

Assuming continuing advances in microlithography and that linewidths continue to halve every 2.3 years, we anticipate that a contemporary modest microprocessor in 1985 will occupy approximately 1 $mm^2$ of silicon surface. On this basis, we require $4 \times 10^4$ $mm^2$ of silicon surface, or in the design proposed in Section IIE we require 4 array elements, each 100 mm X 100 mm. By 1990 each array element could be as little as 25 mm X 25 mm -- comparable in size to the human retina, if semiconductor technology continues its plunge toward ever smaller linewidths.

In summary, the design proposed is conservative in its estimate, and continuing improvements in solid-state technology should in time allow even less silicon per array element, possibly require fewer processing elements of greater computational power, but not otherwise fundamentally change the highly parallel structure of the real-time visual analyzer.

REFERENCES

J. M. Chang and S. K. Chang, "Database Alerting Techniques for an Activity Management System," Proceedings of 1980 International Computer Symposium, Taipei, Taiwan, Republic of China, December 1980.

S. K. Chang and W. H. Cheng, "Database Skeleton and Its Application to Logical Database Synthesis," IEEE Transactions on Software Engineering, Vol. SE-4, No. 1, January 1978, pp. 18-30.

S. K. Chang and J. S. Ke, "Translation of Fuzzy Queries for Relational Database System," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-1, No. 3, July 1979, pp. 281-294.

S. K. Chang, B. S. Lin, and R. L. Walser, "Generalized Zooming Techniques for a Pictorial Database System," Proceedings of the National Computer Conference AFIPS, June 1979, pp. 147-156.

S. K. Chang, J. Reuss, and B. H. McCormick, "An Integrated Relational Database System for Pictures," Proc. Workshop on Picture Data Description and Management, IEEE Computer Society, Chicago, 1977, pp. 142-149.

Y. D. Chang, "Frame Staging Technique for Pictorial Database System," Technical Report KSL-40, Knowledge Systems Laboratory, University of Illinois at Chicago Circle, 1979.

E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," Communications ACM, Vol. 13, No. 6, June 1970.

C. J. Date, An Introduction to Database Systems, Second Edition, Addison-Wesley, 1977.

H. M. Dreizen, "RAIN, A Relational Algebraic Interpreter," Version 2.5 Reference Manual, Technical Report KSL-50, Knowledge Systems Laboratory, University of Illinois at Chicago Circle, September 1980, pp. 129.

L. Dreschler and H. H. Nagel, "Volumetric Model and 3-D Trajectory of a Moving Car Derived from Monocular TV-Frame Sequence of a Street Scene," Int'l Joint Conf. Artificial Intelligence, Vancouver, Canada, August 1981.

C. Enroth-Cugell and J. D. Robson, "The Contrast Sensitivity of Retinal Ganglion Cells in the Cat," J. Physiology (London), Vol. 187, 1966, pp. 517-522.

R. P. Erickson, "Parallel 'Population' Coding in Neural Feature Extraction," In The Neurosciences, Third Study Program, F. O. Schmitt and F. G. Worden, eds., MIT Press, Cambridge, MA, 1974, p. 155.

R. C. Emerson and G. L. Gerstein, "Simple Striate Neurons in the Cat. II. Mechanisms Underlying Directional Asymmetry and Directional Selectivity," J. Neurophysiology, Vol. 40, No. 1, 1977, pp. 136-155.

J. J. Gibson, The Ecological Approach to Visual Perception, Houghton Mifflin, Boston, MA, 1979.

J. J. Gibson, The Senses Considered as Visual Systems, Houghton Mifflin, Boston, MA, 1966.

A. W. Goodwin, G. H. Henry, and P. O.Bishop, "Direction Selectivity of Simple Striate Cells: Properties and Mechanism," J. Neurophysiology, Vol. 38, 1975, pp. 1500-1523.

B. K. P. Horn and B.G. Schunck, "Determining Optical Flow," Proc. Darpa Image Understanding Workshop, April 1981, pp. 144-156.

D. H. Hubel, "Architecture of the Monkey Striate Cortex," Neurosciences Research Program Bulletin, Vol. 15, MIT Press, Cambridge, MA, 1977, p. 327.

G. Johansson, "Visual Motion Perception," Sci. American, Vol. 232, No. 6, June 1975, pp. 76-88.

D. N. Lee, "The Optic Flow Field: The Foundation of Vision," Philosophical Trans. Royal Soc. London, Vol. B, No. 290, 1980, pp. 169-179.

D. Marr, "Early Processing of Visual Information," Philosophical Trans. Royal Soc. London, Vol. B, No. 275, 1976, pp. 483-524.

D. Marr, "Representing Visual Information," In Computer Vision Systems, A. Hanson and E. M. Riseman, eds., Academic Press, New York, NY, 1979.

D. Marr and E. Hildreth, "Theory of Edge Detection," Proc. Royal Soc. London, Vol. B, 1980, pp. 187-217.

D. Marr and H. K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," Proc. Royal Soc. London, Vol. B, No. 200, 1978, pp. 269-294.

D. Marr and T. Poggio, "Cooperative Computation of Stereo Disparity," Science, Vol. 194, October 15, 1979, pp. 283-287.

D. Marr and S. Ullman, "Directional Selectivity and Its Use in Early Visual Processing," Proc. Royal Soc. London, Vol. B, No. 211, 1981, pp. 151-180.

B. H. McCormick, E. Kent, and C. R. Dyer, "A Visual Analyzer for Real-Time Interpretation of Time-Varying Imagery," To be published in Multi-Computer Algorithms and Image Processing, L. Uhr, ed., Academic Press, New York, NY, 1981.

R. S. Michalski, "Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and an Algorithm for Partitioning Data into Conjunctive Concepts," Int'l J. Policy Analysis and Information Systems, 3, 1980.

W. Richards, "The Neuron vs. the Ensemble: An Analysis in Depth," Neurosciences Research Program Bulletin, Vol. 15, No. 3, MIT Press, Cambridge, MA, 1977, p. 439.

W. B. Thompson, "Combining Motion and Contrast for Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 2, No. 26, 1980, pp. 543-549.

S. Ullman, "Analysis of Visual Motion by Biological and Computer Systems," Computer, Vol. 14, No. 8, IEEE Computer Society, August 1981, pp. 57-69.

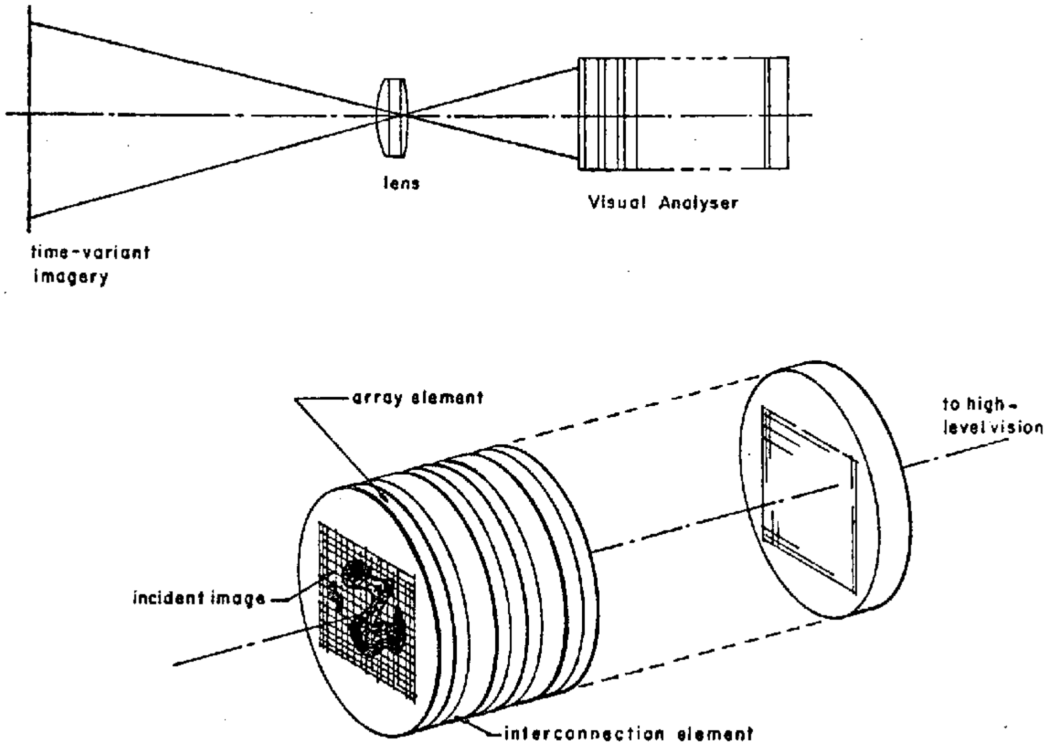



Figure 1. Visual analyzer as stack of array and interconnection elements.

