TR-I-0036

An Integrated Linguistic Database Management System

言語データベース統合管理システム

Kentaro Ogura　Kazuo Hashimoto　Tsuyoshi Morimoto
小倉健太郎　橋本一男　森元逞

1988.8

Abstruct

This paper describes a linguistic database management system which connects object-oriented representation and a relational database(RDB), synergistically deals with copious multi-aspect linguistic data and provides natural language understanding researchers with a user-friendly interface. Natural language understanding research requires a linguistic database which includes multi-aspect information from linguistic phenomena. Requirements for linguistic data can be classified into two types, i.e., micro and macro analysis. Micro analysis of text in detail requires a complex multi-aspect linguistic data search. Macro analysis of global differences between different field linguistic phenomena requires a large database. To achieve two quite different research requirements this database system is arranged as a vertically distributed database system. In workstations, object-oriented representation is adopted to easily effect complex data searches. High performance workstations are used to effectively provide user-friendly interfaces for researchers. In a host computer, RDB which is extended to manipulate multi-value is adopted to treat a large amount of linguistic data and to get statistical linguistic data.

ATR Interpreting Telephony Research Laboratories
ATR 自動翻訳電話研究所

1. Introduction

Recently, an object-oriented database[1][2][3][4][5] has begun to receive considerable attention, but there is not a reliable object-oriented database system to treat a large amount of data. A relational database(RDB)[6][7][8] is the main database used today, but it is pointed out that relational representation is weak in its ability to represent complex objects. Furthermore, it can not treat the "is-a" relationship while object-oriented representation can.

We research into automatic telephone interpretation[9], which enables a person speaking one language to communicate readily by telephone with someone speaking another. In such research, analysis of natural language phenomena by using linguistic data is very important. Linguistic data are sometimes analyzed in detail to understand the use of words and idioms, etc. and sometimes globally for trends. As a result a linguistic database management system (DBMS) must be able to deal with complex data structures as well as a large amount of linguistic data, and also provide a user-friendly interface.

This paper describes a linguistic DBMS which connects object-oriented representation and RDB, synergistically deals with copious multi-aspect linguistic data and provides natural language understanding researchers with a user-friendly interface.

Object-oriented representation is used because it easily represents the complex relationship between each element of linguistic data. RDB is adopted to treat a large amount of linguistic data and to get statistical linguistic data.

A linguistic DBMS which cannot be used easily, will not be used at all. Thus the human-machine interface is important. To effectively use linguistic data, three functions are provided for the human-machine interface, i.e. “search”, “comparison” and “user definition”, to analyze linguistic phenomena. High performance workstations are adopted for the human machine interface. To implement a user-friendly human machine interface, these workstations provide many facilities such as window, menu, mouse, etc. They also utilize object-oriented programming language[10][11]. A human-machine interface based on object-oriented programming language is easily implemented.

To achieve a detailed, multi-aspect analysis of linguistic data, a linguistic database must at least contain basic information, i.e. word information, modification relationships and multilingual comparison data. As a result the linguistic data can be analyzed from the linguistic structure aspect (whole-part aspect), the semantic and syntactic aspect and the multilingual aspect. With research into automatic telephone interpretation, multilingual comparison data is especially required.

Global analysis of linguistic data provides statistical linguistic data. Statistical linguistic data is useful in using probability to create grammar, which promises to connect speech recognition technology with natural language processing technology. In order to acquire statistical linguistic data, compile dictionary or grammar and create a knowledge base to understand natural language, a linguistic database must contain a large amount of linguistic data.

The study of Corpus Linguistics[12] could be divided into two groups. One deals with techniques of compiling and analyzing large computer corpora. The other deals with the exploitation of existing corpora. This study is categorized as the third paradigm of Corpus Linguistics to provide an environment for efficient use of linguistic data.

Most linguistic corpora involve only word information and most linguistic DBMS also manipulate only word information[13]. Brown Corpus[14] is one such corpus. Some linguistic corpora include syntactic information. The CCPP (Computer Corpus Pilot Project) corpus[15] is one such corpus. Some linguistic corpora include only word (concept) comparison[16]. As a Japanese corpus, one compiled by Japan's National Language Research Institute[17] is famous, but is simple linguistic data and has a simple management system. The size of the corpora are as follows.

Brown Corpus	1,000,000 words
Lancaster-Oslo-Bergen Corpus of Written British (LOB)	1,000,000 words
London-Lund Corpus of Spoken English(LLC)	170,000 words
Japan's National Language Research Institute Corpus	5,000,000 words
CCPP Corpus	130,000 words

In Section Two linguistic data treated by an Integrated Linguistic DBMS is described. In Section Three, requirements for an Integrated Linguistic DBMS are presented. Finally, an Integrated Linguistic DBMS is discussed in detail.

2. Linguistic Data

The three main elements of language are syntax, semantics and pragmatics. To treat the relationship between languages, a fourth factor, comparison, must be considered. Word information is, as syntax, the most basic and important information. The modification relationship is, as semantics, the most basic and important information and also important as syntax. Pragmatics, though important, is not determined yet, i.e., what information is required for pragmatics is not yet clear. Thus under the present conditions it is better not to include pragmatic information in a linguistic database. Pragmatic information must be collected by using a linguistic database.

What information must be in a linguistic database is discussed.

2.1. An Outline of linguistic data

Figure 1 shows the structure of linguistic data.

