PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT. © ICOT, 1988

The Research and Development of Natural Language Processing Systems in The Intermediate Stage of The FGCS Project

UCHIDA Shunichi, YOSHIOKA Tsutomu, SUGIMURA Ryôichi,
TANAKA Yûiti, HASIDA Kôiti, and MUKAI Kuniaki
The Second Research Laboratory, ICOT

ABSTRACT

This paper is an introduction to the research activities on natural language processing (NLP) in the Fifth Generation Computer Systems (FGCS) project. Aiming at a verbal interface of computers, the NLP research in this project concerns itself mainly with discourse understanding.

The intermediate stage of the project has been devoted both to the improvement of the existing methods of syntactic processing as a basis for dealing with contextual aspects such as ellipsis and anaphora, and to the establishment of fundamental methods for representing and processing the structure of discourse and context. These activities are accommodated in the development of an experimental discourse understanding system called **DUALS**.

The methods of processing syntactic aspects have been compiled into a general-purpose software library, called the **Language Tool Box** (LTB). The methods in the library can be used, for example, to provide a verbal interface between the user and both expert systems and database systems. In addition to such research as DUALS and LTB, which are conducted on the initiative of ICOT, research conducted by the NLP research groups at the manufacturers cooperating with the FGCS project is outlined.

1 Introduction

The research on natural language processing (NLP) in the FGCS project aims at the implementation of a discourse understanding system. This system should be able to participate in Japanese discourse, and is to serve as a basis for verbal communication between human and computer.

The research in the initial stage attempted to recast in terms of logic programming the current methods for NLP, such as for parsing, to unravel the problems arising there, and to figure out the possibilities of further advancement. In this way, an experimental discourse understanding system called **DUALS-I** was developed in Prolog running on a DEC2060 computer, and was demonstrated at the last FGCS conference, held in November 1984. This system was able to comprehend and answer questions about a small passage which was extracted from a 3rd-grade primary school textbook of Japanese. The passage consisted of 18 sentences/200 words, and included dialogue.

DUALS-I could answer the prepared questions, but its parser, grammar, dictionary, and problem-solving method were quite immature and obviously needed a lot of further development. Nevertheless, a discourse understanding system was compiled into a system on Prolog, which demonstrated the merits of implementing an NLP system in terms of a logic programming paradigm. This not only gave a concrete understanding of the clear view that a logic paradigm provides for software development, but also brought up several new ideas about how to construct a parser and about semantic representation. The research subject was determined in this way in the first half of the intermediate stage.

The initial task of the intermediate stage was a total reconsideration of DUALS-I. The morphological analyzer, the parser, the simplistic context analyzer, the generator, and the dictionary were all to be revised to be more general versions.

As for the underlying hardware, DEC2060, whose memory size and interface functions were very limited, was replaced by PSI-I, a logic-programming machine newly developed at ICOT. A programming language, CIL, was developed and employed. It is an extension of Prolog and is more suitable for describing meaning and situation. This new system, called **DUALS-II** was completed at the end of 1986. The adoption of PSI, which provided a large memory and a multi-window environment with Japanese characters, and the employment of CIL as the programming language meant that the syntactic aspects of the system, such as in the parsing module and the sentence generation module, acquired advanced functions and a clear perspective in the software structures. The semantic aspect of the system was put in order as well, along the line of the development of Situation Theory and Situation Semantics.

Through the development of DUALS-II and the accumulation of software thereby, it became possible to shift the focus of research up to processing of context. The next research target towards the end of the intermediate stage, therefore, is to deal with a larger text with 200 sentences/2000 words, and to accept a much less restricted variety of questions. This means that we face the problem of dealing with a wide range of context. We will attack this problem with **DUALS-III**.

In DUALS-III, with a far greater number of sentences, software that just deals with one sentence at at time no longer works. A systematic design method must be engaged. The functions and knowledge of the syntactic aspects of the system, including the parser, the generator, and their dictionaries, were completely rebuilt from the basic methods, in order to achieve greater generality. Also, realistic contextual processing necessitated a thesaurus or a concept dictionary. For these tasks altogether, it was essential to have plenty of tools and adequate environments for developing each function module.

Syntactic processing has matured to the extent that the software products can also be used for Japanese language processing in domains other than discourse understanding. We decided to extract those software modules for syntactic processing and put them together to make a library of common software tools called **LTB (the Language Tool Box)**, a general Japanese processor.

Context processing is a field that is still wide open from a world-wide perspective. Various methods must thus be investigated upon materials that contain a variety of linguistic phenomena. In the FGCS project, the research group at ICOT and those at cooperating manufacturers are working together in seeking adequate methods for context processing, dividing up the whole work into various subtasks. LTB, a common software tool among these groups, has also been developed through the group's joint efforts.

The research on NLP in the FGCS project is described below. The central topic will be the work concerning DUALS, but the research by the groups at cooperating manufacturers will also be mentioned.

2 The Research Plan of the Intermediate Stage

2.1 The Research Topics and Goal

Regarding discourse understanding, the research goal of the intermediate stage was the experimental development of a software system that could process semantics and contexts. **DUALS-I**, the pilot system in the initial stage, dealt with a sample text containing 18 sentences/200 words, and the new experimental version towards the end of the intermediate stage was to handle a text containing 100 sentences/1000 words.

Many technical tasks must be completed to achieve this goal, including the following research topics.

(1) The analysis functions for handling morphological, syntactic, and semantic information should be improved so as to cover a larger sample text. Processing speed must be increased enough to carry out efficient experiments. The generation and analysis functions ought to fit each other, and share the same syntactic rules.

(2) A dictionary containing morphological, syntactic, and semantic information needed for sentence analysis should be tailored to work efficiently with the functions provided by the analysis modules. This dictionary will become very large, so peripheral software tools must be prepared that can check the consistency of the its contents of the dictionary.

(3) A high-level programming language should be developed to write the analysis module and to describe the intermediate semantic or pragmatic information, so as to avoid cumbersome coding in Prolog or ESP.

(4) A theoretical model both to represent the structures of meaning and discourse and to carry out inferences on them should be devised on the basis of Situation Theory and Situation Semantics.

(5) Practical schemata and methods for system building should be developed to implement discourse processing methods such as those that resolve anaphora and ellipsis.

(6) The method to construct the thesaurus and the concept dictionary should be considered mainly in connection with the given sample text. Words and concepts appearing in the text must be analyzed, classified, and then compiled into a dictionary. The world-knowledge base should also be prepared for understanding the domain talked about in the sample text.

2.2 Implementation of the Research and Development Plan

2.2.1 Policy on Structuring the System

Every one of the above is a very tough research task. The first half of the four-year intermediate stage was devoted to reconsidering each part of DUALS-I, as well as to extending the functions and improving the processing speed of the syntactic aspect of the whole system. Also, a new language **CIL** needed to be developed; it allows **partially specified terms** (terms represented as bundles of attribute-value pairs) and provides a lazy evaluation function. The development of **DUALS-II** included a total revision of the foregoing system, in terms of CIL.

DUALS-II is a re-implementation of DUALS-I, but it is based on more general methods and has a more consistent software structure throughout the entire system. This was the starting point for the research on **DUALS-III**, a system that will handle a sample text ten times larger than that of DUALS-II. The structure of such a discourse understanding system was expected to be something like Fig. 1.

