ICOT Technical Memorandum: TM-0617

TM-0617

Knowledge Base System in a Logic Programming Paradigm

by

H. Itoh, H. Monoi and H. Seki

November, 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

Knowledge Base System in a Logic Programming Paradigm

Hidenori ITOH, Hidetoshi MONOI and Hirohisa SEKI

Institute for New Generation Computer Technology
4-28, Mita 1-Chome, Minato-ku, Tokyo, 108 Japan

April 10th-12th, 1989

Version: Nov. 10, '88

File name: kenkyukai.tex

Abstract

This paper describes about the research and development status of the knowledge base system in logic programming and the parallel processing paradigms at the end of the intermediate stage. Our aim is to realize a prototype of the fifth generation computer system in the final stage. It will consist of a parallel inference subsystem and a knowledge base subsystem. In the final stage, these subsystems will be integrated into the prototype of the FGCS by using the parallel logic programming kernel language, Guarded Horn Clauses, or GHC defined in ICOT in 1985.

1. Introduction

From research and development of the relational database experimental system in the initial stage, we have developed technology for handling databases with large numbers of facts [Murakami 83][Itoh H 87a] and fact database retrieval technology from the user programs written in the logic programming language [Kunifuji 82][Yokota 86a]. These technologies are being used in the knowledge base subsystem on the sequential inference machine[Itoh F 88][Itoh H 88]. In the initial stage, we were waiting for the completion of the inference machines, PSI and CHI, for our research tools and environments. At the beginning of the intermediate stage, we decided that our knowledge base systems should be developed on these inference machines. We believed this approach would be the shortest path to our final goal. The inference mechanism is also one of the most important functions to realize the knowledge base system. So, in our new approach, the knowledge base system on the inference machine is more able to manage both complex logical data structures and simple relational data structures. It is also possible to apply richer logical functions to processing than can be applied to the relational database system on the general purpose processor.

Here, the technologies developed for the knowledge base model on the sequential inference machine are being used in the distributed knowledge base system. The technologies developed for the distributed knowledge base subsystem are being used in the parallel knowledge base subsystem on tightly coupled multi-processors.

At the start of this research, we looked for tightly coupled multi-processors. The multi-PSI machine was one good candidate for the parallel knowledge base experimental subsystem. To use this system for the parallel knowledge base subsystem in the same way as for the parallel inference subsystem, it would be easier to integrate the parallel inference subsystem and the parallel knowledge base subsystem in the final stage than to take another approach. But unfortunately, at the start, it was impossible to use the multi-PSI machine for the parallel knowledge base subsystem, and also the multi-PSI machine did not have shared memory architecture. We therefore decided to develop experimental tightly coupled multi-processors with large scale shared memory units. The tightly coupled multi-processors are general purpose micro-processors at present, but they are to be based on GHC [Ueda 85] and to be connected uniformly by the FGCS network [Goto 87][Itoh H 87b] in the final stage. Therefore, we now plan to develop GHC based VLSI processors for the prototype of the FGCS. We have also developed an application program for the feasibility check of this experimental system. It is a retrieval program for a large number of references stored in the experimental system [Shibayama 88]. We have evaluated the relationships between its parallel retrieval efficiency and the granularity of the knowledge base. We have also developed another application program for reference retrieval using the semantics of the references [Kogochi 88].

The last area is the research and development of the interface system written in GHC between the parallel knowledge base subsystem and the parallel inference subsystem [Yokota 89].

2. Knowledge base system on a sequential inference machine

The approach of building the knowledge base subsystem on the CHI machine is to extend a logic programming language that supports knowledge base functions. We have developed a practical knowledge base system with a large amount of knowledge in order to prove the effectiveness of the functions. The entire system was developed on the CHI machine with a high-performance sequential inference processor and a large capacity memory. The main memory capacity is sufficient for realizing a main memory based knowledge base.

The CHI hardware system configuration is shown in Fig. 1. The performance of the processor in the CHI is about 500 KLIPS (logical inferences per second), and the capacity of the main memory is 320 megabytes. The processor and main memory are connected to a front-end processor for input-output operations. The high performance comes from special hardware for unification, backtracking, clause indexing and sophisticated compiler optimization. The large memory capacity makes it possible to realize a memory based knowledge base system.