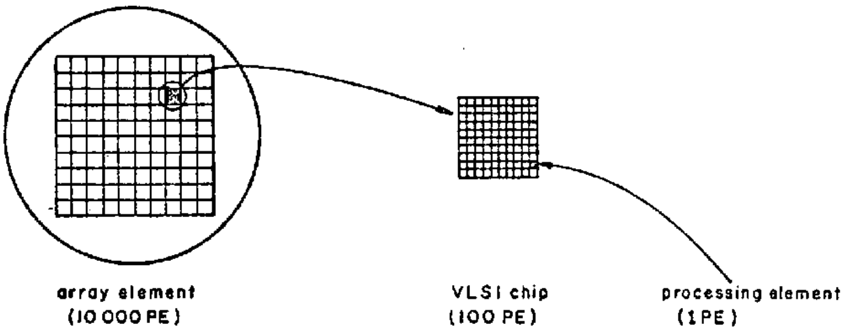



Figure 2. Mosaic construction of an array element.

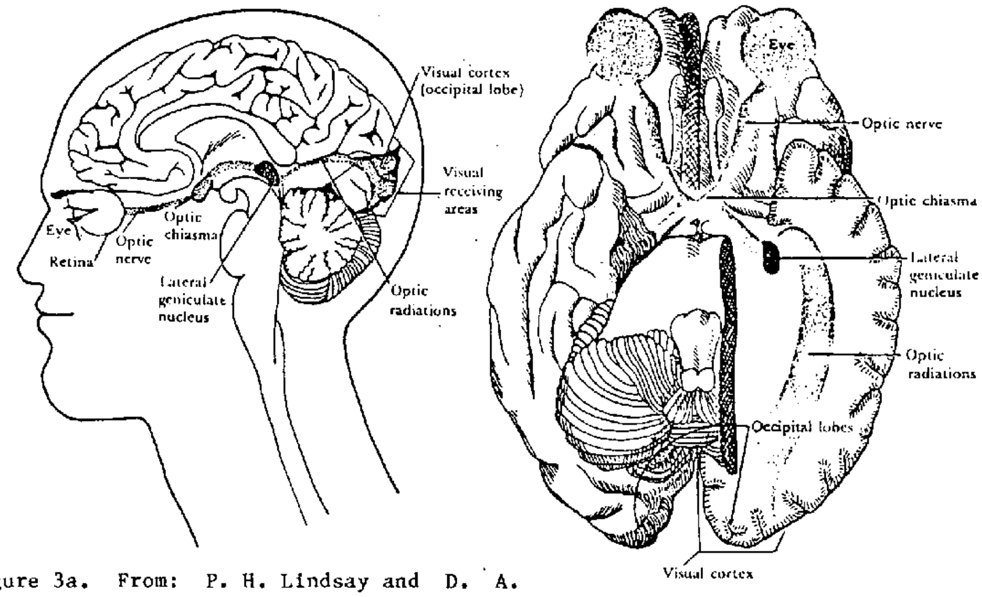



Figure 3a. From: P. H. Lindsay and D. A. Norman, "Human Information Processing, 2nd Ed.", Academic Press, New York, 1977.

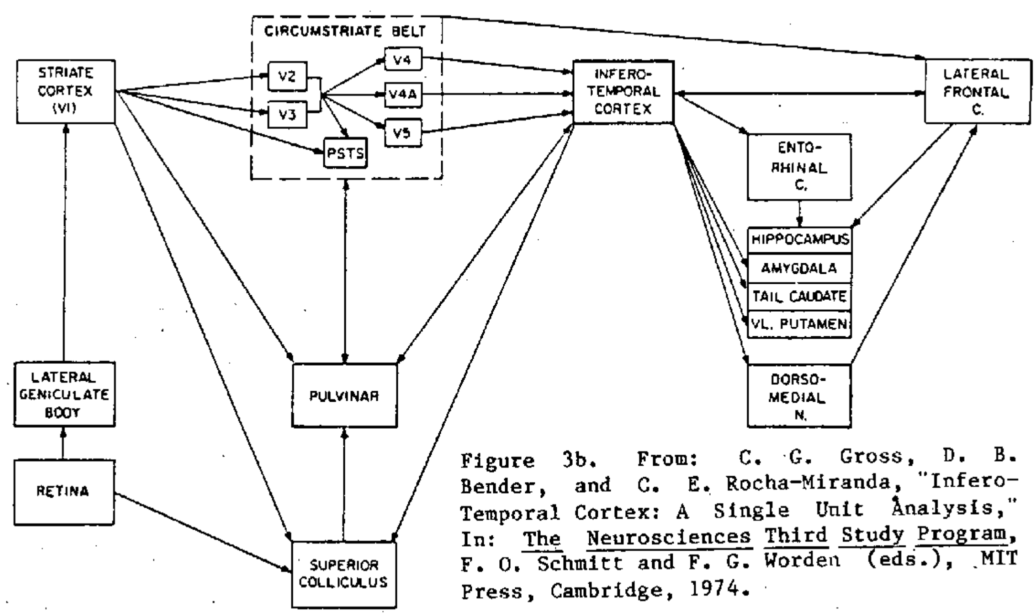



Figure 3b. From: C. G. Gross, D. B. Bender, and C. E. Rocha-Miranda, "Infero-Temporal Cortex: A Single Unit Analysis," In: The Neurosciences Third Study Program, F. O. Schmitt and F. G. Worden (eds.), MIT Press, Cambridge, 1974.

Figure 3. Major structures and connections of the human visual system, shown in anatomical (above) and data path (below) relations. The area labeled "visual cortex" in the upper figure includes the striate cortex (primary visual cortex) and circumstriate belt (secondary visual cortex) of the lower figure. The "mainline" path of the classical visual projection is from retina to lateral geniculate to visual cortex. The many other structures shown in the lower figure are now known to have a variety of visual functions.

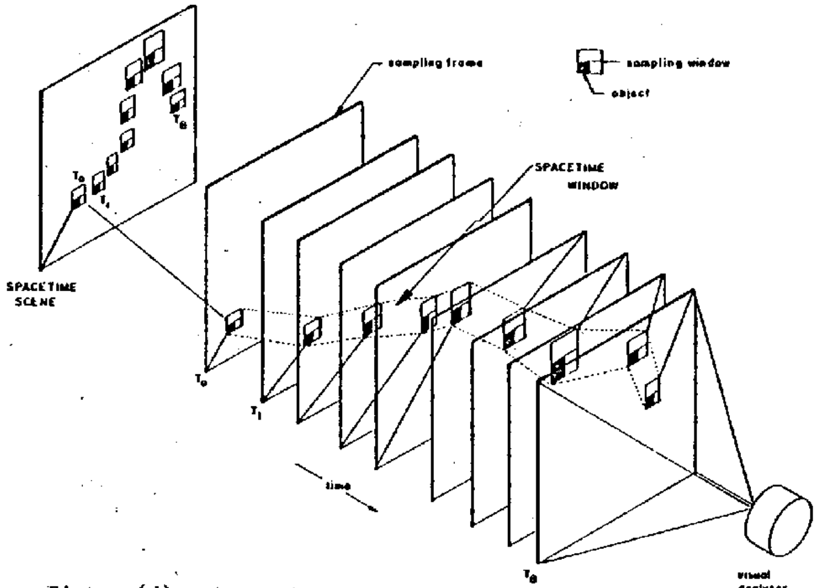



Figure (d). Attention-driven window tracking an abstract event in space-time.

Figure (c). Results of the corner-finding in time-varying imagery applied to the taxicab of figure (a). (From Dreschler and Nagel, 1981)

Figure (a) and (b). First (a) and last (b) from a series of 22 TV frames showing a moving taxicab turning right. (From Dreschler and Nagel, 1981)