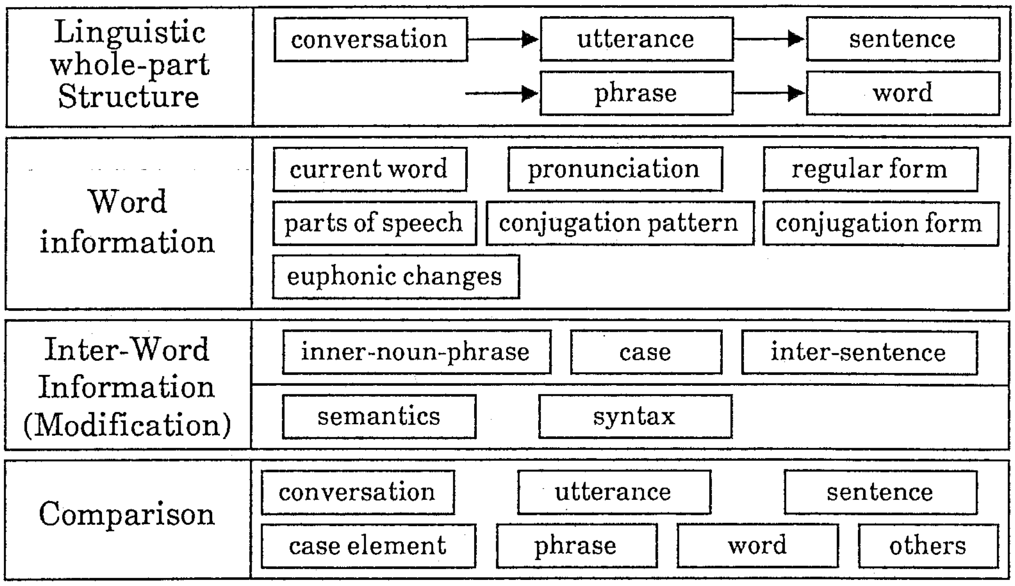

Figure 1: Linguistic Data Structure

To cover all linguistic expression patterns and gather statistical information from the linguistic data, a decision was made to collect 1 million words of linguistic data.

2.2. Details of linguistic data and their aim

(1) Linguistic whole-part information

Linguistic data has a whole-part structure. For example "word", "phrase(*bun-setsu*)", "sentence", "utterance", "conversation", etc. The order in each level is also important in linguistic data. Figure 2 shows the linguistic whole-part structure.

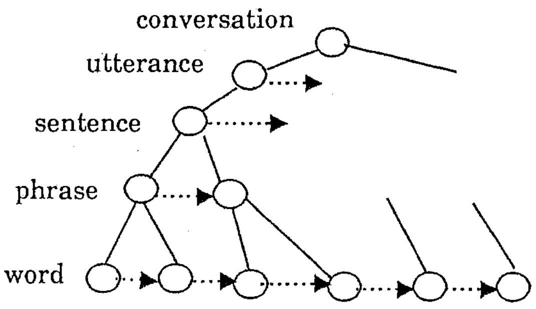

Figure 2: Linguistic whole-part structure

(2) Word information

For research into a language which does not clearly separate words in sentences, word partition information is very important information. Word information is the most basic and important information. This information can be used to compile a grammar and dictionary with probability. Based on these, any other information can be retrieved.

As word information, (a) current word, (b) pronunciation, (c) regular form, (d) parts of speech (e) conjugational pattern, (f) conjugational form, (g) euphonic changes, (h) meaning, (i) super concept, etc. must be provided. (h) and (i) are not involved in our current linguistic data. Details of word information are shown in Appendix 1. Parts of speech and concepts have hierarchies. Figure 3 shows parts-of-speech hierarchy. A complete list of parts-of-speech is shown in Appendix 2.

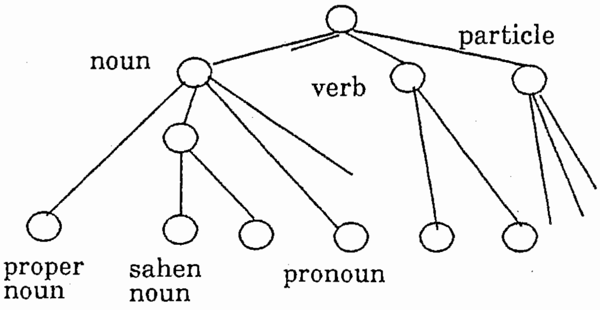

Figure 3: Parts-of-speech Hierarchy

(3) Modification Relationship

The modification relationship shows the semantic and syntactic relationship between modifier and modificand. This information is useful in compiling or modifying a dictionary for a computer. This information can be used to analyze and generate case structures, inner noun phrase structures and sentence structures. Modification relationships can also be used to build a knowledge base (nominal and verbal concept hierarchies).

A semantic relationship is a relationship between a modifier concept and a modificand concept . Some semantic relationships are deep case relationships such as "agent", "object", "goal", etc. Some semantic relationships are not case relationships such as "whole-part", "attribute-object", etc. sixty-three semantic relationships (40 for case relations) are provided. Appendix 3 shows a list of modification semantic relationship.

Five syntactic relationships [(a) Ordinary Case Relationship, (b) Adjective Modification Relationship, (c) Adjective Case Relationship, (d) Adverbial Modification Relationship, and (e) Sentence to Sentence Modification

Relationship] are provided. Appendix 4 shows examples of five syntactic relationships.

(4) Multilingual comparison data

Multilingual comparison data is important for research into translation from one language to another. A transfer dictionary using this data can be compiled. To do this, a several-level comparison must be provided. "Conversation", "utterance", "sentence", "case element", "phrase", "word" level comparison data are provided. Level free comparison data is also provided.

2.3. Theory independence

This linguistic data does not depend on a specific linguistic theory. This linguistic data is designed so that every natural language researcher and linguist can use it. If the data depended on a specific linguistic theory, very few researchers could use it. As syntax, this linguistic data almost depends on school grammar, the general linguistic theory familiar to everyone. As semantics, this data uses a semantic relationship system which has worldwide semantic relationships as core semantic relationships. If a researcher requires linguistic data which depends on a specific linguistic theory, this database system provides him with user-definition facilities to create his own data from the database.

3. Requirements for the Integrated Linguistic DBMS

Requirements for the Integrated Linguistic DBMS to efficiently use linguistic data described above are as follows:

(1)Requirements for human-machine interface:

(A) Quality requirements:

(a) complex data manipulation
(b) "is-a" hierarchy
(c) combinations of various linguistic information
(d) statistical linguistic data
(e) multi-language manipulation

(B) Convenient for users (user workload reduction):

(a) easy linguistic data manipulation (menu etc.)
(b) clear display of linguistic data and search results
(c) to easily specify linguistic data search conditions

(C) To get many users:
- (a) to support from naive users to professional
- (b) user customizing

(2)Requirements for data storage and maintenance:
- (a) a large amount of linguistic data
- (b) easy data modification
- (c) security
- (d) integrity

(1)(A) are mainly requirements for linguistic data. The others are general requirements for DBMS and not specific to linguistic data, but are also very important requirements.

4. Integrated Linguistic DBMS

4.1. Construction of Integrated Linguistic DBMS

Retrieval requirements for linguistic data can be classified two types: one for micro analysis, the other for macro analysis. Micro analysis analyzes features in specific texts. This type of analysis requires a short but complex database search. In this type of analysis some linguistic data is looked at from various aspects. On the other hand, macro analysis analyzes global differences between different fields of linguistic phenomena. This type of analysis requires a full, or at least significant, database search.