The CHI software system configuration [Konagaya 87] is shown in Fig. 2. The software system is composed of three layers: a kernel layer, a language processing layer and an application layer. The kernel layer provides basic functions for multiprocessing and remote input-output operations. The language processing layer provides an interactive programming environment for SUPLOG, a Prolog dialect with multiple name space. The application layer provides special inference rules and facts for special areas, such as DNA sequence matching and a machine translation system. Especially, we have developed the DNA sequence matching program for the application system. The program is one of the typical retrieval applications of collecting all the solutions from a large knowledge base. All processes share the knowledge base systems illustrated in the lower part of Fig. 2, and all processes execute logic programs within their own execution environment illustrated in the upper part of Fig. 2, such as local stacks, global stacks, trail stacks and local knowledge bases.

So, from the user's point of view, the CHI is like a domain-oriented knowledge base machine rather than a Prolog machine, when application layer programs are loaded on the system programs.

We introduced the multiple-multiple name space method to avoid name conflict between inter-processes. An example of multiple-multiple name spaces is shown in Fig. 3. This method gives an elegant way of sharing a knowledge base among inter-processes. To realize it, each process copies the name tables of the shared clause database, when created. The point is that the copied name tables are in the local clause base, so each process can change any name space without affecting other processes.

3. Distributed knowledge base system

The distributed knowledge base system is a kind of co-operative problem solving system in the distributed environment. It is located logically between the knowledge base system on the single processor and the parallel knowledge base system on the tightly coupled processors.

Logical configuration of the distributed knowledge base system which is called is shown in Fig. 4. It consists of the application program, the global knowledge base manager and a number of local knowledge base managers. The application program on this system was developed for the support system for programming.

Physical configuration of the PHI system is shown in Fig. 5. The PSI machines connected by a local area network are used both for the host systems and the knowledge base systems. Each site of the knowledge base has the same logical structure as the software shown in Fig. 5. One global manager and one or more local managers are dynamically assigned to each user or application program shown in Fig. 4. The basic functions in each site of the knowledge base system are based on the deductive database. Therefore, the knowledge base management layer manages an internal database, or IDB, which is a set of rules, and the database management layer manages the external database, or EDB, which is a set of facts. The PHI computes an answer to a query as a set of facts in the EDB using the rules in the IDB.

Recursive query processing in the deductive database and in the distributed environment is one of the key points of research and development. Distributed query processing is based on the communication cost minimization approach, which is the same approach as ordinal distributed database systems. Our recursive query processing is based on a bottom-up approach combined with query transformation procedures in order to reduce the calculation cost [Miyazaki 88]. We adopted a dynamic optimization method in these types of processing. Because a query is also a set of sentences in the logic program, we applied program transformation methods, for example, the partial evaluation method [Sakama 88] and the theorem proving methods, to query transformation procedures. They can be seen as a kind of knowledge compilation method applied to the queries, IDB and EDB for efficient retrieval.

The PHI allows negative literals in bodies of clauses. This extension introduces two difficulties in query processing. One is that we have to specify an appropriate class of databases at first, and then the semantics of negation has to be defined. The other is that we have to consider an efficient query evaluation method which is correct under the semantics.

For the first problem, we consider a class of the so-called stratified data bases which are partitioned into layers. The evaluation of the clause base are decided layer by layer from the bottom layer. Then a higher layer can use the negations decided in the lower layers. We define a new method for the second problem. The ordered linear resolution with tabulation method [Tamaki 86] has already been developed for the definite database, and we combined the idea of negation as failure with it [Seki 88], and applied it to query processing in the PHI system.

4. Parallel knowledge base system

In this system, we assume that knowledge is what is represented by the set of terms. Here, the term is the primitive element of clauses. It may include variables in itself and has a more complex logical structure than data. We have defined new unification based operations on the relational term base for the parallel knowledge base system. Typical new operations are unification-join and unification-restriction [Yokota 86b][Morita 86]. They can be seen as a combination of the unification operations and the relational algebraic operations which are normally used in the relational databases. Because all terms retrieved using this new method are unified with the terms in queries, we call this new method 'retrieval by unification', or RBU.

Unification based query language

A simple example of the unification based query language [Monoi 88b] is shown in Fig. 6. The 'retrieve' in it is a predicate to retrieve knowledge base. The meaning of it is illustrated by the AND-OR tree in the lower part of Fig. 6. The leaves in this tree are the retrieval conditions given in the query or the rules and facts in the knowledge base. Because both rules and facts are stored in the knowledge base in the same logical scheme, it is convenient to select the rules concerning the facts which are expected to be retrieved and to apply instantaneously them to their facts in the same place. It also means that inference processing can be performed even during knowledge base retrieval. It is especially convenient when the shallow inference rules are widely and iteratively applied to a large amount of knowledge. For example, when there are a many kinds of standard tactics in the system of a 'go' game or chess game, it is very convenient to evaluate all of them after moving only a few steps.