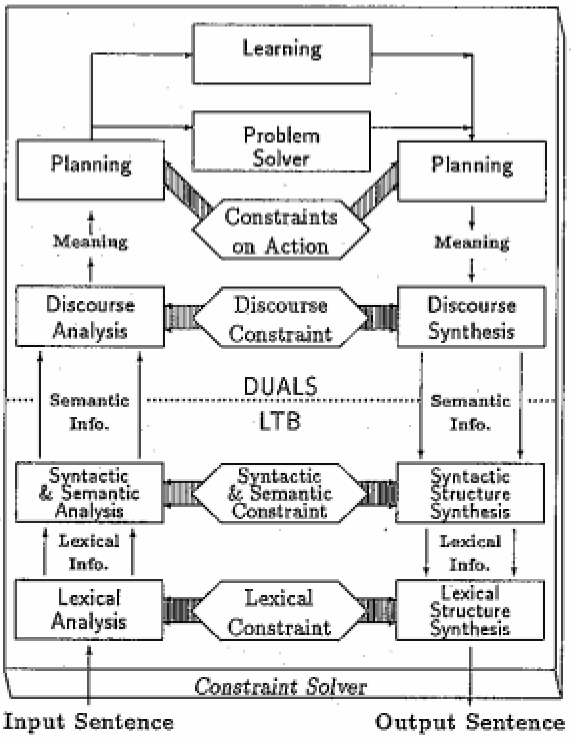


Fig. 1: The Mechanism of DUALS.

The whole system involves the analysis module and the generation module; the former analyzes input sentences, understands them, and produces corresponding discourse structures, and the latter generates response sentences based on the result of problem solving. The processing in each of these modules can be roughly divided into two layers: syntactic and pragmatic. Accordingly, the dictionary and knowledge base for analysis and generation were each regarded as divided into the syntactic part and the pragmatic part.

2.2.2 Syntactic Processing and the General Japanese Processor

Syntactic processing functions, in particular analysis functions, have so far been studied thoroughly not only in the NLP research at ICOT in the initial stage, but also in research and development of machine translation systems all over the world. Since the methods for implementing these functions are comparatively well-known, our main research goal in the syntactic domain was to achieve a higher processing speed. The generation functions have not been studied so well, and hence we planned to implement a software that can generate sentences from an intermediate semantic structure similar to surface sentences. Since the syntactic modules constitute a foundation upon which higher-level functions are piled, and since they can be employed in various types of research on NLP, those modules should be supplemented with, for instance, debugging environments for design and expansion of grammars, so as to form a consistent functional module.

The dictionary was developed in parallel with the lexical analysis of the sample text. The morphological part of the grammar was tailored along the line of a finer-grained approach so as to deal with deep discourse understanding in the future as well. The syntactic part was made so as to cover the sample sentences. For semantic processing, it is important to have a semantic dictionary which contains rules for composing the meaning of words out of morphemes (For example, 'dict' in 'dictionary', 'dictator', and 'contradict' is a morpheme.), lexical property about case-marking, and so forth. The dictionary, the grammar, and those software modules were developed in parallel in order that they could be kept consistent with each other.

It is difficult to maintain consistency between dictionary and grammar, so ours were developed together with their debugging environments. Also the preparation of a corpus of sample sentences and KWIC (key words in context) was considered by which to improve the generality of the description in the dictionary and the grammar. CIL, the language for describing NLP systems, was equipped with a programming environment including a debugger and a compiler, both on a multi-window basis.

The software modules for analysis mentioned so far provide fundamental input-output functions for NLP. They could therefore commonly be employed not only in research on discourse processing, but also in developing natural-language interfaces of expert systems and database systems. We intended to compile those modules into a general Japanese processor (which we call **LTB**: the

Language Tool Box) that would be available not solely in discourse understanding but in a wider domain of research and development.

2.2.3 Contextual Processing

Contextual processing is largely an immature field, and has very little accumulation of achievements in engineering scrutiny. A practical approach must begin by extracting and classifying meaning of words and background knowledge necessary to analyze the subject text and expected questions, and then forming them into a thesaurus and a concept dictionary. Such knowledge is described as 'constraints,' along the line of Situation Semantics.

The research was also intended to cover how to resolve anaphora and ellipsis, how to handle scopes of quantifiers and negation, how to represent the structure of discourse, and so on. This research should be the basis upon which to seek efficient ways to create a system that can understand sentences by using world knowledge and answer questions by problem solving.

Research on this aspect of NLP could be regarded as an attempt to figure out how to cast 'meaning' onto an engineering mechanism. One obstacle is that there is no ready-made structure if we embark on the domain of semantics, and so we are obliged to crawl nearly on a trial-and-error basis, catching at straws scattered by linguistic inquiries. This is totally unlike morphological or syntactic processing, where you can begin with far more refined structures in hand.

Substantial research work on semantics and context must deal with a subject text of a certain degree of complexity, in terms of both the superficial length and the discourse structure. This is because the types of context are practically inexhaustible. Methods for context processing should be examined in a variety of ways upon a variety of sample texts. Research on context processing yields fruits only through comparison across such a wide range of investigation. For this reason we intended to have many groups investigate various texts.

2.3 Organizational Structure of Research and Development

In the FGCS project, the second research laboratory at ICOT and the research groups at cooperating manufacturers are supposed to study various systems for various usages in various situations.

Syntactic processing should meet more strict requirements in terms of both the functional inventory and the processing efficiency. Both the development and the use of LTB, the general Japanese processor, were shared by all these groups. Such sharing of software is easy, because the computer system for research has been unified to PSI/SIMPOS since the beginning of the intermediate stage. LTB is supposed to include also the modules for semantic and contextual processing whose specification has been established clearly enough to share. This is a way to publish successively the concrete achievements of the research; a particularly appropriate way to publish basic research like that of NLP.

DUALS is the central subject of the research at ICOT. ICOT is also supposed to work with research groups at the cooperating manufacturers in the research and development of the common software tools such as LTB. Several research groups at these manufacturers are requested to study semantic processing and contextual processing in dealing with their own sample texts and problem domains.

3 Research at ICOT

3.1 DUALS-II

The research on discourse understanding at ICOT has revolved around the experimental development and extension of DUALS since the initial stage. Fig. 2 illustrates the history of the component methods that have been investigated during this research.

Both DUALS-I and DUALS-II dealt with a sample text extracted from a third-grade elementary school textbook of Japanese. This text, given in Fig. 3, consists of 18 sentences, or about 100 words. There is a great difference in the engaged processing methods and the schemata for software implementation, between DUALS-I, developed in the initial stage, and DUALS-II, developed in the first half of the intermediate stage.

The development of DUALS-I suffered from the shortage of the memory capacity and the poor programming environment available in Prolog on DEC2060. We also had to discover a way to construct syntactic processing along the line of logic programming. The goal to be achieved at that time was to put together the existing methods into a coherent software system, rather than organizing new theories or methods. DUALS-I allows the accumulation of discourse structure through the progress of contextual analysis, and the exploitation of this accumulated knowledge for question-answering.

