TM-0453

An Introduction to the Fifth Generation Computer Systems (FGCS)

by
T. Kurozumi & T. Ichiko

January, 1988

ICOT

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

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

An Introduction to the Fifth Generation Computer Systems (FGCS)

Takashi KUROZUMI

Takao ICHIKO

ICOT
Institute for New Generation Computer Technology
Mita Kokusai Bldg. 21F
4-18 Mita 1-Chome
Minato-ku, Tokyo 108 Japan

1. Introduction

This article presents an introduction to the fifth generation computer systems.

Fifth generation computers are under development to meet the knowledge information processing needs for the advanced information-oriented society of the 1990s. This clearly requires overcoming the technical limitations of conventional computers. A fifth generation computer requires problem solving and inference functions for inference of solutions to problems from accumulated knowledge, knowledge base management functions for acquirement and retrieval of knowledge, intelligent interface functions for fluent conversations using natural language, drawings, images, etc., and intelligent programming functions for automatic conversion of problem specifications submitted in natural language, drawing, etc., into efficient programs.

Fifth generation computers with these functions will have far-reaching impact finding application in all areas of industry and society. However, research and development (R&D) of these computers requires an extremely wide variety of leading-edge technology, and the risks are high. Accordingly, a long-term R&D plan was established for this project that extends over ten years with the initial stage lasting from 1982 to 1984, the intermediate stage from 1985 to 1988, and the final stage will begin in 1989 and the project will end in 1991.

2. What Is a Fifth Generation Computer?

This R&D is founded on the following perceptions: existing computers have reached their architectural limits; the possibility of qualitatively different architectures are in prospect; and a new era in the history of computers is close at hand. The basic architecture of todayś computers (i.e., the von-Neumann architecture) is restricted by computer machine languages. Entire software systems are built on these machine languages, which means that all features of a von-Neumann computer are derived from the machince language.

The hardware and software of fifth generation computer will be determined by a new architecture based on predicate logic. The predicate logic language will be used as the machine language by hardware to perform inference and by software to choose combinations of the basic hardware inference operations.

The basic concept of fifth generation computers are presented in more detail below.

3. Background of R&D of Fifth Generation Computers

(1) Limitations of todayś computers

The basic functions of todayś computers are arithmetic operations. They are suitable for execution of programmable sequences of routine tasks such as numerical calculations. These functions are quite different from manś intellectual activities, which involve performance of inference based on knowledge. In short, the fundamental limitation of todayś computers is that they cannot determine what to do by themselves. That is, they cannot comprehend an unforeseen event nor can they choose the most appropriate action to take.

(2) Meeting demands on computers in the next decade

Research and development of artificial intelligence technology, architectures, and software engineering are all necessary to develop the user-friendly computers as tools for support of human intellectual activities and to improve software productivity in the advanced information-oriented society of the 1990s (Figure 1).

(a) Easy-to-use computers: In an advanced information-oriented society, computers will be used not only in industrial applications, but also in all other fields, especially in everyday life. Thus they must be readily usable by persons without specialized training. Clearly, a man-machine interface close to the human-to-human interface is required.

(b) Support of intellectual activities: Computers working on the behalf of men or supporting intellectual activities, such as decision making in business and R&D are required for maintaining industries and communities that are both logical and creative.

(c) Increasing software productivity: Needs for efficient information processing are increasing immensely with advance and diversification of industrial and social activities. Hence, the advent of the so-called "software crisis" is anticipated and it can only be dealt with by a massive increase in software productivity.

4. Situating Fifth Generation Computers

To date, computers can be classified into the following generations, defined by their respective hardware elements:

- 1st generation	:	Vacuum tubes
- 2nd generation	:	Transistors
- 3rd generation	:	Integrated Circuits (IC)
- 3.5th generation	:	Large-Scale Integration (LSI) devices
- 4th generation	:	Very Large-Scale Integration (VLSI) devices

However, all these computers have a basic architecture that has remained almost unchanged from the beginning until now.

Most conventional computers are based on the "sequential processing and stored program" concept proposed by John von Neumann. Fifth generation computers are being developed under an architecture based on predicate logic suitable for parallel processing in artificial intelligence. In this sense, it may be more appropriate to classify all the conventional computers as generation I and fifth generation computers as generation II (Figure 2).

5. Basic Scheme of a Fifth Generation Computer

(1) R&D targets of a fifth generation computer

The fifth generation computer project aims at developing a Knowledge Information Processing System (KIPS) that can satisfy the requirements described in sec.3 for computers in the next decade. This objective can be divided into the following three targets:

(a) Development of KIPS by using artificial intelligence technology: The results of research on artificial intelligence should consistently be applied and embodied in KIPS as intelligent conversation functions and inference functions using a knowledge base. The results should also be used to define the guiding principles of the new knowledge information processing.

(b) Improvement of software productivity with software engineering technology: Program for KIPS will be large and complicated; thus that can hardly be written in the present programming languages and environment. Therefore, new programming languages based on new principles are required, and a new programming environment, which reduces the burden on human programmers, have to be established. Also software engineering techniques for automatically writing programs need to be developed. These facilities will enable persons without specialized knowledge to write programs easily.

(c) Development of very high-performance hardware: The results so far of research in artificial intelligence indicates that the calculation speed of a fifth generation computer must be an order of magnitude higher than that of todays computers. To attain these speeds, ultrahigh-speed hardware consisting of hundreds of parallel processors must be developed.

(2) Basic architecture of fifth generation computers

The software and hardware of the fifth generation computer must have KIPS functions, i.e., inference functions that use a knowledge base.

Inference is the main power by which human beings understand things consciously or unconsciously. Having knowledge and making inferences are basic conditions for intellectual activity.

A logical definition of inference is "a rational procedure employing knowledge for extraction of unknown information from known information." Inference complies with inference rules. The most basic rule is the syllogism consisting of a major premise (A→B), a minor premise (B→C), and a conclusion (A→C). One method to enable a computer to make inferences is to incorporate this rule into the hardware for automatic execution. This is the method to be used in our fifth generation computers. Writing programs in a logical form for making inference is called logic programming.

In a logic programming environment, most knowledge stored in a computer can be expressed in the forms of predicate logic. The simplest of these forms corresponds to relational expressions in existing relational data bases. Therefore, the present relational data base techniques may be the starting point for developing a knowledge base.

Programs written in the form of predicate logic are well-suited to parallel processing.

We chose a predicate logic language for what is equivalent to the machine language of a conventional computer, which we call the kernel language.

