Europal '90 Invited Paper

What We Have Done with Common Lisp in Japan During 1984 - 1989

Masayuki Ida
ida@cc.aoyama.ac.jp

Computer Science Research Lab.
Aoyama Gakuin University
4-4-25 Shibuya, Shibuya-ku, Tokyo JAPAN 150

Abstract

This paper describes the author's summary of Common Lisp related activities in Japan during 1984 - 1989.

Three major subjects are: an analysis of the activities of Jeida Common Lisp Committee (JCLC) which the author has been chairing for 5 years, a report on Lisp related academic activities in 1989, and a personal perspective.

1 Introduction: The year 1984

The year 1984 is the year CLtL was published. In USA, a convergence of the E-mail discussions gave CLtL to us as a kind of bible with the great effort of Guy L. Steele Jr and several key persons. While in japan, Spice Lisp, VaxLisp and KCL were the available Common Lisps and the author started the translation of CLtL into japanese. It was believed that making a Lisp activity was necessary. As one of the candidates, the author proposed to form a Common Lisp related committee for Jeida (Japan Electronics Industries Development Association). This proposal was accepted as one of the candidates of the next year's projects of them. Two more pre-assembly meeting were done and as a conclusion of several key companies, Jeida Common Lisp committee (JCLC) was scheduled to start in April 1985.

2 The Activities of Jeida Common Lisp Committee

2.1 Purposes

The purposes of JCLC do NOT include the issue of making a domestic standard lisp in Japan. The work was governed by the following:

- Understanding of the Common Lisp specification
- Discussions on implementation techniques
- Investigations of existing Common Lisp implementations
- Communications with international Common Lisp community

• Investigations of Common Lisp applications

As a result, JCLC have played a role of a driving force for Common Lisp.

Since the author has been involved with X3J13 activity to make a USA Common Lisp, this paper also describes some relation to X3J13. The early version of a part of JCLC report was presented at 1987 March Palo Alto meeting of ANSI X3J13 and was registered.

2.2 Organizations Sending Members

Table 1 is an alphabetical list of the organizations sent members. Note that Several subsidiaries of USA companies have played important roles. They all equally supported the activities technically and financially. Table 2 shows the numbers of members.

Table 1. Organizations Sent Members (alphabetic order)

Aoyama Gakuin Univ.	Matsushita(panasonic)	Sanyo
CEC	Meidensha	Sharp
CSK	Mitsubishi	Sinko
DEC Japan	NTT	Sord
Electro-Technical Lab	NEC	Sumitomo
Fuji Electronic	Nippon Data General	Symbolics
Fuji Facom	Nippon Unisys	Toshiba
Fuji XEROX	Omron	Yamatake-Honeywell
Fujitsu	Oki electric	Yokokawa
Hitachi	panaFacom	YHP
JRC	Ricoh	

Table 2. Members of Jeida Common Lisp Committee

Year	Number of members
1985	31 (28 organizations)
1986	37 (30 organizations)
1987	24 (47 for WGs, 27 organizations)
1988	27 (48 for WGs, 26 organizations)

2.3 Brief history

From May 1985, meeting has been held once a month. At monthly meeting of JCLC, along with the technical discussions, the presentations/hearing from almost all of the Common Lisp implementations, related announcements, and discussions have been there.

Besides monthly meetings, two types of activities have been done. One is workshop/Seminor activity to let people know what we do and to obtain a feedback. The other is working group activity to have an intensive discussion on selected issues. To grasp a major course of people, questionnaire survey is considered as important. we made it twice in 1985 and in 1987.

To obtain the reports of JCLC, please contact to Jeida. (address: Japan Electronics Industries Development Association, Kikai-sinkou kaikan, 3-5-8 Shiba-Kouen, Minato-ku Tokyo, Japan 105; The document code for the report of 1985 is 61-A-235II, for the report of 1986 is 62-A-255, 1987 report is 63-A-279, 1988 report is 89-pa-7.)

2.4 Workshop / Forum / Seminor

Five events were held during these 5 years as follows.