Figure 4. The Attention-Driven Window

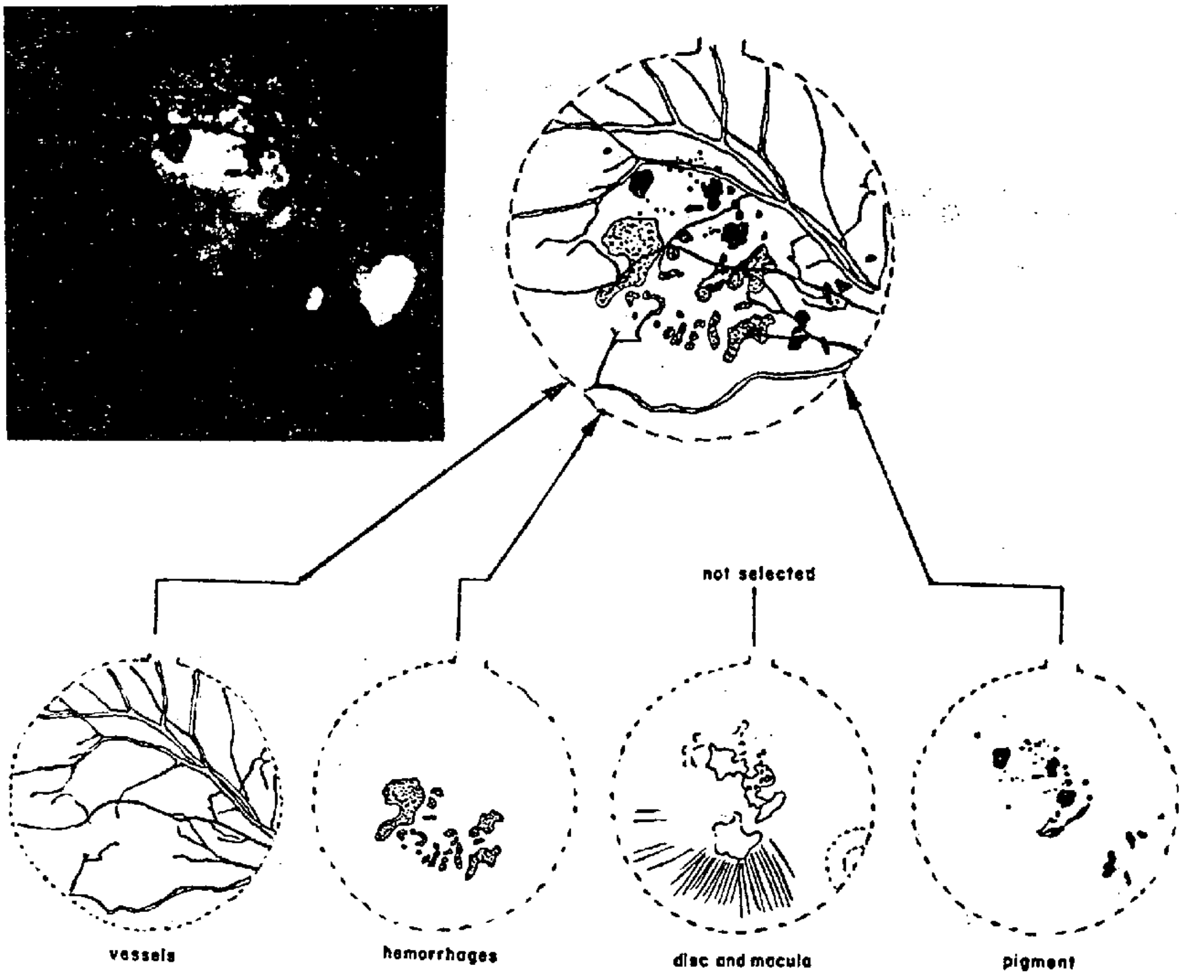



Figure 5. Depth-Stratified Description of the Retina.

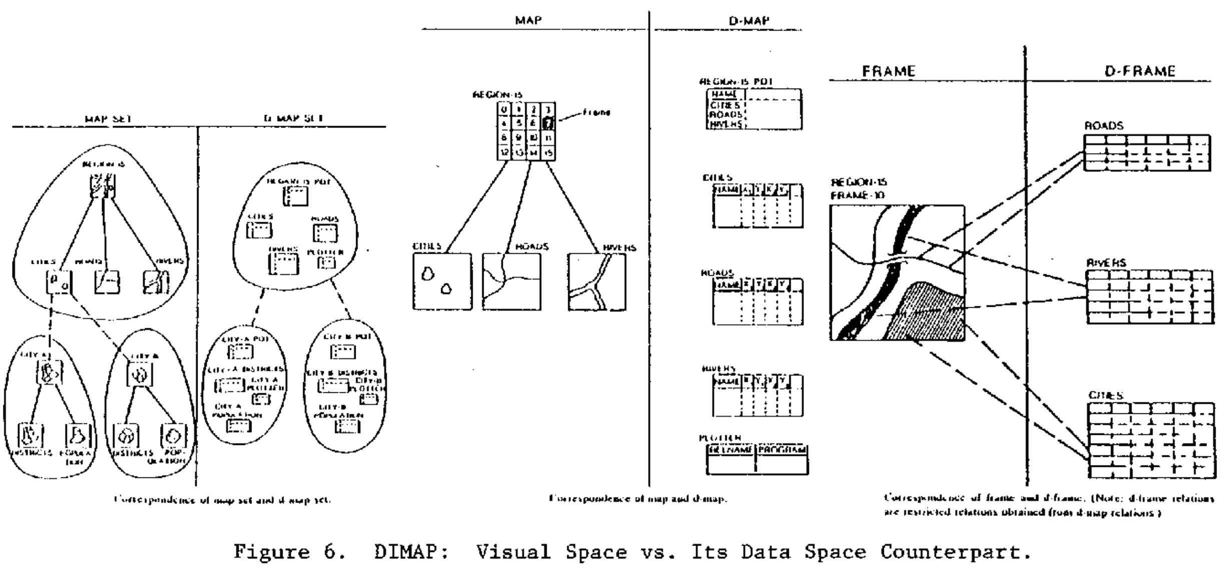



Figure 6. DIMAP: Visual Space vs. Its Data Space Counterpart.

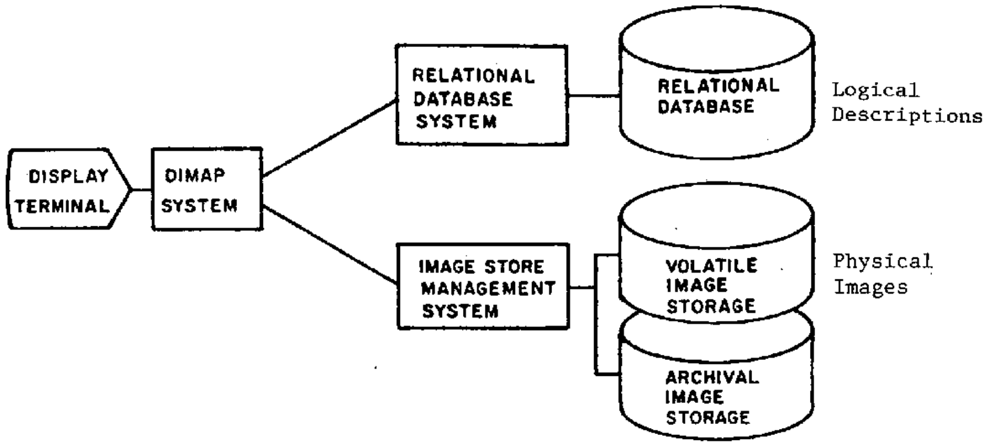



Figure 7. DIMAP: Separation of Logical Descriptions and Physical Images

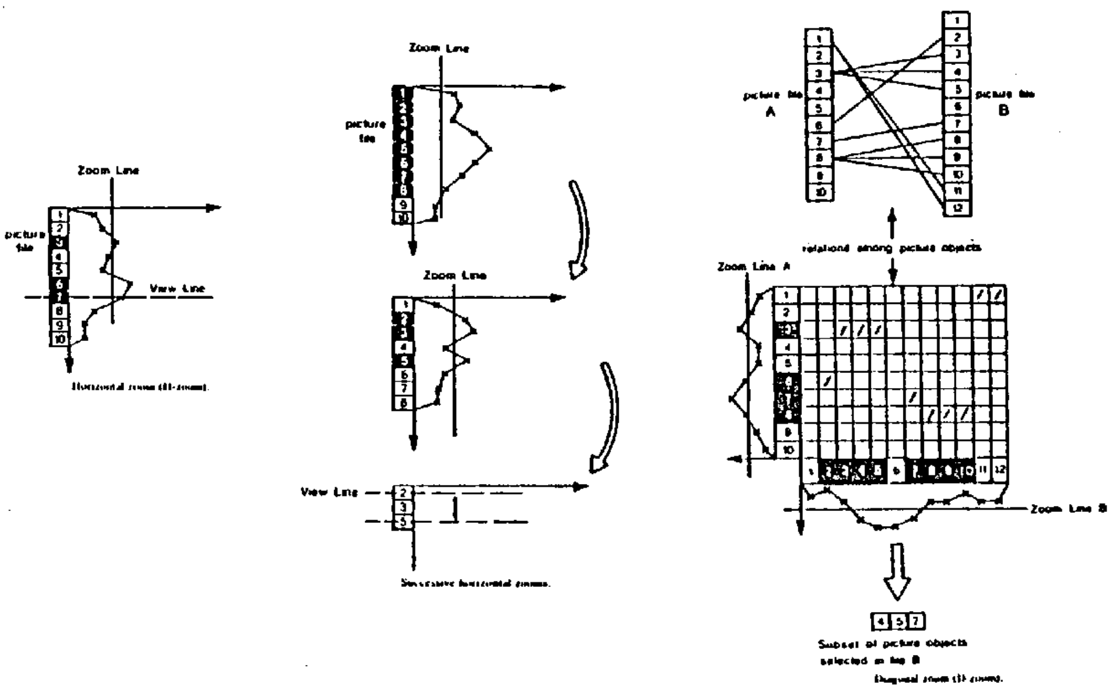



Figure 8. DIMAP: Horizontal Zoom Over Television Data Sets

A part of the paper,

"A Cognitive Architecture for Computer Vision",

written by Prof. McCormic et al.

will be appeard in a Academic Press's publication in 1982.

