TR-0854

Knowledge Information Processing by Highly Parallel Processing

by

S. Uchida, T. Chikayama & K. Nitta

ICOT

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

(03)3456-3191~5

Institute for New Generation Computer Technology

Knowledge Information Processing by Highly Parallel Processing

Shunichi Uchida, Takashi Chikayama and Katsumi Nitta

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

Abstract

This paper intends to introduce the FGCS Project which was conducted for eleven years and the FGCS Follow-on Project which was started in April 1993 as two year project for dissemination of the FGCS technology.

The FGCS project aimed at development of a revolutionary new computer technology combining highly parallel processing and knowledge processing technologies using a parallel logic language KL1 as the kernel language of the new computer technology which is called the FGCS technology.

In the end of the FGCS project, initial goals of the project were successfully achieved and a prototype system of the FGCS was built to demonstrate the establishment of the FGCS technology. The center of the prototype system is highly parallel hardware which executes the KL1 very efficiently and a parallel operating system, PIMOS.

The parallel hardware consists of five models of parallel inference machines (PIMs) having about 1000 element processors in total. The PIMOS is fully written in KL1 and has an efficient parallel programming environment for the KL1.

Using a KL1 and PIMOS environment on the PIMs, many parallel software systems have been developed. They include knowledge processing tools and parallel application systems. Several of them such as a theorem prover, MGTP, have proved that the KL1 and PIMOS environment enabled us to gain linear speed-up which is almost proportional to the number of element processors.

This was probably the first time in the world that important knowledge processing or AI application systems could gain almost linear speed-up using highly parallel processing. Then, the KL1 and PIMOS environment running on the PIMs is now the most powerful parallel processing system for knowledge processing.

Parallel processing of this kind is classified as parallel symbol processing and much wider applicability to not only knowledge processing applications but also more general problems than conventional parallel processing technology.

Thus, it was very desirable that the KL1 and PIMOS environment could be operational on commercially available machines so that many people could use them as a new common infrastructure for advanced research into computer science and technology.

For this reason, Ministry of International Trade and Industry (MITI) decided to place major software developed in the FGCS project in the public domain as ICOT Free Software. Furthermore, MITI and ICOT prepared a new project called the FGCS Follow-on Project.

A general goal of the Follow-on project is a fusion of the FGCS technology and market software and hardware technologies to disseminate the FGCS technology to many researchers in the world.

1 Introduction

Recent progress in parallel and distributed processing using RISC chips and Unix-based operating systems has dramatically changed main streams of market computer technologies.

In these two years, various MIMD parallel machines have appeared in the market. Many of them are intending to be more cost effective number crunchers than conventional vector-type SIMD super computers. More recently, some of them have appeared as super servers and intend to be new central computer systems in distributed systems. They will be used for general management purposes such as database management and administrative decision making instead of main frame computers. Currently, most of these parallel machines include dozens to one hundred element processors. However, some of them are announced that they could be extended up to thousands of element processors.

This dramatic change can be considered as an opening of a new era of computers where parallel and distributed systems shall take over conventional mainframe systems. It is considered to be caused by rapid advance in hardware technology, not by software technology.

Then, most of the parallel machines have to be programmed by conventional languages and controlled by

Unix-based operating systems with some extensions for inter-processor communication supports. Thus, a new parallel programming environment and parallel software technologies are demanded to be able to develop large-scale parallel and distributed software systems efficiently.

The FGCS technology combining parallel processing and knowledge processing technologies is now considered to be most promising technology that can fulfill this demand. From the very beginning, the parallel processing technology contained in the FGCS technology was a born large-scale parallel technology which was intended to cover various knowledge processing applications. It has much wider applicability to more general problems than existing parallel processing technologies.

When we made a plan of the FGCS project about 12 years ago, we anticipated that an era of highly parallel computers in the market would come in about 5 years after the completion of the FGCS project. It means that we anticipated that it would come in 1997 at latest.

The reality seems to have come a little earlier than this anticipation, however, this has made us to possible to continue our effort to further develop the FGCS technology so that we can disseminate it more effectively.

With the completion of the FGCS project, Ministry of International Trade and Industry (MITI) decided to carry out a new project which is called the FGCS Follow-on project. In the Follow-on project, the KL1 and PIMOS environment as well as major software developed in the FGCS project will be ported and operational on MIMD-type parallel machines [1] which are recently appearing in the market.