- 1985.9.11-12: Workshop. 28 persons; hearing from DEC VAX Lisp, IJCAI-85 report, Implementation issues, Subsetting, Objects, questionnaire
- 1986.07.08: Common Lisp /Core seminor. 60 persons.
- 1988.06.06: Common Lisp Symposium. 110 persons; Overview, Exception handling, Kanji proposal, Loop macro, Questionnaire, CLOS
- 1989.06.06: Common Lisp Forum. 250 persons; overview, CLOS, window tools, Lisp vs C, GNU Emacs Lisp, Kanji
- 1990.02.08: Lisp Forum. 120 persons; overview, Real time GC, CLOS applications, X3J13 clean up, window tool kit (CLIM and *YY*)

3 Questionnaire Asking the Directions

3.1 Profile of the 1985 and 1987 Questionnaires

Table 3 shows a profile of the 1985 and 1987 questionnaires.

Table 3. Profile of the two questionnaires

- 1985 -	- 1987 -
350 sent 135 replies 66 Common Lisp Users	200 sent 128 replies 100 CL users
General Survey of AI Tools Common Lisp Pro and Con	Utility of Lisp among AI tools Experiences with Common Lisp

In 1985, 350 questionnaires were sent to researchers and engineers engaged in knowledge information processing, and among 135 replies, 56 are from private companies, 37 from universities, and 42 from public research institutes and others. 82.7% are interested in CL, and 66 users already introduced CL, 21 are planning to introducing CL.

In 1987, 200 Questionnaires were sent to organizations which have active AI sections on October 1987 and summarized on Feb. 1988. The contents are much more weighed on the actual usage/utilization of Lisp than that of 1985. Among 128 replies, 39.1% are from private companies' development sections, 28.1% are from research labs, 10.2% are from universities. All replies but two are from experienced Lispers.

Table 4 shows machines used. Table 5 shows Common Lisp implementations used. It shows wider acceptance of Common Lisp.

Table 4. Machines Used

- 1985 -	- 1987 -
Vax (72)	Vax (56)
Sun (28)	Sun (32)
Xerox-1100 (27)	NEC PC98 (16)
Symbolics (22)	Symbolics (12)
Facom M (17)	Xerox 1121 (12)
ACOS (17)	Hitac (10)
Hitac M (16)	Fujitsu (9)
U-station (15)	Explorer (9)
DEC-20 (15)	Fujitsu PC (8)
-	ELIS (7)

Table 5. Implementations Used

- 1985 -	- 1987 -
Franz Lisp (60)	KCl (42)
Vax Lisp (40)	Franz Lisp (26)
InterLisp/-D (33)	Common Lisp (24)
UtiLisp (33)	Vax Lisp (23)
Zetalisp (18)	UtiLisp (11)
KCl (17)	Interlisp-D (11)
-	GCLisp (11)
-	Zetalisp (8)
-	TAO (7)
-	VOS3 Lisp (7)

3.2 1985 Questionnaire

To know the basic needs, in October 1985, JCLC sent questionnaire to selected organizations, who might be considered to be the most advanced ones in Lisp related works in Japan. This results of the questionnaire assured the direction of the later related activities of us in japan. We realized the needs for Common Lisp is large. Furthermore we also realized that if we can make a guide line for object oriented facility inclusion, japanese character provision, and subsetting consideration on Common Lisp, it will almost cope with the japan domestic needs which were considered at the time of the questionnaire.

The official report was written on March 1987 by Ryu Katayama (Sanyo Electric Co.,Ltd.) who was in charge of the 1985 questionnaire.

Here is a summary of the results:

• Widely used Lisp languages are Franz Lisp (60), VAX Lisp (40), Interlisp/ Interlisp-D(33), UtiLisp (33), Zetalisp (18), Kyoto Common Lisp (17)

• Popular Prolog languages are C-Prolog (46), Prolog-KABA (44), DEC-10 Prolog (23), micro-Prolog (19), Quintus Prolog (17), Prolog/KR (17).

• Familiarly used object oriented languages are smalltalk-80 (38), Flavors (19), Loops (7), Objective-C (8).

• Among other conventional languages, C (89 answers), FORTRAN (51), PASCAL (49), BASIC (30) are also used.

• Major objectives for those AI languages are for development of ES (29.6%), natural language processing (including machine translation) (17.8%), intelligent man machine interface (17.8%), image recognition/understanding (10.4%).