Figure 4 shows a layout of the Integrated Linguistic DBMS. This system is given linguistic data from a collecting system. The linguistic data collecting system collects and analyzes texts. In the analysis process a text is analyzed half-automatically.

To satisfy two user requirements, the Integrated Linguistic DBMS is designed to be a vertically distributed database system. High performance workstations provide an environment for complex linguistic data searches and a user-friendly interface for researchers. Object-oriented representation enables the linguistic DBMS to easily treat complex and multi-aspect linguistic data. The host computer has sufficient capacity for a considerable linguistic database in an extended RDB. Thus linguistic data storage is not a problem. All information is stored in the host computer. The system thus provides easy maintainability of the linguistic database. An extended RDB is adopted to easily allow to access to statistic linguistic data.

To achieve two different searches, micro and macro analysis, this database system is provided with two ways to transfer linguistic data from the extended RDB in the host computer to a workstation. One transfers data for micro analysis.

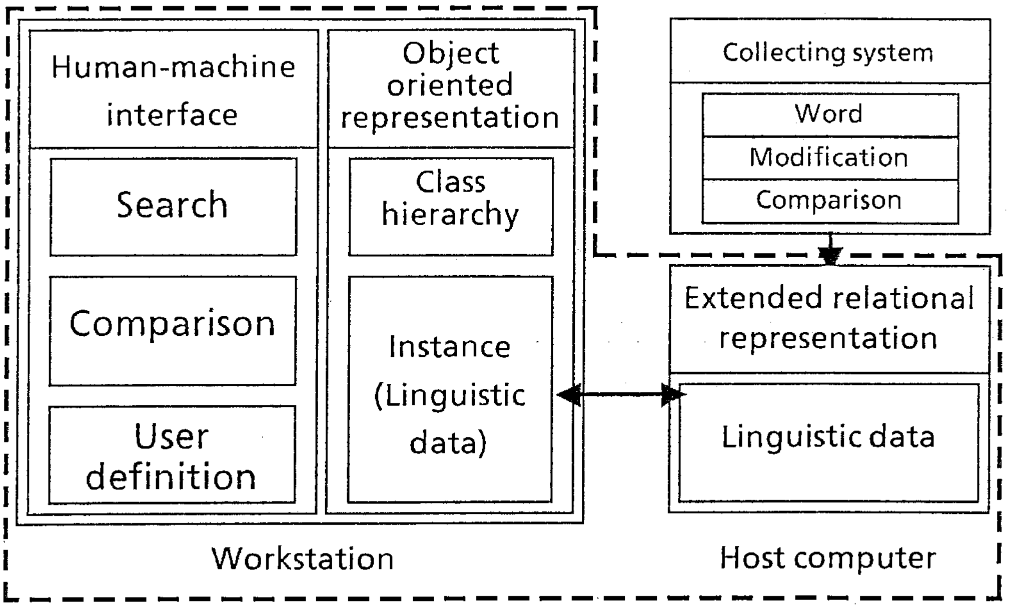

Figure4:
Layout of Integrated Linguistic DBMS

In this mode, necessary multi-aspect linguistic data is transferred from the host computer to the workstation before a sequence of searches is made to analyze linguistic data in the workstation. Thus, linguistic data is retrieved only in the workstation. The other transfers data on demand for macro analysis. In this mode, necessary linguistic data is transferred to the workstation in order to show it to the user when a retrieve is made in the workstation.

To transfer linguistic data from the host computer to a workstation and to represent complex linguistic data compactly, RDB should be extended to deal with field multi-value. Of course SQL(Structure Query Language) must be extended to deal with field multi-value.

Users (linguistic analysts and developers of a language processing system) can access the linguistic database for micro analysis and macro analysis through the workstation interface. Thus they need not be conscious of the extended RDB in the host computer and can use linguistic data as if a large amount of linguistic data is in the workstation.

4.2. Object-Oriented Linguistic Representation

To deal with the above complex linguistic structures linguistic data in a workstation is represented as object-oriented.

Figure 5 shows linguistic data which is represented as object-oriented. Linguistic data is composed of two kinds of data. One is *class* which shows a set of

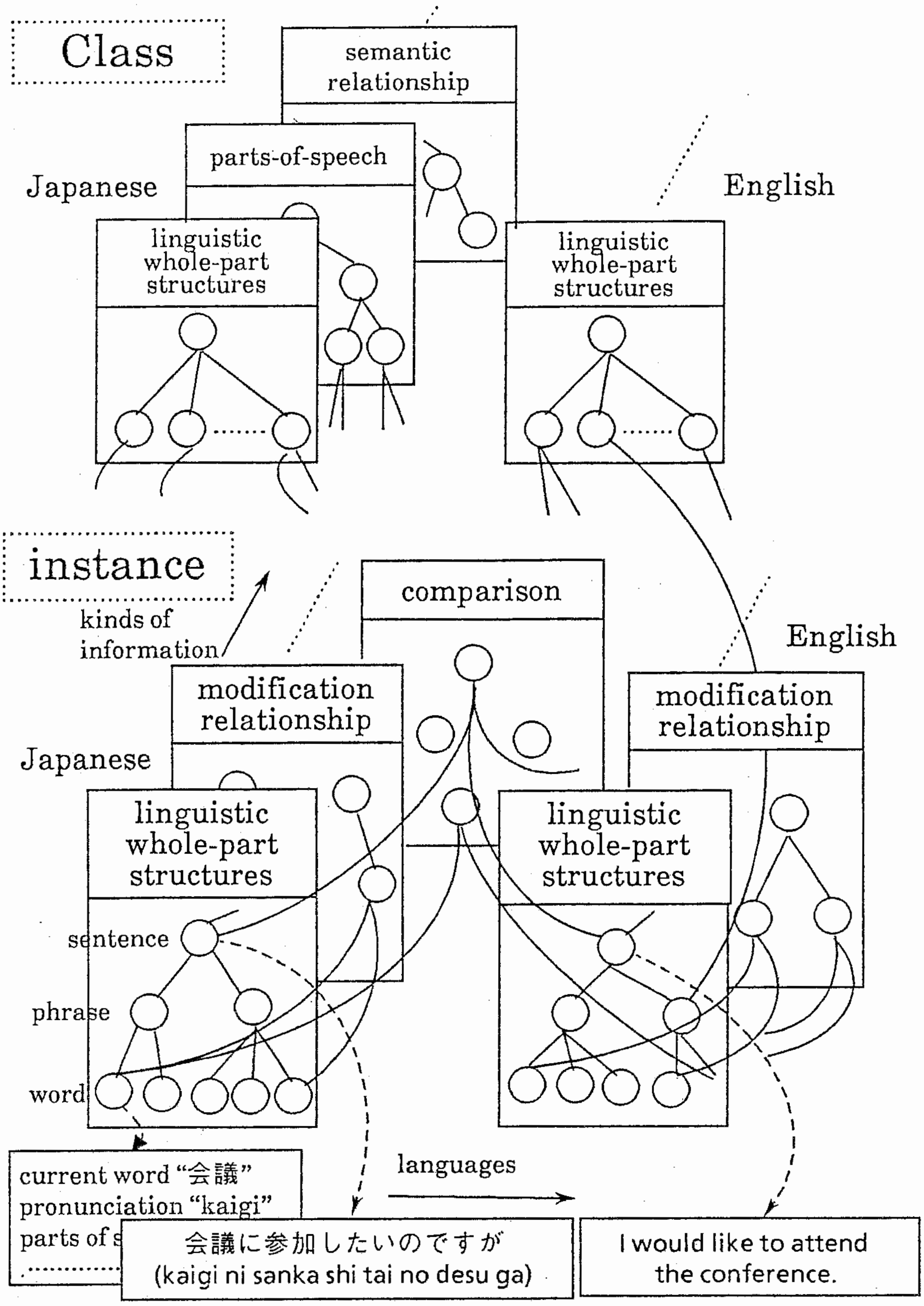
Class
semantic relationship
parts-of-speech
Japanese
English
linguistic whole-part structures
linguistic whole-part structures
instance
kinds of information
comparison
English
modification relationship
modification relationship
Japanese
linguistic whole-part structures
linguistic whole-part structures
sentence
phrase
word
current word "会議"
pronunciation "kaigi"
parts of s
languages
会議に参加したいのですが
(kaigi ni sanka shi tai no desu ga)
I would like to attend the conference.