# FIFTH GENERATION COMPUTER ARCHITECTURE ANALYSIS

Philip TRELEAVEN
Computing Laboratory
University of Newcastle upon Tyne
Newcastle upon Tyne, England

Fifth generation computer systems will represent a unification of research into VLSI processors and into distributed processing. Each computer system will consist of a network of computing elements supporting an individual application or need. VLSI processor research will allow a computing element to provide either a general-purpose or a special-purpose function, and range in power from a main-frame computer to a miniature microcomputer. Distributed processing research will allow a network to be physically dispersed across a country or building, or to be physically close as on a single highly integrated chip. For this spectrum of fifth generation computer systems to be programmed as individual computers and for their computing elements to work together, it is necessary to have a single computer architecture to which they all conform. The basis of this fifth generation computer architecture could be control flow, data flow or reduction computers. Here I examine these three types of computer and present classifications for their underlying concepts, with the aim of analysing their probable contribution to fifth generation computer architecture.

## 1. INTRODUCTION

For more than thirty years the principles of computer architecture design have largely remained static, based on the von Neumann computer. This von Neumann architecture includes:

1. a single computing element incorporating processor, communications and memory

2. a linear organisation of fixed size memory cells

3. a linear address space of cells

4. a low level machine language (instructions perform simple operations on elementary operands)

5. a sequential, centralised control of computation

However, as computing moves from a sequential world to one in which large numbers of computing elements are to be programmed to work together in computer systems then it is necessary to have a new computer architecture for decentralised computing to which all computing elements conform. This I claim is true whether the systems consist of geographically distributed main-frame computers, or in fact are composed of miniature microcomputers on a single VLSI chip.

This new decentralised, or so called fifth generation [1], computer architecture must also handle the rapid developments in microelectronics. For example Mead and Conways' Introduction to VLSI Systems [15] has stimulated the design of novel VLSI processors. These VLSI processor designs are based on the concepts of simplicity and replication. They are implemented by only a few different types of simple computing element, are characterised by simple and regular interconnections of elements, and use extensive pipelining and multiprocessing to increase performance. Examples range from powerful special-purpose devices built from identical elements, to parallel non-von Neumann computer systems built from replicated microcomputers.

Novel decentralised computers which might form the basis of a fifth generation computer architecture are under development in well over thirty laboratories in the United States, Japan and Europe. They may be broadly classified as control flow, data flow and reduction computers. In control flow computers explicit flows of control cause the execution of instructions; in data flow computers the availability of operands triggers the execution of the operation to be performed on them; whereas in reduction computers the requirement for a result triggers the operation that will generate the value.

Although there are these three distinct areas of research, each laboratory has developed its own individual stored program organisation and machine organisation. Across this range of designs there is, however, a significant sharing of concepts. In this paper I classify and analyse these concepts. I start by briefly surveying the special-purpose and general-purpose VLSI processors under development to illustrate the types of computing element that will be found in fifth generation computer systems.

## 2. VLSI PROCESSORS

Microprocessors containing 100,000 transistors are starting to become commonplace [11]. Examples include the recently announced 32-bit microprocessors from Intel, the iAPX 432, and from Bell Laboratories, the MAC-32. The term VLSI processor is normally viewed as being synonymous with such designs. However the making of a large scale single microprocessor in VLSI scaled to submicron dimensions (and the escalating costs of designing and testing such complex processors) becomes self-defeating. The diffusion delays in a wire scale up quadratically while the delay of a MOS switch scales down linearly. These facts, together with the alternative design philosophy advanced by Mead and Conway [15], have led to a whole range of new, and as yet unconventional processors, equally deserving of the term VLSI processors.

These VLSI processors possess one or more of the following properties:

1. The processor is implementable by only a few different types of simple computing element.

2. The processors' data and control paths are simple and regular, so that the elements may be connected by a network with local and regular interconnections. Long distance or irregular communication is thus minimised.

3. The processor uses extensive pipelining and multiprocessing. In this way a large number of elements are active at one time so that the overall computational rate of the simple elements is high.

A whole spectrum of processors are under development based on this philosophy of simplicity and replication, as illustrated by Figure 1.

Special-purpose VLSI processors are usually designed to function as peripheral devices attached to a conventional host computer. They may be a single chip built from a replication of simple cells, or a system built from identical simple chips, or even a combination of these two approaches. Examples of such special-purpose processors include Kung's Systolic Arrays [9], Rivest's RSA Cipher chip [17], and Clark's Geometry Engine [4].

Novel designs for conventional microprocessors are also emerging. Reduced instruction set computers, such as Berkeley's RISC I [16] microprocessor, are reversing the general trend of designing microprocessors, and even main-frame computers, with increasingly complex instruction sets and associated architectures.

Their designers believe that by a judicious choice of a simple set of instructions and corresponding simple machine organisation, such a high instruction rate can be achieved that the overall processing power can exceed that of processors which implement more powerful instructions. This approach is contrasted by that of the MIT SCHEME-79 chip [18] whose high-level instructions are designed to closely match those of the LISP programming language. Either approach can lead to a microprocessor with a short design time.

The remaining two categories of processor, shown in Figure 1, aim to exploit VLSI by constructing computer systems from a number of identical general-purpose elements. In the tree-machine category the design centres on the machine organisation (i.e. the way a machine's resources are configured and allocated). In the non-von Neumann computers category the design centres on the program organisation (i.e. the way machine code programs are represented and executed).

A tree-machine is a collection of simple computing elements connected together as a binary tree. There is no global communication, only communication between a parent and its child in the tree, and between the root of the tree and the external world. This computer organisation gives rise to integrated circuits having regular interconnections, local communication, and many repetitions of a single cell. These integrated circuits, in turn, may be assembled into regular patterns at the printed-circuit board and backplane level to construct machines with thousands of processors. Examples of such tree-machines are being investigated at Berkeley [7] and Caltech [3].

In the final area the VLSI processors' program organisation is not based on the traditional (von Neumann) control flow organisation, but on alternative naturally parallel organisations such as data flow [6] or reduction [14]. These program organisations are distinguished by the form of instructions, by the way instructions manipulate their arguments, and by the pattern of control in programs. Examples of such non-von Neumann computers include Arvind's Dataflow machine [2] and Mago's Cellular Tree Machine [14], based on reduction.

```
special-purpose <------------spectrum-----------> general-purpose

    |            |            |            |            |

 special-     special-      novel         tree-      non-von
 purpose      purpose    conventional   machines     Neumann
  chips       systems   microprocessors             computers
```

Figure 1: Spectrum of Novel VLSI Processors.

For fifth generation computer systems, if all these special-purpose and general-purpose VLSI processors are to be potential building blocks for larger systems, then it is necessary for them to conform to some overall computer architecture. That is they must conform to some overall program organisation and to some overall machine organisation.

## 3. PROGRAM ORGANISATIONS

I use the term program organisation to cover the way machine code programs are represented and executed in a computer architecture. This examination of the control flow, data flow and reduction program organisations starts by classifying their underlying mechanisms [20].

### 3.1. Classification

For a program organisation there are two fundamental computation mechanisms, which I refer to as the data mechanism and the control mechanism. The data mechanism defines the way a particular argument is used by a number of instructions. There are three sub-classes:

1. by literal - where an argument is known at compile time and a separate copy is placed in each accessing instruction.

2. by value - where an argument, generated at runtime, is shared by replicating it and giving a separate copy to each accessing instruction, this copy being stored as a value in the instruction.

3. by reference - where an argument is shared by having a reference to it stored in each accessing instruction.

The control mechanism defines how one instruction causes the execution of one or more other instructions, and also the resulting control pattern.

There are again three sub-classes:

1. sequential - where in general a single thread of control signals an instruction to execute and is passed from one instruction to another.

2. parallel - where control signals the availability of arguments and an instruction is executed when all its arguments (e.g. input data) are available.

3. recursive - where control signals the need for arguments and an instruction is executed when one of the output arguments it generates is required by the invoking instruction. Having executed it returns control to the invoking instruction.

To facilitate comparisons of control flow, data flow and reduction, simple program representations for the statement "a=(b+1)*(b-c)" are used. Although this example is simple, the concepts it illustrates are equally applicable to more complex operations and data structures.

### 3.2. Control Flow

A control flow program organisation has a "by reference" data mechanism, with references embedded in instructions being used to access the contents of shared memory cells. Traditional control flow has a "sequential" control mechanism with a single thread of control being passed from instruction to instruction. There are however other forms of control flow with a "parallel" control mechanism [8,10].

