TM-0077

A Personal Perspective on Some Aspects of the FGCS —Preliminary Considerations for Fifth-Generation-Computer Networks—

by
Akihito Taguchi

September, 1984

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

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

PREFACE

The Fifth Generation Computer Systems project, called the FGCS for short, is a Japanese national project (1982-1991). ICOT (Institute for New Generation Computer Technology) is the central organization of this project. The FGCS is aimed at creating new basic technology required to build intelligent systems capable of knowledge processing; they are expected to come into widespreaded use during and beyond the 1990's. Such a system is called a KIPS (Knowledge Information Processing System).

The same philosophy and technology as those for the FGCS project will be adopted in the communication network field, as well as in other information-processing fields.

We have now a new question: what will the requirements of a new-generation communication network be as the infrastructure for fifth generation computer systems? The FGCS project is designed only to carry out a portion of the overall research work necessary to build KIPSs; not explicitly included in this project is communication network technology. However, at the final stage of this project, the investigation of total systems, including communication networks, will be essential for developing practical KIPS applications.

This article is intended only as a preliminary consideration about the requirements of communication networks for fifth generation computer systems.

I should like to thank Kazuhiro Fuchi, Director of ICOT Research Center, Kunio Murakami, Chief of ICOT First Laboratory, and Toshio Yokoi, Chief of ICOT Third Laboratory, for providing the opportunity to pursue this work.

CONTENTS

(1) What are Fifth Generation Computers? What is the FGCS Project?

* The FGCS is aimed at KIPS (Knowledge Information Processing System)

  Logic Programming (PROLOG) is the kernel of the FGCS

Today, we obviously need more natural friendly man/machine interfaces and more intelligent processing; in the near future, computers will be applied not only in data processing but also in knowledge-information processing. On the other hand, we have the seed (VLSI technology) for implementing more advanced sophisticated computer architectures, which are essential for realizing such intelligent processing.

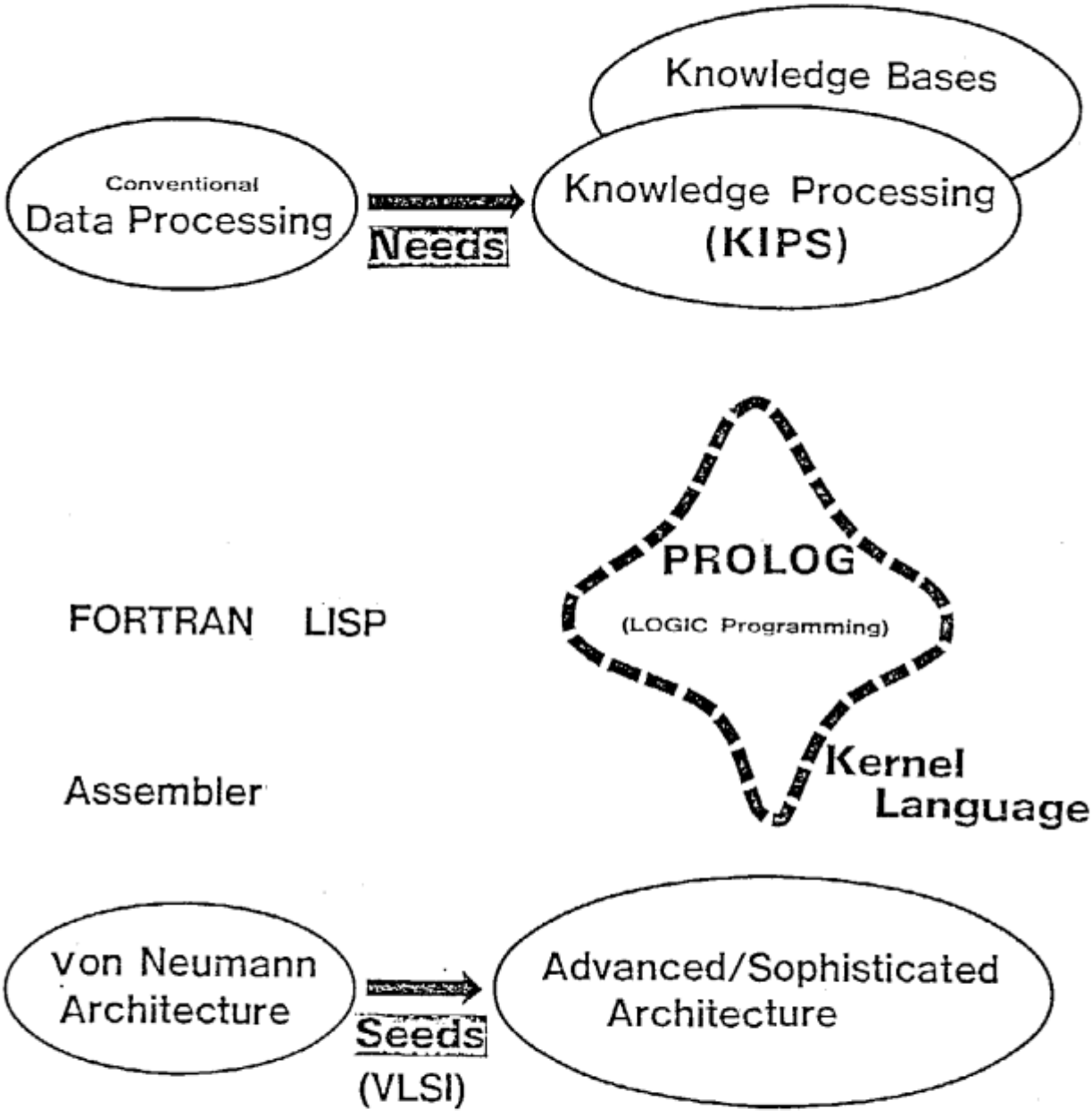
In order to satisfy such a need, the FGCS is aimed at establishing the basic technology required for developing knowledge information processing systems (KIPS). However, it is not clear which programming language is the best to link the need and the seed. Knowledge processing may be based on symbol processing; so, we have now two alternatives, functional programming languages (e.g., LISP) and logic programming languages (e.g., PROLOG). PROLOG is based on predicate logic (Horn logic), and equipped with higher-level functions, such as inference capability (e.g., such as backtracking and non-deterministic processing) and relational database functions, which are useful for developing KIPS applications.(*1) Consequently, logic programming languages are adopted as this missing link and as the basis of the research and development in the FGCS; we call them "Kernel Languages (KLs)", which will be firmware-interpreted PROLOG-like languages. Kernel Languages are regarded as the kernel of the FGCS, because they regulate almost all of the architecture and software of fifth generation computers.

PROLOG is still in its early childhood; the invention of advanced logic programming languages, based on PROLOG, is one of the main R&D themes of the FGCS. Higher-level user languages will also have to be provided for use in developing various KIPS applications; these will be evolved from logic programming languages and possibly based on object-oriented programming concept as ICOT's ESP or Xerox's LOOPS. Object-oriented programming is regarded as a natural evolution or as an upper-level concept of logic programming.

(*1)
The inference capability of PROLOG is considered to be realized by the non-deterministic execution and through trial-and-error, based on backtracking. The following examples would give some suggestions helpful to understand the fact that a programming language can possess the inferential capability, which might sound strange.

Ex1: Some kinds of problems can easily (almost semi-automatically) be solved by using simultaneous algebraic equations.