Here we consider a case where the knowledge base is very large. For processing efficiency, it is also necessary to process in parallel.

Compilation and parallel execution

Fig. 7 shows the flow of query compilation and its parallel execution processing. When one of the idle processing elements is determined dynamically by the other processing elements to be a receiver of the queries, this processing element then compiles the queries into RBU operations and decomposes them into many pieces which are units of parallel processing. They are then delivered to the other processing elements. The query language between the knowledge base and user programs is defined according to these ideas [Monoi 88].

Hardware configuration

For the parallel execution of a large number of their decomposed pieces, each of which is very large, we have developed the experimental system shown in Fig. 8. It is composed of eight processing elements, and eight multi-port page-memory units [Tanaka 84a][Tanaka 84b]. Each processing element is connected with a page-memory unit by a port through a switching network. Because a logical space lies on all the page-memory units, each processing element can access same logical space at the same time in page-memory units through its own port in parallel [Monoi 88a][Sakai 88].

Fib. 9 is the parallel processing timing diagram. The processing time is composed of compiling, decomposing, delivering and processing time. If the knowledge base is very large, the whole processing time using all the processing elements in parallel is reduced to about one eighth of the whole time using only one processing element. The system's memory stores many references for different papers.

5. Interface system between GHC and the parallel knowledge base system

The features of the GHC are the don't-care nondeterministic and-parallel processing, and the data stream-type input-output processing. When a process in the don't care nondeterministic and-parallel method finds a resolution, then it stops to find the other resolutions. It is inconsistent with the all solution collecting mechanism which is required in the knowledge base system. To avoid this problem, the special predicates which are the processes in GHC are provided. They continuously receive all the resolutions from the knowledge base system in a stream which is the second feature of GHC. We have developed the special predicate processing system and have embedded it into GHC. This interface system will be applied in the parallel knowledge base system which will be enhanced to the knowledge base clusters in the prototype of the FGCS. At present, for the evaluation of it we have developed the better-first search parallel production system on it (see Fig. 10). The detail of it had been descrived in [Yokota 89].

6. Conclusion

This paper have described at the current status of research and development of the knowledge base subsystem. In the intermediate stage, we investigated and performed experiments on four knowledge base mechanisms and the application programs on the parallel knowledge base system required for constructing the prototype of the FGCS.

Here, the knowledge base systems are summarized as follows.

(1) The knowledge base system on the CHI machine provides a very high performance knowledge retrieval mechanism, a practical memory based knowledge base, and a hierarchical clause base for a multi-process environment. In this system, multiple-multiple name spaces play an essential role in hierarchical knowledge representation and avoiding inter-process name conflict.

(2) We have developed a distributed knowledge base system based on deductive databases. Recursive query processing is based on the calculation cost minimization approach, which comes from the logic programming transfer methods, because the query, IDB and EDB are defined by the sets of sentences in the logic programming language.

(3) In parallel knowledge base research, the experimental hardware and knowledge base management software have been developed. An application program for a feasibility check of the system has also been developed. The system has been proven to manipulate sets of terms efficiently in parallel. The hardware configuration proved useful for large knowledge bases.

(4) We propose to introduce a special predicate processer for the interface between the GHC processes and the knowledge base processes. It receives all the resolutions from the knowledge base processes in stream. We have developed a parallel application program to check the feasibility of the interface system.

In conclusion, the first and second research areas has been finished at the end of the intermediate stage. The relational database system on the general purpose machine developed in the initial stage, the knowledge base system on the CHI machine and the distributed knowledge base system on the PSI machines developed in the intermediate stage are milestones towards our final goal. The parallel knowledge base subsystem, the interface system between GHC and the parallel knowledge base subsystem, and the semantics retrieval application program will be enhanced and combined to realize clusters, which are one of the components of the prototype of the Fifth Generation Computer System in the final stage. The various technologies developed in the intermediate stage will be incorporated in the FGCS.

Acknowledgement

We would like to express our gratitude to the other members of the third laboratory of the ICOT Research Center. Each system described in this paper has been developed with the close co-operation of manufacturers. Thanks goes also to the manufacturers' people who were engaged in the implementations. We are inbedted to the members of the KBM Working Group for their fruitful discussions.