DUALS-II, though sharing the same basic structure with DUALS-I, engages a largely different set of methods. Between these two versions of DUALS, the grammar was extended from a simple version written in terms of LFG to DCG-representation of Watanabe grammar,

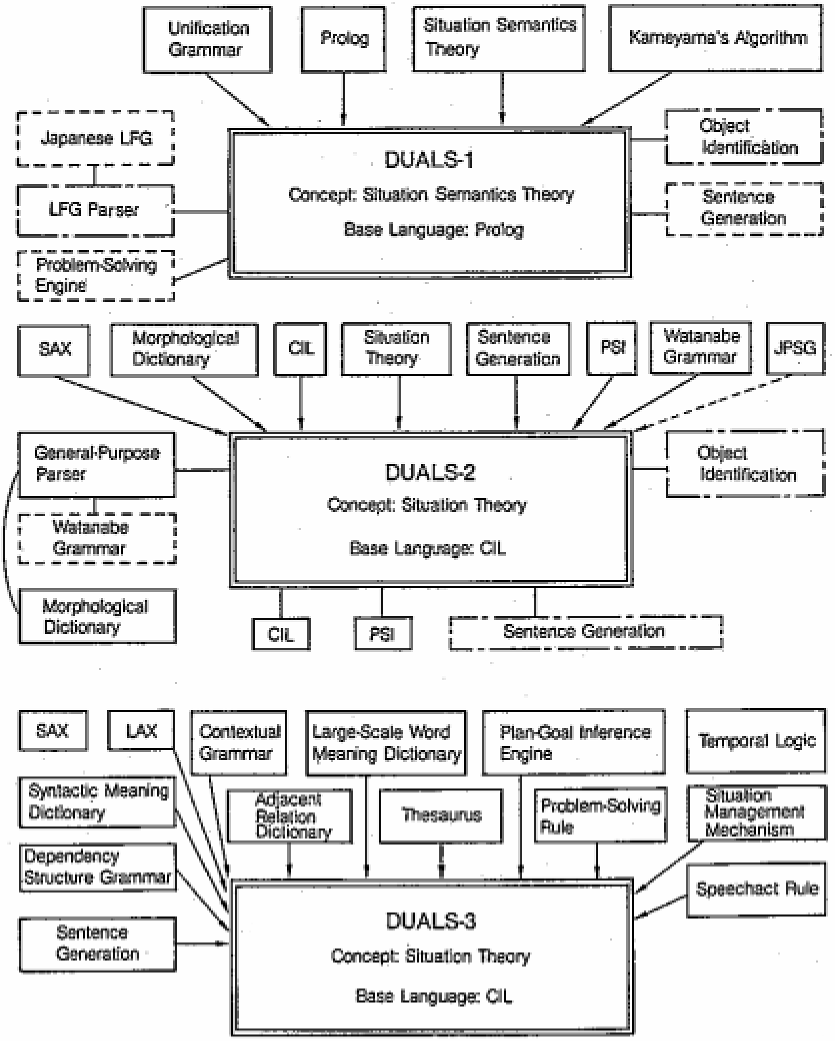


Fig. 2: Progress in DUALS R&D.

while the number of rules increased from 200 to 300. The parser was also revised from BUP, a bottom-up depth-first version, to SAX, a bottom-up breadth-first one. SAX had more functions and worked better. The central parts of semantic processing were improved, being rewritten in terms of CIL, a language for semantic description.

As Fig. 4 shows, CIL is a sort of Prolog, extended to allow partially specified terms (terms as bundles of attribute-value pairs) and a lazy evaluation control (in which an instantiation of a variable triggers the evaluation of the subprograms previously attached to that variable). The partially specified terms of CIL greatly reduce the restriction on the numbers and the locations of arguments in Prolog terms, thus providing a clear view in the description of the system and in the representation of the intermediate semantic data.

The central part of these two versions of DUALS was anaphora resolution, which employed Kameyama's algorithm to find the candidates of antecedents of noun phrases by reference to postpositions. The knowledge representation scheme focused on situations, based on Situation Theory. This representation was referred to by an inference mechanism for problem solving. With the small size of the sample text, however, this mechanism ended up to be a patchwork of ad hoc rules and representations specific to the text. Capturing generality about this aspect of processing was shelved until DUALS-III.

あと一時間でマニラに着こうと言う時、どうしたことか、急にエンジンから白いけむりが吹きだしました。	Only an hour before arriving Manila, suddenly white smoke began to billow from an engine.
これを発見したき長のロールさんは、はっとしました。	Seeing this, Captain Roll was alarmed.
もしも火でも吹こうものなら、ひこうきは、ばく発してしまいます。	If the fire spreads, the plane could explode.
下は、広広と広がる太平洋です。	Below lies the broad expanse of the Pacific Ocean.
そうしたら、五十人の乗客のいのちはどうなることでしょう。	What will become of the 50 passengers?
ロールさんは、急いで、スチュワーデスのふちがみさんをよびました。	Captain Roll hurriedly called a flight attendant, Ms. Hutigami.
そして、いざというときの用意をするようにめいじました。	He instructed her to prepare for an emergency.
ふちがみさんの顔がひきしまりました。	A tension showed up on Ms. Hutigami's face.
「お客様にお知らせするんですか。」	"Shall I tell the passengers?"
「いや、何とかこのままとんでみる。お客様には、知らせないほうがいい。」	"No, I'll try to keep on flying." said the captain, "Do not let the passengers know."
ふちがみさんは、そうじゅう室を出ると、にっこりとほほえみながら、「みなさん、これから、きゅうめいぐをつけるくんれんをいたします。」と言いました。	Ms. Hutigami came out of the cabin and said with a smile, "Ladies and gentlemen, it's time to conduct the life jacket drill."

Fig. 3: The Text for DUALS-I and -II.

$$< PST > ::= \{\alpha_1/\beta_1, \ldots, \alpha_n/\beta_n\}$$

where $n \geq 0$, $\alpha_i \neq \alpha_j$ $(i \neq j)$

```
Example : {relation/lend,
           who/taro,}
           to_whom/hanako,
           what/{ object/car,
                  owner/taro},
           fee/{$/10},
           date/{ year/1988,
                  month/dec,
                  day/1},
           Period/{month/1}}
```

PST is an expansion of Herbrand Term:

$$\begin{array}{ccccccccl} P & (& X & , & Y & , & Z) & \Rightarrow & \{\, arg_0/P, \\ \vdots & & \vdots & & \vdots & & \vdots & & \quad arg_1/X, \\ arg_0 & & arg_1 & & arg_2 & & arg_3 & & \quad arg_2/Y, \\ & & & & & & & & \quad arg_3/Z\} \end{array}$$

Fig. 4: Partially Specified Term.

DUALS-II makes good use of the window system provided by SIMPOS, the operating system of PSI, and provides an efficient interactive interface that supports the Japanese language. Fig. 5 shows the result of a syntactic analysis, and Fig. 6 the internal representation of syntactic and semantic information obtained from this analysis.

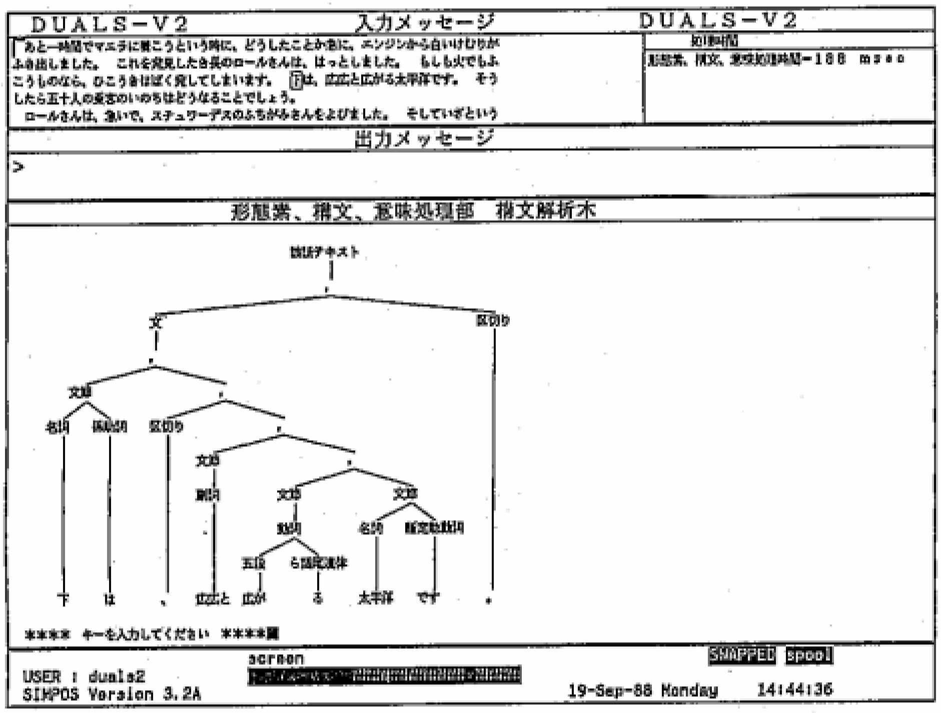
The algorithm and dictionary for syntactic processing in DUALS-II work much faster than those of DUALS-I. They repond to a question with a delay of only a few seconds. For each sentence in Fig. 3, parsing takes 1 to 2 seconds, problem solving 2 to 5 seconds, and generation less than 0.5 seconds.