Prolog is a predicate logic language used in our system in the same way as a high-level programming language in existing computers. The Prolog-based kernel (machine) language has to be modified and extended for KIPS functions in our fifth generation computers. Application-oriented languages with knowledge representation and modularity functions are being developed as users' programming languages.

VLSI techniques progress each year. A typical result of this progress is that the capacity of a semiconductor memory chip has increased four-fold every three years. New research in computer architecture, especially on parallel processing machines, is associated with such software concepts as predicate logic, functional, and object-oriented languages. The progress of VLSI techniques supports this research. This project aims at developing the ultrahigh-speed parallel processing hardware described in (1) using the VLSI techniques and the kernel as machine language, as well as artificial intelligence, software engineering, and architecture techniques under a logic programming environment.

Under the basic architecture described above, a fifth generation computer must have the following functions to attain the R&D targets described in (1) (Figure 3):

(a) Problem solving inference functions: These perform deductive and inductive inference including speculation based on incomplete knowledge.

(b) Knowledge base management functions: These express, acquire, store, and retrieve various knowledge required by inference functions.

(c) Intelligent interface functions: These are man-machine interface functions that can flexibly converse with humans in natural languages.

(d) Intelligent programming functions: These automatically express given problems in efficient programs.

6. A Developed Fifth Generation Computer

The final form of a developed fifth generation computer consists of software and hardware that are interfaced by the kernel language (Figure 4).

(1) Software

The software consists of a basic software system and its demonstration system. The former is the knowledge information processing base using high-level languages based on predicate logic. The latter demonstrates whether or not the results of the fifth generation computer project are sufficient components of KIPS.

The basic software system consists of the following modules:

(a) Problem solving inference module: This is the nucleus of the OS for a fifth generation computer. It controls inference machines, performs deductive and high-level inference operations, and solves problem efficiently.

(b) Knowledge base management modules: This assumes extensive control over the knowledge information base. It has knowledge acquirement functions including the learning function, which determines whether or not specific items of knowledge are worthy of being stacked.

(c) Intelligent interface module: This interprets human language and processes graphic data to play the role of a man-machine interface.

(d) Intelligent programming module: This minimizes the burdens on human programmers by fully utilizing the most advanced software engineering technique. Its ultimate goal is automatic programming.

The demonstration system for basic software demonstrates whether or not the results of the fifth generation computer project are suitable for a variety of KIPS applications. When the development of a fifth generation computer is completed, a greater number of application systems incorporating a wide range of knowledge will be developed in place of this demonstration system.

(2) Hardware

The hardware consists of the following super-high performance machines developed in VLSI with parallel processing techniques.

(a) Inference machine: Has basic inference functions such as the syllogism function and executes program written in the kernel language.

(b) Knowledge base machine: Processes knowledge data expressed in a variety of complicated forms.

(c) Intelligent interface machine: Processes acoustic and graphic data.

7. Applications and Impact of Fifth Generation Computers

Research in artificial intelligence pursues development of intelligent functions. Continued effort is required for an extremely long period of time, but results may find a vast range of applications. The applications of a KIPS containing these results will help shape the main stream of an advanced information-oriented society in the future. Some applications may be completely new, while others are advanced versions of conventional applications. In such a society, fifth generation computers will be required for knowledge information processing.

The purpose of a fifth generation computer is to form a KIPS that helps solve human problems in probing the unkown world and to greatly expand the scope of human intellectual activity.

Fifth generation computers will remarkably improve the relatively low productivity of primary and tertiary industries rather than that of secondary industry. They will also aid in coping with the problems of resource use and high average age, etc., expanding the scope of their impact on society (Figure 5).

The basic techniques required for realizing the previous scheme were developed during the initial stage and during the first year (1985) of the intermediate stage of this R&D. We are now working on subsystems in accordance with the intermediate stage plan. The results so far have demonstrated that commitment to predicate logic is sound and that construction of intelligent systems based on it is feasible.

8. Initial Stage R&D Targets and Results

To attain the goal of developing basic techniques in the initial stage (1982 to 1984), the targets below were established, which pertain to the inference subsystem, knowledge base subsystem, basic software, and pilot model for software development.

As a result, the basic techinques necessary for subsystem development in the intermediate stage were, on the whole, developed satisfactorily during the initial stage.

(1) Inference subsystem

(a) Target: The inference subsystem and the knowledge base subsystem form the nucleus of fifth generation computer hardware. One target of the initial stage was to settle certain basic issues in hardware architecture. So we studied parallel inference, data flow mechanisms, and abstract data type mechanisms.

Other targets of the initial stage were the evaluation of methods for developing the techniques by software simulation and the creation /evaluation of small-scale experimental systems.

(b) Results: Data flow and reduction based architectures were evaluated using software simulator for 64 parallel processing elements. Also, a hardware simulator consisting of 8 to 16 element processors was created and used to evaluate the data flow, reduction, and modified reduction (complete-copying/clause-unit-processing) methods. The simulations proved that sufficient parallelism can be obtained by these methods. An abstract data type mechanism was evaluated for the language specifications of the kernel language.

(2) Knowledge base subsystem

(a) Target: A knowledge base machine stores, retrieves, and efficiently updates knowledge data expressed in a knowledge representation or the kernel language. Finding basic techniques for development of such a machine was a target of the initial stage. As such, three mechanisms had to be researched and developed: the knowledge base basic mechanism, the parallel relational knowledge processing mechanism, and the relational data base mechanism. Other targets were to evaluate methods for building and managing a knowledge base logically, to determine a basic architecture for parallel relational knowledge computation and a basic architecture for a relational data base machine (by simulating the machine), and finally to construct and evaluate the machine.

(b) Results: A relational data base machine called Delta was constructed. This machine consists of hierarchical storage, four parallel relational algebraic computation engines, and multiple processors having distributed functions. Delta connects with the sequential inference machine through an interface that transfers relational algebraic commands and data.

(3) Basic software system

The basic software system is the nucleus of fifth generation computer software. The system consists of the 5G kernel language, problem solving inference software, knowledge base management software, intelligent interface software, and intelligent programming software, all of which are described below.

(i) 5G language

(a) Targets: The kernel language version 0 (KL0) is a sequential language specifying the interface between the core hardware and software of the pilot model for application software development. The target was KL0 prototype design and development by the end of 1983.

The kernel language version 1 (KL1) is a parallel language stipulating the interface between the hardware and software of the inference subsystem for the intermediate stage. Determining the basic language specifications of KL1 was another target of the initial stage.