Ex2: Some kinds of problems can easily be solved by mathematical induction. This means that mathematical induction, in itself, possesses a certain intellectual capability.



FGCS is aimed at KIPS (Knowledge Information Processing System). Logic Programming (PROLOG) is the kernel of FGCS

* FGCS Computer Architectures are regulated by logic programming language (PROLOG)

PROLOG-like logic programming languages, called Kernel Languages (KLs), serve as the native machine languages of fifth generation computers: so, they will regulate the architectures of fifth generation computers. As a result, their architectures should be extremely high-level and far different from a conventional von Neumann architecture: such functions as symbol/list processing (e.g., pattern matching), backtracking and parallel execution control will be directly interpreted by hardware or firmware as well as basic mechanisms, such as stack management and garbage collection.

Kernel Language 0 (KL0), based on sequential PROLOG (e.g., DEC10 PROLOG), has been developed as the machine language of a sequential PROLOG machine PSI (Personal Sequential Inference Machine), which is now under development by ICOT. Kernel Language 1 (KL1) is also being developed: it will be appropriate for parallel PROLOG machines. High-level parallel processing mechanism will be required to develop a high-performance computer, which will support parallel inference used as a principal method in problem-solving. The research and development of efficient parallel execution engines for logic programming languages, as based on data-flow mechanism or reduction mechanism, is the main R&D theme of the FGCS: the research in ICOT is currently proceeding under the name PIM (Parallel Inference Machine).

Almost all of existing "general-purpose" computers are, in fact, originally intended for use only in numerical processing. On the other hand, fifth generation computers executing logic programming languages (e.g., PROLOG) are expected to be truly for general-purpose, because they are equipped with the elemental mechanisms, such as symbol processing and resolution, regarded as the basis of various kinds of human intellectual activities.

In addition to logic-programming-based machines, such as PSI and PIM, a relational database machine is being developed in ICOT as the first step towards a knowledge base machine. A logic programming language (e.g., PROLOG) is also one of powerful query languages for relational databases, which are also based on first-order predicate logic.

FGCS Computer Architectures are regulated by Logic Programming Language (PROLOG)

- **Logic Programming Language Oriented Architecture**
  - Inference (Resolution, Backtracking, etc.)
  - Unification / Pattern -matching
- **Very High-level Architecture**
  - Symbol/List Processing
  - Parallelism (Data-flow mechanism, etc.)
- **Truly General Purpose Computer**
  - For Knowledge Processing (Based on Symbol Processing)
  - For Inferential Problem Solving (Based on Resolution)

* The FGCS is aimed at creating new software technology, based on Predicate Logic

For accomplishing the goal of the FGCS project, it is essential to incorporate various research fields, as listed below:

.Artificial Intelligence

.Knowledge Engineering

.Automatic Programming or Mathematical Theory of Programs (Specification, Synthesis and Verification of Programs)

.Relational Database

.Linguistics

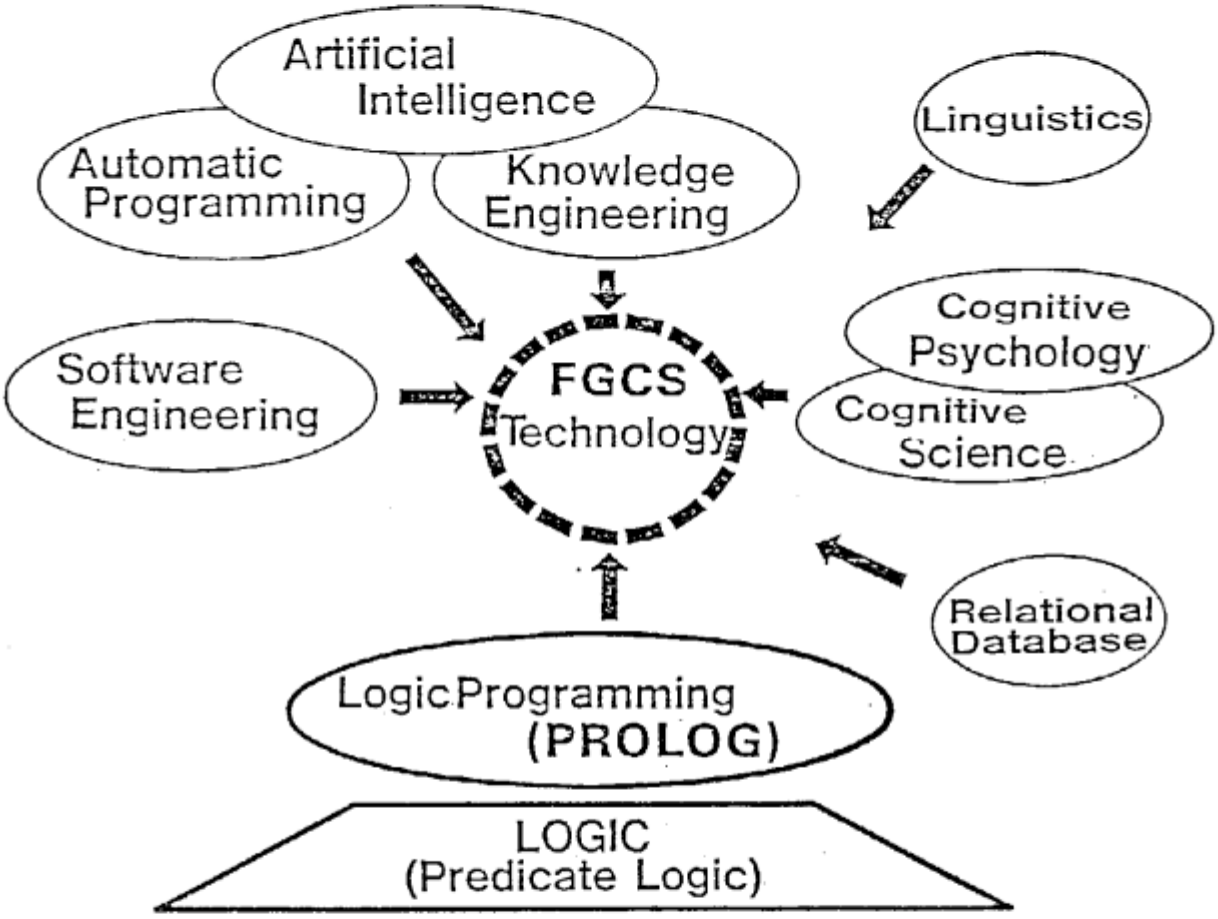
.Cognitive Science or Cognitive Psychology

.Software Engineering

It should be noted that predicate logic is essential to almost all of these researches; this is why the FGCS has adopted logic programming (PROLOG) as the basis of the research and development of KIPSs or fifth generation computers instead of functional programming (e.g., LISP).

FGCS software technology will necessarily claim high-level sophisticated computers, capable of parallel executions; such sophisticated computer architectures will surely be made possible by rapid evolution of VLSI technology.

One of the motivating factors behind the FGCS is the so-called software crisis; it became even more serious in spite of the evolution of software engineering in the 1970's. So, the revolutionary improvement of the programming environment is also one of the major goals of the FGCS: an intelligent programming support system will be developed as a KIPS application.