Figure5: Object-Oriented Linguistic Data Representation

things of the same concept and has hierarchies, the other is *instance* which shows actual linguistic data.

There are various classes of hierarchies in linguistic data including a parts-of-speech hierarchy, a semantic relationship hierarchy, a linguistic whole-part structure hierarchy, a concept hierarchy, etc. These hierarchies may be different in other languages. If there are differences, these hierarchies must be provided. Appendix 5 shows class definitions for the linguistic whole-part structure hierarchy.

Instance also has various aspects from different kinds of information, i.e., linguistic whole-part structure, modification relationship, comparison, etc. There are also different aspects in other languages.

Figure 6 shows examples of object-oriented linguistic data. Class hierarchy is shown by superclass. The subclass relationship, i.e. relationship from superclass to subclass is automatically built by the system. The slot conditions which instances of a class must have can be described in a "has-slots" slot. For example an instance of sentence class has an instance of utterance class as a parts-of slot (instance variable). Values of an instance object slot are instance objects or class objects. Some slots have single value. Some slots have multi-value. This representation provides a simple and flexible framework for linguistic data.

```
An example of CLASS (Sentence Class)
(SENTENCE
    (superclass LINGUISTIC-ELEMENT)
    (has-slots
        (part-of        :slot-value (instance-of UTTERANCE))
        (composed-of    :slot-value (instance-of PHRASE))
        (corresponding  :slot-value
                (instance-of CORRESPONDING-RELATIONSHIP)))

Examples of INSTANCE
(sentence1
    (instance-of      SENTENCE)
    (part-of          utterance1)
    (composed-of      phrase1 phrase2)
    (corresponding    corresponding-relationship1)
(phrase1
    (instance-of      PHRASE)
    (part-of          sentence1)
    (composed-of      word1 word2))
(word1
    (instance-of      WORD)
    (part-of          phrase1)
    (current-word     "会議")
    (pronunciation    "kaigi")
    (parts-of-speech NOUN) .......)
```

Figure 6: Examples of Object-Oriented Linguistic Data

4.3. Object-oriented Human-Machine Interface

In order to perform complex searches and easily employ a human-machine interface, a human-machine interface using object-oriented representation is being implemented. A human-machine interface based on object-oriented representation has the following advantages.

(1) representational ability for complex searches

Object-oriented representation allows treatment of the "is-a" relationship. Searches which use superclass are acceptable.

(2) easy implementation of user-friendly interfaces

Object-oriented representation conveniently displays linguistic data as objects in a workstation. By making the linguistic data objects mouse sensitive, the user can easily examine or modify them. Three functions for a linguistic database human-machine interface are provided.

Three Main Functions for Human-Machine Interfaces
(1) search
(2) comparison
(3) user definition

These three functions will now be discussed in detail.

4.3.1. Search

"Search" is the most important of the three functions. Linguistic data has many aspects and as stated above is very complex. Thus various types of searches are necessary. It is very difficult and wasteful to provide as built-in functions all searches which users require. A general framework to search the database is required. Users can then define the search required within this framework. Appendix 6 shows the general search framework in the Integrated Linguistic DBMS.

4.3.1.1. Simple Search

An example of a simple search pattern is as follows.

```
(defsearch
    Japanese-English-word-corresponding-relationship; search name
    "a search pattern example"; comment
    (object-pattern (var corresponding-relationship)
      (instance-of  corresponding-relationship)
      (Japanese     (var J-word))
      (English      (var E-word))); matching object pattern
    (object-pattern (var J-word)
      (instance-of  Japanese-word)); matching object pattern
    (object-pattern (var E-word)
      (instance-of  English-word)); matching object pattern
    (return-form         corresponding-relationship))
                 ; return form specification
```

A search pattern is usually a rough specification for a search. More parameters can be specified for an actual search. Of course a search pattern details can be specified. For example, regular-form of J-word and regular-form of E-word are specified as "焼く(yaku)" and "burn" in actual search.

To easily specify the actual search specification a tool is provided. This is like a frame editor and can modify slot values.

The search pattern provides a general search framework of a corresponding relationship between a Japanese word and an English word. In this example of an actual search, the Japanese word "焼く(yaku)" and the English word "burn" are specified. Words in English corresponding with the Japanese word "焼く(yaku)" can be "grill", "burn", "heat", "broil", "bake", etc. Thus the context in which the Japanese word "焼く(yaku)" can be translated into "burn" in English can be examined in this example.