References

[Chikayama 84] Chikayama, T., "Unique Features of ESP", in *Proc. Int. Conf. Fifth Generation Computer Systems*, pp.292-298, 1984

[Goto 87] Goto, A., "Parallel Inference Machine Research in FGCS Project", in *Proc. of the US-Japan AI Symposium 87*, pp. 21-36, 1987

[Habata 87] Habata, S., Nakazaki, R., Konagaya, A., Atarashi, A. and Umemura, M., "Co-operative High Performance Sequential Inference Machine: CHI", in *Proc. ICCD'87*, New York, 1987

[Itoh F 88] Itoh, F., Shimakawa, K., Togo, K., Matsuda, S., Itoh. H. and Oba, M., "Design, Implementation, and Evaluation of a Relational Database Engine for Variable Length Records", in *Database Machines and Knowledge Base Machines*, Kluwer Academic Publishers, pp.269-282, 1988

[Itoh H 87a] Itoh, H., Abe, M., Sakama, C. and Mitomo, Y., "Parallel Control Techniques for Dedicated Relational Database Engines", in *Proc. 3rd Int. Conf. Data Engineering*, pp.208-215, 1987

[Itoh H 87b] Itoh, H. and Uchida, S., "Parallel Inference Machine and Knowledge Base Machine", in *Proc. TEMCON 87-IEEE Region 10 Conference*, pp.841-845, 1987

[Itoh H 88] Itoh, H., Takewaki, T. and Yokota, H., "Knowledge Base Machine Based in Parallel Kernel Language", in eds. Kitsuregawa and Tanaka, in *Database Machines and Knowledge Base Machines*, Kluwer Academic Publishers, 1988

[Kakuta 85] Kakuta, T., Miyazaki, N., Shibayama, S., Yokota, H. and Murakami, K., "The Design and Implementation of Relational Database Machine Delta", in *Proc. Int. Workshop on Database machines '85*, 1985

[Konagaya 87] Konagaya, A., Nakazaki, R. and Umemura, M., "A Co-operative Programming Environment for a Back-end Type Sequential Inference Machine CHI", in *Proc. Int. Workshop on Parallel Algorithms and Architectures*, East Germany, pp.25-30, 1987

[Kunifuji 82] Kunifuji, S. and Yokota, H., "Prolog and Relational Database for Fifth Generation Computer Systems", in *Proc. Workshop on Logical Bases for Data Bases*, Gallaire, et al.(eds.), ONERA-CERT, 1982

[Miyazaki 88] Miyazaki, N., Haniuda, H., Yokota, K. and Itoh, H., "Query Transformations in Deductive Databases", ICOT-TR 377, 1988

[Monoi 88a] Monoi, H., Morita, Y., Itoh, H., Sakai, H. and Shibayama, S., "Parallel Control Technique and Performance of an MPPM Knowledge Base Machine Architecture", in *Proc. 4th Int. Conf. Data Engineering*, pp.210-217, 1988

[Monoi 88b] Monoi, H., Morita, Y., Itoh, H., Takewaki, T., Sakai, H. and Shibayama, S., "Unification-Based Query Language for Relational Knowledge Bases and its Parallel Execution", in *Proc. Int. Conf. Fifth Generation Computer Systems*, 1988

[Morita 86] Morita. Y., Yokota, H., Nishida, K. and Itoh, H., "Retrieval-By-Unification Operation on a Relational Knowledge Base", in *Proc. of 12th Int. Conf. on Very Large Databases*, pp. 52-59, 1986

[Murakami 83] Murakami, K., Kakuta, T., Miyazaki, N., Shibayama, S. and Yokota, H., "Relational Database Machine: First Step to a Knowledge Base Machine", in *Proc. 10th int. symp. Computer Architecture*, pp.423-426, 1983

[Sakai 88] Sakai, H., Shibayama, S., Monoi, H., Morita, Y. and Itoh, H., "A Simulation Study of a Knowledge Base Machine Architecture", in *Database Machines and Knowledge Base Machines*, Kluwer Academic Publishers, pp.585-598, 1988

[Sakama 87] Sakama, C. and Itoh, H., "Partial Evaluation of Queries in Deductive Databases", Workshop on Partial Evaluation and Mixed Computation, 1987

[Seki 88] Seki, H. and Itoh, H., "A Query Evaluation Method for Stratified Programs under the Extended CWA", ICOT Technical Report TR-337, 1988, also in *Proc. 5th Int. Conf. and Symp. Logic Programming*