• On the development of ES, 62.8% organizations are developing ES by their own, and among them 46.8% are also developing ES development tool (shell).

• Languages for developing shell: Lisp (55 answers), Prolog (26), C (19).

• Widely used shells: OPS5 (18), KEE (8), ZEUS (7), ART (5), BRAINS (4).

• Complaints for Lisp: slow execution speed (32), poor graphics (28), inability to handle Japanese characters (26), lack of object oriented facilities (18), large size of required memory (16), inefficient programming tools such as editor or debugger (16).

• Comments on language specifications of CL that appreciates compiler oriented design: lexical scope or function closure (20.3%), sufficient data types (8.6%), while point out the large memory use (21.4%), needs for object oriented facilities (18.7%), requests for supporting Japanese characters (16.0%), etc.

• Needs for subsetting: 60.2% answers requests some suitable organization or committee should deal with the standardization of CL subsets.

3.3 1987 Questionnaire

Here is a summary of the 1987 questionnaire.

- Lisp implementations used are shown in Table 5.
- Application area: Expert system building (59), Intelligent man/machine interface (26), Natural language processing (15).
- The ratio of Lisp among development languages:
  Lisp is not/seldomly used (9), Lisp is the major language (53), C is used with Lisp (47).
- Development type: Research (65), prototyping (66), Internal use (27), Commercial product (36), others (8).
- Advantages of Common Lisp: its flexibility (43), productivity (39), rich environment (19).
- Problem with Common Lisp: slow speed (32), large memory requirement (22), irritating GC (13).
- Manys complain about the functionality of debugging facilities, 60% complain about the graphic interface, Satisfaction level with japanese handling is half and half,

- The top four reasons to choose Common Lisp; 1. defacto standardness, 2. compatibility of application software, 3. rich functionality, 4. suitable for large scale development.
- The top four claims on Common Lisp disadvantages are; 1. huge implementation, 2. complex specification, 3. slow interpreter, 4. function naming with long spelling.
- Among the proposal to improve CL, the extension of CL specification for the following items are significant;
  - object oriented facility,
  - loop facility,
  - multi process,
  - multi national character handling,
  - window tool kit,
  - graphic functions.

  (With this result, the author encouraged X3J13 to adopt CLOS, Loop, character extension proposals. Also started a window tool kit project.)

4 JCLC Working Groups Activities

Under the committee there were several working groups for each year.

• object oriented working group (OOWG 1985, 1986, reformed as OOSG 1987, 1988) OOWG was headed by Toru Ishida (NTT) for 1985 and 1986, and by Kiyoki Ohkubo (pfu) for 1987, 1988.

• subset working group (subsetWG 1985, 1986) headed by Katsuhiko Yuura (Hitachi).

• kanji working group (kanji WG 1986) headed by Fumio Motoyoshi (ETL).

• bboard working group (bboardWG 1986)

• application working group (apWG 1987, 1988, 1989) headed by Katsuhiko Yuura.

• technology working group (techWG 1987, 1988, 1989) headed by Fumio Motoyoshi.

The early summaries of OOWG, subsetWG and kanjiWG are contributed by the chair of each WG.

The formation of the WG was based on the results of the 1985 questionnaire. That is, subsetting, object oriented facilities, japanese character manipulation. As Dr. Yuasa of Kyoto University brought us a copy of the archives of Common Lisp discussions, we formed a bboardWG in 1986 to translate the contents. Temporary groups were occasionally gathered to translate Loop facility documents, exception handling documents, and so on.

4.1 Subset WG for 1985 - 86

4.1.1 Overview

During 1985 and 1986, Subset WG made a CL/Core design and was reported and registered to X3J13. CL/Core was based on Ida's private proposal and 19 times meetings refined it. Huge man powers were spent. On 1986.07.08 Common Lisp/Core Seminor was done with 60 attendants. On 1986.08.05 at Boston, CL/Core was presented.

The trial was made to reduce the size of implementation and to make Common Lisp slim. But, since large applications need rich functionality and small applications does not always be necessary to standardize, along with the growth of memory size of personal computers, Common Lisp/Core effort did not give an actual implementation yet. Rather, several existing Common Lisp from Japanese Manufacturer learned Common Lisp with this activities and extended to full level.