(b) Results: KL0 was developed as the machine language of the pilot model. It has Prolog functions to which frequently used functions have been added and from which seldom used functions have been deleted. The added functions are, for example, improved execution efficiency, parallel execution, and string manipulation functions. The deleted functions are, for example, those of program definition and data base management. Furthermore, a system description language called Extended Self-contained Prolog (ESP) was developed separately. ESP has module creation (object-oriented) and marco expansion functions. By developing KL0 and ESP separately, the conflicting needs for higher efficiency and flexible description could be harmonized on a high level. The effectiveness of ESP may have been sufficiently proved by SIMPOS, which is described later.

KL1 is for advanced parallel processing. Therefore, it must support strong independence between minutely divided execution units and assume precise control over communication and synchronization between units. The first KL1 developed is based on concurrent Prolog and has functions such as AND parallel, OR parallel, set/stream conversion, module creation, and meta-inference. However, the environment for the OR parallel function and read-only annotations were found to be too complicated for implementation, as were, as a result of testing prototype processing systems. Therefore, a second KL1 named Guarded Horn Clause (GHC) was developed by adding semantic rules to the language specifications and thereby deleting unnecessary functions.

(ii) Problem solving inference software

(a) Targets: Problem solving inference software must have functions requiring unexplored techniques; e.g., high-level inference functions including speculation based on incomplete knowledge, cooperative problem solving functions, etc.

Primary R&D targets for this software have been assigned to the intermediate and final stages. The target of the initial stage was to develop various prototypes of problem solving software for obtaining the information requisite to reaching the primary goals.

(b) Results: Methods to develop parallel inference, meta-inference, and problem solving inference functions were evaluated. Concerning the parallel inference function, how to develop a parallel execution interpreter for pure Prolog programs was evaluated. The meta-inference function infers and controls inference itself. For this function, a simulation language was developed. To improve the execution efficiency of meta-inference programs, methods to use partial calculations and apply the results for numerical formula processing were evaluated. Concerning the problem solving inference function, techniques for solving particular problems on actual use were evaluated. Problems dealt with were distributed-type problems, problems in CAD of electronic circuits, etc.

(iii) Knowledge base software

(a) Targets: Three targets were determined to cope with the trend toward larger-scale knowledge bases. The first target was to evaluate methods to connect a knowledge base and inference machine. To evaluate the methods, a program managing a large-scale relational data base had to be developed. The second target was to develop a complicated knowledge representation system intended for a particular field. The third target was to develop an experimental system using knowledge to evaluate our R&D results.

(b) Results: An experimental data base management system named Knowledge Acquisition-oriented Information SuppliER (KAISER) was developed for an orientation to management of a large-scale knowledge base. The development of KAISER was based on knowledge representation according to predicate logic and used relational data base techniques closely associated with logic representation. KAISER consisted of modules to manage the internal data base for a Personal Sequential Inference (PSI) machine and modules to manage the external data base for Delta. The modules are for knowledge acquisition, knowledge manipulation, knowledge accumulation, knowledge interface, and knowledge conversation. The knowledge manipulation and accumulation modules are equivalent to a conventional relational data base management program. The knowledge interface module translates retrieval requests expressed in a logical language into the relational data base manipulation language. The knowledge conversation module provides necessary knowledge for the user through natural conversation. The knowledge acquisition module acquires new knowledge, stores it in a knowledge base, and updates the contents of the knowledge base. Its functions are based on logical formatted knowledge acquisition.

Two programming languages instead of a fixed knowledge representation system were prepared for flexible knowledge representation. One is named Mandala, the other Complex Indeterminate Language (CIL).

Mandala is based on the logical language KL1 and has a variety of functions supporting programming techniques used to create object-oriented, rule-oriented, and data-oriented programs.

CIL is a problem-oriented knowledge representation language. It enabies to modelling the process of understanding the content of a conversation on the computer. The representation format of CIL is so universal that CIL will find many new applications. In short, this language is Prolog reinforced by frames with constraint.

The prototypes of a Japanese sentence generation support system and a CAD system for logic design in electronics were created and evaluated as experimental expert systems. The applicability of logic, object-oriented, and parallel languages to these systems was also evaluated.

(iv) Intelligent interface software

(a) Targets: The main target of the initial stage was to perform research on natural language processing for flexible conversation between men and computers. Other targets associated with natural language processing were the development of a data base for natural language processing, development of language analysis techniques, and study of the recognition functions of an intelligent interface.

(b) Results: A data base for a natural language (dictionary) differs in contents and creation method from a dictionary for use by humans having common sense and high intelligence. In the initial stage, therefore, a small-scale experimental electronic dictionary was developed to determine the basic specifications of a full-scale electronic dictionary. Then, functions such syntactic analysis, semantic analysis, word analysis, sentence interpretation, and sentence generation functions were researched and developed. The functions comply with Lexical Functional Grammer (LFG) and Generalized Phrase Structure Grammer (GPSG). As a syntax analyzer, a high-performance program utilizing the features of logical language was developed and named Bottom Up Parser (BUP). BUP can be applied to grammer belonging to a free grammar family of extended contexts. Therefore, a syntax description system for defining individual syntactic rules was also developed.

An experimental sentence interpretation system (named DUALS) was developed for the intelligent interface. This system reads and understands sentences and responds to questions according to situational semantic logic. The system has syntactic, semantic, and contextual analysis functions that comply with LFG and can process the kind of sentences that appear in Japanese text books for 3rd-year elementary school students. Besides these functions, problem solving and sentence composition functions were also incorporated in this system.

(v) Intelligent programming software

(a) Targets: One target concerning intelligent programming software was to develop an ESP-based system that supports smooth modular programming. Another was to develop and evaluate a software test and management program that supports efficient programming stages; i.e., designing, coding, testing, debugging, modifying, maintaining, and managing programs efficiently.

(b) Results: The above-mentioned object-oriented language ESP was developed as a primary function for modular programming. The programming system was developed and installed in a PSI machine, which is now being used.

The specifications of experimental systems were evaluated as those of software verification programs. The experimental systems are a software development consultation system, a hierarchical logic program verification system, etc. Furthermore, prototype for a few of these systems were developed.

(4) Pilot model for software development

(a) Targets: The model is a Sequential Inference Machine (abbreviated SIM here) having a partially improved version of the existing von-Neumann architecture and is a tool for efficiently developing software for fifth generation computers. Developing SIM hardware and software was a target of the initial stage.

SIM hardware is a firmware-based machine supporting KL0. When executing software written in the kernel language, this hardware must exhibit execution speed several times faster than the execution speed of a large-scale general-purposes machine and provide larger addressable space than 10M bytes. SIM software is the OS for SIM. If combined with intelligent programming software, this OS becomes a programming operating system.