Figure 2 shows both sequential and parallel control flow programs. Each instruction consists of an operator followed by one or more operands which are literals or references. For instance a dyadic operation such as "+" is followed by three operands, the former two "b and 1" provide the input data

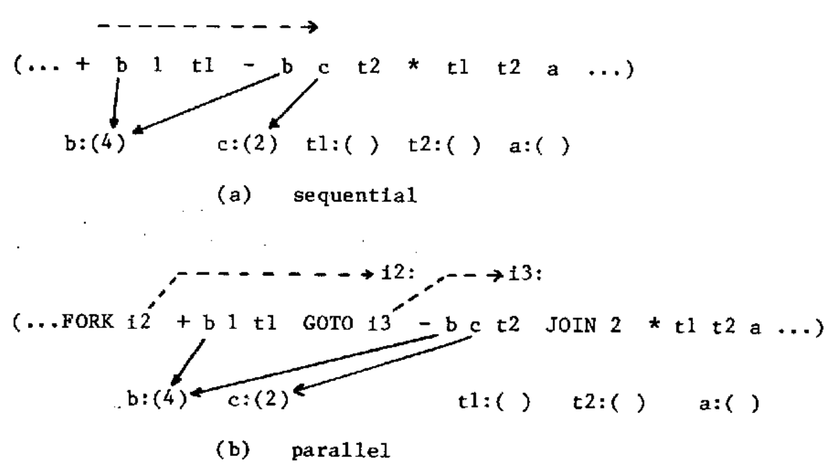



Figure 2: Control Flow Programs for a=(b+1)*(b-c).

and the latter "tl" is the reference to the shared memory cell for the result. Evaluation of control flow instructions is implicitly sequential. Explicit sequential or parallel patterns of control are caused by control operators such as GOTO and FORK-JOIN. As shown in Figure 2b, the FORK spawns the subtraction instruction at address "i2" and passes an implicit flow of control to the addition instruction. These two parallel threads of control are synchronised by the JOIN instruction, which releases a single thread to activate the multiply instruction.

The main features of control flow, resulting from the choice of data and control mechanisms, include: (i) data is passed indirectly between instructions via references to shared memory cells, (ii) literals are stored in instructions, and (iii) execution is implicitly sequential, with additional explicit sequential and parallel patterns being obtained from control operators. Due to these features, the form of sharing supported is "update in place", which means control flow organisations incur overheads when evaluating simple expressions but have advantages when manipulating larger data structures.

## 3.3. Data Flow

A data flow program organisation has a "by value" data mechanism and a "parallel" control mechanism. Both mechanisms are supported by a single scheme, data tokens, which convey data from a producer to a consumer instruction, and also contribute to the activation of the consumer instruction.

In Figure 3 each data flow instruction consists of an operator, two inputs which are either literals or "unknown" operands defined by empty bracket "( )" symbols, and a reference such as "i3/1" defining a consumer instruction and argument position for the result data token.

Figure 3 illustrates the sequence of execution for the program fragment "a=(b+1)*(b-c)" using a black dot to indicate the availability of a data token. When an instruction is enabled for execution all its arguments are "known". The operator then consumes the data tokens, performs the required operation, and using the embedded reference stores a copy of the result data token into the consumer instruction(s).

The main features of data flow, resulting from its choice of mechanisms, include: (i) data is passed directly between instructions, (ii) literals are again stored in instructions, (iii) execution consumes data tokens, so the values are no longer available to be re-used, and (iv) there is no concept of shared data as embodied in the traditional notion of a variable. Due to these features, the form of sharing supported is that of taking "separate copies", which means data flow is very efficient for the evaluation of simple expressions, but is less so for manipulating structures which need to be updated in place. In addition, due to the data token scheme, flows of data and control are identical causing problems for the conditional evaluation of alternatives.

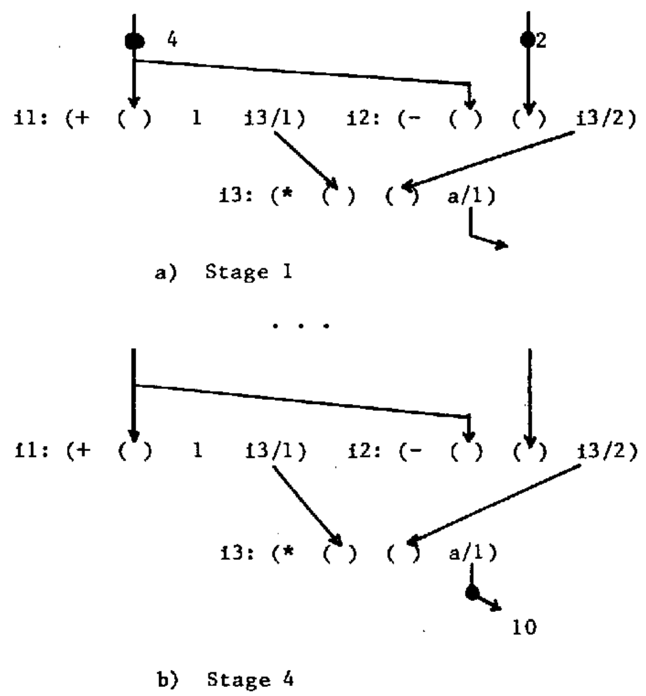



Figure 3: Data Flow Program for a=(b+1)*(b-c).

Control flow and data flow programs are built from fixed size instructions whose arguments are atoms - simple operators and operands. Higher level program structures are built from sequences of these primitive instructions, plus explicit control operators such as CALL and RETURN. In contrast, reduction programs are built from expressions. Higher level program structures, instructions and arguments are all expressions, removing the need for explicit CALL-RETURN operators.

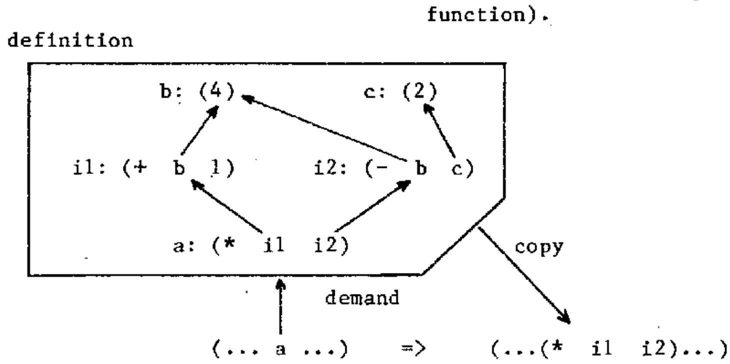
### 3.4. Reduction

There are two basic types of reduction program organisation called string reduction and graph reduction. String reduction [14] has a "by value" data mechanism and graph reduction [12] has a "by reference" data mechanism. Both forms of reduction have a "recursive" control mechanism. Demanding the result of the definition "a" where "a=(b+1)*(b-c)" means we require that the embedded reference "a", causing the execution, is rewritten in a simpler form. Because of these attributes only one definition of "a" may occur in a program and all references give the same value, a property known as referential transparency.

The basis of a string reduction program organisation is that each instruction accessing a particular definition will take and manipulate a separate copy of the definition. Each instruction consists of an operator followed by literals or embedded references used to demand the corresponding input operands. As illustrated in Figure 4, at Stage 1 some instruction, containing the reference "a", demands the value associated with the definition "a". This causes a copy of the definition to be loaded into the instruction, overwriting the reference "a". Next the multiplication operator demands the values associated with "i1" and "i2", and suspends while the addition and subtraction operators demand their inputs. When, as shown at Stage 3, all the arguments of an expression are atoms, the expression is reduced, until at Stage 5 the multiplication is replaced by the "10" which is the value of "a".

The basis of a graph reduction program organisation is that each instruction accessing a particular definition will manipulate references to the definition. That is, graph manipulation is based on the sharing of arguments using pointers. As illustrated in Figure 5, at Stage 1 some instruction demands the value associated with the definition "a", but instead of taking a copy of the definition execution merely traverses the reference, with the aim of reducing the definition and returning with the actual value. One of the ways of identifying the original source of the demand for "a", and thus supporting this return, is to embed a reference in the definition. This traversal of the definition and reversal of the references (as shown in Figure 5) is continued until all the arguments of an expression are atoms and the expression can be reduced, and the result returned.

The main features of reduction include: (i) program structures, instructions and arguments are all expressions, (ii) there is no concept of data storage such as a variable, (iii) there are no additional sequencing constraints over and above those implied by demands for operands, and (iv) demands may return both simple or complex arguments such as a function (as input to a higher order function).




(a) Stage 1

. . .

(...(* (+ 4 1) (- 4 2))...) => (...(* 5 2)...) => (...10...)

(b) Stage 3 to 5

Figure 4: String Reduction Program for a=(b+1)*(b-c).

Due to the individual features of the two forms of reduction, string reduction has advantages when manipulating simple expressions and graph reduction has advantages when larger structures are involved. Both forms are constrained by the absence of explicit output arguments.

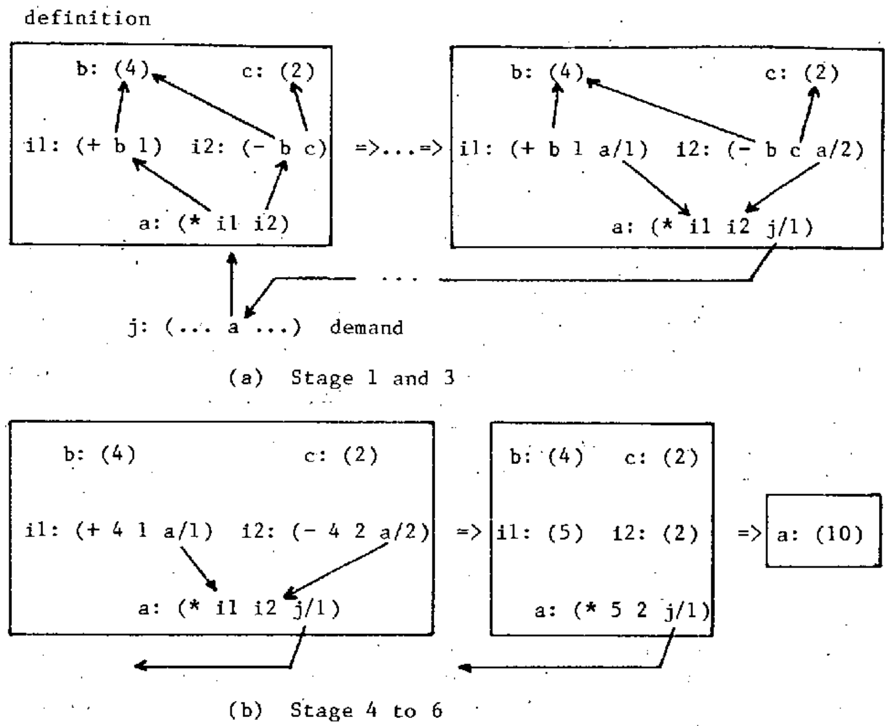



Figure 5: Graph Reduction Program for a=(b+1)*(b-c).

