COLING 82, J. Horecký (ed.)
North-Holland Publishing Compar

A SOFTWARE ENVIRONMENT FOR DEVELOPING NATURAL LANGUAGE UNDERSTANDING SYSTEM

Fumio Mizoguchi and Shozo Kondo

Department of Industrial Administration
Science University of Tokyo
Noda 278, Chiba
Japan

This paper deals with a software environment for developing the natural language understanding system. The system called Multi-Layered Software Environment (MLSE) is proposed for providing a designer of a language understanding system with wide varieties of design alternatives in software components which derive from Computational Linguistics. The MLSE is a collection of module packages for building the tools for language understanding system. That is, by integrating the Computational Linguistics methods into the MLSE scheme, we have emphasized the layered approach to build the software environment as the basis for developing the language understanding system. In order to illustrate the strategy of the MLSE scheme, we have discussed the case study for designing Question and Answering system. Based upon this case study, we have developed a new language understanding system called Japanese Q & A system (JQAS) which was a product of the MLSE scheme. The MLSE has implemented in variety of LISPs such as Inter-Lisp on DEC-20 and Lisp F3 on IBM370.

The JQAS has domain specific Q & A system for computer documentation and explanation system.

INTRODUCTION

This paper is focussed on the necessary software environment for designing Natural Language Understanding (NLU) systems . Here,the notion of software environment is to provide a designer of an understanding system with wide varieties of design alternatives in software components which derive from the computational linguistics methods. Therefore,the goal of this paper is to propose a scheme for the software environment which is appropriate for the design of a NLU system. In order to accomplish this goal, the Multi-Layered Software Environment (MLSE) which we propose in this paper is described and implemented for the development of a NLU system.

In order to build a NLU program,we must formulate the problem to fit the existing tools of Knowledge Engineering, or build the tools that will be applicable to the problem. For this purpose, there have been a number of attempts to support these design activities such as E-LINGOL and ATN. As for knowledge representation, KRL, FRL, Unit and RLL are proposed as general purpose semantic representation systems. If the tools for NLU

systems are suitable to the problem domain, the necessary task of the designer is to select the best tool among them and to formulate the problem in terms of the format which are specified in the tool. Problably, this design process is highly dependent upon the software or tool which is applied to the problem. Then, the design activities will focus on how to formulate the "Domain Semantic" in NLU.

SOFTWARE ENVIRONMENT

The present study proposed to support the design of natural language understanding system by considering the following components.

1. As the basic implementation language, we select Lisp F3 written in FORTRAN which is not machine dependent. Our computing facilities are both IBM 3031 of the Science University of Tokyo and M200 of the Tokyo University of Computer Center. However, the implementation is carried out by Inter-Lisp of DEC-20.

2. The necessary Lisp functions are developed by the F3 users. As a result, the user of Lisp F3 can extend the facilities of Lisp function to meet his requirements. The Lisp system at this stage includes the same level of Inter-Lisp.

3. In order to support the coding and debugging, we have designed the prettyprint and zoom editor as useful functions. These functions are considered as the primitive functions for developing the modules library.

4. There are a number of module programs which are further developed into sub-components of the Computational Linguistic. These programs are mainly parser, generator and semantic representation system.

5. The natural language understanding system,especially, Question & Answering system is developed by combining these modules with scheduling function which is called Agenda. Therefore,the design of Agenda is important part for developing the various control structures which deal with language processing.

To develop these several components, we have prepared the scheme called MLSE which is shown in Fig.1. MLSE consists of five layers which correspond to the above components.

The main layers from CLL to UFL in Fig.1 are the basis for developing the rest of two layers which are regarded as a tool level for natural language understanding system. In case of a Module Package Layer(MPL), there are two kinds of program packages. One is a collection of module programs for pattern matchers which are further served as components of parser, generator and Frame base system. These pattern matching modules can treat a variety of data structures from Lisp basic data to user defined data such as Frame and semantic nets. The other is derived from the pre-existing tools such as natural language interface and knowledge representation system. In this respect, we have developed ATN base semantic nets system for Japanese. Therefore, MPL is a large collection of tool module program from components to tool levels. These modules are consistent in coding style which is useful for a notion of module library.

The top level of the layer is called Language Understander Layer (LUL) which is built as domain specific language processing system. The LUL is the language system which is implemented by using the components of MPL with user's defined main program. That is, by using the package programs from MPL, we

can develop the domain specific tool for the language processing system. In the scheme, LUL is integrated into the specific knowledge base systems such as System-1, System-2 ... and so on. The system corresponds to the actual Q & A system.

The present implementation is carried out for designing Japanese Q & A system for instructing computer documentation system. That is, JQAS is a kind of instructional system for supporting the user's documentation and retrieval of necessary information on MLSE programs.