The FGCS is aimed at new, Software Technology, based on Predicate Logic

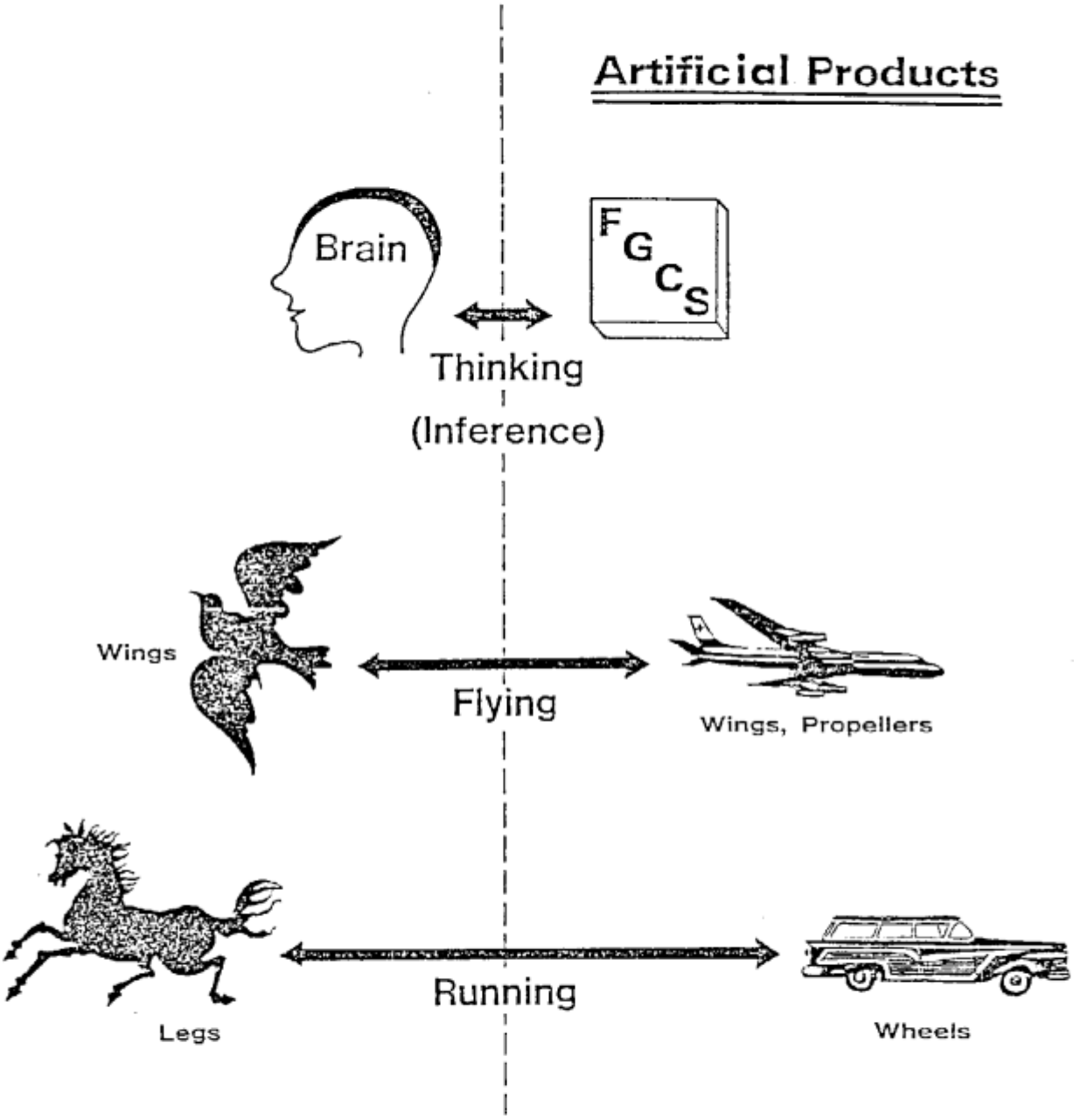
FGCS is AI computer, thinking as a man does !

(The more sophisticated, The more similar)

Generally speaking, the more sophisticated or higher-level the function of an artificial product becomes, the more similar it becomes to the natural. For example, while the running mechanism of a car, moving by means of wheels, is quite different from that of a horse, the flying mechanisms of a airplane and a bird are relatively similar; flying can be said to be higher-level in function than running. As another example, it is well known that the structure of a camera is similar to that of the eye, although their focusing mechanisms are different: a camera lens is analogous to the biological crystalline lens.

Therefore, a fifth generation computer system (i.e., KIPS), which thinks as a man does, is conceived of as being extremely similar to the brain of a human beings, for thinking is far more sophisticated in function than flying or seeing. At least, it is regarded as one of the fruitful approaches to try to make computers imitate how the human brain works. Research into human intellectual mechanisms, as carried out in cognitive science or cognitive psychology, will therefore be essential for realizing fifth generation computers. Especially, research into the mechanism of understanding natural languages will be very important and fruitful.

In order to perform intelligent processing, computer systems (i.e., AI computers) must, to some extent, understand what they are doing. This means that they are requested to be equipped with some knowledge/ knowledge bases. For example, in processing or understanding natural languages, they need to know about the real world; i.e., they must have a kind of common sense, or they need to know the situation in which natural languages are used.



FGCS is AI Computer, thinking as a man does! (The more Sophisticated, The more Similar)

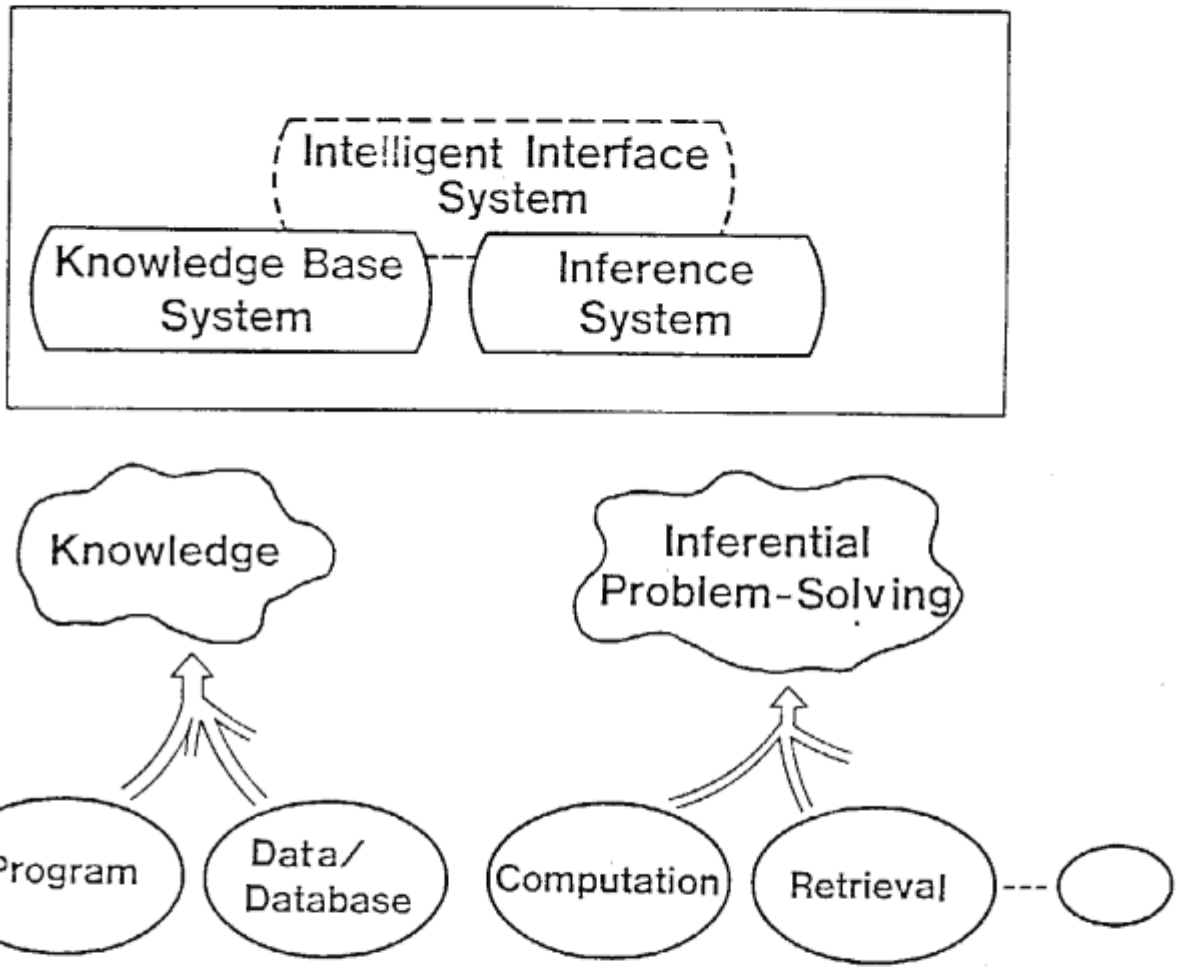
Knowledge and KIPS (Knowledge Information Processing Systems)