4.3.1.2. Combination Search

Users can also retrieve from multi-aspect condition (we call this as "combination search"). The following example is a combination search of a modification relationship and a corresponding-relationship.

```
(defsearch
    Combination-C&M; search name
    "a combination search pattern of modification and
     corresponding relationship"; comment
    (object-pattern (var corresponding-relationship)
        (instance-of corresponding-relationship)
        (Japanese    (var J-word1))); matching object pattern
    (object-pattern (var J-word1)
        (instance-of Japanese-word)
        (regular-form   "焼く(yaku)"); grill
        (rel-element-of (var modification-rel))); matching object pattern
    (object-pattern (var modification-rel)
        (instance-of modification-relationship)
        (sem-rel   (var sem-rel))); matching object pattern
    (object-pattern (var sem-rel)
        (instance-of semantic-relationship)
        (sem-rel-name   OBJECT)
        (modifier (var J-word2))); matching object pattern
    (object-pattern (var J-word2)
        (instance-of Japanese-word)
    (regular-form   魚))
                          ; matching object pattern [魚(sakana)=FISH]
  (return-form    corresponding-relationship))
                          ; return form specification
```

This search pattern provides a specific search which finds the corresponding relationship of a Japanese modificand that the modifier is a kind of 魚(sakana)=fish, the modificand itself is 焼く(yaku) and the semantic relationship is OBJECT (魚 is OBJECT of 焼く).

4.3.1.3 Search Using Class Hierarchy

Although instance objects (actual linguistic data) are connected with only leaf as class objects, class objects but not leaf in the hierarchy can be specified in search patterns. For example "verb", which is not leaf class in the parts-of-speech hierarchy and has two subclasses, i.e., "intransitive verb" and "transitive verb", can be used in searches. It is convenient when a search need not make a distinction between "intransitive verb" and "transitive verb". It allows the user not to specify a search in detail.

4.3.1.4. A Search Pattern Definition Tool

This framework for a general search allows examination of a complex linguistic database from various aspects. A user can register his search patterns in the search pattern files. This search pattern definition specification increases the user's workload. Thus a search pattern definition tool must be provided to easily define a search. To do this, a search pattern definition tool which provides skeletons by using class definitions is provided. The user then only fills these skeletons to define a search. An expert user can handle the linguistic database with this framework. This system is treated for expert users. As it would be difficult for a naive user to handle the linguistic database with this framework, frequently used search patterns are built-in.

4.3.2. Comparison

The second function of the human-machine interface is comparison. This function allows display of comparable linguistic data showing the correspondence between them. For example, Japanese and English sentences which have the same content can be displayed showing various levels of correspondence. Comparison of similar and dissimilar things is useful in analyzing linguistic data. From the point of view of transferring speech from one language to another, multi-language data comparison is very important.

4.3.3. User definition

Previous linguistic databases did not provide all the data users wanted and so specifying may not be preferred by some users. To partially solve this problem, a user definition function is provided. User definition functions effectively allow the user to define the database. As a first step, redefinition of parts of speech, modification of semantic relationships and word regular forms are implemented. Current redefinition functions are intended to rename a concept, create a superclass concept, and move, remove, or merge concepts.

4.3.4. Statistic Facilities

As statistic facilities, this system provides "count" and "group" functions which are usually supported by SQL in RDBMS. In this system, these functions are supported as part of the search function. Figure 7 shows an example of a search using the statistic function.

```
Search pattern in workstation
(defsearch    "count words by parts of speech"
    (object-pattern  (var J-word)
        (instance-of Japanese-word)
        (parts-of-speech %group))
    (return-form %count))
Corresponding SQL  representation in RDBMS
    select count(*)
    from Japanese-word
    group by parts-of-speech
```

Figure 7: A typcal search using statistic function

```
Search pattern in workstation
(defsearch    "count words "
    (object-pattern  (var J-word)
        (instance-of Japanese-word)
        (parts-of-speech auxiliary-verb)
        (regular-form (or (れる  られる)) %group))
    (return-form %count))
Corresponding SQL  representation in RDBMS
    select count(*)
    from Japanese-word
    where regular-form in <れる, られる>
    group by regular-form
```

Figure 8: A typical search using the statistic function

%count and %group are provided in search pattern representation language. In the case of Figure 7, the result of the search are the word counts of each part of speech. Figure 8 shows that the grouped field also has a restriction. The result is a word count of"れる" and "られる".

For grouping with hierarchy, %group* are provided. If %group* is specified in parts-of-speech, the result is not only the lowest level parts-of-speech counts but also higher level parts-of-speech counts. Figure 9 shows an example of a search using %group*.

Search pattern in workstation

```
(defsearch    "count words "
    (object-pattern  (var J-word)
        (instance-of Japanese-word)
        (parts-of-speech %group*))
    (return-form %count))
```

The result

```
noun                       159
     proper noun            10
     common noun           126
          sahen noun         8
...........
numeral                     12
..............
```

Figure 9: A typical %group*

Actual computations of these statistic functions are efficiently made in RDBMS in host computer.

4.4. Extended RDBMS

The linguistic data collected by the collecting system is stored in the RDB of the host computer. To effectively treat the whole-parts relationship the RDB should be extended. The RDB must have functions to deal with multi-value. To acquire these functions the SQL is extended.

Conversation, utterance, sentence, phrase, word, semantic relationship, syntactic relationship and each level of corresponding relationships are represented by "Relation". Actual linguistic data are presented by "Tuple". Utterance speaker, word parts-of-speech, semantic relationship are represented by "Attribute".

Correspondence between the RDB representation and the Object-Oriented representation is as follows.

RDB representation	Object-Oriented representation
Relation	lowest class
Tuple	instance
Attribute	instance variable

To use the functions of RDBs on the market can make easier development of functions like data modification, security, integrity, and statistic processing.

5. Concluding Remarks

In natural language research, a linguistic database is important in order to really know how a language is spoken or written. A linguistic DBMS is also very important for using the linguistic database effectively. One approach to implementing a linguistic DBMS is shown. Linguistic DBMSs should have two functions, i.e., database management and human-machine interfacing. In this system, database management is by means of a RDB which is already on the market. Human-machine interface is by means of a high performance workstation. With these two functions a linguistic database can be effectively implemented. By representing linguistic data elements as objects linguistic data can be looked at from various aspects or combinations of aspects and then retrieved.

In the future, it is intended that the speech database being developed in an other department of these laboratories will be connected with this database management system. This will provide a good research environment for both natural language and speech researchers.