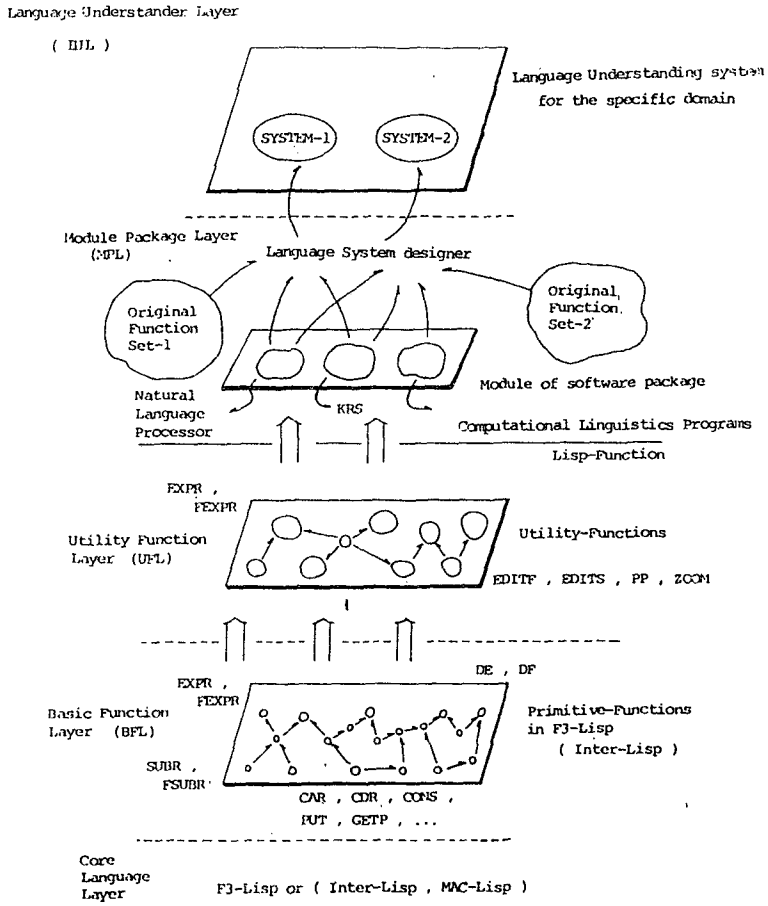


Fig.1 A Scheme for Multi-Layered Software Environment

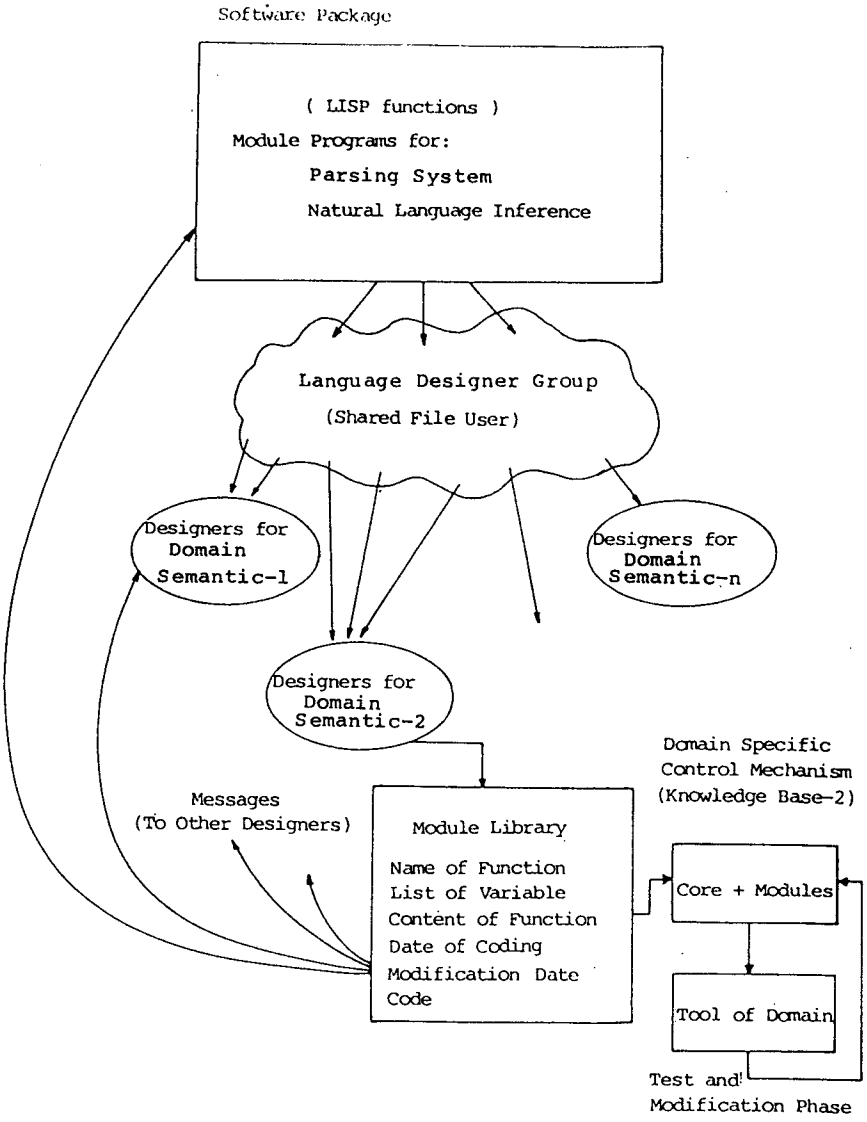


Fig.2 Design Process for NLU System

DESIGN PROCESS

The design process within MLSE is described by the scheme which is shown in Fig.2. This scheme is obtained from the generalization of the design process in an interactive system by introducing the notion of " shared package user's group " that have the common files and utilize the basic componets of a NLU program. As is shown in Fig.2, the activities are divided into three parts.

The first is program packages which consist of various program modules for computational linguistic methods. These modules include the components for a paser and a knowledge representation system.

The second is the shared package user's group that develops cooperatively the NLU system. The communication among the group is made by the use of a mail system.

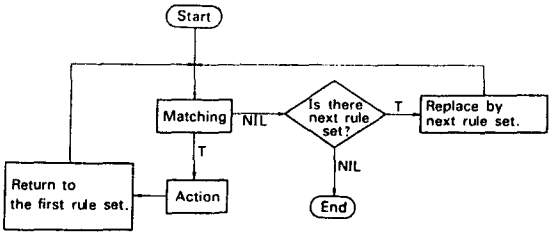
The third corresponds to the construction of NLU system by using the modules programs. This implementation is carried out by combining the package into the core program which fits the purpose of the NLU system. If there are some changes in the package program,then the user must accept the message on the code modification by documentation.

The entire design process is usually step by step in order to meet the requirement of a given problem domain. Therefore, there are a number of messages passing among the designers to accomplish the knowledge base system. This feedback loop is described in terms of a testing and modification phase in Fig.2. Probably, the main activities of this process are to document the module library which is a collection of LISP code, name of function, list of variable,coding date and so on. By looking at these documentations, the designer can build the domain specific tool without constructing the common programs for a NLU system.

CASE STUDY TO DEVELOP SIMPLE AGENDA SYSTEM

In this section, we shall show details of how to build the tools by using components within the MLSE scheme. The tools that we have selected as case studies are a design of various Agenda systems. We believe that agenda is one of the important system for building a NLU system.