KIPS consists of an inference system and a knowledge base system. In addition, KIPS may also be equipped with a natural intelligent man/machine interface; it should be pointed out that such an intelligent interface system is itself a KIPS application.

Knowledge is a higher-level concept, including both programs and data (or database). Programs and data differ in representation rather than in semantics: programs and data are viewed as intensional and extensional representations of the same knowledge, respectively. For example, trigonometric-function SIN(x) can be represented either as a program (e.g. in Taylor expansion) or as data (e.g. in trigonometric-function table). Moreover, the retrieval of data and the computation of programs are integrated into the broader concept of inferential problem-solving through knowledge bases. Logic programming languages (e.g. PROLOG) can represent and manipulate programs and data uniformly as knowledge.

Some kinds of knowledge will also naturally be represented in relational models, and it should be noted that PROLOG can be used to represent relational models. Other various kinds of knowledge would, however, better be represented in other than PROLOG or a relational model. Various techniques of knowledge representation, as shown below, are now being studied; these are considered to be based on logics and will possibly be incorporated into future logic programming languages.

- Mathematical expressions for representing mathematical knowledge
- Logical expressions for representing logical relations between facts (PROLOG is the most basic example.)
- Procedure-oriented programming languages for representing procedural knowledge (e.g. programs)
- Compound data structures, such as "Frame", "Semantic network", "Object" and so on, for representing complex relations between facts (e.g. hierarchical relations)



Knowledge and KIPS

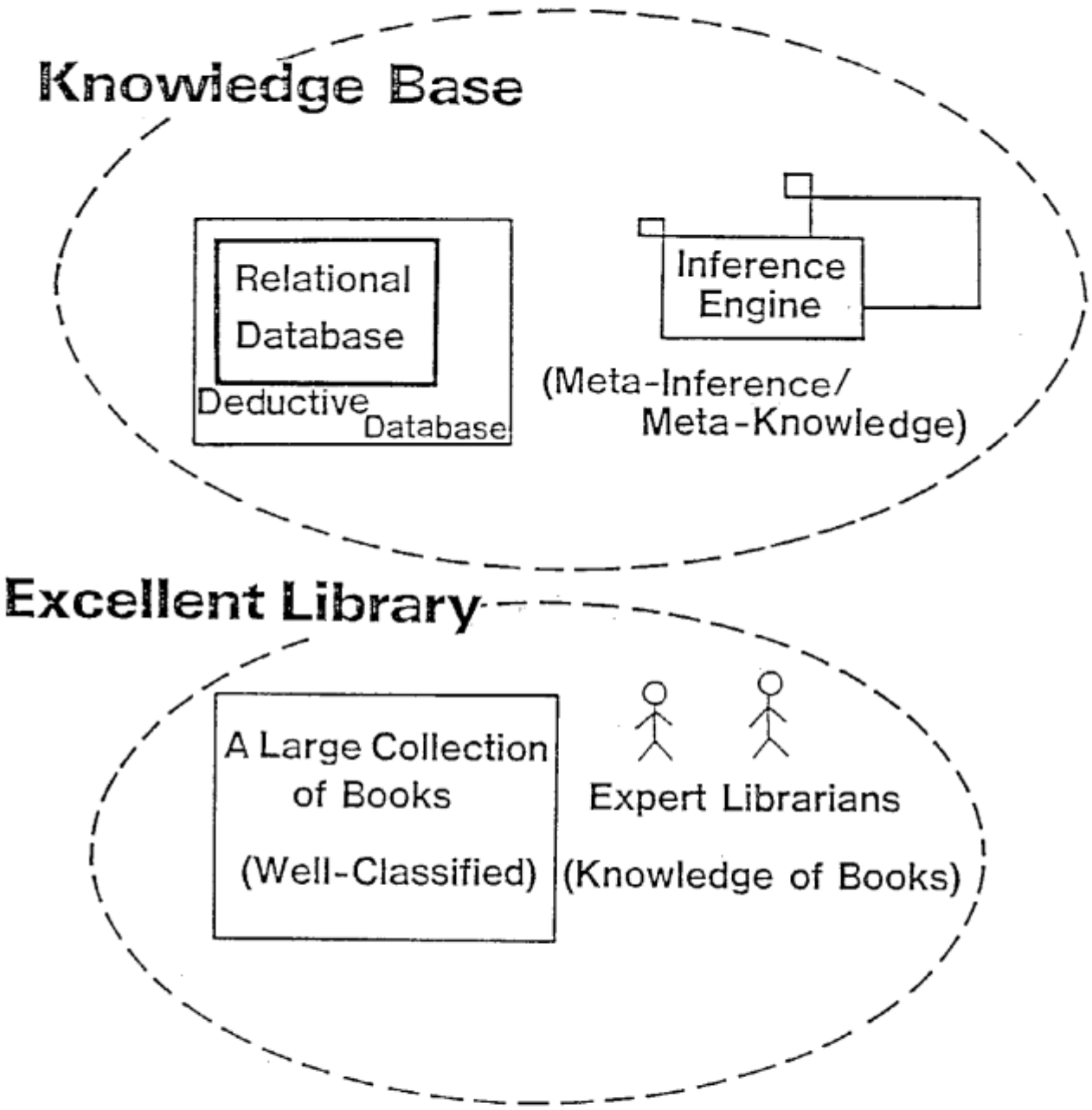
(Knowledge Information Processing System)

* Knowledge Base is similar to a library with good librarians

A PROLOG program, consisting of a set of facts (relations) and rules, can be considered to be a first approximation to the knowledge base. It is more powerful and flexible to construct models organically connecting individual relations than relational databases; a conventional relational database, also based on first-order predicate logic, can only be viewed as a set of facts (relations).

Formerly, the meaning of "data" or "database" was known only to the user (i.e., a program); a database or a data-file only contained "something". On the contrary, a knowledge base is like an organism: it knows the meaning of the stored information. For example, a rule contained in a PROLOG program can be considered to organically represent the meaning and relationship of individual facts/relations.