The program of DUALS-II is about 11,000 lines long. Approximately half of it is written in CIL, of which about 4,000 lines are for morphological, syntactic and semantic analysis, about 500 lines are for handling ellipsis and anaphora, and about 500 lines are for problem solving. The other half of the program is written in ESP, the language which describes the underlying PSI/SIMPOS system. This half consists of about 2,000 lines for sentence generation, about 2,300 lines for CIL interpreter, and miscellaneous utility subroutines. Besides DUALS-II itself, there are peripheral software tools for building grammars and dictionaries. This as a whole constitutes a very large software system.

3.2 DUALS-III

The research on DUALS-III aims at a more proper treatment of verbal context, based upon the methods of processing and software implementation accumulated through the development of the first two versions of DUALS. DUALS-III is expected to deal with a sample text containing about 200 sentences/2000 words, and to answer questions of a much greater variety. Part of the sample text and some of the expected questions are given in Fig. 7.

This enlargement of the system necessitated a new pe-



DUALS－V2　　入力メッセージ　　DUALS－V2

「あと一時間でマニラに着こうという時に、どうしたことか急に、エンジンから白いけむりがふき出しました。　これを発見した機長のロールさんは、はっとしました。　もしも火でもふこうものなら、ひこうきはばく発してしまいます。　下は、広広と広がる太平洋です。　そうしたら五十人の乗客のいのちはどうなることでしょう。
　ロールさんは、急いで、スチュワーデスのふちがみさんをよびました。　そしていざという

処理時間
形態素、構文、意味処理時間＝188 msec

出力メッセージ

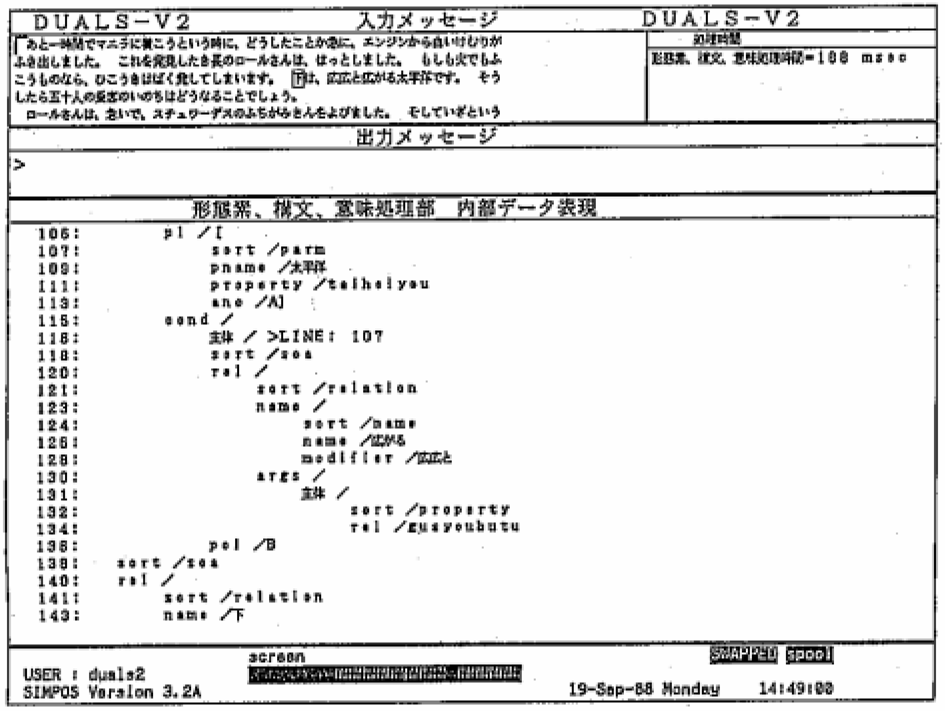
```
>
```

形態素、構文、意味処理部　構文解析木

＊＊＊＊　キーを入力してください　＊＊＊＊

```
                    screen                                   SNAPPED spool
USER : duals2
SIMPOS Version 3.2A                          19-Sep-88 Monday     14:44:36
```

Fig. 5: The Result of a Syntactic Analysis.



DUALS－V2　　入力メッセージ　　DUALS－V2

「あと一時間でマニラに着こうという時に、どうしたことか急に、エンジンから白いけむりがふき出しました。　これを発見した機長のロールさんは、はっとしました。　もしも火でもふこうものなら、ひこうきはばく発してしまいます。　下は、広広と広がる太平洋です。　そうしたら五十人の乗客のいのちはどうなることでしょう。
　ロールさんは、急いで、スチュワーデスのふちがみさんをよびました。　そしていざという

処理時間
形態素、構文、意味処理時間＝188 msec

出力メッセージ

```
>
```

形態素、構文、意味処理部　内部データ表現

```
 106:         pl /[
 107:              sort /parm
 109:              pname /太平洋
 111:              property /taiheiyou
 113:              ane /A]
 115:         cond /
 116:              主体 / >LINE: 107
 118:              sort /soa
 120:              rel /
 121:                   sort /relation
 123:                   name /
 124:                        sort /name
 126:                        name /広がる
 128:                        modifier /広広と
 130:                   args /
 131:                        主体 /
 132:                             sort /property
 134:                             rel /gusyoubutu
 136:              pol /B
 138:    sort /soa
 140:    rel /
 141:         sort /relation
 143:         name /下
```

```
                    screen                                   SNAPPED spool
USER : duals2
SIMPOS Version 3.2A                          19-Sep-88 Monday     14:49:00
```

Fig. 6: The Internal Structure of a Sentence.

ここで、考えてみなければならないのは、自然界には、さまざまな種類の生物たちが、それぞれの環境に応じて生きているということである。そして、これらの生物たちは、互いに影響を与え合い、複雑に絡み合った関係を保ちながら、生活しているということである。

森という環境を例にとってみよう。

森には、いろいろな動物が住んでいる。獣も鳥も虫も、もっと小さな微生物もいる。彼らは、そこに森があるから生活しているといっていい。

森の植物は、動物たちに食物を提供している。昆虫たちは、木や草の葉を食べたり、花の蜜や、木の幹から出る樹液を吸ったりして生活している。小鳥たちのあるものは、木の実や新芽を盛んについばむ。シカやサルは、木の葉や実を好んで餌にする。

What should be considered here is that a variety of creatures live in nature, adapting themselves to their own environments, and that these creatures make their lives influencing each other and keeping complicated relationships.