4.1.2 A Survey Report

- Excerpted from the report written in 1986 by Katsuhiko Yuura (Central Research Laboratory, Hitachi,Ltd.) -

1. Background and Purpose: Most Japanese Lisp users on personal computers(pc) do not use all functions of the full set of Common Lisp so that good performance is a constant concern. To set up an international Lisp standard for pc, the authors would like to propose a subset, which does not include seldom used functions and those that make a system inefficient.

2. Discussions and Proposal Activities: Discussions for the subset started in December 1985, and the authors decided on four steps for the completion of Common Lisp/Core. The first step was the review of Ida's personal proposal for a subset (IPSJ WGSYM 34-4), and the second step was the examination of the necessities for each function and the difficulties in implementing them. In the third step the basic issues of Common Lisp/Core were decided, and in the fourth step the functions were selected by the vote of WG members.

3. Basic Issues of Common Lisp/Core: Common Lisp/Core preserves the arms and legs of Common Lisp, because it is important to be able to transfer programs in the subset to those in the full set easily, as well as to enable subset users to grow into full set users naturally. The following features were selected in the third discussion step: scope and extent rules including lexical closure features, keyword parameters, the principles of type hierarchy and generic functions, some useful and characteristic data types (bignum, ratio, structure and readtable), some package features.

On the other hand, Common Lisp is characterized by rich functions, so that the aim for the number of functions in Common Lisp/Core is about a half of the full set, which is over that of Utilisp (developed at University of Tokyo in 1981, one of the most famous Lisps' in Japan) or Franzlisp. The functions which do not correspond to the "arms and legs" features were selected in the fourth discussion step, giving higher priority to functions having high necessity than to functions that can be implemented easily.

Common Lisp/Core includes **356 functions and 20 variables and constants**, whereas Common Lisp includes 622 functions and 101 variables and constants. Major deleted items are most system parameter constants, complex numbers, most package features, local functions, adjustable arrays, hashtables and pathnames.

4.2 Object oriented WG for 1985 - 88

4.2.1 Overview

In 1985, the three proposals in the USA (CommonLoops from Xerox, Object Lisp from LMI, and Snyder proposal from HP), and New Flavors from Symbolics were investigated. Also the discussions on E-mails were brought by Ken Kahn and analyzed. Every process was reported to US Common Lisp key persons. The members felt CommonLoops was one of the best and later on the focus has been on CommonLoops.

Several versions of Portable CommonLoops have been brought by G. Kiczales with the courtesy of Xerox PARC. They were re-sent to OOWG members, including universities and software vendors. PCL has been used as an executable model to analyze.

In 1987, CLOS was the subject to discuss. X3J13 87-002 was translated into Japanese. some results of the discussions were sent to US committee and used on improvement of the draft. In the discussion, implementation topics were also there.

In 1988, 88-003 MOP was translated into Japanese. CLOS book was published, which contains tutorial articles for technical details and translation of 88-002R. Several opinions were sent to US committee and some were adopted.

4.2.2 A Report for 1985 and 1986 Activities

- Excerpted from the report written by Toru Ishida (NTT Electrical Communication Laboratories) -

OOWG was organized in April 1985 as the first working group of the JCLC. Although there were more urgent problems for the Japanese Lisp Society, we started the OOWG so early because:

- The Object Oriented Programming paradigm was the center of attention in Japanese academic society at that time. However, notion and its effectiveness were not well understood by industrial society.

- Standardization of new programming paradigms requires a lot of experience. Thus, we thought discussion should begin as soon as possible.

In 1985, six companies joined the OOWG. The members reviewed and summarized on-going discussions of the United States, and reported to the JEIDA Lisp Workshop and the IPSJ (Information Processing Society of Japan) Special Interest Group of Symbolic Processing.

Since 1986, fourteen companies have been involved. Several members of the OOWG attended the Lisp Standardization Meeting in Boston in August. In addition, an extended JEIDA meeting with Gregor Kiczales, was held in October.

4.3 Kanji WG for 1986

4.3.1 Overview

Kanji WG made a design to cope with japanese characters with the major contributions from Fuji Xerox and Symbolics along with the japanese companies. The author and the Kanji WG have a desire to send their specification to X3J13 as a base for the extension of Common Lisp for the multi national or the multi character sets treatment standard.