Roughly speaking, a knowledge base is analogous to an excellent library: while the latter has not only a large collection of books but also professional librarians, the former needs to be equipped with some knowledge and inference capability in addition to a database. As librarians have knowledge about the books in their library, a knowledge base must have knowledge of the knowledge (facts/rules) stored in its own database. Such knowledge is called meta-knowledge; inference that uses meta-knowledge is called meta-inference. It should be noted that PROLOG is useful to implement such a hierarchically structured knowledge base, because a PROLOG program itself can be manipulated by another PROLOG program.



Knowledge Base is similar to a Library with good librarians

* KIPS Applications in The FGCS

At first, fifth generation computers or KIPSs will be used in intelligent office automation environments, especially for providing more advanced intelligent man/machine interfaces, in order to enhance the productivity of human intellectual activities in offices.

A practical intelligent programming support system is also an R&D theme of the FGCS project as one of KIPS applications. All knowledge required for programming processes, such as programming languages, programming methodologies and existing program-packages, will be prepared as knowledge bases. Moreover, automatic programming techniques, such as automatic verification, automatic synthesis and specifications of programs, will be incorporated into such an intelligent programming system. These techniques have been studied in theory but not in practice, because it is very difficult to apply them to practical programs in conventional procedural languages, such as FORTRAN. On the contrary, a logic programming language, such as PROLOG, seems to be adequate to apply them, because it is high-level and has non-procedural (declarative) semantics. For example, logic programming languages can be viewed as program specification languages; or, at least, the former are very close to the latter.

KIPS applications, called expert systems or consultation systems, will take over some of the intellectual activities of human experts by accessing large-scale knowledge bases. For example, MYCIN (Stanford University) and R1 (DEC Corp.) are both well-known; the former is for assisting medical examination in place of a medical specialist, and the latter for configuring VAX-11 computer systems.

Today, a number of microcomputers or LSIs are built into various equipments (e.g., for engine control in automobiles); fifth generation computers will similarly be embedded in various devices. In future, computers will be applied not only in data processing but also in knowledge-information processing; this means that fifth generation computers will pervade the various new fields, including human creative activities and intellectual works. Pioneering new applications may also be important in the FGCS project.

KIPS Applications in the FGCS

- Intelligent Programming Support Systems
- Advanced Office Automation Systems
- Natural Language Processing Systems
  - ➡ Intglligent Man/Machine Interfaces
- CAD (VLSI-CAD)
- Consultation Systems/Expert Systems

* What is Logic Programming (PROLOG programming) like ?

Logic programming languages (e.g. PROLOG) are rather different from procedure-oriented languages. For example, PROLOG dispenses with the concept of "control flow", which is common to conventional procedure-oriented languages. A PROLOG programmer is like the pilot of an airplane. While a driver of a motorcar must be conscious of the surface route (course) towards his destination, a pilot of an airplane need not: a pilot only needs to know his destination, because an airplane is not constrained by the road. Similarly, a PROLOG programmer need only declaratively describe logical relations and conditions ("what to do") inherent in his problem to be solved; he is not obliged to specify the solving procedure ("how to do"). It should be pointed out that a PROLOG program is easy to read or understand owing to its declarative semantics. Sentences (called "clauses" in PROLOG) whose conditions are satisfied will non-deterministicly be selected and executed.

A PROLOG system might be said to be similar to a production system; its inference capability is realized by means of top-down, depth-first search and backtracking. In other words, PROLOG may be similar to conventional parametric or problem-oriented programming languages, which are non-procedural and easy to use. The former is, however, for general problem-solving while the latter are intended for specific use in a very narrow field as if they were boats in a lake, which are free but restricted to the inside of the lake.

A PROLOG program consists of a set of facts (relations) and rules, while a relational database is a set of facts (relations). In other words, a PROLOG program can be viewed as a deductive database, evolved from a relational database.

Logic programming may also be similar to solving some kinds of arithmetic problems by using simultaneous algebraic equations: making equations is analogous to describing logical relations in PROLOG. Once equations are made, the arithmetic problems can easily (almost semi-automatically) be solved.

Logic programming based on predicate logic, of course, has much in common with writing in a natural language, because predicate logic can be viewed as a formulation of part of the natural language. One future principal programming process will be interactive "knowledge programming", based on logic programming: this will involve constructing a model or a program while accessing a number of knowledge bases, such as program libraries and document files. In other words, the future programming process will be closer to the writing process of a research article.

What is Logic (PROLOG) Programmig like ?

- Similar to Parametric or Problem-Oriented Languages
- Similar to constructing Relational Models
- Similar to using Simultaneous Algebraic Equations
- Similar to Writing in Natural Language

Jet Plane Airplane Helicopter	Air Transport	Logic Programming Language	**PROLOG**
Automobile Motorcycle Bicycle	Surface Transport	Procedure–Oriented Programming Language	Pascal Fortran Assembler

PROLOG PROGRAM can be viewed as Evolution of Relational Database

A PROLOG program consists of a set of facts (relations) and rules. A relational database, also based on first-order predicate logic, can only be viewed as a set of facts (relations).

The rules are considered to represent some relationships among the facts. For example:

Example 1: ancestor(X, Y):- father(X, Y).

This means, "For any X and any Y, X is an ancestor of Y, if X is father of Y."

Example 2: ancestor(X, Y):- father(X, M), ancestor(M, Y).

This means, "For any X and any Y, X is an ancestor of Y, if there exists some M whose father is X and who is an ancestor of Y." It should be noted that "ancestor" is recursively defined. Recursive definition has been a concise and powerful technique even in conventional programming languages; it is still more natural and native in PROLOG programming. Recursive definition is considered to be essential for knowledge representation and knowledge processing.

PROLOG Program can be viewed as Evolution of Relational Database

```
father (alec, ken).          mother (maggie, ken).
father (alec, tom).          mother (maggie, tom).
father (alec, ted).          mother (maggie, ted).
father (sammy, teddy).       mother (mary, teddy).
father (sammy, jimmy).       mother (mary, jimmy).
father (charles, emmy).      mother (jane, emmy).
father (charles, nora).      mother (jane, nora).
father (richard, alec).      mother (elizabeth, alec).
father (richard, sammy).     mother (elizabeth, sammy).
father (athur, charles).     mother (victoria, charles).

ancestor(X,Y):-father(X,Y).                          [*]
ancestor(X,Y):-father(X,M),ancestor(M,Y).            [*]

pair(HB, WF):-father (HB, CHD),mother(WF,CHD).       [*]
```

[*] Rules

Examples of Usages

(1) ?-father(X,tom).

-->X=alec

(2) ?-ancestor(X,tom).

-->X=alec

-->X=richard .

* PROLOG has powerful facilities for developing KIPS applications

PROLOG language/system possesses prinitive but essential knowledge-processing capabilities, such as inference function and symbol-processing. Therefore, it might be viewed as a first approximation to a KIPS.

The following table enumerates the outstanding features of PROLOG for developing KIPS applications.