[Shibayama 87] Shibayama, S., Sakai, H., Monoi, H., Morita, Y. and Itoh, H., "*Mu-X*: An Experimental Knowledge Base Machine with Unification-Based Retrieval Capability", in *Proc. France-Japan Artificial Intelligence and Computer Science Symposium 87*, pp.343-357, 1987

[Takewaki 86] Takewaki, T. and Itoh, H., "Parallel Control Techniques for Retrieval Processes in the Parallel Logic Programming Language and their Evaluation", ICOT TR-255, 1987

[Tamaki 86] Tamaki, H. and Sato, T., "OLD Resolution with Tabulation", in *Proc. of 3rd ICLP*, 1986

[Tanaka 84a] Tanaka, Y., "MPDC: Massive Parallel Architecture for Very Large Databases", in *Proc. Int. Conf. Fifth Generation Computer Systems*, pp.113-137, 1984

[Tanaka 84b] Tanaka, Y., "A Multiport Page-Memory Architecture and A Multiport Disk-Cache System", in *New Generation Computing*, OHMSHA, 2, pp.241-260, 1984

[Ueda 85] Ueda, K., "Guarded Horn Clauses", in *Logic Programming '85*, E. Wada (ed)., Lecture Notes in Computer Science 221, Springer-Verlag, 1986

[Yokota 84] Yokota, H., Kunifuji, S., Kakuta, T., Miyazaki, N., Shibayama, S. and Murakami, K., "An Enhanced Inference Mechanism for Generating Relational Algebra Queries", in *Proc. 3rd ACM SIGACT-SIGMOD Symp. Principles of Database Systems*, pp.229-238, 1984

[Yokota 86a] Yokota, H., Sakai, K. and Itoh, H., "Deductive Database System Based on Unit Resolution", in *Proc. 2nd Int. Conf. Data Engineering*, pp.228-235, 1986

[Yokota 86b] Yokota, H. and Itoh, H., "A Model and an Architecture for a Relational Knowledge Base", in *Proc. 13th Int. Symp. Computer Architecture*, pp.2-9, 1986

[Yokota 89] Yokota, H. et.al, " a Relational Knowledge Base", to be appeared in *Proc. 5th Int. Conf. Data Engineering*, 1989