(b) Results (SIM hardware): The SIM hardware consists of basic and extended systems. The basic system consists of a PSI machine and a LAN. This system was developed in the initial stage to assist software development. The extended system consists of a Cooperative High-performance sequential Inference (CHI) machine placed behind a PSI machine and peripherals such as graphic I/O units. A relatively long period was allowed for developing the extended system because the system had to have high performance and advanced functions.

Both PSI and CHI machines have a tag architecture and use a microprogram control method. The design of the PSI machine has placed emphasis on supporting I/O control functions of the OS, but that of the CHI machine behind PSI has minimized these functions and placed emphasis on high-speed execution of programs written in logical languages. The execution speed of the PSI machine is 30K Logical Inference Per Second (LIPS), and that of the CHI machine 200K LIPS or higher. The storage capacity of PSI is 40 bits x 16M words, and that of CHI is 36 bits x 64M words. The PSI machine cycle time is 200 nanoseconds, and the CHI counterpart is 100 nanoseconds. The main device techniques used in PSI are TTL and NMOS techniques, and those in CHI are CML and CMOS techniques. By the end of 1986, approximately 100 PSI machines were in use for software development by the fifth generation computer project.

(b) Results (SIM software): SIM software consists of a programming system and an OS. Thus it is called the SIM Programming and Operating System (SIMPOS). SIMPOS divides into four layers. Among these, three layers, the kernel, supervisor, and I/O controller, form the OS; the one layer above the other three is the programming system. SIMPOS has been designed according to the policies below.

(A) System design based on a single framework: The machine, language system, OS, and programming system should be structured in the same logical programming framework.

(B) Conversational system having multi-window functions: A multi-window system including a mouse and keyboard is an extremely effective conversation means in an intelligent system.

(C) Data base function: A logical language and a relational data base are very harmonious. Therefore, a new programming system and a new OS fully utilizing data base functions should be developed.

(D) Japanese language processing: People should be able to communicate with computers in their own language. To be usable in Japan, computers must be able to process the Japanese language.

(E) Compatibility: As many existing software modules as possible should be reusable to create a large-scale program. Therefore, functions providing the necessary compatibility should be available.

9. Intermediate Stage R&D Targets and Current Situation

The intermediate stage (1985 to 1988) is important for this project because the success or failure of project will be determined in this stage. In this respect, the R&D targets of this stage were formulated according to the following policies.

(a) The basic software techniques necessary for problem solving inference, knowledge base management, intelligent interface, and intelligent programming functions should be developed.

(b) As a hardware system supporting the above basic software functions, a number of small-to-medium scale subsystems (inference and knowledge base subsystems) should be developed. Then, their functions should be tested.

(c) A parallel software environment and a network system should be created as an infra-structure for supporting software development.

The R&D targets of the intermediate stage and the current situation are described below for hardware, software, and development support systems.

(1) Hardware system

The final fifth generation computer hardware consists of inference and knowledge base subsystems. The desired execution speed of the hardware is in the range 100M to 1G LIPS.
Research and development of the hardware system in the intermediate stage is based on the R&D results of mechanisms for inference and knowledge base subsystems and of SIM in the initial stage. Also, the R&D follows the parallel-language specifications of KL1.
The R&D goal in the intermediate stage is to determine an advanced parallel-machine architecture for the inference and knowledge base subsystems. This goal is closely associated with the research and development resource and execution management software (an OS for parallel machines) and parallel processing software that are executed in the parallel-processing machines.

(A) Inference subsystem

The first R&D target of the inference subsystem in the intermediate stage is to study the mechanisms developed in the initial stage (e.g., data flow and reduction mechanisms) with KL1 specifications, evaluate the results, and extend the mechanisms. As well, the other R&D targets listed below are now being pursued with the goal of developing a Parallel Inference Machine (PIM) suitable for a parallel logical language.

(a) PIM architecture: The PIM architecture should be determined for a network of approximately 100 element processors. Then, prototype hardware should be developed and its operations tested.

(b) Component modules: The prototypes of component modules performing fine-grained parallel processing in an element processor should be developed. The modules must operate according to methods implemented by mechanisms such as data flow and reduction.

(c) Large-scale PIM architecture: A network of approximately 1000 element processors should be simulated by software. Also, the mechanism connecting a PIM and a Knowledge Base Machine (KBM) should be investigated.
During the first year of the intermediate stage (1985), we evaluated in detail the data flow, reduction, complete copy, and clause unit methods that were implemented in the initial stage. At the same time, the architecture of PIM executing KL1 was evaluated, as were the component modules. Furthermore, basic PIM specifications were determined, and a prototype software simulator for modelling PIM was developed.

(B) Knowledge base subsystem

Based on R&D results of the initial stage (RDBM), R&D of the knowledge base subsystem in the intermediate stage should clarify techniques for developing the knowledge computation mechanisms that KBM ought to be equipped with. Then, the architecture of parallel KBM should be determined, and prototype hardware developed. Furthermore, techniques for developing mechanisms supporting distributed knowledge bases should be clarified. These targets are detailed below.

(a) KBM architecture: Techniques for developing knowledge computation mechanisms should be investigated. Also, the prototype of a knowledge computation engine having functions such as sorting variable-length records and collating character strings should be developed.

(b) Distributed knowledge base architecture: Multiple knowledge bases may be connected by a high-speed bus or Local Area Network (LAN). To make these knowledge bases which appear logically as a single knowledge base, a control mechanism for distributed KBM is necessary.

This mechanism should be investigated. Also, a software simulator and a verification system should be developed.

(c) Large-scale KBM architecture: The parallel processing mechanisms of KBM should be investigated, and prototypes of a software simulator and an operation test simulator should be developed.

In the intermediate stage, we evaluated in detail the connection between the relational data base machine, Delta, which had been developed in the initial stage, and a PSI machine. We also evaluated a model architecture for KBM and prepared for the development of a prototype test system.

(2) Basic software system

The final R&D goal of the basic software system is to develop the nucleus of fifth generation computer software. In the intermediate stage, the R&D targets of the basic software system are to extend Kernel Language version 1 (KL1), design the specifications of Kernel Language 2 (KL2), research and develop new software techniques such as parallel-inference control, implement the element technique developed in the initial stage, and develop a full-scale prototype system. These six R&D targets in the intermediate stage are detailed below.

During the first year of the intermediate stage, the implementation specifications of KL1 were determined, and prototypes of portions of processing systems were developed. To implement the element technology, a sample program was designed and developed, and a test system was designed and partially developed.