As a summary of this section the relationship of these data and control mechanisms to the program organisations is shown in Figure 6.

| Control Mechanisms | | Data Mechanisms: by value (& literal) | Data Mechanisms: by reference (& literal) |
|---|---|---|---|
| | sequential | | von Neumann control flow |
| | parallel | data flow | parallel control flow |
| | recursive | string reduction | graph reduction |

Figure 6: Data Mechanisms and Control Mechanisms.

## 4. MACHINE ORGANISATIONS

I use the term machine organisation to cover the way a machine's resources are configured and allocated to support a program organisation.

### 4.1. Classification

An examination of research into novel computer architectures under development reveals three basic classes of machine organisation [23]. I call these centralised, packet communication, and expression manipulation. Since these machine organisations relate closely to the way program execution is represented the three are often equated and confused with, respectively, control flow, data flow and reduction program organisations.

1. Centralised - this organisation consists of a single processor, communication and memory resource, as shown in Figure 7. It views an executing program as having a single active instruction which passes execution to a specific successor instruction. The state of execution is often held in registers or stacks.

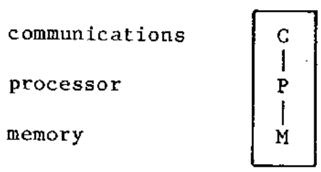



Figure 7: Centralised Machine Organisation.

2. Packet communication - this organisation consists of a circular instruction execution pipeline of resources in which processors, communications and memories are interspersed with "pools of work". This is illustrated by Figure 8. The organisation views an executing program as a number of independent information packets all of which are conceptually active, that split and merge. For a parallel computer, packet communication is a very simple strategy for allocating packets of work to resources. Each packet to be processed is placed with similar packets in one of the "pools of work". When a resource becomes idle it takes a packet from its input pool, processes it and places a modified packet in an output pool, and then returns to the idle state. Additional parallelism is obtained either by having a number of identical resources between two pools; or by replicating the circular pipelines and connecting them by the communications.

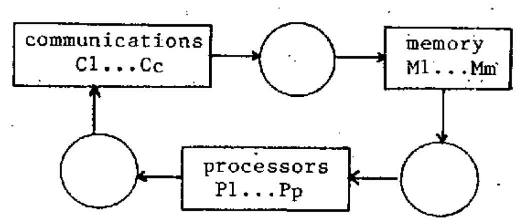



Figure 8: Packet Communication Machine Organisation.

3. Expression manipulation - this organisation consists of identical resources usually organised into a regular structure such as a vector or tree, as shown in Figure 9. Each resource contains a processor, communication and memory capability. The organisation views an executing program as consisting of one large nested program structure, parts of which are active while other parts are temporarily suspended. In an expression manipulation organisation the adjacency of items in the program structure is important, and the memories in this machine structure maintain the adjacency of items in the program structure. Each resource performs work in its part of the overall program structure.

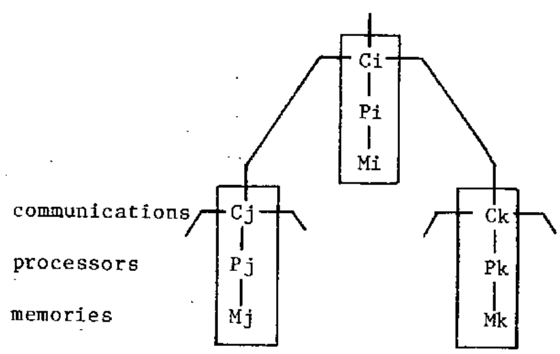



Figure 9: Expression Manipulation Machine Organisation.

### 4.2. Centralised

The most natural usage of a centralised machine organisation is in the support of a sequential control flow program organisation. This is clearly the basis of all traditional computers. The advantage of this organisation is its familiarity and simplicity, both for resource allocation and implementation. Its disadvantage is the difficulty of extending the organisation to utilise parallelism.

When considering the use of a centralised machine organisation to support other program organisations, then for data flow it is difficult to perceive of a suitable organisation because of the need to record the large set of potentially executable instructions. However for a reduction program organisation, because of sequential computation rules, various centralised machine organisations are possible [13].

For any machine organisation supporting reduction, there are two basic problems: firstly, managing dynamically the memory of the program structure being transformed, and secondly keeping control information about the state of the transformation. Solutions to the memory management problem include: (i) representing the program and instructions as strings, e.g. "((*) ((+) (b) (1)) ((-) (b) (c))" which can be manipulated without altering the meaning of the surrounding structure, and (ii) representing the program as a graph structure with pointers, and using "garbage collection". Solutions to the control problem include: (i) to use control stacks which record, for example, the left ancestors of an instruction, and (ii) pointer reversal where the ancestor is defined by a "reversed" pointer stored in the instruction. An example of a centralised organisation is the string reduction machine [13] implemented at the GMD Laboratory in West Germany.

## 4.3. Packet Communication

Specific packet communication machine organisations may be used to support control flow, data flow or reduction program organisations. There are two subsidiary forms of packet communication organisations, which I refer to as "token storage" and "token matching", distinguished in the way instruction execution is synchronised. In the token storage scheme tokens are actually stored into an instruction, or a copy of the instruction, and an instruction executes when it has received all its tokens. In the token matching scheme a "matching" mechanism is used to group together tokens destined for the same instruction and to activate the instruction when the group is complete. Organisations based on these two schemes are illustrated by Figure 10 and 11.

In Figure 10 the token packets are in the input pool of the update unit. This unit takes in single tokens and stores them in the memory unit. Certain of these tokens may complete the inputs for an instruction, thus enabling it for execution. For these instructions the update unit places their addresses in its output pool. The fetch unit uses these instruction addresses to retrieve the corresponding instructions and place them in its output pool for execution.

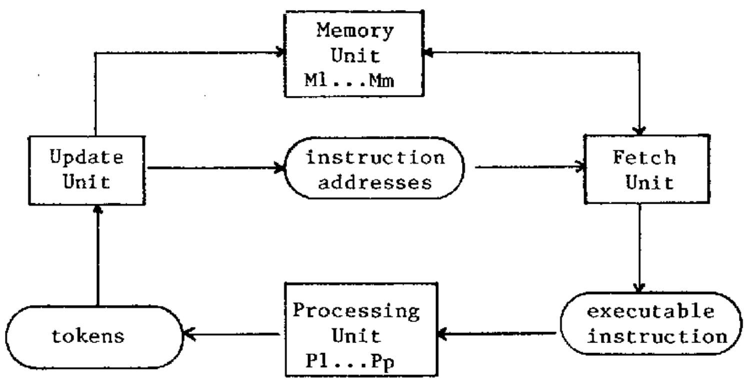



Figure 10: Packet Communication with Token Storage.

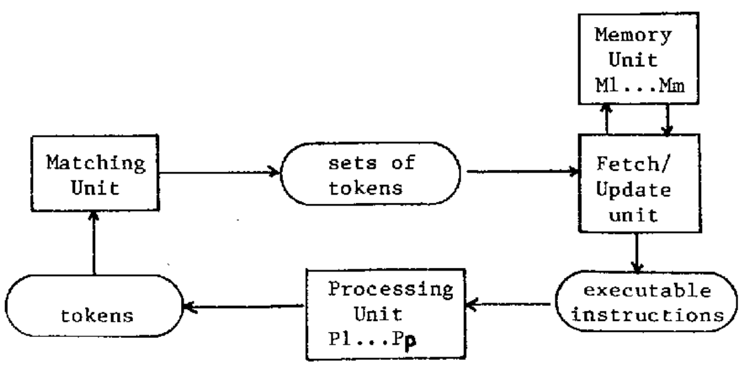



Figure 11: Packet Communication with Token Matching.

Examples of this scheme include: the LAU system [19] which uses control flow concepts to support data flow, the MIT data flow computer [6], and the AMPS system [12] which supports graph reduction.

In Figure 11, where synchronisation is based on a matching mechanism, token packets form the input pool of the matching unit. This unit forms them into sets, temporarily storing the set until complete whereupon the set is released to the fetch/update unit. This unit forms executable instructions by merging the values from a set of tokens with a copy of their consumer instruction.