An Integrated Linguistic DBMS is now being implemented on VAX8700/ULTRIX and Symbolics Lispmachines. The VAX8700/ULTRIX used as the host computer collects texts, analyzes them and stores the results in an extended RDB. Symbolics Lispmachines are used as high performance workstations to create a user-friendly environment. FLAVORS are based to implement the object-oriented human-machine interface. A prototype human-machine interface system was implemented. An extended RDB and transfer functions between an extended RDB in the host computer and workstations are now being implemented.

The first goal is Japanese to English interpretation. For this reason chiefly Japanese language data is being collected. This linguistic data is also useful to human translators, language teachers and contrastive linguistics researchers.

References

[1] K. Dittrich, U. Dayal (ed.), Proc. of the International Workshop on Object-Oriented Database Systems, 1986

[2] A. Ege, C. A. Ellis, "Design and Implementation of GORDION, an Object Base Management System", Proceedings Third International Conference on DATA ENGINEERING, 1987

[3] J. Diederich and J. Milton, "ODDESSY: An Object-oriented Database Design System", Proceedings Third International Conference on DATA ENGINEERING, 1987

[4] D. Woelk and W. Kim, "Multimedia Information Management in an Object-Oriented Database System", Proceedings of the 13th VLDB Conference, Brighton, 1987

[5] F. Bancilhon, "Object Oriented Multilanguage Systems: the Answer to Old and New Database Problems?", France-Japan AI and CS sympo87, Cannes, Nov., 1987

[6] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks", Comm. of ACM Vol. 13, No. 13, 1970

[7] Z. M. Ozsoyoglu, Li-Yan Yuan, "A Design Method for Nested Relational Databases", Proceedings Third International Conference on DATA ENGINEERING, 1987

[8] V. Linnemann, "Non First Normal Form Relations and Recursive Queries: An SQL-Based Approach", Proceedings Third International Conference on DATA ENGINEERING, 1987

[9] A. Kurematsu, "Prospect of a Basic Study for The Automatic Telephone Interpretation", First International Symposium on Advanced Man-Machine Interface Through Spoken Language, 1988

[10] A. Goldberg and D. Robson, SMALLTALK-80 The Language and its Implementation, Addison-Wesley, Reading, MA, 1983

[11] Symbolics, Symbolics Common Lisp – Language Concepts, 1986

[12] J. Aarts and W. Meijs, Corpus Linguistics II: new studies in the analysis and exploitation of computer corpora, Rodopi Amsterdam, 1986

[13] L. F. Farina,"LDMS: A Linguistic Data Management System", Computers and the Humanities 17 ,1983

[14] Brown University, Brown Corpus, Technical Report, Brown University, 1967

[15] University of Nijmegen, TOSCA The Nijmegen Research Group for Corpus Linguistics", 1987

[16] H. Czap, C. Galinski, Terminology and Knowlege Engineering, Indeks Verlag, 1987

[17] The National Language Research Institute in Japan, CL Study Vol. 1, 1987 (in Japanese)

[18] H. Arita, K. Kogure, I. Nogaito, H. Maeda and H. Iida, "Media-Depend Conversation Manners - Comparison of Telephone and Keyboard Conversations - (メディアに依存する会話の様式 - 電話会話とキーボード会話の比較-)" IPSJ Technical Report NL61-5, 1987 (in Japanese)

[19] H. Arita and H. Iida, "Discourse Structure of Task-Oriented Dialogue in Japanese - Comparison of Telephone and Keyboard Conversations - (日本語におけるタスクオリエンティッドな対話の構造 -電話対話と端末間対話の比較-)", IEICE Technical Report NLC87-10, 1987 (in Japanese)

[20] H. Arita, K. Kogure, I. Nogaito, H. Maeda and H. Iida, "Comparison of Telephone and Keyboard Conversation (電話対話と端末対話の比較)", ATR Technical Report TR-I-0016, 1987 (in Japanese)

[21] K. Hashimoto, K. Ogura and T. Morimoto, "A Man-Machine Interface for an Integrated Linguistic Database Management System (言語データベース統合管理システムのマンマシンインタフェース)", 37th IPSJ Meeting (forthcoming), 1988 (in Japanese)

[22] K. Hashimoto, K. Ogura and T. Morimoto, "A Man-Machine Interface for an Integrated Linguistic Database Management System (言語データベース統合管理システムのマンマシンインタフェース)", ATR Technical Report (forthcoming), 1988 (in Japanese)

[23] H. Iida, I. Nogaito and T. Aizawa, "Analysis of Telephone Conversations through an Interpreter (電話会話の特徴分析)", IEICE Technical Report NLC86-11, 1986 (in Japanese)

[24] H. Iida, K. Kogure, I. Nogaito and T. Aizawa, "Analysis of Telephone Conversations through an Interpreter (電話会話の特徴分析)", ATR Technical Report TR-I-0002, 1987 (in Japanese)

[25] H. Iida, M. Kume, I. Nogaito and T. Aizawa, "Collection of Interpreted Telephone Conversation Data (通訳を介した電話会話収集データ)", ATR Technical Report TR-I-0007, 1987 (in Japanese)

[26] N. Inoue, K. Ogura and T. Morimoto, "The Problems of Semantics Relation and the Consideration (係り受け意味関係の問題点とその考察)", IEICE Technical Report NLC87-25, 1988 (in Japanese)

[27] N. Inoue, K. Ogura and T. Morimoto, "Semantics Relations for the Language Database (言語データベース用格·係り受け意味体系)", ATR Technical Report TR-I-0029, 1988 (in Japanese)

[28] N. Inoue, K. Ogura and T. Morimoto, "Semantics and Syntactics Relations described in the Integurated Linguistic Database (言語データベース用単語間の関係データ)", 37th IPSJ Meeting (forthcoming), 1988 (in Japanese)

[29] K. Kita and T. Morimoto, "Automatic Idiom Extraction from Text Database (テキスト·データベースからの慣用表現の自動抽出)", 37th IPSJ Meeting (forthcoming), 1988 (in Japanese)

[30] T. Morimoto, K. Ogura and H. Iida, "Constructing Linguistic Database for Automatic Telephone Interpreting Research (自動翻訳電話研究用言語データベースの収集について)", 36th IPSJ Meeting 4U-5, 1988 (in Japanese)

[31] K. Ogura, "Construction of Data for Language Comparison (言語対比データの構築について)", IEICE Spring Meeting 1642, 1987 (in Japanese)