With this project, many new and interesting software systems developed in the FGCS project will be used by many people in the world, and they will be further developed in many ways in many places in the world.

In this paper, major achievements of the FGCS project and research activities to be carried out in the Follow-on project will be described.

2 Major achievements of the FGCS Project

The FGCS project was started in April 1982 as a Japanese national project. This project was unique among other national projects because it aimed at contribution to the advance of global computer science and technology through the development of revolutionary computer technology which was far advanced from market technologies of those days. ICOT was established as a central research institute to carry out this project.

In this project, the fifth generation computer was defined that it would have an inference mechanism using knowledge bases for its kernel function, and would fully use highly parallel processing technology for its implementation as shown in Fig.-1.

After the eleven-year research and development effort, the FGCS project achieved its initial goals and established the FGCS technology. To attain the goals, many new ideas, theories, small to large software and hardware technologies were created, evaluated, improved and extended. Finally, they were consistently integrated into an FGCS prototype system as shown in Fig.-1 and Fig.-2. It is probably the world's fastest and largest-scale computer for knowledge information processing which is actually being used for practical applications.

To discuss many element technologies contained in the prototype system from macroscopic scientific viewpoint, we roughly divide them into two categories: one is technologies related to parallel symbol processing and the other is parallel knowledge processing.

2.1 Major achievements for parallel symbol processing

The lower two layers of the prototype system in Fig.-2 can be regarded as a highly parallel symbol processing system. These two layers were built as a more complete and practical system than the upper two layers. This system is currently being used for further development of parallel knowledge processing tools and parallel symbol and knowledge processing applications.

The lowest layer of the prototype system in Fig.-2 is five models of PIMs and a KL1 language processor. The largest model is the PIM model p which has 512 element processors. The next model is the PIM model m having 256 element processors. Their inter processor networks are difference each other as shown in Fig.-2. Currently, PIM model p is most frequently used and attained 150M LIPS. [2]

The KL1 is a parallel logic language and provides us with an automatic memory management function like Prolog and LISP. In addition to this, it provides us with an automatic process synchronization function based on a dataflow model and can exploit very fine grain parallelism. The PIMs have dedicated hardware mechanisms to support efficient execution of the KL1.

The second layer is the basic software. The PIMOS is a main system of this software and has functions for hardware and software resource management and execution control which were designed in principle to be scalable up to one million element processors. It also provides a comfortable parallel programming environment for the KL1 including new visual performance debugging tools.

Using this KL1 and PIMOS environment on the PIMs, many parallel software systems have been developed to built the basic software including PIMOS itself and many other software systems included in the upper two layers.

[1] As KL1 programs are compiled into C programs, they will run even on a small personal computer in principle.

[2] LIPS stands for Logical Inference Per Second. One LIPS corresponds to 50 to 100 IPS.