Let us take forest as an example of the environment.

In a forest, many kinds of animals live, including beasts, birds, insects and much smaller ones. We can say that they can keep their lives because the forest is there.

Plants in the wood feed animals. Insects live by eating leaves of trees and grasses, and sipping honey of flowers and sap coming out of tree stems. Some small birds peck nuts and sprouts vigorously. Deers and monkeys like to eat leaves and nuts of trees.

Fig. 7a: Sample Text for DUALS-III (part).

rusal of syntactic processing. With a problem domain of this scale, it is impossible for one person to capture the entirety of a grammar and a semantic dictionary all through their developmental stages. The maintenance of consistency across the whole system, therefore, requires not only appropriate schemata for classification and arrangement of the relevant knowledge, but also efficient debugging environments for the systems under construction. CIL, which is used in parsing and semantic processing, should run more efficiently and should be equipped with more powerful debugging tools.

We needed to develop new, properly-arranged versions of the thesaurus, concept dictionary, and so on, and to extend the morphological dictionary and semantic dictionary, in order to implement a more powerful semantic processing and to apply methods for context processing to the longer text. The former rules for handling ellipsis and anaphora should be reorganized into a more systematic structure. We need a proper problem-solving mechanism which can work in cooperation with those rules. An appropriate knowledge base must be developed which will be exploited by this problem solver. The corresponding parts of DUALS-II ware very simplistic, for instance with ad hoc inclusion of contextual processing into sentence analysis and generation. The difference between these two versions of DUALS could be looked upon as a typical example of quantity requiring quality.

The most important feature of DUALS-III is the accommodation of ideas from constraint programming and Situation Semantics. A major reason for believing that these theories can cope with contexts is that they are theories that handle partial information particularly well. The most stubborn problems blocking the way to the proper treatment of discourse include context sensitivity, ambiguity, and non-monotonicity, all of which follow from the partiality of information. The approach of DUALS-III and its successor versions, therefore, is an attempt to grasp the root of these ubiquitous problems.

In particular, the constraint-based approach poses an interesting but radical view of a NLP system in which analysis and generation share everything, including not only grammar and dictionary tableware, but even processing software. In other words, such an ultimate constraint-based system has no particular program module for parsing, generation, problem solving, or anything else. Nor does it have, for instance, any grammar strictly for analysis or strictly for generation. Instead it has only one general constraint solver to deal with all the types of constraints, morphological, syntactic, semantic, and pragmatic, along a variety of processing directions including those of analysis and generation. Such a unification between analysis and generation and across the types of constraints gives us a new extended view of the NLP system and is quite beneficial in many respects. To mention only a few of many engineering aspects, first, it would make the inference mechanism smart. Second, it would also bring about a sophisticated classification and structuring of the behavior of the constraint management system, which can process various sorts of constraints (such as a morphological constraint and a pragmatic constraint) simultaneously.

This ultimate constraint-based system is somewhat beyond the scope of DUALS-III, however, and should be reserved for its successors. The alternative intermediate goal of DUALS-III is to apply the constraint paradigm to semantic and pragmatic processing. This goal is to be reflected in the problem-solving module, which is a general constraint solver rather than a procedural description of

誰が考えてみるのですか。	Who should consider?
自然界の生物たちの生活の様子はどうであると言っていますか。	What is said about the lives of creatures in nature?
環境の例として何をとりあげていますか。	What is taken as an example of environment?
動物たちは、なぜ、森に住んでいるのですか。	Why animals live in the forest?

Fig. 7b: Sample Questions for DUALS-III.

how to deal with individual constraints. This problem solver therefore encompasses not only inferences for the comprehension of input sentences, but also inferences for the preparation of the answers to questions. The above ultimate goal is expected to be implemented in the future by extending and improving this problem solver.

Needless to say, to what extent such a constraint-based approach and Situation Semantics could really fit and contribute to practical software construction should become clear only through the experimental development of DUALS-III and the evaluation of its performance. But at any rate, DUALS-III, facing a number of novel problems, is regarded as the real first step of proper scrutiny into discourse understanding. Fig. 8 presents the structure of DUALS-III and component methods employed therein.

3.3 LTB: a General Japanese Processor

LTB consists of a syntactic analyzer, a sentence generator, dictionaries and grammars referred to by them, and a system of CIL, a language for semantic description. LTB is written in CIL and ESP, and runs on PSI-II/SIMPOS.

LTB is a collection of software modules that not only are in charge of syntactic processing for DUALS-III, but also are intended to serve for a wider variety of usage. Many aspects such as the classification of vocabulary and semantic representation, however, do not yet have established methods. Those parts of LTB are being devised, both the linguistic generality and the specific requirements raised by DUALS-III being taken into consideration.

Since the dictionaries and grammars are to be improved and extended iteratively in the course of their development, systems relating to them also include debugging environments. However, the debugging environments of the processing modules of LTB, having been developed separately, are different in the module interfaces and the ways they operate. Besides, they are rather insufficiently documented. Both the standardization of them and the documentation are currently in progress. A unified operative environment called **LTB-Shell** is also being developed, which provides the user with facilities for combining the processes of LTB modules in a variety of ways. The present structure of LTB is illustrated in Fig. 9.

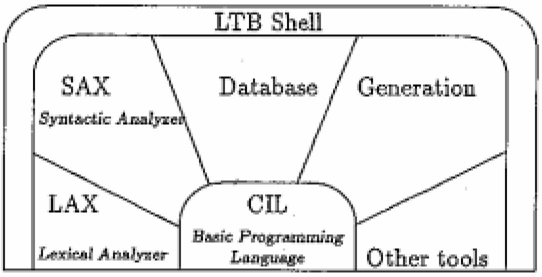


Fig. 9: The Structure of LTB

The major components of LTB are described below.

3.3.1 Dictionary

The dictionary is required to have as wide a coverage of vocabulary as possible, and at the same time to contain deep knowledge. The dictionary in LTB, called the master dictionary, is currently limited to a vocabulary of about 4,000 words, and the emphasis is placed on the description of deep knowledge relating to the sample text that DUALS-III is to work on.

The content of the master dictionary includes syntactic information of words and phrases such as the parts of speech and the types of inflection, and semantic information such as thesaurus codes and semantic constraints. The semantic properties of words are divided into three types. The first type concerns several hundreds of function words, for which the semantic properties are specified in a direct correspondence to the syntactic functions those words perform. The second type concerns those words whose semantic properties are defined in terms of primitive concepts: approximately 2,000 verbs and adjectives of Japanese origin. The words dealt with in the third type of description have their meaning specified in relation to the concepts of the other words.