Let us start with the first example of simple agenda system which is shown in Fig.3. In this flow chart, the agenda system operates within a control mechanism called the matching-act cycle. In matching, it finds the rule to be executed, and in action it executes this rule which is described by the action sides of the rule form. The matching-act cyle is repeated until either no rule can be found or an action element of rule will stop the processing. The LISP code of this flow chart is shown in Fig.3. The definition called PS-TOP is the function for dealing with the matching-act cycle in the agenda system. Although this coding example is simple, the design of agenda system is started in this way. In this coding, the other functions are not included, but it is clear in order to show the control structure of agenda system. Therefore, this example is regarded as the basic component of agenda system.




```
<PS-TOP
(LAMBDA NIL
        (* PS-TOP CONTROLS SYSTEM FLOW OF PURE PRODUCTION SYSTEM)
        (PROG (RULES MODEL RULES* DB RULE RULE-NAME)
         LOOP1
              (START)
         LOOP2
              (SETQ RULES* RULES)
         LOOP3
              <COND ((NULL RULES*)
                     (COND ((END) (GO LOOP1)) (T (RETURN NIL>
              (SETQ RULE (CAR RULES*))
              (SETQ RULE-NAME (CAR RULE))
              (* IF PATTERN-MATCH-DB RETURNS TRUE , ACTIONS OF THE MATCHED RULE
                 IS ACTIVATED)
              (COND ((PATTERN-MATCH-DB (CADR RULE))
                     (ACTIVATE-ACTIONS (CAR (CDDDR RULE)))
                     (PRINT-RESULT)
                     (GO LOOP2)))
              (SETQ RULES* (CDR RULES*))
              (GO LOOP3))))>
```

Fig.3 A flow chart for control structure of a simple agenda system and LISP code

CONCLUSION

As for the scheme of software environment to meet the requirements for a design of NLU system, we have proposed a five layered environment which has hierarchical, multi-level structures of software systems. The scheme is called MLSE and these characteristics are following:

1. There exists the core language which served as the assembly language in NLU systems. In this case, LISP is selected.
2. In case of designing the NLU systems, the utility functions such as structural and zoom editor are some of the important functions. In this respect, LISP is already developed in these functions. These facilities are important to develop the module packages.
3. There exists a number of NLU module packages which may be further used as the components of tools.

We believe that these three characteristics are applicable for designing NLU system.

REFERENCE

Winograd,T."Language as a cognitive process"Reading,Mass.:Addison-Wesley,(draft).