CHI Hardware System Configuration

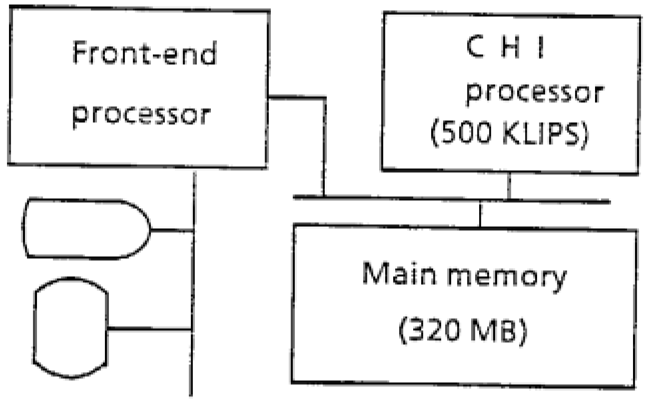

Fig. 1

CHI Software System Configuration

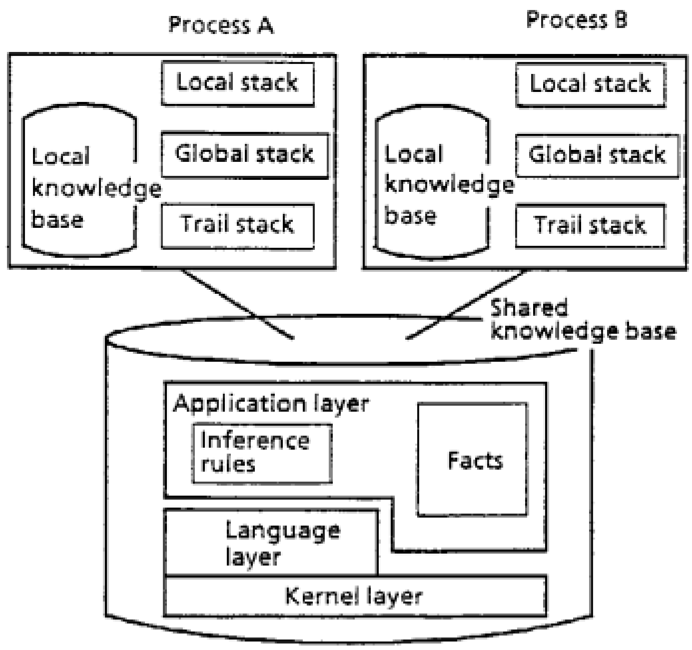

Fig. 2

Multiple-multiple Name Spaces

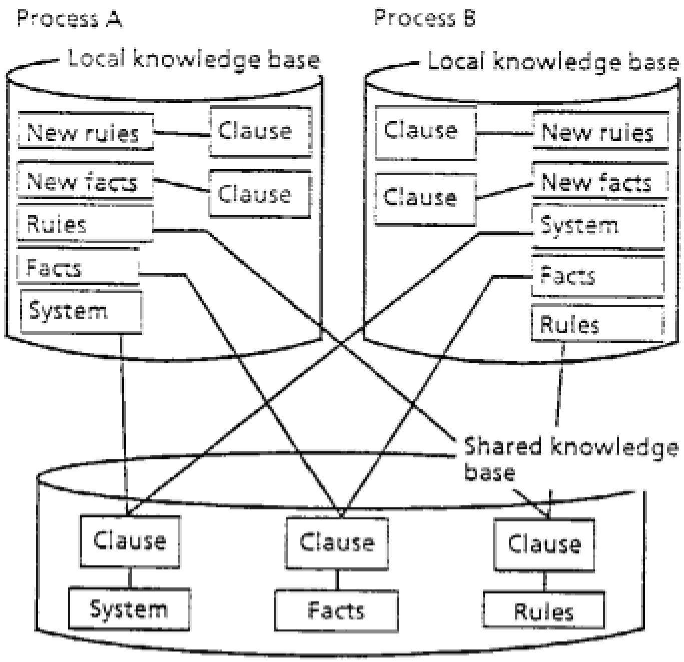

Fig 3

Logical Configuration of the PHI System

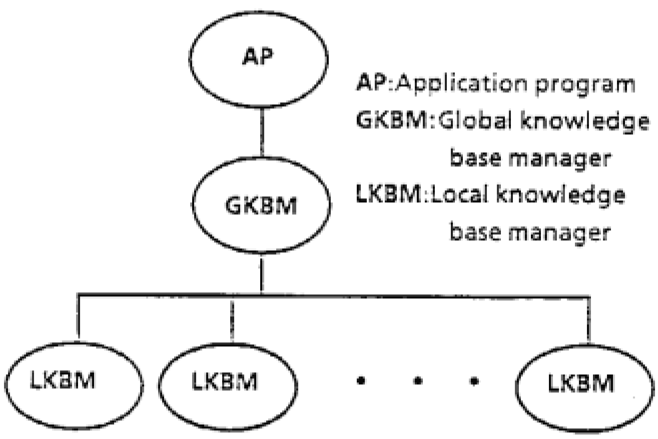

Fig 4

Physical Configuration of the PHI System

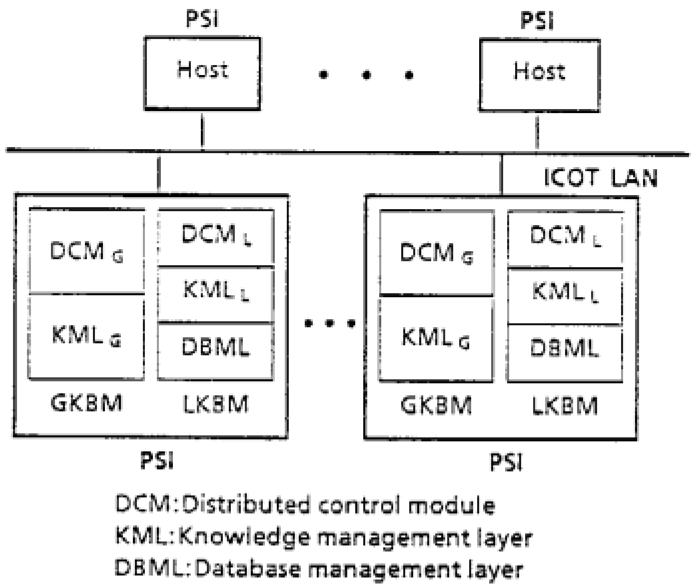

Fig 5

Retrieval Predicate and Its AND-OR Tree

retrieve (a(X,Y),((b(X,Z),(c(Y,Z),y = f(__));d(X,Y,a)))

';' means OR(∨).