No	PROLOG Features	Explanations
1	Equipped with natural paradigms PROLOG possesses primitive but essential features required to incorporate and unify all of the excellent qualities of existing programming languages.	In PROLOG, paradigms (e.g., recursive definition of data and procedures, lazy evaluation and so on) are natural and native. PROLOG is suitable to both object- and procedure-oriented approaches.
2	Equipped with advanced symbolic processing capability Unification of PROLOG is more powerful for symbolic processing than conventional pattern matching.	Knowledge processing is carried out on the basis of symbol/list manipulations.
3	Equipped with deductive database function PROLOG and relational database are both based on first-order logic. Data/databases and programs can be uniformly represented and manipulated in PROLOG.	The uniformity of data and programs is the first step toward knowledge baseses. It is easy to introduce meta-knowledge (programs) regarding to the use of various kinds of knowledge (programs or data/databases).
4	Equipped with inference capability This capability is realized by means of top-down, depth-first search and backtracking.	A PROLOG system might be said to be similar to a production system.
5	Appropriate for efficient hardware implementations, especially parallel machines (based on data-flow or reduction mechanisms).	The rapid evolution of VLSI technology will make these sophisticated machine architectures possible.

(Note: From material by T. Yokoi (ICOT TM-0026))

PROLOG has powerful facilities for Developing KIPS Applications

- Equipped with Natural Programming Paradigms

  PROLOG possesses Primitive but Essential Features, Required to Incorporate and unify all of the excellent Qualities of existing Programming Languages.

- Equipped with Advanced Symbolic Processing Capability

  Unification of PROLOG is more Powerful for Symbolic Processing than conventional pattern matching.

- Equipped with Relatinonal Database Function

  PROLOG and Relational Database are both based on First-Order Predicate Logic. Data/Databases and Programs can be uniformly represented and manipulated in PROLOG.

- Equipped with Inference Capability

  This capability is realized by means of Top-Down, Depth-First Search and Back-tracking.

- Appropriate for Efficient Hardware Implementations,

  especially Parallel machines (Based on Data-Flow or Reduction Mechanisms).

(2) What Impact will the FGCS have on Computer Network Technology?

■ FGCS Impact on Computer Network Technology

The same philosophy and technology as those for FGCS will be adopted in the communication network field, as well as in other information-processing fields.

Future intelligent networks will have the capability of autonomic intelligent self-control, such as adaptive routing and adaptive flow-control. These intelligent controls will be performed through knowledge processing by KIPSs installed in networks.

Future intelligent networks will support knowledge communications. Knowledge communication implies, firstly, intelligent services provided by intelligent servers and knowledge bases, and secondly, the conversion of knowledge representations, such as protocol translation and media conversion. These will be much more intelligent than conventional capabilities, such as code conversion and data compression.

We have now a new problem to be studied: what will the requirements of a new generation communication network be as the infrastructure supporting fifth generation computers?

At first, fifth generation computers will possibly be used for intelligent office automation environments, especially for providing more advanced man/machine interfaces. As an office automation system can be realized by personal computers and various types of local area networks, fifth generation computers will possibly require a new type of communication network systems as their own nervous systems. At the final stage of the FGCS project, the investigation of total systems will also be necessary for realizing an integrated intelligent system, called co-operative problem solving system; it will consist of several fifth generation computers (or KIPSs), cooperating through communication networks in order to accomplish their common purposes. As another example, the integration of a fifth generation computer (for intelligent processing), a database machine and a super-computer, which are scattered around the communication network, will be required for building a practical KIPS application (e.g., a nuclear-reactor diagnostic expert system).

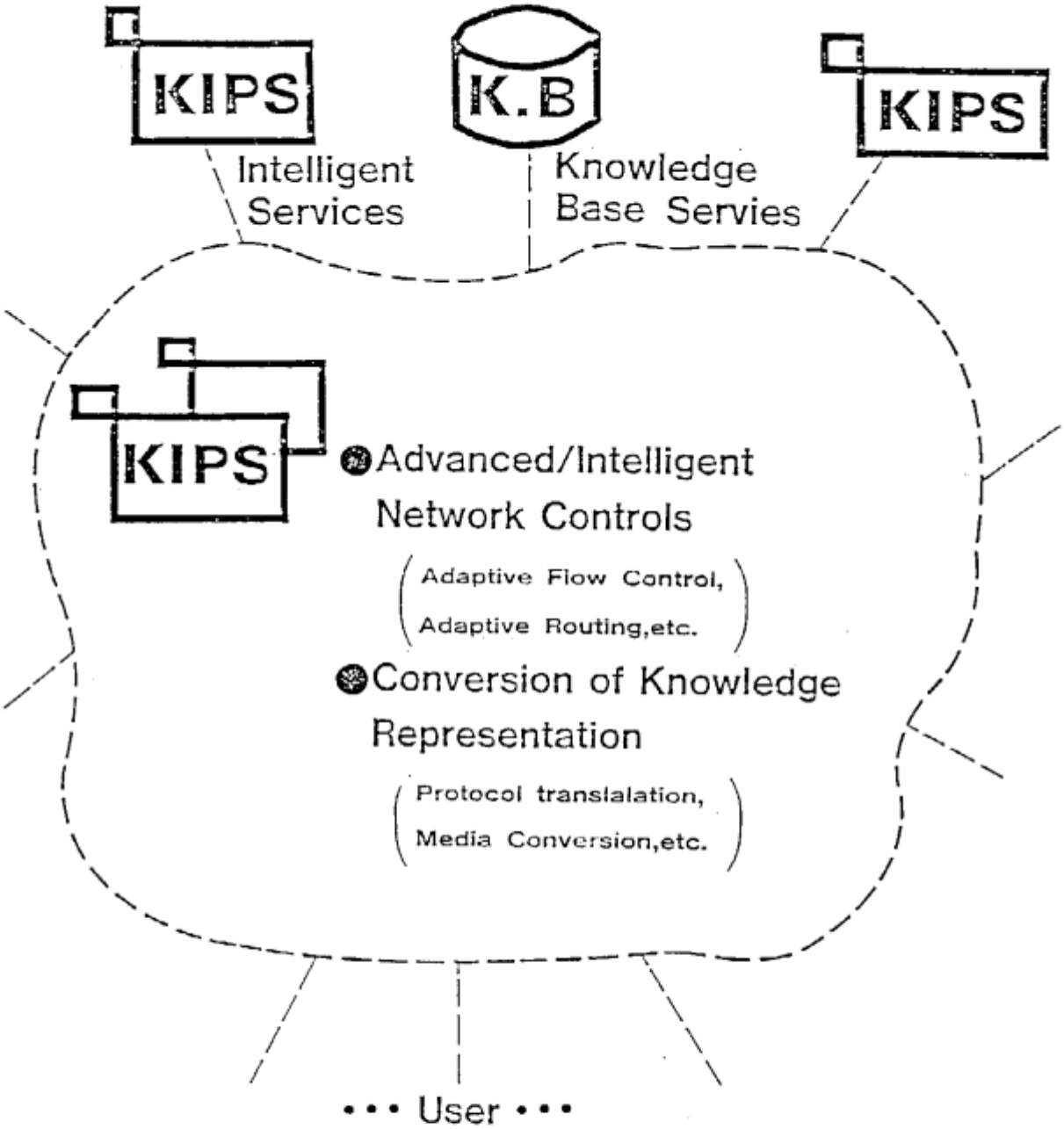
FGCS Impact on Computer Network Technology

- **FGCS** Technology Applied to Future Intelligent Network Systems
  - ➡ For Advanced Network Control
    For Knowledge Communication
- Requirements of Communication Networks as the Infrastructure for FGCS Computers

  (Various LANs for Personal Computers in OAs
  ? for FGCS Computers)

Future intelligent networks will have the capability of autonomic self-control, such as adaptive routing and adaptive flow-control. These intelligent controls over network systems will be performed by KIPSs installed in networks themselves, as IBM's YES/MVS exerts interactive control over the MVS operating system as an aid to computer operators. YES/MVS is a KIPS application (an expert system) that runs continuously in real time. By the way, for such network controls, gathering of statistics from various nodes and propagation of instructions are essential; the group communication facility (broadcasting protocol of logical-level layer), as supported by INI (Internal Network in ICOT), will be useful for these purposes. Extra transmission required for such controls will hardly disturb other usual service transmissions, firstly because of the future use of high-capacity transmission media such as optical fibre, and secondly because the load level of future networks will need to be held far below the saturation point in order to guarantee reasonable throughput and transfer-delay.