The master dictionary is to be improved in terms of both the vocabulary and the depth or precision. This extension is planned to be based upon a knowledge-base

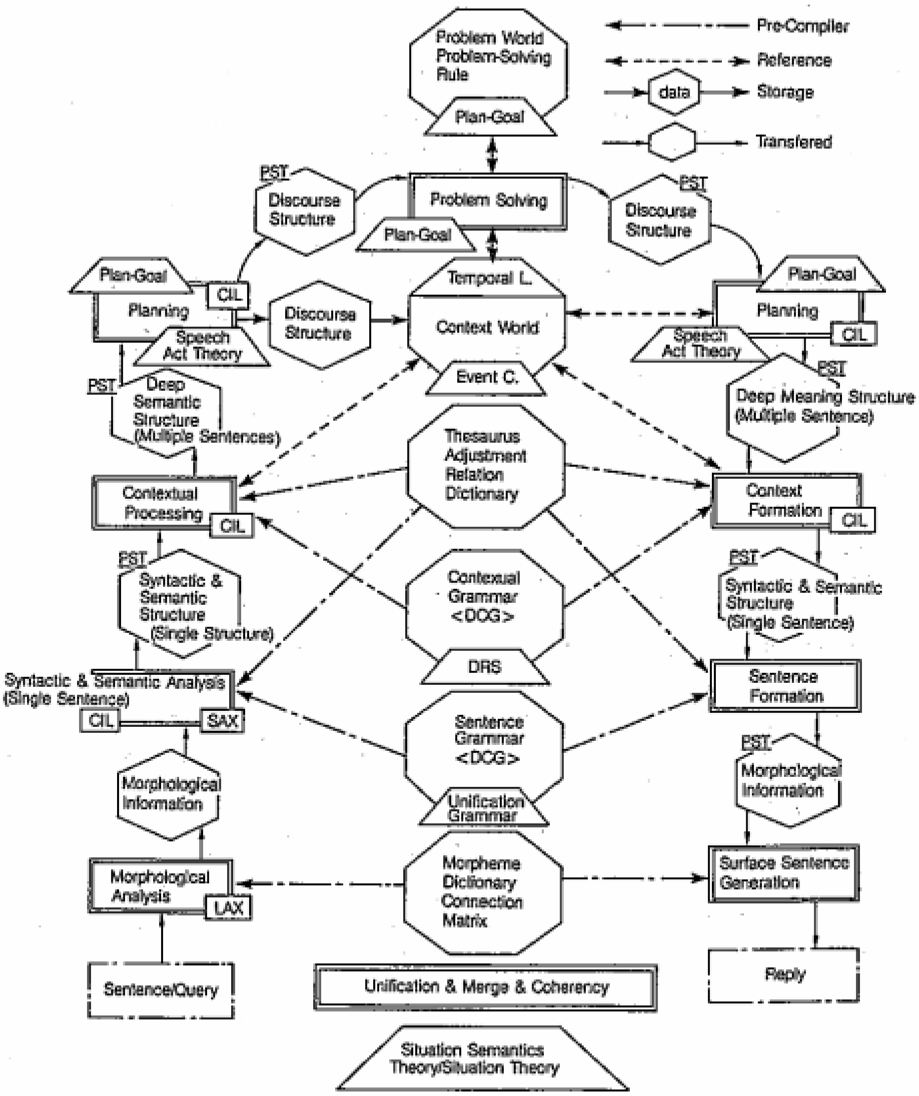


Fig. 8: The Structure of DUALS-III.

management system called Kappa, which is being developed separately from LTB.

3.3.2 Grammar

The morphology grammar is formulated according to Morioka grammar. The purely morphological part of the grammar is described by a regular language, and the semantic part is formulated so that semantic composition should be carried out by unification. The version for analysis includes about 1,000 entries of dependent words, about 3,000 lines of coding in extended DCG plus CIL, and about 1,000 entries, amounting to about 30,000 lines, of independent words. The version for generation is expected to have approximately 4,000 entries.

The syntax grammar is described in DCG, and the corresponding semantic composition is again supposed to be processed by unification. Its version for analysis has about 200 rules and about 1,000 lines of extended DCG program. The version for generation is supposed to include about 100 rules, although the number of rules depends on the format of internal semantic representation.

Obviously, both the analysis version and the generation version of a grammar should preferably be derived from one and the same description, as pointed out earlier. In this respect, a master grammar of morphology is currently being developed that will unify information about part of speech, inflection, connection between words, and so on.

3.3.3 Programming Language

CIL may be considered to be a general-purpose knowledge representation language, because of the special data structure and the lazy evaluation control it provides.

As already described, CIL deals with **partially specified terms (PSTs)** as a basic data structure, and allows successive augmentation of attribute-value pairs in PSTs through unifications. Thanks to this feature, programmers do not have to worry about the positions and the numbers of arguments as they have to in the case of Prolog. PSTs are appropriate for representing parse trees and semantic structures, since they permit recursive occurrences of PSTs as values for attributes.

Another important feature of CIL is a sort of daemon mechanism which invokes the previously specified subprograms upon the instantiation of the variable to which the program was associated. This lazy evaluation mechanism allows declarative description, thus providing a better perspective of programming. It renders debugging much more difficult, however, and debugging functions must hence be enriched accordingly.

CIL is executed after being compiled down to ESP. Speeding up the execution requires some invention. The current system has acquired a practical efficiency, through iterative optimization of the compiler and improvements to the implementation of execution-time routines. This system includes a debugger, and provides the top-level programming interface of LTB.

3.3.4 Morphological and Syntactic Analysis

The module for morphological analysis is called **LAX**. This module consists of a grammar editor to tailor a regular grammar, a transducer which transforms this grammar into an execution form, and an analyzer which executes the analysis based on it. The analyzer looks up a dictionary organized in TRIE structure. The analyzer uses a breadth-first method called the layered stream, to search for the solution. This method is consistent with parallel processing.

The parsing module, which is called **SAX**, consists of a grammar editor to write a grammar in terms of DCG, a transducer to transform this into an execution form, and an analyzer to execute parsing. The DCG format employed here allows CIL programs in extraconditions. As in LAX, the search of the solution exploits the layered stream method.

Both the morphological analysis system and the syntactic analysis system are equipped with their own debugger, which enables easy revision of grammars.

3.3.5 Text Generation

The generation module includes an editor to handle a grammar containing tree-rewriting rules, a transducer of this grammar, a tree-rewriting system, and a morphological processor. Given an intermediate representation of the syntactic structure of a Japanese sentence to be generated, this module produces a surface sentence. Just like the analysis module, the generation module has its own debugger, which makes the grammar easy to revise.

4 Research at Cooperating Manufacturers

As has been mentioned, several groups at the cooperating manufacturers are also conducting related research at the request of ICOT. This research is briefly described below.

4.1 An Intelligent Text-information Retrieval System [21 to 25]

This research aims verify and advance methods for natural language understanding and for inference using the result of understanding texts. For this purpose, an intelligent text-information retrieval system is being developed which will retrieve parts of a text that semantically match against the query. The system accepts a Japanese sentence as a query, and analyzes its semantic structure. Next, it infers a keyword retrieval command on the semantic level to get the passages in the given text which have similar semantic structure to that of the query. Then, the system analyzes the meaning of the retrieved parts of the text, compares them with the meaning of the query, and judges the relatedness between them. Finally, the system shows the retrieved parts of the text in the order of decreasing relatedness. Inference on the semantic level makes it possible to achieve better retrieval than using only the lexical connection such as in a word-thesaurus.

4.2 A Knowledge Representation System for Understanding Discourse and Meaning [26, 27]

This project studies ways to represent and use world knowledge for understanding conversational contexts. The world knowledge is represented in terms of "frames" and "rules." Input sentences are processed by the inference mechanisms. As a specific problem of understanding utterances, that of resolving ambiguities for anaphoric references is being studied.

To understand input sentences, the system extracts candidates for referents of the input expressions from the preceding context, and selects consistent candidates. It then examines their preference by observing causal relations among events in the task domain. The algorithm has been implemented as an experimental system on PSI, with "guidance for VTR operation" as a tentative task.