In 1986, the first version of our Kanji specification was compiled and the WG was closed. The follow up job was inherited to technology WG.

4.3.2 A Survey Report

- Excerpted from the report written in 1986 by Motoyoshi, F. (Electrotechnical Laboratory) -

We have discussed the way to embed Japanese characters in Common Lisp for a year. We first decided on the policy to be followed in the whole discussions: "a Japanese character should be treated as a character of Common Lisp."

All members of the working group agreed this policy and many users prefer it to others. Then we drew up guidelines to be respected within the limits of above policy. The are

1) a program which is written without regard to Japanese characters should run as expected when it is given data which include Japanese characters,

2) any multi-byte character set other than Japanese characters can be implemented in the same manner,

3) a program which does not deal with Japanese characters should run without so much loss of efficiency for speed and memory.

According to above principles, we have made a proposal to embed Japanese characters in Common Lisp as described in the following. The basic idea is very simple and is that "let the value of 'char-code-limit' be large enough for the system to include all Japanese characters." This means that one can even define a read macro to any Japanese character.

By taking that way, we need only one change to CLtL, which is concerning to character printing width. The point is that the set of fixed-pitch characters for font 0 is limited to graphic standard characters. Instead, we introduce a function 'write-width' for getting the print width.

No problem occurs when a system has a uniform internal structure for strings, however, some system may have two or more types of strings internally for memory efficiency. There may be some loss of efficiency for users who use only standard characters in such a system.

A new type of string is introduced in such cases and is used mainly in declarations. The name is 'internal-thin-string', which is a vector of special characters distinguished from others by a predicate 'internal-thin-char-p'. This does not mean that 'internal thin character' should be a proper subset of 'string character'. They may represent a same character set and this case is equivalent to the above one for uniform structure of string.

We have described the common topic to any multi-byte characters so far. We also discussed the problem specific to Japanese, where our main concerns are related to the meaning of each Japanese character. The major problem is that of characters whose print forms are similar to those of standard characters. We reached to the agreement that any system should have at least one mode where any characters other than standard characters have not special meanings.

4.4 Technology WG for 1987 –

Technology WG was formed in 1987. It focuses on the technological issues of Common Lisp.

In 1987, 18 persons from 18 organizations joined. Exception handling and Scheme were the subjects for 1987 activities. Pitman's proposal was translated and discussed.

In 1988, the kanji proposal of 1986 was traced. X3J13 extensions and it were compared. Investigations on existing implementations were done for 9 implementations, that is, Lucid CL, DG CL, VAX Lisp, TAO, VOS3 Lisp, HI-UX lisp, NCL Unisys, Xerox CL, V10 Lisp and Symbolics CL. Most implementations are found to be satisfactory level, except two implementations. The result was sent to X3J13.

In 1989, real time GC algorithms are the main subjects. Reading several academic papers were done.

4.5 Application WG for 1987 –

Application WG was formed in 1987. It focuses on the utilization, usage, user's point of view of Common Lisp. In 1987, 14 persons from 14 organizations joined. Questionnaire and its analysis were the subjects.

In 1988, 12 persons joined. The productivity of Lisp and C were analyzed and reported using 10 small software projects including file handling, code conversion, data handling, search, pi calculation, and some typical symbolic processing applications.

In 1989, the productivity of CLOS is the main issue. Several sample projects were written in CLOS.

5 Common Lisp Implementations / Related Products in japan.

The year 1989 is a hot year for Common Lisp community as a result, meaning several years from 1984 were effectively spent. To show data, here are a table for a history and a list of recent topics.

Table 6 is the list of Common Lisp implementations and some related products during 1984 - 1989.

Table 7 shows the list of Lisp related presentation at a recent academic conference (Annual Conf. of Information Processing Society Japan, Oct. 1989). This conference covers all the Computer related topics from application, education, social issues, semiconductors, architecture, network, programming languages, OS, and so on. Then, the fact that among 1168 presentations 82 papers are on Lisp is quite good evidence to indicate "Lisp is used in various fields." Especially expert system area and CAD area are significant. 82 out of 1168 papers means *"Lisp is 7 % of total presentations at 1989 IPSJ annual conf."* The invited speech of the conference was "Common Lisp Object System" by Gregor Kiczales of Xerox PARC. The author believes that the year 1989 was a hot year for Common Lisp community as a result of several years efforts from 1984.