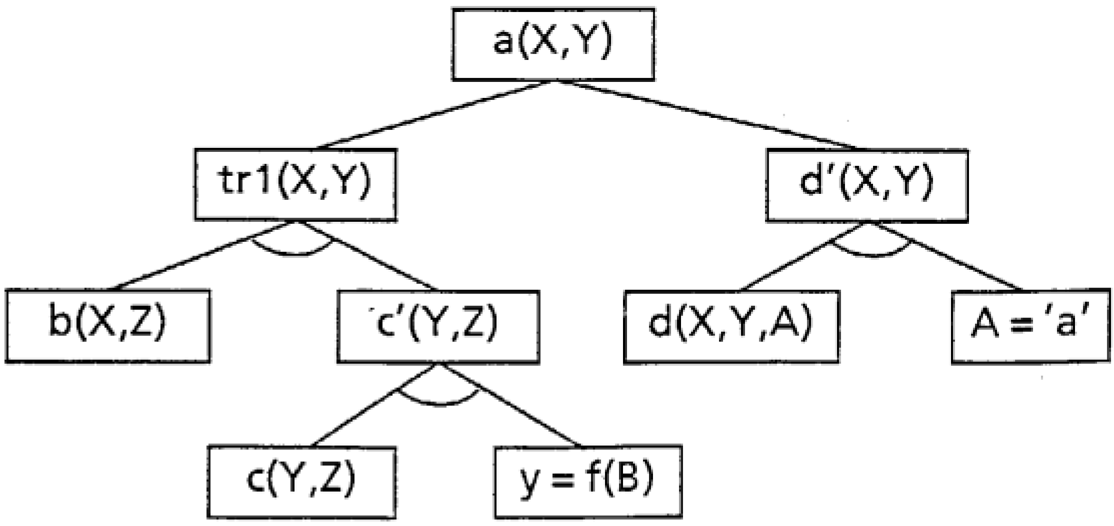

Fig 6

Compilation and Parallel Execution

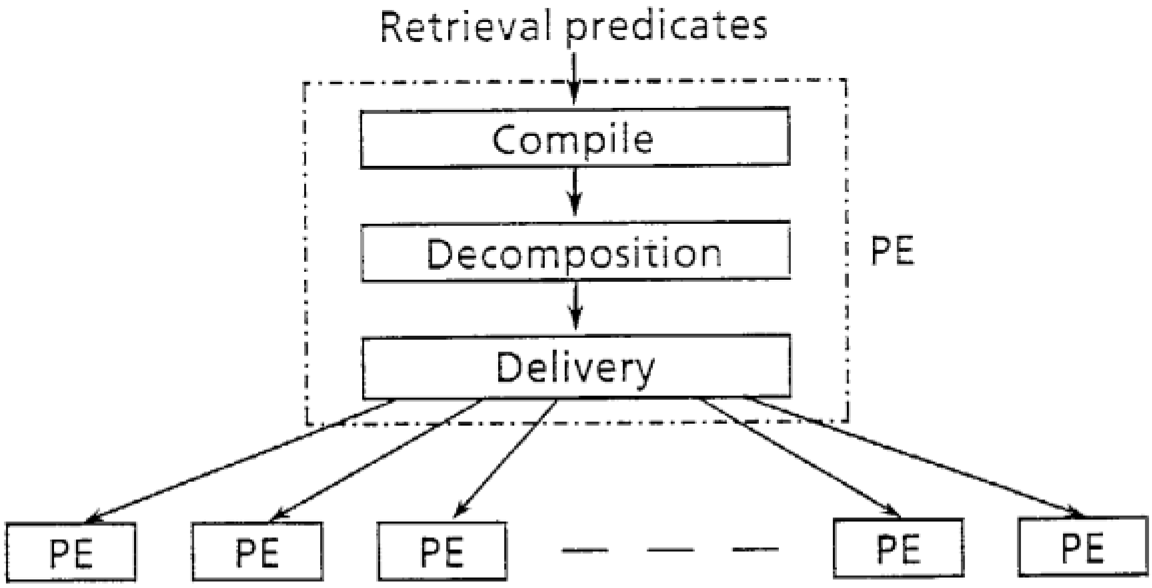

Fig 7

Hardware Configuration
of the Parallel Knowledge Base System