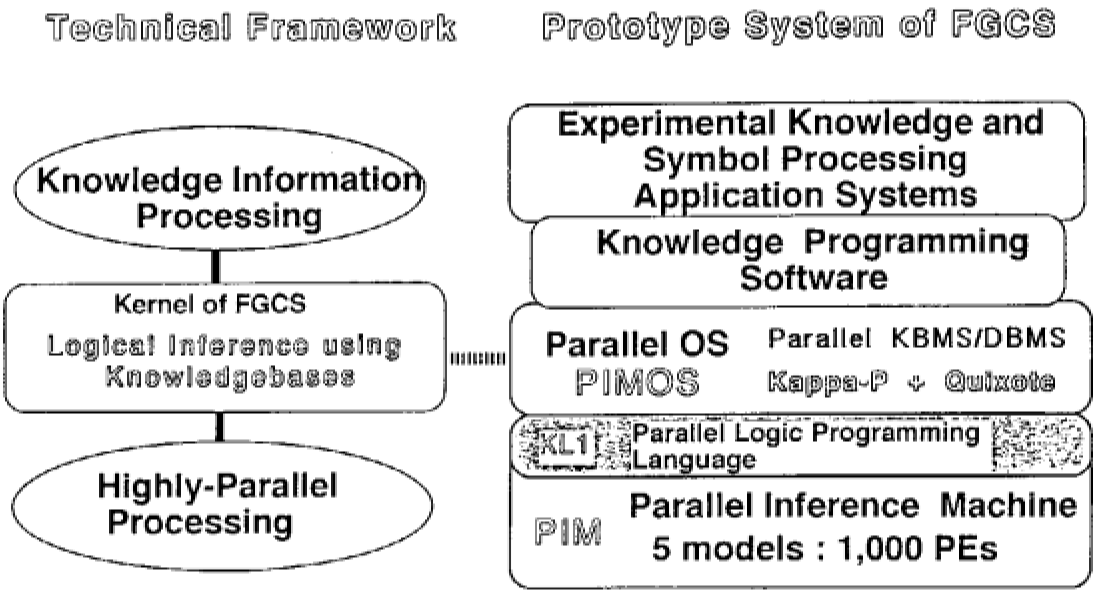

Figure 1: Technical Framework and Prototype System

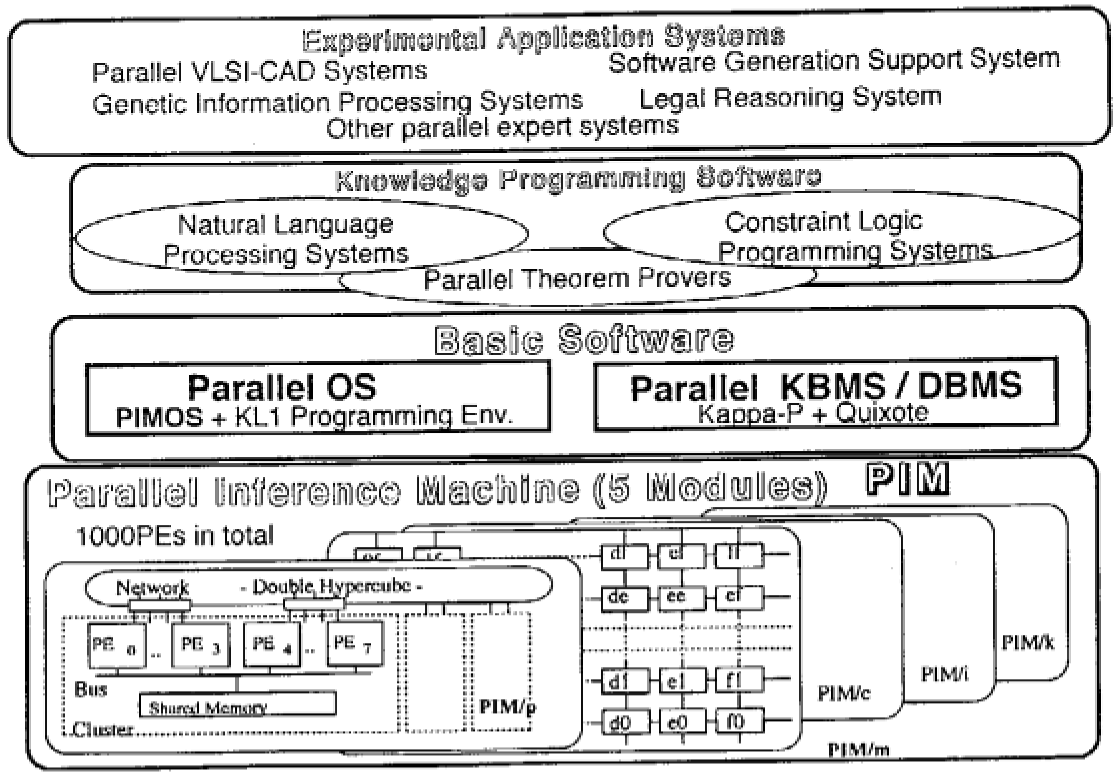

Figure 2: Organization of Prototype System

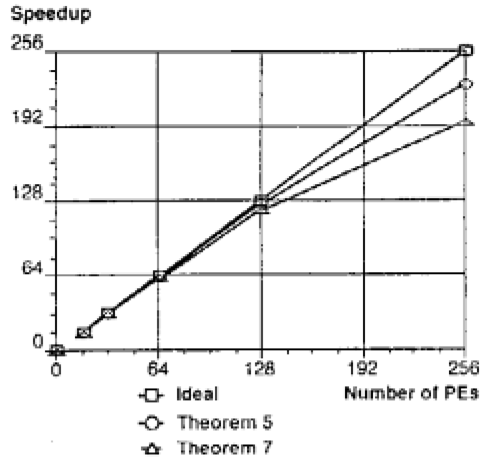

Figure 3: Speed-up ratio for the theorem prover MGTP/N on PIM/m

Through this parallel software development, this environment was proved to be very efficient and productive. For example, the first version of PIMOS which is an operating system for real use parallel processing systems was completed in half an year. In the development of logic simulator, a part of parallel LSI-CAD experimental system, took only 3 man-month as though this system took complex algorithm suited for parallel processing, called virtual time method. This term is less than 1/10 of the term in case of parallel software development using usual technologies.

As performance tunings of the parallel software systems were hurriedly proceeded for demonstrations at the FGCS'92 conference, we could get surprising results that some application systems succeeded in gaining the speed-up which was almost proportional to the number of element processors.

For example, a parallel theorem prover, MGTP attained about more than 200 times speed-up using the PIM of 256 element processors as shown in Fig.-3

The MGTP has solved several open problems in some theories on quasigroup and is probably the world's fastest theorem prover now.

Generally, many of the parallel software systems developed for knowledge processing or AI applications were evaluated to have gained 20 to 100 times speed-up on the PIM compared to similar systems running on usual workstations or mainframe computers.

This was probably the first time in the world that important knowledge processing or AI application systems could gain almost linear speed-up using highly parallel processing. With this evaluation, we could concluded that parallel symbol processing technology we developed are very promising to accelerate the new era of parallel computers.

2.2 Major achievements of knowledge processing technology

One general guideline of knowledge processing research in the FGCS project was to use "mathematical logic" as the base of knowledge representation and knowledge base management.