(A) 5G Kernel Language

(a) KL1: KL1 supports the hardware system developed in the intermediate stage. Some processing systems in KL1 are useful for creating a parallel programming environment. In the intermediate stage, these systems and the programming support system in KL1 should be developed, used to reach the other R&D goals, and evaluated. The results should be reflected in the specifications of KL2.

(b) KL2: KL2 supports the prototype system to be developed in the final stage. In the second half of the intermediate stage, the language specifications of KL2 should be determined. The specifications will be KL1 language specifications with knowledge representation functions, etc.

(B) Problem solving inference software modules

Techniques for developing software modules that will be the nucleus of the OS controlling and managing a PIM should be developed. Also, an inductive inference function and an inference function supplementing knowledge should be investigated. These R&D targets are detailed below.

(a) Parallel inference software: Techniques for controlling advanced parallel processing should be developed. Then, software controlling problem solving parallel inference machines that make deductive inferences at high speed should be developed.

(b) Basic software for advanced inference: Prototype software having advanced inference functions and some learning functions should be developed. Advanced inference functions such as inductive inference and analogical inference are necessary for developing an intelligent system.

(c) Basic software for cooperative problem solution: Prototype software should be developed that enables cooperation among multiple problem solving programs and performs inference by supplementing knowledge.

(C) Knowledge base management software modules

Techniques for developing knowledge base management functions should be developed. These functions are knowledge acquisition that processes both known and incompletely known knowledge, knowledge representation, knowledge storage, and knowledge use that processes distributed knowledge sources. This R&D target is detailed below.

(a) Knowledge representation and use system: The knowledge programming language designed in the initial stage should evolve to a knowledge representation language for use in a specific field and treatment as a common base. Also, a prototype of a knowledge base creation support system having knowledge base editing functions should be developed.

(b) Basic software for knowledge acquisition: Tools for acquiring knowledge from experts should be developed. Also, techniques for composing rules based on inductive inference should be researched.

(c) Basic software for distributed knowledge base management: The KAISER developed in the initial stage should be developed further, a model of distributed knowledge bases should be created, and the algorithm of managing multiple knowledge bases as a single logical knowledge base should be determined.

(D)Intelligent interface software modules

Software supporting flexible conversations between a computer and human beings should be researched and developed. This software must have the functions of semantic analysis support, sentence analysis that can cross reference and process omissions, and sentence generation that creates natural sentences from semantic expressions. Also, a pilot model for researching versatile conversation functions should be developed. The conversation functions must process natural languages, acoustic data, graphic data, etc. These targets are detailed below.

(a) Support of dictionary creation: Software supporting the creation of and reference to entries and their semantic variations in a dictionary should be developed.

(b) Japanese language GPSG system: A Japanese language parser based on the GPSG theory should be developed.

(c) Context interpretation system (extended): A prototype of a context interpretation system should be developed by introducing new language theories such as syntactics and situation semantics.

(d) Conversation system (Pilot model): A prototype conversation system should be developed that models changes of knowledge about the conversation partner in a knowledge base during conversation. Also, techniques for processing acoustic and graphic I/O data should be investigated.

(E) Intelligent programming software modules

Techniques for coding new software in logical languages should be developed, as should techniques for programming in specification description Japanese language and formatted logical language as well as for converting and verifying programs. Techniques for supporting an overall programming procedure ranging from design to maintenance should also be developed. These targets are detailed below.

(a) Specification description natural language system: Japanese language constrained in grammar and vocabulary should be used to describe specifications. Therefore, a system is required that analyzes the syntax and meaning of input specification description Japanese language and converts the language to formatted specification descriptions.

(b) Software knowledge management system: Knowledge about programs developed in SIM should be stacked in a knowledge base, and an intelligent programming support system using this knowledge base should be developed.

(c) Basic software for program conversion, verification, and generation: A prototype system should be developed that converts programs to optimize them logically, tests the conversion results, and generates programs according to the results.

(d) Overall programming support system: A prototype system supporting an entire programming process that ranges from design to maintenance should be developed.

(F) Demonstration system for basic software

The R&D results of the basic software should be applied to a number of fields. Also, prototype software testing of the application results should be developed to feed the test results back to the basic software.

(3) Development support system

Advanced R&D requires a suitable R&D support system. In the initial stage, a pilot model for software development was created. In the intermediate stage, parallel software development tools should be developed as an infra-structure to further facilitate R&D.

(a) Parallel software development machines: Parallel software machines (abbreviated M-SIMs here) should be developed to support the development of a variety of parallel-inference software based on KL1. An M-SIM is the CPU of the small-to-medium-scale PSI machine developed in the initial stage. A number of closely connected M-SIMs make up a multi-processor software system for parallel software development. This system can also be used to test the effectiveness of parallel processing. The system should become usable in the middle of the intermediate stage.

(b) Improvement and extension of SIM: The functions of the SIM software and hardware developed in the intial stage should be improved and extended to become more useful tools.

(c) Development support network system: A Local Area Network (LAN) should be extended and improved for more efficient development of software. Also, a wide area network should be prepared for exchanging data and software between R&D groups, thereby improving R&D efficiency.

During the intermediate stage, SIMs were improved and extended, miniaturization and parallelization promoted, and development support networks created.

10. Concluding Remarks

Based on the R&D concept, ICOT is publishing its R&D results, welcoming visiting researchers, and holding active communication with overseas R&D institutes according to the following policies:

(1) ICOT researchers are sent to leading R&D and academic institutes in the United States and Europe to investigate technology, exchange opinions, and present papers.

(2) Experts visiting from other countries are welcomed. Selected experts are invited for short periods to exchange ideas.

(3) This exchange also takes place with overseas government-sponsored R&D institutes that are engaged in high-technology computer development projects.

The R&D results on the fifth generation computer are to be disseminated for the purpose of establishing creative and technical foundations of computer science in Japan. To meet this purpose, the results are collated and their use promoted. In May 1986, R&D results of the initial stage, such as the PSI machine and SIMPOS, have gone through the procedure for making them available for use. Since then, their dissemination has been actively promoted. This will expand the scope of the environment for developing fifth generation computer application software and thus greatly contribute to establishing the foundations of widespread use of fifth generation computers. For more details on current R&D results and their perspective, we refer to our recent paper.