4.3 A Dialogue Model based on Situation Semantics [28 to 31]

Through the development of a dialogue model, the key technologies to establish man-machine dialogue using natural language (Japanese in this case) are being investigated. They are the utterance recognition and the cooperative response functions, both based on a planning mechanism. A principle in cooperative dialogue has also been studied and tailored into rules. By this strategy, some mechanisms are expected to be designed which will extract meaning from the utterances in a daily conversation (particularly the meaning related to indirect speech acts), and also for cooperative response generation in a dialogue focusing upon the user's final goal.

The current stage is that of software development in CIL on PSI-II. In parallel with this task, modelling of dialogue scenes in connection with the situation semantics is also studied.

4.4 A Task-Oriented Dialogue Understanding System [32 to 34]

This project aims to develop a theory of task-oriented dialog understanding and to design and implement intelligent natural language interface systems. The work focuses on the representation and management of contextual information in discourse and its use for purposes including the following.

- Understanding anaphoric expressions
- Interpreting the semantic contents of user's utterances
- Planning responses to the user
- Controlling and guiding the dialog (including the selection of the topics)

The work has strong connections with Situation Theory. The situation to represent the contextual information about an utterance is called "utterance situation," which contains:

- Who the addressor is
- Sho the adressee is
- What the uttered expression is
- What the situation described by the utterance is
- What the background of the utterance is

A knowledge representation language called LAST provides the facilities to represent situations and to make inferences on situations. Another important work is the research on identifying a "user model."

4.5 A Summarization Support System based on World Knowledge [35 to 38]

To realize intelligent communication facilities, context processing and inference based on knowledge are important. This research focuses on a semantic representation model and a summarization support system. The semantic representation model uses an intermediate language to represent contextual information. The summarization support system uses the following four components.

- World knowledge compiled as a thesaurus, a hierarchical structure of frames, is exploited for anaphora resolution.
- The context processing creates frames of contents which embeds the resulting anaphoric relations.
- A contextual structure is constructed out of the relationships among simple sentences.
- The summarizing process applies evaluation rules to the created representations, and abstracts the parts that are regarded as important.

The resultant summary is presented to the user in the form of a text, a summary-table or a picture.

4.6 A Support System for Generating Controlled Japanese Texts [39]

Natural language, Japanese in particular, accompanies a great degree of ambiguity. In order to resolve ambiguities in texts, a prototype system, "A Support System for Generating Controlled Japanese Texts." has been developed.

This system consists of a morphological processing module, a syntactic processing module and a semantic processing module. Each such module presents existing ambiguities to the user in the form of internal representation displayed on a CRT.

Syntactic analysis is based on KAKARIUKE relation, in which sentences are analyzed in terms of relationships between modifiers and heads. One of the KAKARIUKE rules says that a modifier should modify the nearest head.

A unique and the most plausible sentence structure could be generated by exploiting the knowledge-base and user's instructions. The semantic processing module is currently being installed.

5 Final Remarks

We decided to use NLP as a way to create an intelligent interface. In the initial stage of the FGCS project, we intended to integrate NLP with phonetic, visual, and other types of processing. A re-examination of the plan at the beginning of the intermediate stage of the project, however, showed us that this integration was difficult to implement. As for NLP itself, we recognized that there were too many problems to be solved before we could use an NLP system to create a practical interface. The intermediate stage thus restricted itself to research on discourse understanding in Japanese.

The development of a large-scale dictionary was entrusted to Electronic Dictionary Research Laboratory (EDR), which was established at the beginning of the intermediate stage. EDR is thus in charge of the research and development of dictionaries of Japanese, English, technical terms, and so on, consisting of several hundred thousand entries. ICOT decided not to pursue such a wide coverage of vocabulary, but instead to focus on semantic and contextual processing. The contribution of ICOT in the joint research with EDR should be to elucidate the problems the users of the dictionary face. Problems that relate, for example, to the representation schemata and concept classification for building the concept dictionary and thesaurus.

The research on discourse understanding enjoys little heritage of achievements from the foregoing engineering inquiries. For instance, the research on machine translation has not so far regarded discourse as the central subject of study. We were thus obliged to accumulate by ourselves research achievements concerning semantic representation based on Situation Semantics, compilation of dictionaries reflecting linguistic studies on vocabulary classification, systematic software for syntactic processing, and so on. This is how we have come to be ready for a proper study on contextual processing, and how concrete outcomes such as LTB have been brought about.

Starting with a reorganization of DUALS-III, the research of its descendant versions in the final stage will be aimed at establishing more sophisticated methods of processing ellipsis and anaphora, of knowledge representation and problem solving for deeper semantic processing, and particularly of handling verbal communication in terms of speech acts. All of these methods should be accommodated in a declarative format of constraints, rather than procedures. The constraint-based implementation should yield an exponential augmentation of the coverage of DUALS. Besides, the coverage of the system will be augmented by increasing the computational efficiency by means of parallel processing on PIM/PIMOS, a parallel inference machine. Kappa, a knowledge-base management system, might also be employed in order to cope with the greater amount of knowledge accompanying the extension of the grammar, dictionary, and knowledge-base of the domain world.

The software of LTB will also be revised in terms of KL1, a parallel logic programming language, to cope with increasing computational complexity. In order for LTB to handle a greater amount of knowledge, a knowledge-base management system should be used on a PSI machine for the time being. It is planned that LTB should eventually engage a knowledge-base management system to be developed on the parallel inference machine.

Our NLP research is in a sense an integration of all the research about knowledge information processing. It exploits as tools the software and hardware products of the FGCS project, in order to replicate the human linguistic behavior with the greatest possible precision. This research will therefore be a comprehensive benchmark test for the FGCS project.

Acknowledgements

The NLP research in the FGCS project is being conducted jointly by many researchers at ICOT, the cooperating manufacturers, and universities, among others. Thanks are firstly due to those who have given support and helpful comments, including Dr. Fuchi, the director of the research laboratories at ICOT, Mr. Yokoi, the former chief of the second research laboratory at ICOT and the current director of EDR, Prof. Tanaka at Tokyo Institute of Technology, who is also the chairman of the natural language working group at ICOT, and Mr. Takizuka, a former researcher at the second research laboratory at ICOT and presently a researcher at KDD Kamifukuoka R&D Laboratories. Special thanks go to many people at the cooperating manufacturers in charge of the joint research works: Dr. Amano and Mr. Ukita at Toshiba Co., Mr. Sugiyama and Mr. Akiyama at Fujitsu Limited, Mr. Dazai and Mr. Kondoh at Mitsubishi Electric Co., Mr. Komorida and Mr. Yasukawa at Matusita Electric Industrial Co., Mr. Obuchi and Mr. Miyoshi at Sharp Co., and others.

References

[1] Yokoi, T., Mukai, K., Miyoshi, H., and Tanaka, Y. "Research Activities on Natural Language Processing of the FGCS," *Proc. of FJCC'86*, 1986.

[2] Kimura, K., Sugimura, R., Takizuka, T. and Mukai, K. "Danwa Rikai Jikken System DUALS dai 2-han no Sekkei to Jissou (Design and Implementation of Discourse Understanding System DUALS-V2 *in Japanese*), *Proc. of the 3rd National Conference of Japan Society for Software Science and Technology*, pp.33-36, Tokyo, 1986

[3] Matsumoto, Y. and Sugimura, R. "A parsing system based on Logic Programming," *Proc. of the 10th IJCAI*, 1987.

[4] Mukai, K. and Yasukawa, H. "Complex Indeterminates in Prolog and their Application to Discourse Models," *New Generation Computing, 3*, OHMUSHA, Ltd. and Spring Verlag, 1985.