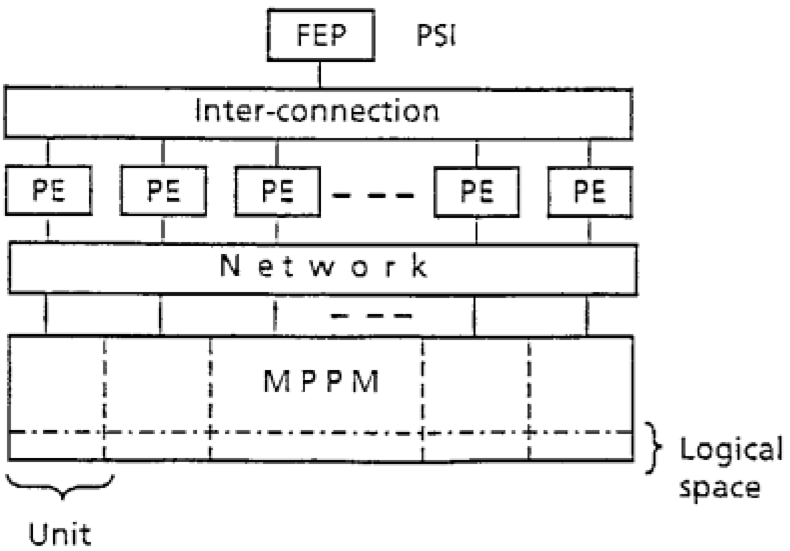

Fig 8

Parallel Processing Timing Diagram

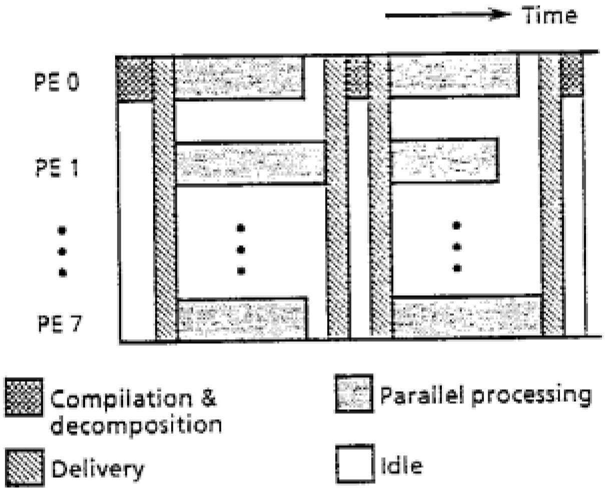

Fig. 9

GHC Interface System

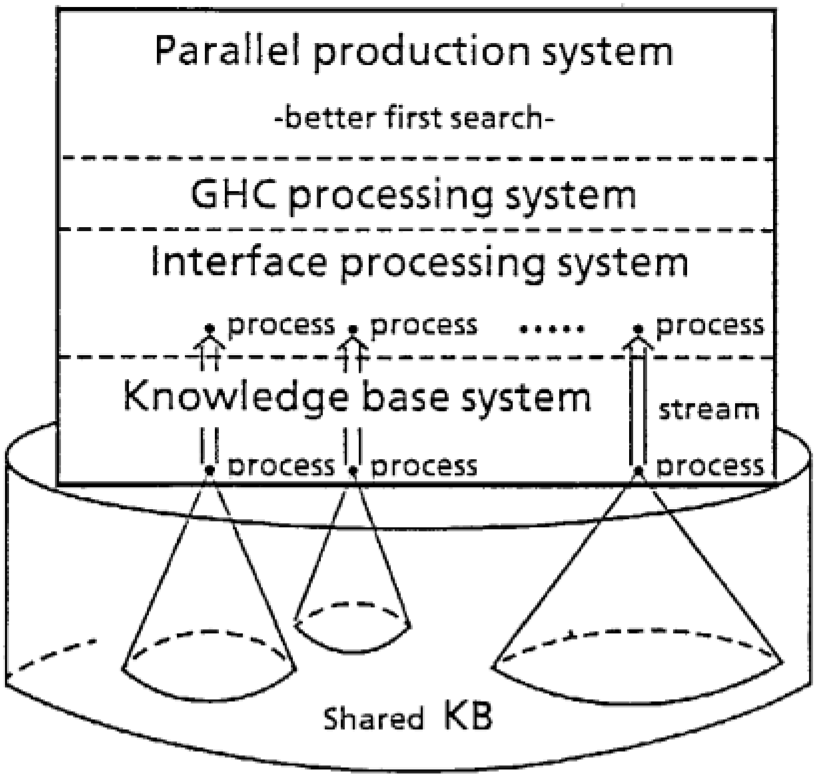

Fig. 10