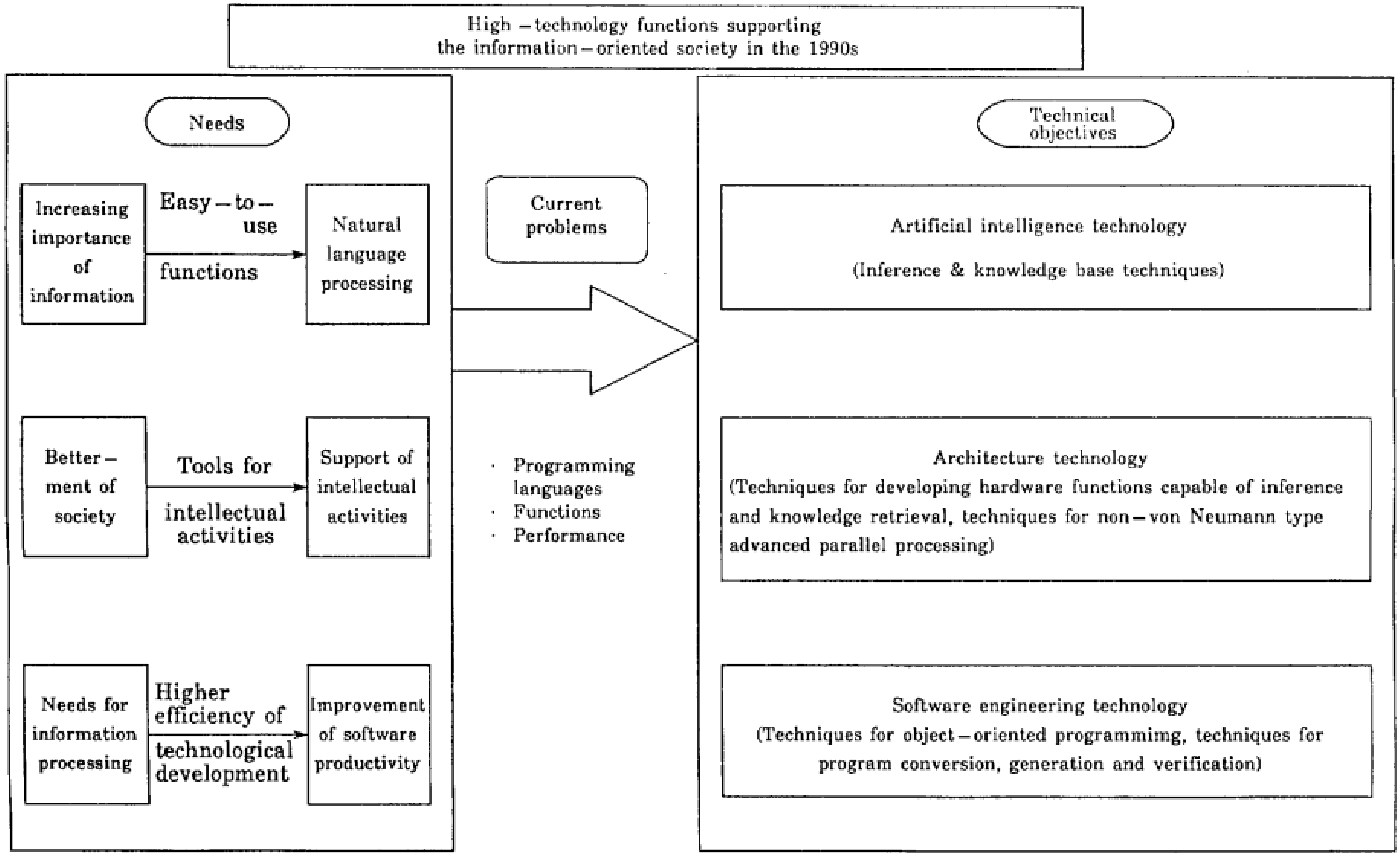

Figure 1 Background of 5th Generation Computer R&D
(Why is a fifth generation computer necessary?)