In the second layer, a parallel DBMS, Kappa-P and a KBMS, *QUIXOTE* [3] are included. They are less complete systems than the KL1 or PIMOS. They are experimental systems to evaluate new concepts and methods.

The Kappa-p is a parallel DBMS. Its unique feature is the employment of the nested-relational model which enable us to use much more flexible internal date structures than the regular relational model. Thus, it is suitable for the management of complex databases such as natural language dictionaries and various biological date bases. The Kappa-P was intended to be used as an engine for a knowledge base management system which would be built on the Kappa-P.

For knowledge representation, two languages have been developed and used for some applications. One is a deductive and object-oriented language called *QUIXOTE*. The other is a constraint logic programming language GDCC.

In the FGCS Project, knowledge representation languages based on first-order logic have been promoted. The *QUIXOTE* is the final version of logic based languages developed in the FGCS project. It has object-oriented features to provide us with flexibility in making various module structures. The qxt is now being used to describe rules and cases for legal reasoning system and biological knowledge analysis in for the sequence alignment system.

In the knowledge processing research, the theorem prover, MGTP is considered as a high-level inference engine which can execute expressions of first order logic as a knowledge programming language. Thus, the 200 times speed-up of the MGTP has greatly enhanced our confidence that our guideline and research direction are promising.

Among 20 parallel application systems in the fourth layer, a legal reasoning system, HELIC-II is one of the most interesting applications. The HELIC-II has two types of high-level inference engines as show in Fig-4; one is a rule-based inference engine. The other is a case-based inference engine. The rule based inference engine is a simple extension of the MGTP. Then, the HELIC-II is considered as an application of the theorem prover.

In a current version of the HELIC-II, some penal codes are described in a logic based knowledge description lan-

[3] *QUIXOTE* is the name of the KBMS and also the name of the knowledge representation language embedded in the KBMS.