While fifth generation computers will perform knowledge-processing instead of data-processing, intelligent networks will support knowledge-communications instead of conventional data-communications: they will transmit not representations but semantics. Knowledge communication implies, first, intelligent services provided by intelligent servers and knowledge bases around the network, and second, the conversion of knowledge representations, such as protocol translation and media conversion; the latter may also be performed by KIPSs installed within the network itself.



A Prospect for Future Intelligent Networks

* Some Requirements of Communication Networks as the Infrastructure for FGCS Computers

What will a new generation communication network, as the infrastructure for fifth generation computers, be like? As an office automation environment can be realized by personal computers and various types of local area networks, fifth generation computers will possibly require new network environments. For example, a cellular-radio network is intended to be used as the basis of mobile-information-systems (such as mobile-office-facilities), which are one of the R&D themes in the UK's Alvey fifth-generation-computer program; this national cellular-radio network, planned in the UK, would comprise some 400 base stations. (Electronics July 12, 1984)

Only some requirements of such communication networks will briefly be discussed here, though most of them are now obscure and indefinite.

Generally, KIPSs or fifth generation computers should be capable of the high-speed processing of a large volume of data/knowledge, while accessing to large knowledge bases locally or through communication networks; e.g., for realizing advanced man/machine or man/network interfaces. So, first of all, knowledge communications will require the capability of transmitting interactively a large volume of information.

In addition to the future use of high-transmission-capacity communication media (e.g., optical fibre), the load level of future generation networks will need to be held far below the saturation point in order to guarantee reasonable throughput and transfer-delay; the low-level network load will be claimed not only for real-time applications (e.g., voice communications and tele-conference), but also for expedited transmission of control information and statistics to be required for intelligent network control/management. For gathering and propagating of such control information and statistics from/to various network nodes, the group communication facility (broadcasting protocol of logical-level layer), provided by INI (Internal Network in ICOT), will be useful as well.

Broadcasting facility (group communication facility) is generally appropriate for concurrent or nondeterministic processing over the network environment, e.g., for executing distributed PROLOG programs (this might be analogous to simultaneous query to distributed databases) and for co-operative problem solving systems or distributed problem solving systems, including intelligent network control systems. Before long, various broadcasting physical communication media will be available; e.g. satellite communication and wireless communication.

Voice need not be transmitted so accurately in our face-to-face conversation. Fifth generation computers will be equipped with intelligence, e.g., default reasoning and common-sense reasoning. This means that even fuzzy (or noisy) transmission facility may be convenient and efficient in certain special cases; the transmission cost will be much lower only to exchange acceptably accurate information. In order to reduce transmission volume, the situation or context of communication should be taken into consideration as we do in our conversation; this technology will be more sophisticated than that used now in tele-conference communications.

Some Requirements of Communication Networks as the Infrastructure for FGCS Computers

- Capability of Interactive Transmission of Large Volume
  - ➡ Knowledge Communications
    Advanced Interactive Man/Machine Interfaces
- Low-Level Network Load
  - ➡ Intelligent Network Control
    Real-time or Interactive Applications
- Broadcasting Facility (Group Communication Facility)
  - ➡ Network Control/Management
    Non-deterministic Problem Solving
    Concurrent Problem Solving
    (Co-operative Problem Solving Systems)
- Fuzzy/Noisy Transmission
  - ➡ Extremely Interactive High-Volume Communications
    (Complemented by FGCS Inference Capability)

* Examples of FGCS Technology Transfer

(as Replacements for Conventional Techniques, e.g., Petri-Net and FAPL)

Software technology adopted for the FGCS will be effective as well as conventional techniques, currently used in network technology (e.g., petri-net, state-transition-diagram and IBM's FAPL).

It should be noted that most of these new techniques are based on predicate logic and so will naturally be incorporated into logic programming languages.

Several knowledge-based systems for configuring or maintaining computer or network systems have already been proposed, e.g., R1 (or XCON from DEC), for configuring VAX-11 computer systems (Mcdermott,J., R1: A Rule-Based Configurer of Computer Systems, Artificial Intelligence 19 (1982)), and ACE, for the maintenance of telephone-networks (Vesonder,G.T.,et.al., ACE: An Expert System for Telephone Cable Maintenance, Proc. of IJCAI'83 (1983)).

Examples of FGCS Technology Transfer

(as Replacement for Conventional Techniques, e.g, Petri-Net/FAPL)

- Expert Systems or Consultation Systems
  for Diagnosing and Configuring Network-Systems

- Computational Linguistics, such as DCG (Based on PROLOG) and ATN,
  for Protocol Specification and Translation

- Logic (e.g., Temporal Logic)
  for Protocol Specification

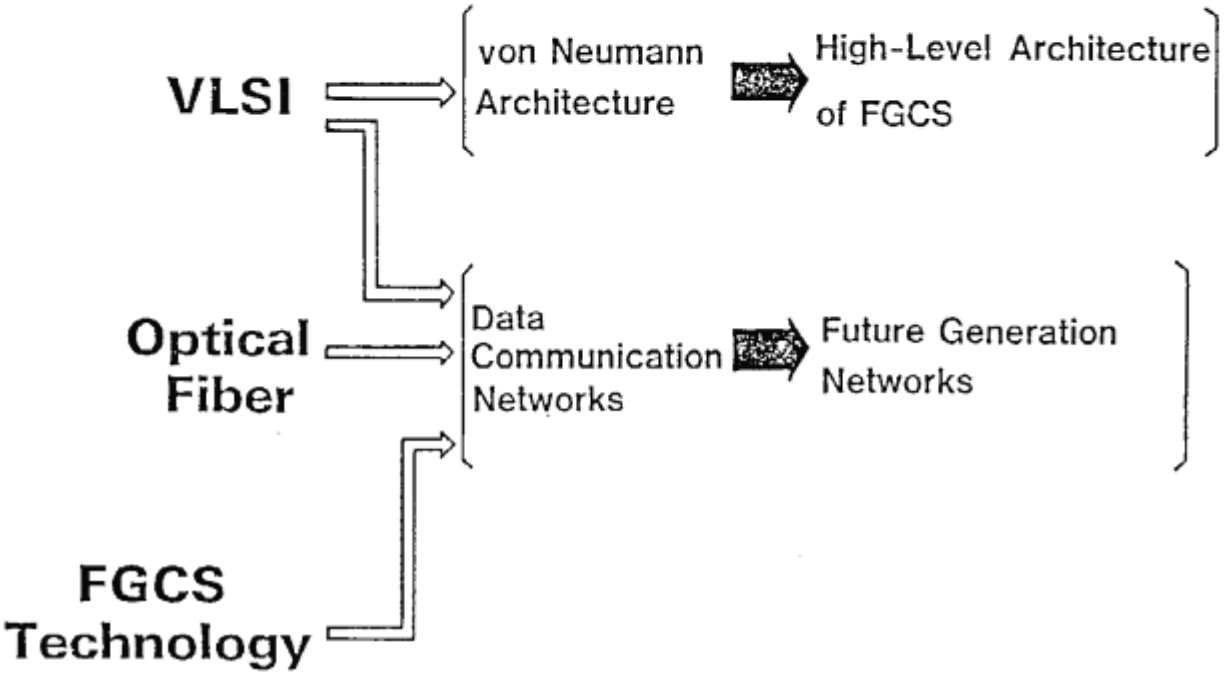
- Logic Programming and Object-Oriented Programming
  for Modeling of Distributed Processing Systems