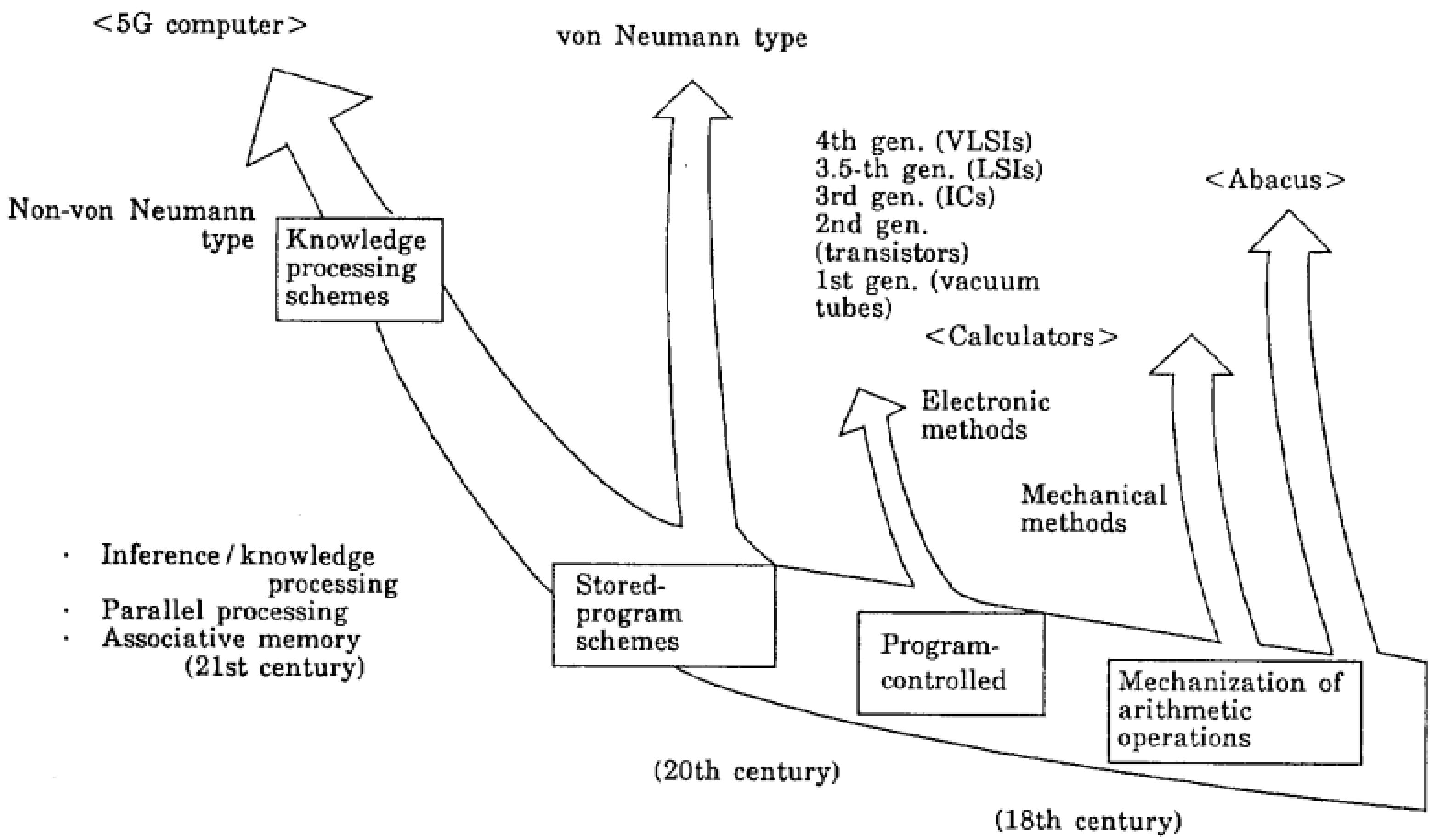

Figure 2 Situating 5G Computers in the Hierarchy of Computer Technology

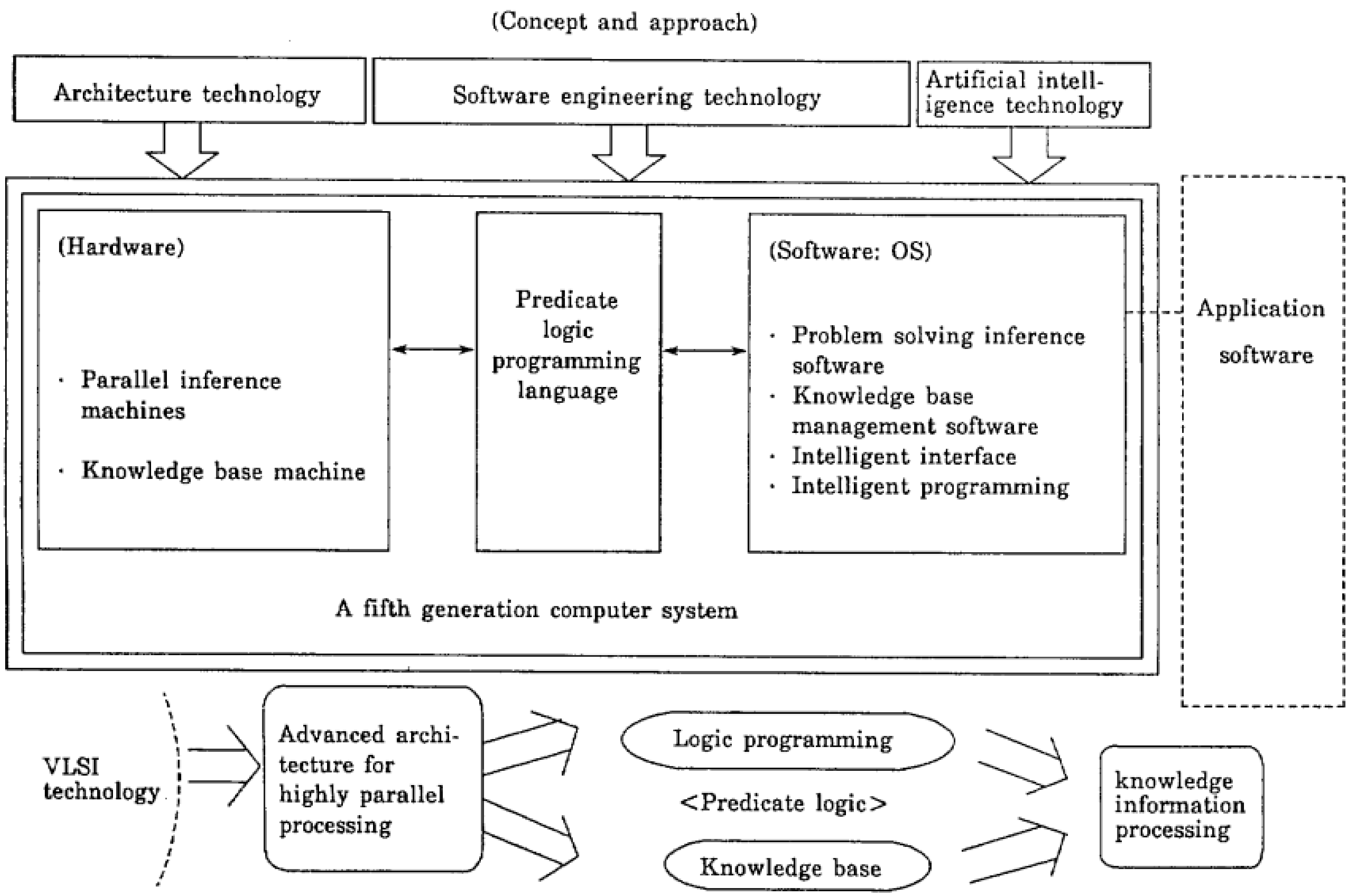

Figure 3 Basic Structure of a Fifth Generation Computer