6 Where do we go ?

General comments on future works are as follows. Nowaday, Lisp, or Common Lisp is used not only by researchers and developers in Computer Sciences and Mathematics, but also by people in real application fields directly and indirectly. The culture level of Lisp in Japan seems to be not matured yet with the author's point of view. Much more works on applications, educations, development environments are needed. As a kernel of such works, to tell people 'why' aspect is important.

The author feels the primary role of the frontier activities described here and directed by the author, becomes an end and some cultivating activities should start. Much more on user side by people in every field will be needed and expected. We started to discuss about how to inherit the output of our activities and seem to come to the end with a conclusion that we should make a publicly accessible database for internet as one of the works to do. We think it should contain the document which describes the language, public source codes, and many useful information.

On the other hand, we are facing the needs to have a common interface among various Common Lisp implementations. There is no standard for window toolkits or GUI/UIMS. Furthermore, lots of window systems are there. To cope with this situation, as one of the trials, *Yy* project has started to provide a common window tool kit on several window systems.

Acknowledgements

This paper contains the results of contributions of many persons. The author would like to express sincere thanks to all the anonymous contributors especially from JCLC. Communications with X3J13 members assisted a lot on several stages for decisions. This paper is not the official report from Japanese national organizations, but a private one.

Table 6. Common Lisp Implementations and some related software in Japan

1984
- Nihongo VAX LISP (DEC Japan -C-)
- Kyoto Common Lisp (Kyoto University -RC-)

1985
- ESHELL/FM (Fujitsu Ltd. -A-)
- KBMS (NTT -A-), EXCORE (NEC -A-)
- GCLisp (CEC), KEE (CSK)

1986
- KS-300 Series (Nihon Unisys Ltd. Japanized TI-Explorer I, II -I-)
- KEE/KS-300 (Nihon Unisys Ltd. -I-)

1987
- VOS3LISP (Hitachi Ltd. -C-)
- Lisp System (Fujitsu Ltd. -C-)
- TAO (NTT Intelligent Tech. Co., Ltd. -C-)
- Symbolics Common Lisp (Nihon Symbolics Inc. -I-)
- ABCL/1 (Tokyo Institute of Technology -RA-)
- Nihongo San Marco Lisp Explorer (CEC -A-)

1988
- Hi-UX LISP (Hitachi Ltd. -C-)
- Lucid Common Lisp 3.0 /NEWS (CSK Corp. -I-)

1989
- VOS3LISP E2 (Hitachi Ltd. -O-)
- ELIS Common Lisp (NTT Intelligent Tech. Co. Ltd. -C-)
- Nihongo Common Windows (CSK Corp. -A-)
- DIET (Matshushita Electric Industrial Co., Ltd. english text understanding -RA-)
- MES (Matshushita Electric Industrial Co., Ltd. expert shell -RA-)
- Office Expert (Fujitsu Ltd. Office resources layout and management -A-)
- YY on X (Aoyama Gakuin University -RA-)

1990(Scheduled)
- NX-LISP (NEC Corp. -C-)
- Hi-UX Lisp E2 (Hitachi Ltd. -O-)

-A-	:	Application software on top of Common Lisp
-I-	:	Imported from abroad and Japanized
-C-	:	Originally developed Common Lisp implementation
-O-	:	Originally developed Common Lisp implementation with CLOS chapt.1 and 2
-R-	:	Research Output

Caution:
1. The above list does NOT include imported products which have no modification by japanese distributors. Two examples are Allegro Common Lisp and ART. (they may be world-wide uniform products.)
2. PC related subset Common Lisps are not included.
3. The list does not try to list up all the applications developed using Common Lisp. Some are included to record milestones.