Examples of this scheme include: the GCF control flow computer [8], a computer [10] supporting both control flow and data flow, the Irvine dataflow computer [2], and the Manchester data flow computer [24].

### 4.4. Expression Manipulation

As with packet communication, specific expression manipulation machine organisations may be used to support control flow, data flow and reduction program organisations. A major advantage of expression manipulation is that the organisation is sympathetic to the utilisation of VLSI. VLSI encourages the use of the following machine organisation ideas [21]: (i) a replication of identical computing elements, each with capabilities for processing, communication and memory; (ii) a group of elements being functionally equivalent to a single element, to overcome problems of increasing miniaturisation; (iii) concurrency to counteract the simplicity of the individual computing elements; and (iv) locality of reference to reduce system-wide communications. Expression manipulation clearly matches these ideas. However, a disadvantage of expression manipulation is the lack of a simple strategy for dynamic resource allocation.

Examples of expression manipulation organisations: for control flow are the Berkeley [7] and Caltech [3] tree-machines, and the RCF computer [22]; for data flow is the DDM1 computer [5]; and for reduction is the Cellular tree-machine [14] which supports Backus' FP class of languages as its instruction set.

As a summary of this examination Figure 12 shows the program and machine organisations for various computer architectures under investigation.

## 5. CONCLUSION

Just as Mead and Conway's VLSI design philosophy [15] provides a whole new way to "design using electronics", fifth generation computer architecture is likely to provide a whole new way to "build computer systems". I believe fifth generation computers will consist of replicated general-purpose computing elements, as well as special-purpose computing elements that allow individual computers to be specialised. Hence these computing elements will need to co-operate at both the program organisation and machine organisation levels.

For the program organisation it is significant that control flow, data flow and reduction regard the "by value" and "by reference" data mechanisms, and the "sequential", "parallel" and "recursive" control mechanisms as sets of alternatives. This results in each program organisation having specific advantages and disadvantages for program representation and execution. For example, in comparing "by value" and "by reference" data mechanisms the former is more effective when manipulating integers and the latter is more effective when manipulating arrays. Thus each program organisation is, although Universal in the sense of a Turing machine, somewhat restricted in the classes of computation it can efficiently support.

| Machine Organisation | Program Organisation: control flow | data flow | reduction |
|---|---|---|---|
| centralised | von Neumann computers | | GMD reduction machine [13] |
| packet communication | LAU [19]<br>GCF [8] | Irvine [2]<br>MIT [6] | Utah<br>AMPS [12] |
| expression manipulation | Newcastle<br>RCF [22] | Utah<br>DDM1 [5] | Cellular machine [14] |

Figure 12: Examples of Computer Architectures.

We may speculate that it should be possible and indeed desirable for general-purpose computing to design computer architectures whose program organisation is a synthesis of both triples of data and control mechanisms [20].

For the machine organisation it is clear that centralised, packet communication and expression manipulation organisations gravitate toward, respectively, control flow, data flow and reduction. However other pairings of the machine organisations and the program organisations are viable and as indicated above, are being investigated by various groups. When evaluating the three machine organisations against the motivations for fifth generation computer architecture it would seem that the utilisation of parallelism would preclude centralised organisations in favour of the other two organisations. In addition, VLSI motivates an organisation in which a replication of identical computing elements can be plugged together to form a larger parallel computer. But it is also necessary for a computing element to have a centralised organisation so that it can function independently. Thus the three machine organisations, rather than being competitive, seem in fact to be complementary organisations. Each organisation is based on a sequential building block; a computing element containing a processor, communications and memory. The centralised organisation defines how a single computing element must be able to function as a self-contained computer. The packet communication organisation shows how concurrency within a computing element may be increased by replicating resources. Lastly, the expression manipulation organisation specifies how a group of computing elements may be interconnected, at a system level, to satisfy the VLSI attributes of replication.

## 6. ACKNOWLEDGEMENTS

Numerous people have contributed to the research presented in this paper. Firstly I wish to thank my colleagues in the Computer Architecture Group at the University of Newcastle upon Tyne, in particular David Brownbridge and Richard Hopkins. Secondly I thank Wayne Wilner of Xerox PARC for stimulating the Groups current work on VLSI computer architecture. Thirdly, Brian Randell is thanked for commenting on an early version of this paper. Lastly, I wish to thank the Distributed Computing Systems Panel of the UK Science Research Council, which not only funds our research but has also been largely responsible for establishing and encouraging fifth generation computing research in the UK.

## 7. REFERENCES

[1] Anon : "Interim Report on the Study and Research on Fifth-Generation Computers", Japan Information Processing Development Center, (1980).

[2] Arvind et al: "A Processing Element for a Large Multiple Processor Dataflow Machine", Proc. Int. Conf. on Circuits and Computers, (October 1980).

[3] Browning S.A. and Seitz C.L.: "Communication in a Tree Machine", Proc. Caltech Conf. on VLSI, (January 1981).

[4] Clark J.H. : "Structuring a VLSI System Architecture", LAMBDA Magazine, vol. 2 (1980) pp. 25-30.

[5] Davis A. L. : "The Architecture and System Method of DDM1: A Recursively Structured Data Driven Machine", Proc. Fifth Int. Symp. Computer Architecture, (April 1978) pp. 210-215.

[6] Dennis J. B. : "The Varieties of Data Flow Computers", Proc. First Int. Conf. on Distributed Computing Systems, (October 1979) pp. 430-439.

[7] Despain A.M. and Patterson D.A.: "X-TREE: A Tree Structured Multiprocessor Computer Architecture", Proc. Fifth Int. Symp. on Computer Architecture, (April 1978) pp. 144-151.

[8] Farrell E.P. et al : "A Concurrent Computer Architecture and Ring Based Implementation", Proc. of the Sixth Int. Symp. Computer Architecture, (April 1979) pp. 1-11.

[9] Foster M.J. and Kung H.T. : "The Design of Special-Purpose VLSI Chips", IEEE Computer Magazine (January 1980) pp. 26-40.

[10] Hopkins R.P. et al : "A Computer Supporting Data Flow, Control Flow and Updatable Memory", Computing Laboratory, The University of Newcastle upon Tyne, Technical Report 144, (1979).

[11] Johnson R.C.: "32-bit Microprocessors inherit mainframe features", Electronics, (February 1981) pp. 138-141.

[12] Keller R. M., et al : "A Loosely-coupled Applicative Multi-processing System", Proc. Nat.Comp. Conf., (1978) pp. 861-870.

[13] Kluge W. E. : "The Architecture of a Reduction Language Machine Hardware Model", Gesellschaft fur Mathematik und Datenverarbeitung MBH Bonn, Tech. Report ISF-Report 79.03, (August 1979).

[14] Mago" G. A. : "A Network of R Microprocessors to Execute Reduction Languages", Int. Journ. of Computer and Information Sciences, vol. 8, no. 5 and vol. 8, no. 6, (1979).

[15] Mead C.A. and Conway L.: "Introduction to VLSI Systems", Addison-Wesley (1980).

[16] Patterson D.A. and Sequin C.H. : "RISC I: a reduced instruction set VLSI computer", Proc. Eighth Int. Symp. on Computer Architecture (May 1981) pp. 443-458.

[17] Rivest R.L. : "A Description of a Single-Chip Implementation of the RSA Cipher", LAMBDA Magazine, vol. 4 (1980) pp. 14-18.

[18] Sussman G.J. et al : "Scheme-79 - Lisp on a Chip", IEEE Computer Magazine (July 1981) pp. 10-21.

[19] Syre J. C. et al : "Pipelining, Parallelism and Asynchronism in the LAU System", Proc. 1977 Int. Conf. on Parallel Processing, (August 1977) pp. 87-92.

[20] Treleaven P.C. and Hopkins R.P. : "Decentralised Computation", Proc. of the Eighth Int. Symp. Computer Architecture, (May 1981) pp. 279-290.

[21] Treleaven P.C. (ed.) : "VLSI: Machine Architecture and Very High Level Languages", Proc. of the Joint SRC/University of Newcastle upon Tyne Workshop, Computing Laboratory, University of Newcastle upon Tyne, Technical Report 156, December 1980, (a summary in SIGARCH vol. 8, no. 7, December 1980)

[22] Treleaven P.C. and Hopkins R.P. : "A Recursive (VLSI) Computer Architecture", Computing Laboratory, University of Newcastle upon Tyne, Technical Report 161, (March 1981).

[23] Treleaven P.C. et al : "Data Driven and Demand Driven Computer Architecture", Computer Laboratory, University of Newcastle upon Tyne, Technical Report 168, (July 1981).

[24] Watson I. and Gurd J. : "A Prototype Data Flow Computer with Token Labeling", Proc. Nat. Comp. Conf., vol. 48, (1979) pp. 623-628.

# ALGORITHMS, ARCHITECTURE, AND TECHNOLOGY

Jonathan Allen
Massachusetts Institute of Technology
Research Laboratory of Electronics and
Department of Electrical Engineering and Computer Science
Cambridge, MA 02139 U.S.A.