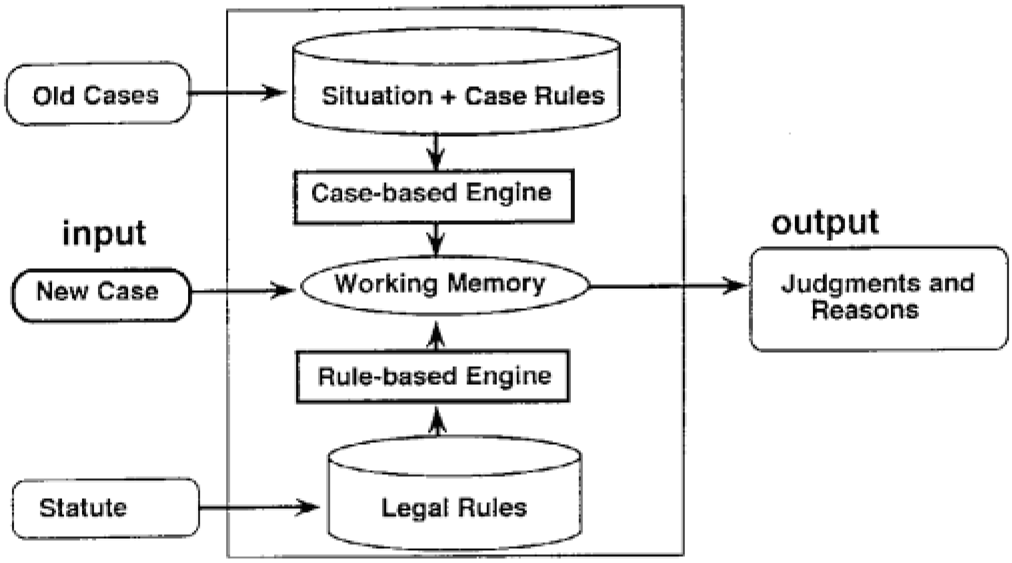

Figure 4: Organization of the legal reasoning system, HELIC-II

guage and stored in the legal rule database. In addition to this, about one hundred old cases on past criminal events are described and stored in the case rule database. Using the mechanism shown in Fig.-4, if the HELIC-II is given a new criminal event, it looks up these databases and find out similar old cases and answers back a list of possible judgments.

To produce the answer, the HELIC-II has to make various sophisticated search operations in its databases based on "similarity matching". In the search operations, effective parallel processing is indispensable to shorten elapsed time of the search operations. Without the powerful parallel symbolic processing environment, the HELIC-II must not have been implemented so successfully.

Through various knowledge programming experiences, we have become confident that deductive languages with some extensions will be very feasible for practical knowledge programming purposes.

Current knowledge programming languages and environments are usually provided as expert systems' shells. Usually, they are too much dedicated to some specific applications or specific conventional languages. Then, it is very difficult to efficiently apply parallel symbolic processing because they are not based on any well-defined computational model.

Deductive languages and their extensions will be able to overcome this difficulty. As shown in Fig.-5, we can predict that they will open a next new era of knowledge processing computers and provide us with practical knowledge programming tools and environments which can be applicable to social, cultural, and cognitive scientific problems in the next century.

3 The FGCS Follow-on Project

As the end of the project period approached, it became obvious that the project would produce many valuable scientific and technological results. As the project targeted precompetitive technological goals, many of the results seemed to be too advanced or immature for industries to commercialize them immediately.

This was the first time that a MITI sponsored project produced such important scientific research results which MITI must consider how to deal with seriously. MITI asked its advisably committee to review achievements of the project. The committee gathered and examined both domestic and international opinions by sending questionnaires to hundreds of people involved with new computer technologies.

After nearly half a year discussion, the committee decided to advise MITI to carry out a two year follow-on project. Based on this advice, MITI and ICOT formulated a plan for the follow-on project which started from April 1993.

This project aims to make the major software systems and basic research results, developed in the FGCS project, widely available to the research community and industry as new tools or a new infrastructure for developing advanced computer technologies centered on parallel knowledge processing.

The goals of the follow-on project are as follows;