Table 7. Lisp related presentations at a Japanese academic conference

====== Applications on top of Lisp based shell ======
Development of the Intelligent Sightseeing Guidance System (ESHELL)
A scene-builder using frames and rules for expert-systems (KBMS)
Expert system developing support facilities of KBMS (KBMS)
Design Expert System for Kitchen Layout (ART)
Evaluation on the Representation for Design Operators (EXLOG)
A Telephone Sales Expert System and Its Implementation (GC-lisp, KBMS)
An expert system for verification of switching sequences (KEE)
Chemical Graph Database based on Function View (G-Base)
Layered String Octree for Solid Object Indexing (G-Base)
====== Applications on Lisp ======
Intelligent Consultation with Graphics by ''MISAKO'' (Lisp)
An Object-oriented Knowledge Base System for Aiding Computer Utilization (PCL)
Expert System for Fault Diagnosis by Using online Data on Power Distribution system (Lisp)
Troubleshooting Expert System of Counter Circuit on Object (Lisp)
Study on schedule Optimization Methods through Knowledge-Based and Neural-Network-Based
Approaches (Lisp), An Implementation of HPS with PHIGS and TeX (ELIS)
An Object-oriented Graphics Package for Drawings based on Hierarchical Data structure (Lisp)
Japanese Full Text Search and Various Expression (Common Lisp)
A Study of Editting Support System in Logical Document Structure (Lisp)
Object Manipulation Language for an Object-Oriented Database Management system - Odin (Lisp)
An Automatic Programming System for Sequence Control (lisp)
Programming guide system (Common Lisp), Processor Architecture Simulator PASIM (VOS3 Lisp)
A Design Verification System for sequential Control Circuits (VOS3 Lisp)
Estimation System of Machine Cycle Time for Large Scale Computers (VOS3 Lisp)
Development of a Performance Evaluation System for designing Hardware Functions (Common Lisp)
====== Lisp based Experimental Research ======
A User Model Construction Method for Editor (emacs lisp)
Lazy Updating Algorithm of ATMS Labels (Common Lisp)
Frame System Based on Fuzzy Set Theory (Franz Lisp)
The Implementation and Evaluation of Strategy Annotated Term Rewriting System (Lisp)
A Script Interpreter on the Platform CANAE of User Interface Development (Citrus Lisp)
TOP-1 Operating System (lisp), Memory Management for Linked Data Structures (lisp)
A 'Process+Field' Based Cooperation Model (KCL), Event-Driven Real-Time AI Engine (ELIS)
A Study on Object Oriented Languages for Knowledge Processing (ELIS)
A Study on Introducing Constraint to Object-Oriented Programming (ELIS)
An Extension of Object-Oriented Data Model for Simulations of Parallel Processor Systems (ELIS)
A Consideration about fusion of programming paradigms (ELIS)
Writing functional specification using FDSS (Franz lisp)
Transformations between different formal specifications (Lisp)
The hybrid system of the Parallel Lisp machine and Neuro engines (Neuro Lisp)
Online of an Experimental Spoken Language Translation System from Japanese to English (Common Lisp)
====== Lisp Implementations / Environments ======
Development of X Window System on ELIS 8200 (Common Lisp)
Realization of Window Environment for Lisp-based Object-oriented Language KPL (KCL)
Construction Toolkit for Graphical User Interface in TAO (ELIS)
An Implementation of Multiprocessing-Lisp by the Lisp Machine SYNAPSE (Synapse Lisp)
Queue-based Multiprocessing Lisp (Common Lisp), A RISC Architecture for AI Languages (Lisp)
Development of Multi-Tasking Common Lisp on OS/2 (lisp)
RPC facility of on-board Lisp machine (ELIS), Concurrent Logic Language on TAO/ELIS (ELIS)
Multiple Programming on ELIS Common Lisp (ELIS)
Preprocessing Mechanism on ELIS Common Lisp Interpreter (ELIS)
Object-Oriented Programming with HiOBJ2 based on CLOS (HiLisp)
A Revised Implementation Method for the cUtiLisp System (UtiLisp)
The Performance Evaluation Method for Analyzing Characteristics of LISP System (Fujitsu Lisp)
A Lisp Program Development Support Tool Based On Stepwise Refinement Technique: FASET (Common Lisp)
Hardware Development of the MacELIS Desktop Lisp Machine (ELIS)
Single Board LISP Processor for the New-ELIS (ELIS)