## Abstract

With the advent of VLSI, the design of high-performance digital systems changes from the realization of an algorithm on a given architecture fabricated in a fixed technology to an integrated process wherein the mutual interaction of algorithms, architecture, and technology establishes a balanced design satisfying the system goals. In this paper, these factors are examined in terms of the computational needs of Fifth Generation Computer Systems, with emphasis on applications in signal processing, speech generation, and speech understanding.

The evolving silicon planar technology is providing capability for the fabrication of individual circuits containing millions of devices. Complex systems cannot exploit this technology, however, without the limitation of architectural elements to restricted classes of canonical forms which can be modularly expanded to reflect the natural hierarchy of the task domain. A specific domain where such a basis set of functional units is particularly useful is that of spectral analysis and filtering, where the technology now permits full utilization of all possible parallelism. For example N point FFTS can exploit up to (N/2)Log2N complex "Butterfly" units, and such a high performance architecture in turn demands a technological base requiring wafer-scale integration and accompanying redundancy techniques dictated by yield considerations. In this way, the modular size of units, the architecture of interconnection, the technological base for testing and restructuring, are all seen to interact in complex ways. Whereas in previous systems, natural modular elements were built from smaller separate elements, now the characteristic limitations of the technology establish the degree of redundancy and size of modular elements, thus strongly constraining the overall architecture. Today redundancy techniques are applied at the chip level, but large new systems will require system level utilization of redundant elements. These large systems will also require a testing architecture and technology capable of comprehensive testing of modules at fabrication time for use by interconnection processes, but also self-testing during field utilization to continually establish the viability of these complex computational engines. Technological innovations may also lead to three-dimensional structures that will permit new interconnection strategies and architectural forms.

The massive parallelism afforded by the evolving technology reveals, however, the need for fundamental understanding of algorithmic representation. As usually expressed, algorithms confound the competence or semantics of a task with its performance. In a world restricted to single-sequence von Neuman machines this was not a severe limitation, but now the system designer requires the systematic means to explore space/time/power tradeoffs as various performance options all the while retaining the basic competence of the prescribed task. New results providing the theoretical basis for these performance explorations are finding use in design systems that provide the system designer with a comprehensive choice of options, hence merging a class of algorithms for a given task with a corresponding set of architectural possibilities. The availability of extensive parallelism has also encouraged new compilation techniques for the design of data path units incorporating distributed arithmetic-logic capability over a set of registers. VLSI design strategies of this sort thus encourage the designer to exploit parallel operations on specialized registers in a way not possible with previous general-purpose architectures. Future design system will discover such task-specific parallelism and automatically compile architectures that reflect these possibilities.

The interacting constraints due to technology, architecture, and algorithm provide a particularly rich set of design options in systems for digital signal processing, speech generation from text or concept, and speech understanding. Systems capable of supporting interactive design for these applications will soon permit the realization of high performance complex engines. It is of especial interest that the techniques described here for use in Fifth Generation Computer Systems will provide the basis or both the requisite design systems and the application systems produced by their utilization.

Major Contributors of Research and Planning Committee

of Fifth Generation Computer Systems

(1979 - 1981)

| | |
|---|---|
| Aiso, H. | Keio University |
| Amamiya, M. | Musashino Electric Communication Lab., NTT |
| Amou, K. | Toshiba Corp. |
| Anraku, Y. | Oki Electric Industry Co., Ltd. |
| Aoyama, A. | Hitachi Ltd. |
| Asada, A. | Sharp Co., Ltd. |
| Aya, H. | Mitsui Engineering & Shipbuilding Co., Ltd. |
| Chiba, N. | Nippon Electric Co., Ltd. |
| Fuchi, K. | Electrotechnical Lab. |
| Furukawa, K. | Electrotechnical Lab. |
| Fusaoka, A. | Mitsubishi Electric Co., Ltd. |
| Goto, A. | University of Tokyo |
| Goto, S. | Yokosuka Electric Communication Lab., NTT |
| Hakozaki, K. | Nippon Electric Co., Ltd. |
| Hatano, Y. | Hitachi Ltd. |
| Hattori, T. | Japan Software Industry Association |
| Hayashi, H. | Fujitsu Lab. |
| Hirose, K. | Waseda University |
| Hotta, H. | Nippon Electric Co., Ltd. |
| Iikawa, S. | Mitsubishi Electric Co., Ltd. |
| Iizuka, H. | Seikei University |
| Inoue, H. | University of Tokyo |
| Ishii, O. | Electrotechnical Lab. |
| Ishizaki, S. | The Fuji Bank Ltd. |
| Ishizaki, Shu. | Electrotechnical Lab. |
| Itoh, N. | Oki Electric Industry Co., Ltd. |
| Itoh, T. | Tohoku University |
| Kadowaki, Y. | Hitachi Ltd. |
| Kamijo, T. | Information Technology Promotion Agency |
| Kaneoka, K. | Japan Information Processing Association |
| Karatsu, H. | Technology Consultant |
| Kidode, M. | Toshiba Research Center |
| Kitsuregawa, Y. | University of Tokyo |
| Kodera, T. | Sharp Co., Ltd. |

| | |
|---|---|
| Konishi, K. | Yokosuka Electric Communication Lab., NTT |
| Kunii, T. | University of Tokyo |
| Kunifuji, S. | Fujitsu Ltd. |
| Kurita, S. | Japan Electronic Computer Co., Ltd. |
| Kurokawa, T. | Toshiba Research Center |
| Kuwabara, M. | Nippon Telegraph & Telephone Public Corp. |
| Makinouchi, A. | Fujitsu Lab. |
| Matsui, K. | Institute for the Future Technology |
| Mizoguchi, F. | Tokyo University of Science |
| Mizoguchi, T. | Mitsubishi Electric Co., Ltd. |
| Mizuno, Y. | Nippon Electric Co., Ltd. |
| Moto-oka, T. | University of Tokyo |
| Motoyoshi, F. | Electrotechnical Lab. |
| Nagao, M. | Kyoto University |
| Nakajima, R. | Kyoto University |
| Nakashima, H. | University of Tokyo |
| Nobukuni, H. | Nippon Telegraph & Telephone Public Corp. |
| Odaka, T. | Hitachi Ltd. |
| Ohta, H. | Mitsubishi Electric Co., Ltd. |
| Osuga, S. | University of Tokyo, Institute of Space and Spacenautical Science |
| Sakaki, K. | Mitsubishi Electric Co., Ltd. |
| Sakama, Y. | Yokosuka Electric Communication Lab., NTT |
| Sakamura, K. | University of Tokyo |
| Sato, K. | Electrotechnical Lab. |
| Sato, T. | Electrotechnical Lab. |
| Sekino, A. | Nippon Electric Co., Ltd. |
| Shimada, T. | Electrotechnical Lab. |
| Shiraishi, T. | Sharp Co., Ltd. |

| | |
|---|---|
| Sohma, Y. | Fujitsu Lab. |
| Sugimoto, M. | Fujitsu Ltd. |
| Suwa, M. | Electrotechnical Lab. |
| Takei, K. | Toshiba Research Center |
| Takeuchi, A. | Mitsubishi Electric Co., Ltd. |
| Takizawa, M. | Japan Information Processing Development Center |
| Tamura, H. | Electrotechnical Lab. |
| Tamura, K. | Electrotechnical Lab. |
| Tanaka, Hi. | University of Tokyo |
| Tanaka, Ho. | Electrotechnical Lab. |
| Tanaka, Yu. | Hokkaido University |
| Tanaka, Ya. | Gakusyu-in University |
| Tokoro, M. | Keio University |
| Tomaru, Y. | Matsushita Communication Industry Co., Ltd. |
| Torii, K. | Electrotechnical Lab. |
| Tsutsumi, Y. | The Nihon Keizai Shinbun |
| Uchida, G. | Technology Consultant |
| Uchida, S. | Electrotechnical Lab. |
| Ueda, K. | Matsushita Research Institute of Tokyo |
| Uehara, T. | Fujitsu Ltd. |
| Uemura, Y. | Nippon Electric Co., Ltd. |
| Uraki, T. | Hitachi Ltd. |
| Wada, E. | University of Tokyo |
| Yamada, H. | Fujitsu Lab. |
| Yamamoto, K. | Japan Information Processing Development Center |
| Yamamoto, M. | Nippon Electric Co., Ltd. |
| Yamazaki, H. | Oki Electric Industry Co., Ltd. |
| Yasuhara, H. | Oki Electric Industry Co., Ltd. |

| | |
|---|---|
| Yokoi, T. | Electrotechnical Lab. |
| Yoneda, E. | Toshiba Corp. |
| Yoneda, N. | University of Tokyo |
| Yonezawa, A. | Tokyo Institute of Technology |

# Author's Index

## Organizing Committee

| | | |
|---|---|---|
| Chairman: | Tohru MOTO-OKA, | University of Tokyo |
| Program Chairmen: | Hideo AISO, | Keio University |
| | Kazuhiro FUCHI, | Electrotechnical Laboratory |
| | Hajime KARATSU, | Technology Consultant |
| Secretary General: | Kinko YAMAMOTO, | Japan Information Processing Development Center |