1. Porting of the KL1, PIMOS and major software sys-

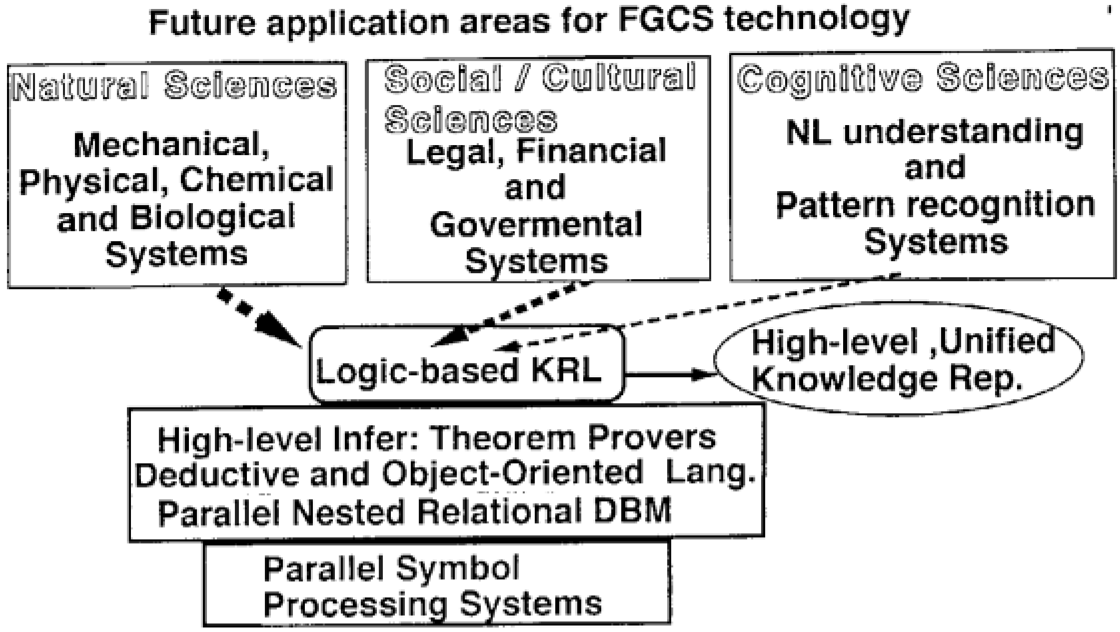

Figure 5: Prospect of knowledge processing based on the FGCS technology

tems to widely available Unix-based sequential and MIMD parallel systems

2. Further research and development of knowledge processing software and parallel application software using the PIMs and software systems written in KL1.

In the FGCS'92 conference held in June 1993, MITI and ICOT announced that major software developed in the FGCS project should be placed in the public domain as ICOT Free Software (IFS). [4] Since then, 77 programs including the PIMOS and most of new application systems have been released. Till now, we have had more than 8000 time accesses to the IFS by anonymous FTP through the Internet.

Achievements of the Follow-on project will also be placed in the public domain as a part of the IFS. This will complete another goal of the FGCS project that the project should contribute to the global computer science.

3.1 Porting the KL1 and PIMOS environment to stock hardware and OS

As a target system and language for the porting, the Unix operating system and the language C were chosen because of their incomparable popularity in the computer research world.

The language C is, unfortunately, not ideal for KL1 implementation. However, as our principal objective of this implementation is to provide a portable system, we attached greater importance to ease in porting to a wide range of systems than to absolute performance. Unix is not ideal either for parallel language implementations, as its standard inter-process communication mechanisms are rather weak. More attractive extensions have been proposed recently that may enable more efficient implementation but again, we make more of portability and will stick basically to a very standard subset of Unix features. Although we place priority on portability over all else, the system would not become useful without attractive efficiency.

Considering these conditions, we started development of the KLIC system. It is regarded as a tool or an environment to create a KL1 and PIMOS environment on Unix-based machines. The KLIC system includes a compiler which compile KL1 programs into C programs as shown in Fig.-6.

The KL1 language specification will remain basically the same. Some of the meta-level programming features, such as the *shoen* construct, may be altered slightly to work well with the Unix system.

Currently, the KLIC system plan to cover three different kinds of Unix systems as the hardware platform of the system.

Sequential Systems: Sequential Unix systems, such as low-price personal workstations, would be effectively used in the early software development phases.

Multi-Processor Systems: Unix-based multi-processor systems would provide high performance. An implementation on systems with

[4] For the information on IFS, please mail to ifs@icot.or.jp.