[32] K. Ogura, N. Shinozaki and T. Morimoto, "Tools for linguistic data compilation (言語データベース収集支援)", 36th IPSJ Meeting 4U-4, 1988 (in Japanese)

[33] K. Ogura, K. Hashimoto and T. Morimoto, "An Integrated Linguistic Database Management System (言語データベース統合管理システム)", 37th IPSJ Meeting (forthcoming), 1988 (in Japanese)

[34] A. Shimazu, S. Naito and H. Nomura, "日本語文意味構造の分類 -名詞句構造を中心に-", IPSJ Technical Report NL47-5, 1985 (in Japanese)

[35] N. Shinozaki, K. Ogura and T. Morimoto, "Quality Control for Linguistic Database (言語データベースの品質管理)", 36th IPSJ Meeting 4U-3, 1988 (in Japanese)

[36] N. Shinozaki, K. Ogura and T. Morimoto, "Simulating Conversations for Linguistic Database (言語データベース作成のためのシミュレーション会話 - 会話データのリアリティ改善-)", 37th IPSJ Meeting (forthcoming), 1988 (in Japanese)

[37] K. Yoshimoto, "Classification of Japanese Parts of Speech(日本語品詞の分類)", ATR Technical Report TR-I-0008, 1987 (in Japanese)

IEICE:	*The Institute of Electronics, Information and Communication Engineers of Japan (*電子情報通信学会*)*
IPSJ:	*Infomation Processing Society of Japan (*情報処理学会*)*

Appendix 1. Details of word information

(a) current word (CW):
word itself appearing in sentences
It is important to analyze how words are used and how often words are used.

(b) pronunciation (P):
word pronunciation from context
It is important to analyze relationship between sounds and characters.

(c) regular form (RF):
In accordance with Japanese dictionary custom the third base (conclusive base) is used as regular form in Japanese for words.
This information allows to search various word conjugations as the same word.

(d) parts of speech (POS):
A hierarchical system of Japanese "parts of speech" including 24 parts of speech at the lowest level was compiled. The system almost followed that of Japanese school grammar, which is familiar to Japanese.
This is the most basic information as syntax. A grammar can be create based on this information.

(e) conjugational pattern (CP):
shows how the word is conjugated.

(f) conjugational form (CF):
shows how the word is connected to the next word or modality. 6 forms were prepared, i.e., first (negative) base, second (continuative) base, third (conclusive) base, fourth (attributive) base, fifth (conditional) base, sixth (imperative) base.

(g) euphonic changes (EC):
three euphonic changes, i.e., the "i" sound change, the geminate consonant sound change and the nasal sound change are provided.
It allows to analyze what context causes an euphonic change.

(h) meaning (M):
Word “meaning” is very important as semantic information, but it is not clear yet how to treat word meaning.
For example a word may has two meanings in some dictionary and three meanings in other dictionary. Thus word meaning is not provided.

(i) super concept (S):
shows hierarchical meaning relationships.
For example “mamal” is a super concept of “dog” and “bird” is a super concept of “duck” and “animal” is a super concept of “mamal” and “bird”. It

is also not clear yet how to create meaning hierarchy. Thus super concept is not provided.

An example of word information about a sentence fragment is as follows.

CW	P	RF	POS	CP	CF	EC
会議	kaigi	会議	noun			
に	ni	に	case particle			
参加	sanka	参加	sahen noun			
し	shi	する	subsidiary verb	sahen	2nd	
たい	tai	たい	auxiliary verb		4th	
の	no	の	case particle			
です	desu	です	auxiliary verb		3rd	
が	ga	が	conjugation particle			

An example of word information

APPENDIX 2. Parts of Speech Hierarchy

- noun(名詞)
  - proper noun(固有名詞)
  - common noun(普通名詞)
    - sahen noun(サ変名詞)
    - adjective noun(形容名詞)
    - other noun(その他の普通名詞)
  - nominal(数詞)
  - pronoun(代名詞)
- verb(動詞)
  - main verb(本動詞)
  - subsidiary verb(補助動詞)
- adjective(形容詞)
- adverb(副詞)
- affributive(連体詞)
- conjunction(接続詞)
- interjection1(間投詞)
- interjection2(感動詞)
- auxiliary verb(助動詞)
- particle(助詞)
  - case particle(格助詞)
  - nominal particle(準体助詞)
  - topic particle(係助詞)
  - adverbial particle(副助詞)
  - parallel particle(並立助詞)
  - conjugation particle(接続助詞)
  - sentence-final particle(終助詞)
- prefix(接頭辞)
- suffix(接尾辞)
- sign(記号)

APPENDIX 3. Modification Semantic Relationship

1 Agent
2 Unvolitional Agent
3 Unvolitional Subject
4 Object
5 Experiencer
6 Originator
7 Recipient
8 Partner
9 Accompanyment
10 Opponent
11 Time At
12 Time From
13 Time To
14 Time Duration
15 Space At
16 Space From
17 Space To
18 Space Through
19 Source
20 Goal
21 Cause
22 Tool
23 Material
24 Manner
25 Condition
26 Purpose
27 Role
28 Range
29 Degree
30 Predicate
31 Comparison
32 Topic
33 Nomination
34 Content
35 Evaluation
36 Concession
37 Addition
38 Circumstance
39 Viewpoint
40 Selection
41 Example
42 Possessor
43 Author
44 Apposition
45 Whole
46 Part
47 Reference for Relation
48 Delimiter
49 Attribute Value of Object
50 Attribute
51 Object of Attribute
52 Attribute Value
53 Metaphoral Substance
54 Metaphor
55 Indirect Case
56 Complementizer
57 Pseudo Complementizer
58 Insertion
59 Parallel
60 Or
61 Connective
62 Adverbial Particle
63 Others

Appendix 4. Modification syntactic relationships

(a) Ordinary Case Relationship:

ex. He *wants* it.

(b) Adjective Modification Relationship:

ex. She is a beautiful *girl.*

(c) Adjective Case Relationship:

ex. The *book* which is on the desk is mine.

(d) Adverbial Modification Relationship,

Predicate Modification Relationship:

ex. He *speaks* too fast.

(e) Sentence to Sentence Modification Relation:

ex. *You would have succeeded* if you had tried harder.

Undelined words, phrases or clauses modify italic words, phrases or clauses.