- Parallel Processing Architecture (Data-flow, etc.)
  for Communication Processer

* New Generation Intelligent Networks Supported by and Making use of Revolutionary New Technology

While fifth generation computers will make good use of and be supported by the rapid evolution of VLSI technology, the intelligent knowledge-communication networks of the new generation will be made possible by optical fiber technology in addition to VLSI and FGCS technology.

Petroleum has become still more valuable since the invention of automobiles and the start of mechanical industries than before: not so much oil would be needed only for lamps. Similarly, new generation communication network systems will necessarily make good use of revolutionary new technology, such as VLSI and optical fiber, as well as fifth generation computer systems.



New Generation Intelligent Network Supported by and making use of Revolutionary New Technology

BIBLIOGRAPHY

[Aze1] Azema,P., et. al.,"Specification and Verification of Distributed Systems Using PROLOG Interpreted Petri Net", 7th ICSE, 1984

[Bac1] Backus,J.,"Function Level Computing",IEEE Spectrum, 1982

[Bow1] Bowen,D.L., "DEC system-10 PROLOG USER'S MANUAL", University of Edinburgh, Dept. of AI, 1981

[Bux1] Bux,W., "Local Area Subnetworks: A Performance Comparison", IEEE Transactions on Communications Vol.COM-29 No.10, 1981

[Chi1] Chikayama, T., "ESP Reference Manual", ICOT TR-044, 1984

[Clo1] Clocksin,W.F, "AN INTRODUCTION TO PROLOG", ARTIFICIAL INTELLIGENCE (Edited by O'Shea,T. and Eisenstadt,M.) HARPER & ROW, 1984

[Clo2] Clocksin,W.F and Mellish,C.S., "Programming in Prolog", Springer-Verlag, 1981

[Cod1] Codd,E.F.,"Relational Database:A practical Foundation for Productivity",CACM Vol.25 No.2, 1982

[Com1] Computer Vol.16 No.10 (Knowledge Representation), 1983

[Deg1] DeGroot, D., "Restricted And-Parallelism", FGCS'84, 1984

[Fis1] Fischer,G. and Schneider,M.,"Knowledge-Based Communication Processes in Software Engineering", 7th ICSE, 1984

[Fuc1]* Fuchi,K.,et.al.,"An Invitation to Cognitive Science", Nippon Hosou Kyoukai, 1983

[Fuj1]* FUJITSU Vol.35, No.3 (Car Electronics), 1984

[Gfe1] Gfeller,F., "INFRANET: Infrared Microbroadcasting Network for In-house Data Communication", 7th ECOC, 1981

[Gri1] Griesmer,J.H. and et.al., "YES/MVS: A Continuous Real Time Expert System", AAAI-84, 1984

[Ico1] "RESEARCH REPORT ON FIFTH GENERATION COMPUTER SYSTEMS PROJECT", ICOT, 1984

[Inm1] "OCCAM Programming Manual", INMOS Ltd., 1984

[Koe1]* Koeta,I.,Saitoh,T. and Inose,H., "Protocol Specification in Temporal Logic", Proc. of 28th National Conference of Information Processing Society of Japan, 1984

[Kow1] Kowalski,R.A.,"Logic for Problem Solving",North Holland, 1980

[Kow2] Kowalski,R.A.,"PROLOG AS A LOGIC PROGRAMMING LANGUAGE", AICA, 1981

[Les1] Lesser, V.R. and Corkill, D.D.,"THE DISTRIBUTED VEHICLE MONITORING TESTBED: A Tool For Investigating Distributed Problem Solving Networks", AI MAGAZINE Fall 1983

[Mcd1] Mcdermott,J.,"R1: A Rule-Based Configurer of Computer Systems",

Artificial Intelligence 19, 1982

[Mot1] Moto-oka, T. and Fuchi, K., "THE ARCHITECTURES IN THE FIFTH GENERATION COMPUTERS", Proceedings of the IFIP 9th World Computer Congress, 1983

[Nor1] Norman,D.A.,Simon,H.A.,et.al.,"Perspectives on Cognitive Science", Ablex Publishing Corporation, 1981

[Per1] Pereira,F.C.N. and Warren,D.H.D., "Definite Clause Grammars for Language Analysis--A Survey of the Formalism and a Comparison with Augmented Transition Networks", Artificial Intelligence 13, 1980

[Pere1] Pereira,L. and Nasr,R., "DELTA-PROLOG: A Distributed Logic Programming Language", FGCS'84, 1984

[Ren1] Rentsch,T.,"Object Oriented Programming", Sigplan Notices Vol.17 No.9, 1982

[Sch1] Scherr,A.L., "A Perspective on Communications and Computing", IBM Systems Journal Vol.22 No.1 (1983)

[Scu1] Schultz,G.,et al, "Executable Description and Verification of SNA", IEEE Transactions Vol.COM-28 No.4, 1980

[Sis1] Simons,G.L.,"Towards Fifth-Generation Computers", NCC Publications, 1983

[Ste1] Stefik,M.,et.al., "Knowledge Programming in LOOPS", AI MAGAZINE Fall 1983

[Tag1] Taguchi,A.,et.al.,"INI: Internal Network in ICOT and its Future", ICCC'84, 1984 (To appear)

[Tag2] Taguchi,A.,"Writing in a Foreign Language and Programming in Warnier's Methodology -A Study of Programming Processes-", ICOT TM-0057, 1984

[Tag3] Taguchi, A.,"A Study of Processes of Writing in a Non-Native Language -Analogy to Warnier's Programming Methodology-", 29th National Conference of Information Processing Society of Japan, 1984

[Uch1] Uchida,S. and et.al.,"Outline of the Personal Sequential Inference Machine PSI", New Generation Computing Vol.1 No.1, 1983

[Ves1] Vesonder,G.T.,et.al.,"ACE: An Expert System for Telephone Cable Maintenance",Proc. of IJCAI'83, 1983

[War1] Warren,D.S.,et.al., "Executing Distributed PROLOG program on a Broadcast Network", Int. Symp. on Logic Programming, 1984

[Yam1]* Yamashita, M., et.al.,"DATA FLOW TYPE COMMUNICATION CONTROL ARCHITECTURE", Institute of Electronics and Communication Engineers of Japan EC-84, 1984

[Yok1] Yokoi,T. "A Perspective of the Japanese FGCS Project",ICOT Tech. Rep. TM-0026, 1983

[Zan1] Zaniolo,C., "OBJECT-ORIENTED PROGRAMMING IN PROLOG", Int. Symp. on Logic Programming, 1984

(Note) *: In Japanese