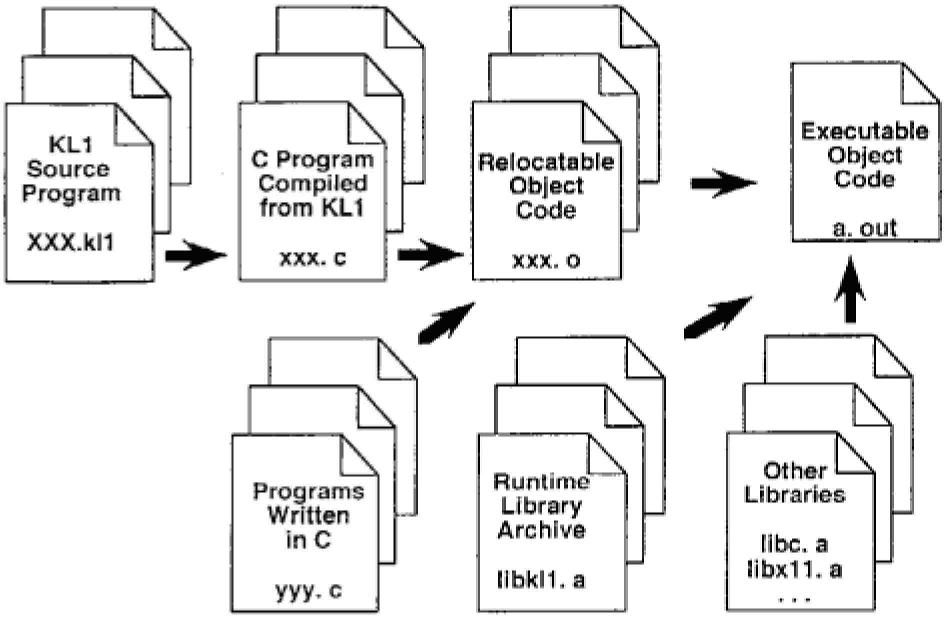

Figure 6: Compilation and execution of KL1 programs using the KLIC system

enough processors will outperform PIM systems in the future.

Network-Connected Systems: Sequential Unix systems connected with standard ethernet would provide reasonable performance without too much hardware cost. Many multi-processor systems connected together with high-performance networks would provide maximum performance.

Our development and release schedule is roughly as follows.

- The first sequential system for application users is expected in October '93. This version will provide a reasonable software development environment for debugging and performance analyses.
- The release of the first parallel system is planned in the second quarter of '94.

Various improvements are planned after these releases. For higher execution efficiency, enhanced compilation-time analysis is planned. For a better software development environment, various utilities for debugging and performance analysis will be ported from the PIMOS programming environment. Automatic load distribution libraries will also be ported.

3.2 Further development of knowledge processing technology

Another important task of the follow-on project is to further develop knowledge processing technologies based on the mathematical logic to be able to provide a more practical knowledge programming environment making full use of the power of parallel symbol processing provided by the PIMs.

We plan to conduct research themes as follows;

- Knowledge representation languages:
  1. A knowledge representation language, *QUIXOTE*
  2. A parallel constraint programming language
  3. Theorem provers, MGTP
- Parallel knowledge processing applications:
  1. Genetic information processing systems
  2. A legal reasoning system

Research activities of these themes will be managed to have a common general goal which is the collection of knowledge programming experiences and the development of important element technologies for knowledge programming.

As we are extending our application area for natural scientific area to other areas as shown in Fig.-5. We have recognized the fact that even if we could succeed in the

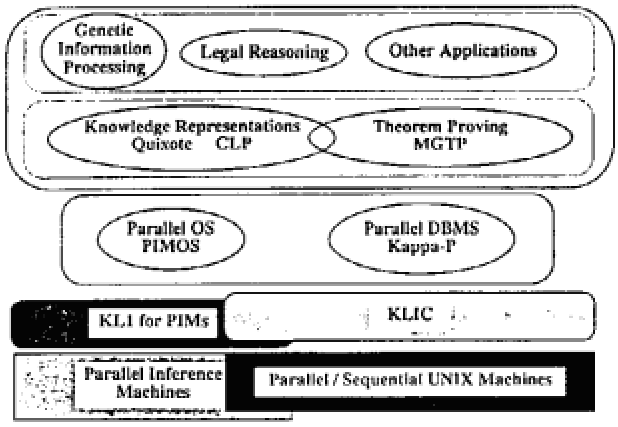

Figure 7: Research themes in the Follow-on project

development of a powerful deductive and object oriented knowledge programming language like $\mathcal{QUIXOTE}$ and use it, we should have a semantic gap problem between this language and various knowledge fragments we want to program.