[5] Mukai, K. "A system of Logic Programming for Linguistic Analysis Based on Situation Semantics," *Proc. of the Workshop on Semantic Issues in Human and Computer Languages.*, CSLI, 1987.

[6] Mukai, K. " Partially Specified Term in Logic Programming for Linguistic Analysis," *Proc. of FGCS'88*, 1988.

[7] Okunishi, T., Sugimura, R., Matsumoto, Y., Tamura, N., Kamiwaki, T., and Tanaka, H. "Comparison of Logic Programming Based Natural Language Parsing Systems," *Natural Language Understanding and Logic Programming, II*, Dahl, V. (ed.), North-Holland, pp. 1-14, 1988.

[8] Sugimura, R. "Japanese Honorifics and Situation Semantics," *Proc. of the 11th COLING*, pp. 507-510, 1986.

[9] Sugimura, R., Miyoshi, H., and Mukai, K. "Constraint Analysis on Japanese Modifying Relations", *Natural Language Understanding and Logic Programming,II*, North-Holland, pp. 93-106, 1988.

[10] Hasida, K. "Dependency Propagation: A Unified Theory of Sentence Comprehension and Generation," *Proc. of the 10th IJCAI*, pp. 664-670, 1987

[11] Hasida, K. "Izondenpa (Dependency Propagation, *in Japanese*)," *Proc. of the 29th Programming Symposium*, pp. 147-158, 1988.

[12] Sakai, K. and Sato, Y. "Boolean Gröbner Bases," *ICOT Technical Memo No.488*, 1988.

[13] Yamasaki, S., Sugimura, R., Akasaka, K., and Matsumoto, Y. "Koubun Kaiseki System SAX no Debug Kankyou (Debugging environment of SAX system, *in Japanese*)," *Proc. of the 2nd Annual Conference of Japanese Society for Artificial Intelligence*, pp. 411-414, 1988.

[14] Sugimura, R., Akasaka, K., Kubo, Y., Sano, H., and Matsumoto, Y. "Ronri-gata Keitaiso Kaiseki LAX (Logic Based Lexical Analyzer LAX, *in Japanese*)," *Proc. of the Logic Programming Conference '88*, ICOT, pp. 213-222, 1988. (English version will appear in *The Lecture Notes on Computer Science*.)

[15] Ikeda, T. et. al, "Sentence Generation in LTB (*in Japanese*)," *Proc. of the 5th National Conference of Japan Society for Software Science and Technology*, 1988.

[16] Tanaka, Y. and Yoshioka, T. "Overview of the development of dictionary and Lexical Database," *Proc. of FGCS'88*, ICOT, 1988.

[17] Takizuka, T. and Sugimura, R. "LTB Shell no Kousei (Configuration of LTB Shell *in Japanese*)," *Proc. of the 37th National Conference of Information Processing Society of Japan*, pp. 1074-1075, 1988.

[18] Uchida, S. (ed.) *ESP Guide*, ICOT TM-338, Aug. 1987.

[19] Uchida, S. "Inference Machines in FGCS Project," *Proc. of International Conference IFIP TC-10, VLSI'87*, Aug. 1987, also ICOT TR-278.

[20] Yokota, K. Kawamura, M., and Kanaegami, A., "Overview of the Knowledge Base Management System KAPPA," *Proc. of FGCS'88*, ICOT, 1988.

[21] Sugiyama, K., Akiyama, K., Ibuki, K., Kawasaki, M. "IRIS: An Intelligent Information Retrieval System based on Natural Language Understanding" (in Japanese), *IPS Research Report of Natural Language Research Group*, Information Processing Society of Japan, Nov. 1986.

[22] Akiyama, K., Sugiyama, K., Itoh, H. and Onodera, H. "Initial study on transportability of IRIS (An intelligent information retrieval system)" (in Japanese), *Proc. of the 35th National Conference of Information Processing Society of Japan*, pp. 1429-1430, Sep. 1987.

[23] Ibuki, J., Sugiyama, K., Tamada, I. and Kawasaki, M. "Natural Language for Content Retrieval" (in Japanese), *Proc. of the 4th National Conference of Japan Society for Software Science and Technology*, pp. 347-350, 1987.

[24] Kiyama, K. "The Issue of Intelligent Information Retrieval to Textbase," (in Japanese), *IPS Research Report of Database Systems Research Group*, Information Processing Society of Japan, Mar. 1988.

[25] Akiyama, K. "Generation Method Keyword Retrieval Commands in IRIS" (in Japanese), *Proc. of the 37th National Conference of Information Processing Society of Japan*, Sep. 1988.

[26] Kinoshita, Sano, Ukita, Sumita, and Amano "Knowledge Representation and Reasoning for Discourse Understanding," *Proc. of the Logic Programming Conference 1988*, pp. 205-212.

[27] Ukita, Sumita, Kinoshita, Sano, and Amano "Preference Judgement in Comprehending Conversational Sentences with Multi-Paradigm World Knowledge," *Proc. of the FGCS'88*, 1988.

[28] Shimada, H., Kondoh, S., Dazai, T. "A Dialogue Mechanism in IDS (Intelligent Dialogue System)" (in Japanese), The Institute of Electronics Information and Communication Engineers, 1986.

[29] Kondoh, S., Shimada, H., and Dazai, T. "Planning Mechanism based on Dialogue Model (in Japanese)," *Proc. of the 33th National Conference of Information Processing Society of Japan*, pp. 1207-1208, 1986.

[30] Kondoh, S., Imamura, M. "Cooperative Responses based on Dialogue Model" (in Japanese), Information Processing Society of Japan, 1988.

[31] Imamura, M., Kondoh, S., Dazai, T. "A Dialogue Model based on Means-Ends Analysis" (in Japanese), *Proc. of the 36th National Conference of Information Processing Society of Japan*, pp. 1201-1202, 1988.

[32] Motoike, S., Noguchi, N., Yasukawa, H. "The Use of Circumstantial Information in a Task-Oriented Dialogue" (in Japanese), *Proc. of the 5th National Conference of Japan Society for Software Science and Technology*, 1988.

[33] Noguchi, N., Takahashi, M., Yasukawa, H. "Generating Natural Language Responses Appropriate to Conversational Situations — In the Case of Japanese," in Furukawa, K. et al. (Eds.) *Logic Programming '87: Proceedings of the 6th Conference (LNCS Vol. 315)*, Springer Verlag, 1988.

[34] Yasukawa, H., Suzuki, H., Noguchi, N. "Knowledge Representation Language based on Situation Theory," *Proceedings of France-Japan Artificial Intelligence and Computer Science.*

[35] Kita, Komatu, Yasuhara "Summarization support system COGITO" (in Japanese), *IPS Research Report of Natural Language Group*, Information Processing Society of Japan, 1986.

[36] Komatu, Katoh, Yasuhara, Shiino "Summarization support system COGITO — Structural analysis of text" (in Japanese), *IPS Research Report of Natural Language Group*, Information Processing Society of Japan, 1987.

[37] Komatu, Katoh, Yasuhara, Shiino "Summarization Support System COGITO — Summarizing Module" (in Japanese), *Proc. of the 36th National Conference of Information Processing Society of Japan*, 1988.

[38] Katoh, Komatu, Yasuhara, Shiino "Summarization Support System COGITO — Man-machine Interface module" (in Japanese), *Proc. of the 36th National Conference of Information Processing Society of Japan*, 1988.

[39] Obuchi, Y., Hamada, A., Miyoshi, H., and Akiyama, H. "Standardizing Japanese Grammar and its Support System" (in Japanese), *Proceedings of the 36th National Conference of Information Processing Society of Japan*, 1988