APPENDIX 5. Class Definition

```
(DEFCLASS  言語構成素  ;  Linguistic-Element
    (HAS-SLOTS
        (
          (COMPOSED-OF)
          (PART-OF)
          (NEXT-IS)
          (PREV-IS)
          (AFTER)
          (BEFORE)
          )))

(DEFCLASS  全テキスト  ;  Full-Text
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (COMPOSED-OF  :SLOT-VALUE  (INSTANCE-OF  会話)
                        :MULTIPLE  T)
          )))

(DEFCLASS  会話  ;  Conversation
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (PART-OF  :SLOT-VALUE  (INSTANCE-OF  全テキスト)
                        :MULTIPLE  NIL)
          (COMPOSED-OF  :SLOT-VALUE  (INSTANCE-OF  発話)
                        :MULTIPLE  T)
          (タイトル  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (会話者    :SLOT-VALUE  SYMBOL  :MULTIPLE  T)
          (日付け    :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (メディア  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (方向      :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (領域      :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          )))

(DEFCLASS  発話  :  Utterance
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (PART-OF  :SLOT-VALUE  (INSTANCE-OF  会話)
                        :MULTIPLE  NIL)
          (COMPOSED-OF  :SLOT-VALUE  (INSTANCE-OF  文)
                        :MULTIPLE  T)
          (発話者  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          )))
```

```
(DEFCLASS  文  ;  (Japanese) Sentence
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (PART-OF  :SLOT-VALUE  (INSTANCE-OF  発話)
                                 :MULTIPLE  NIL)
          (COMPOSED-OF  :SLOT-VALUE  (INSTANCE-OF  文節)
                                 :MULTIPLE  T)
          (対応  :SLOT-VALUE  (INSTANCE-OF  対応)
                                 :MULTIPLE  NIL)
          )))

(DEFCLASS  文節  ;  Phrase
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (PART-OF  :SLOT-VALUE  (INSTANCE-OF  文)
                                 :MULTIPLE  NIL)
          (COMPOSED-OF  :SLOT-VALUE  (INSTANCE-OF  単語)
                                 :MULTIPLE  T)
          )))

(DEFCLASS  単語  ;  Word
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (PART-OF  :SLOT-VALUE  (INSTANCE-OF  文節)
                                 :MULTIPLE  NIL)
          (PREV-IS  :SLOT-VALUE  (INSTANCE-OF  単語)
                                 :MULTIPLE  NIL)
          (NEXT-IS  :SLOT-VALUE  (INSTANCE-OF  単語)
                                 :MULTIPLE  NIL)
          (出現単語  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (読み      :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (正規表現  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (品詞      :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL
                     :REF-TREE  品詞定義)
          (活用型    :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (活用形    :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (関係      :SLOT-VALUE  (INSTANCE-OF  関係)
                                 :MULTIPLE  T)
          )))
```

```
(DEFCLASS  対応  ;  Corresponding Relationship
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (日本語  :SLOT-VALUE  (INSTANCE-OF  文)
                                :MULTIPLE  NIL)
          (英語    :SLOT-VALUE  (INSTANCE-OF  英文)
                                :MULTIPLE  NIL)
          (独語    :SLOT-VALUE  (INSTANCE-OF  独文)
                                :MULTIPLE  NIL)
          (仏語    :SLOT-VALUE  (INSTANCE-OF  仏文)
                                :MULTIPLE  NIL)
          )))

(DEFCLASS  英文  ;  English Sentence
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (PART-OF  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL)
          (COMPOSED-OF  :SLOT-VALUE  SYMBOL  :MULTIPLE  T)
          (対応  :SLOT-VALUE  (INSTANCE-OF  対応)
                                :MULTIPLE  NIL)
          (出現文  :SLOT-VALUE  STRING  :MULTIPLE  NIL)
          )))

(DEFCLASS  独文  ;  German Sentence
    (SUPER-CLASS  言語構成素))

(DEFCLASS  仏文  ;  French Sentence
    (SUPER-CLASS  言語構成素))
```

```
(DEFCLASS  関係  ;  Relationship
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (構文関係  :SLOT-VALUE  (INSTANCE-OF  構文関係)
                        :MULTIPLE  NIL)
          (意味関係  :SLOT-VALUE  (INSTANCE-OF  意味関係)
                        :MULTIPLE  NIL)
          )))

(DEFCLASS  構文関係  ;  Syntactic Relationship
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (関係名  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL
                        :REF-TREE  係り受け構文関係)
          (MODIFIER  :SLOT-VALUE  (INSTANCE-OF  単語)
                        :MULTIPLE  NIL)
          (MODIFICAND  :SLOT-VALUE  (INSTANCE-OF  単語)
                        :MULTIPLE  NIL)
          )))

(DEFCLASS  意味関係  ;  Semantic Relationship
    (SUPER-CLASS  言語構成素)
    (HAS-SLOTS
        (
          (関係名  :SLOT-VALUE  SYMBOL  :MULTIPLE  NIL
                        :REF-TREE  係り受け意味関係)
          (HEAD  :SLOT-VALUE  (INSTANCE-OF  単語)
                        :MULTIPLE  NIL)
          (MODIFIER  :SLOT-VALUE  (INSTANCE-OF  単語)
                        :MULTIPLE  NIL)
          )))
```

Appendix 6. A general search framework in Integrated Linguistic DBMS

```
(defsearch <<search-name>> <<comment>>
    <<matching object pattern list>>
    <<return form specification>>)
<<search-name>> ::= <atom>
<<comment>> ::= <string>
<<matching object pattern list>>
   ::=     <<matching object pattern>> |
           <<matching object pattern>>
           <<matching object pattern list>>
<<matching object pattern>>
   ::=    (object-pattern <<object-name>>
           <<slot condition>>%)
<<object-name>> ::= <<variable>>
<<slot condition>>
   ::=    (<<slot-name>> <<value__restriction>>)
<<slot-name>> ::= <atom>
<<value__restriction>>
   :: =    <<value>> | (not <<value>>) |
          (or <<value>>$) |(multi <<value>>$) |
          (multi+ <<value>>$)
<<value>> ::= <atom> | <<variable>> | ? | *
<<variable>> ::= (var <atom>)
<<return form specification>>
   ::=    (return-form <<level>> [ - <<region>> [+]
           <<region>>] )
<<level>> ::= conversation | utterance | sentence |
                   phrase | word | character | relation
<<region>>::= <number>
```

? means matching any object in a single value slot.

* means matching all objects in a multi-value slot.

% means reputation equals more than 0.

$ means reputation more than 1.