For example, legal rules or systems are usually believed very logical and consistent from a viewpoint of our daily life. However, they are not complete nor well ordered from a knowledge programming viewpoint. Thus, implicit background knowledge has to be programmed.

To fill this gap, we should need to create new schemes, new theories, new methodologies with which we can restructure or reorganize these knowledge fragments and related background knowledge so that we can program them. In the next century, this problem shall be a main research theme for knowledge processing and the FGCS technology will be an indispensable tool to carry out this research.

4 Conclusion

As described above, achievements of the FGCS project can be considered to be classified into two classes, namely, parallel symbol processing technology and knowledge processing technology.

As far as the achievements classified into parallel symbol processing are concerned, the KLIC system will disseminate them very effectively. It is very fortunate that many MIMD parallel machines have started to appear quite recently and will help the dissemination.

As small KL1 programs run on a personal computer, this will be beneficial for students to learn the KL1 as an interesting example of parallel programming languages. On a large-scale parallel machine, practical application systems are expected to be developed linking KL1 programs with C programs, Fortran programs and so forth on the Unix environment.

However, dissemination of the achievements classed into knowledge processing will need more effort because the knowledge programming environment we intends has not been integrated as concrete as the KL1 and PIMOS environment. It is still a group of several independent programs such as the theorem prover, MGTP, the knowledge representation language $\mathcal{QUIXOTE}$, the experimental application system, HELIC-II, and so on. We hope our further effort to develop these programs will be able to attract many researchers in many places.

Fortunately, dissemination using a scheme of ICOT Free Software (IFS) is well received by world's researchers. It is very encouraging that more than 900 people have accessed the IFS from more than 30 countries and about 800 people from domestic sites although most of the IFS programs will not run without the PIMs. With the KLIC system, these people shall be able to have an environment in which they can actually run these programs. We hope that IFS programs will be a useful infrastructure for their research and will be further developed in many places.

References

[Uchida et al.) 1993] S. Uchida, R. Hasegawa, K, Yokota, T. Chikayama, K. Nitta and A. Aiba, "Outline of the FGCS Follow-on Project", New Generation Computing, Vol. 11, No. 2, OHMSHA and Springer-Verlag, 1993.

[Uchida (Ed.) 1992] S. Uchida (Ed.), "Proceedings of the FGCS PROJECT EVALUATION WORKSHOP", ICOT Technical Memo TM-1216, Sep. 1992.

[Shapiro and Warren (Ed.) 1993] E. Shapiro and David H.D. Warren(Ed.), "Special Section: The Fifth Generation Project: Personal Perspectives, Launching the New Era, and Epilogue", Communications of the ACM, Vol.36, No.3, Mar. 1993.

[van de Riet (Ed.) 1993] R. P. van de Riet(Ed.), "Special Issue: FGCS Conference Tokyo, 1992 ", Future Generation Computer Systems, Vol.9, No.2, North-Holland, Jul. 1993.

[Uchida 1992] S. Uchida, "Summary of the Parallel Inference Machine and its Basic Software", Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul.1-5, 1992.

[Taki, et al. 1992] K. Taki, "Parallel Inference Machine, PIM", Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul.1-5, 1992.

[Chikayama, 1992] T. Chikayama, "Operating System PIMOS and Kernel Language KL1", Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul.1-5, 1992.

[Yokota and Yasukawa 1992] K.Yokota and H. Yasukawa, "Towards an Integrated Knowledge-Base Management System", Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul.1-5, 1992.

[Aiba and Hasegawa 1992] A. Aiba and R. Hasegawa, "Constraint Logic Programming System", Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul.1-5, 1992.

[Hasegawa and Fujita 1992] Ryuzo Hasegawa and Masayuki Fujita, "Parallel Theorem Provers and Their Applications", Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul.1-5, 1992.

[Fujita, et al. 1992] Masayuki Fujita, John Slaney and Frank Bennett, "Automatic Generation of Some Results in Finite Algebra", Proc. Int. Conf. on IJCAI'93, Aug.29-Sep.3, 1993.

[K. Nitta et al. 1992] K. Nitta, K. Taki, and N. Ichiyoshi, "Experimental Parallel Inference Software", Proc. Int. Conf. on Fifth Generation Computer Systems, Tokyo, Jul.1-5, 1992.