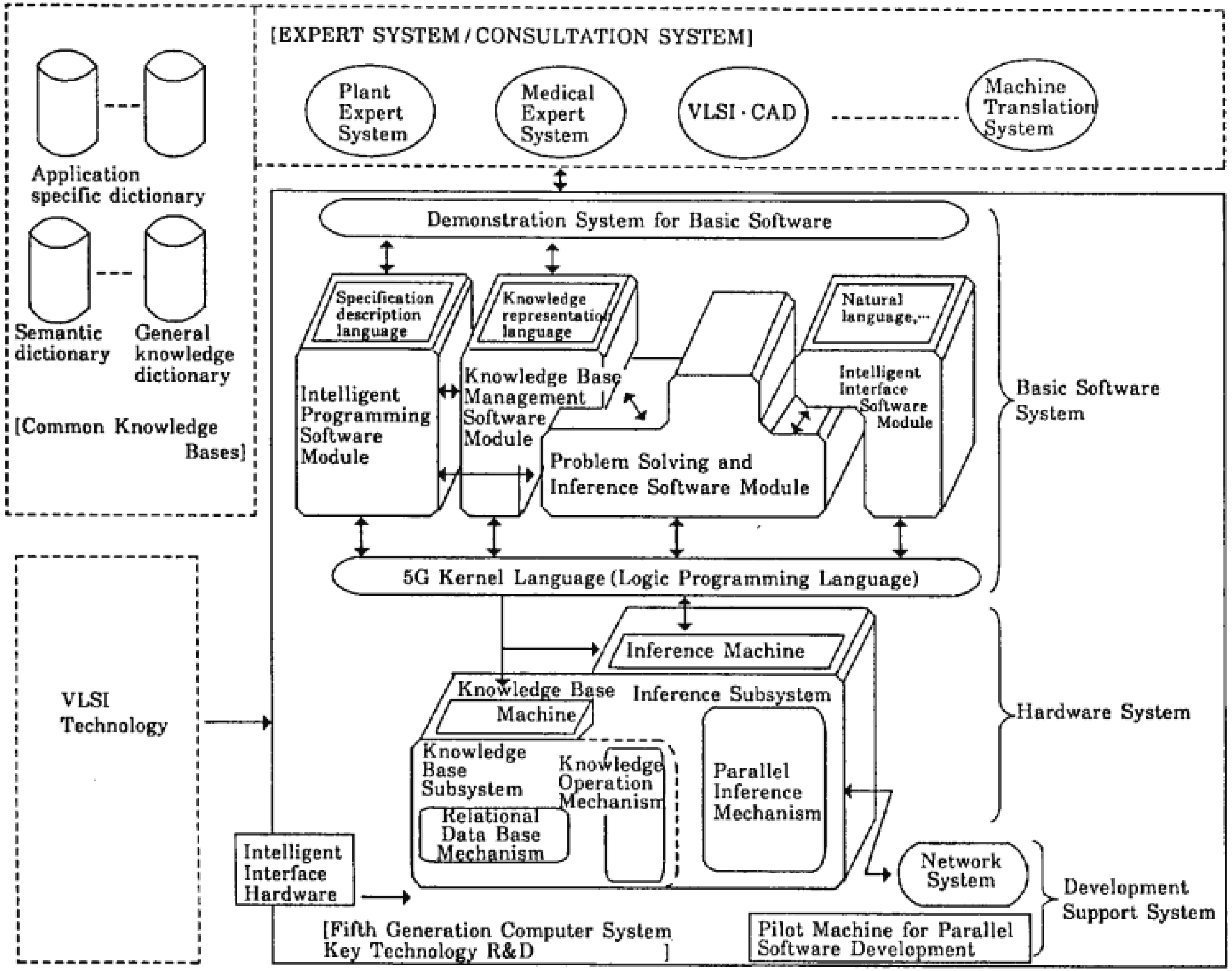

Figure 4 Basic Configuration Image of Fifth Generation Computer System

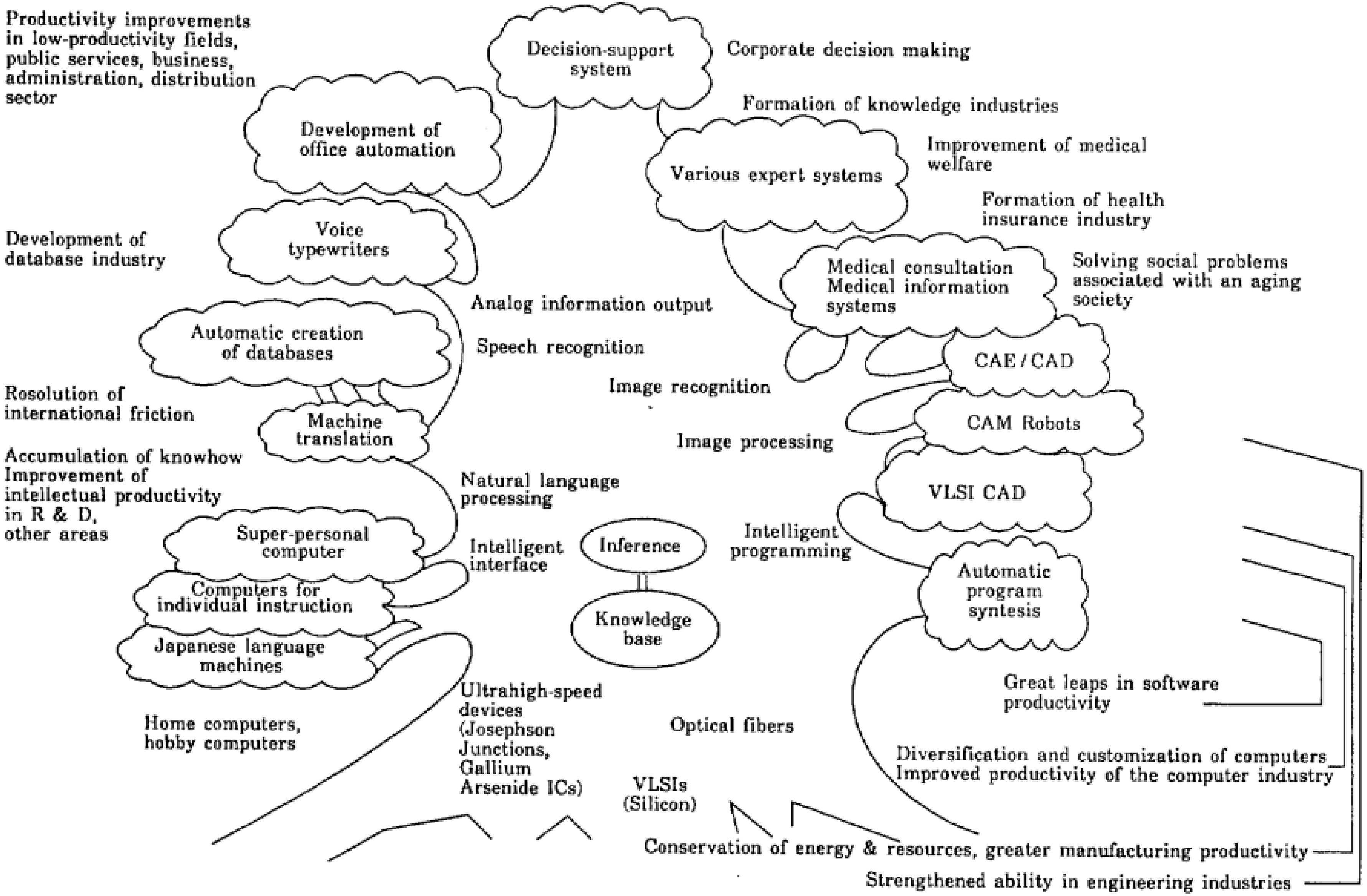

Figure 5 Social Impacts of 5G Computers