

## A NATURAL LANGUAGE INTERFACE USING A WORLD MODEL

Yoshio Izumida, Hiroshi Ishikawa, Toshiaki Yoshino,  
Tadashi Hoshiai, and Akifumi Makinouchi

Software Laboratory  
Fujitsu Laboratories Ltd.  
1015 kamikodanaka, Nakahara-ku, Kawasaki, 211, Japan

### ABSTRACT

Databases are nowadays used by varied and diverse users, many of whom are unfamiliar with the workings of a computer, but who, nevertheless, want to use those databases more easily. Rising to meet this demand, authors are developing a Japanese language interface, called KID, as a database front-end system. KID incorporates a world model representing application and database knowledge to help make databases easier to use. KID has the following features: (1) parser extendability and robustness, (2) independence from the application domain, (3) ease of knowledge editing, (4) independence from the database. This paper focuses on the first three features. KID has already been applied to the fields of housing, sales, and drug testing, thus confirming its transportability and practicality.

### INTRODUCTION

KID (Knowledge-based Interface to Databases) is a Japanese-language database interface (Izumida, 84). KID has the following four features.

#### Extendability and robustness

Natural language sentences employ a wide variety of expressions. A parser must always be extended to understand new sentences. A parser which can understand one set of sentences is often incapable of understanding another set of sentences. In KID, parsing rules are grouped into packets and the parsing mechanism is simple, thus making KID highly extendable. The system must be robust, in order to handle conversational sentences, which often contain errors and ellipses. To interpret these ill-formed sentences, semantic interpretation must play a leading role. KID has an integrated knowledge base called the world model. The world model represents the semantic model of the domain of the discourse in an object-oriented manner. Several systems (e.g., Ginsparg, 83) use a semantic model to interpret ill-formed sentences, but the use of the semantic model is unclear. We have made the semantic interpretation rules clear according to the structure of the world model and syntactic information of the input sentences. This helps the parsing of ill-formed sentences.

#### Independence from the application domain

The system must be easily adaptable to

different applications. The domain-dependent knowledge must be separate from the domain-independent knowledge. In many systems (e.g., Waltz, 78 and Hendrix, 78), the domain-dependent knowledge is embedded within the parsing rules, thus reducing the system's transportability. In KID, the domain-dependent knowledge is integrated into the world model separately, therefore giving KID high transportability.

#### Ease of knowledge editing

The world model contains various kinds of knowledge, and the editing of this knowledge must be easy to accommodate various levels of users. KID provides users with the world model editor, this having a separate user interface for each user level. The world model editor makes the customization and extension of the KID system easy.

#### Independence from the database

The system must be able handle changes in the database system and schema easily. In TEAM (Grosz, 83), the schema information is separate, but the user must be familiar with the database schema such as files and fields. In KID, the mapping information between the model of the domain and the database schema is described in the world model, so the user does not have to worry about any changes in the database schema. Knowledge of the query language of a database system is represented separately as production rules. Thus, the user only has to change these rules if there are changes in the database system.

In this paper we will focus on the first three features of KID. Firstly, we will explain the world model, then the overall structure of the KID, the morphological analyzer (required to process Japanese-language sentences), the model-based parser, semantic interpretation and the flow of the parsing process, knowledge for customizing KID and, lastly, the evaluation of KID and its results.

### WORLD MODEL

The world model represents the user's image of the application domain. The user's image does not match the database schema, because the database schema reflects the storage structure of the data and the performance consideration of the database system. The world model represents the user's image as classes and relationships between them.

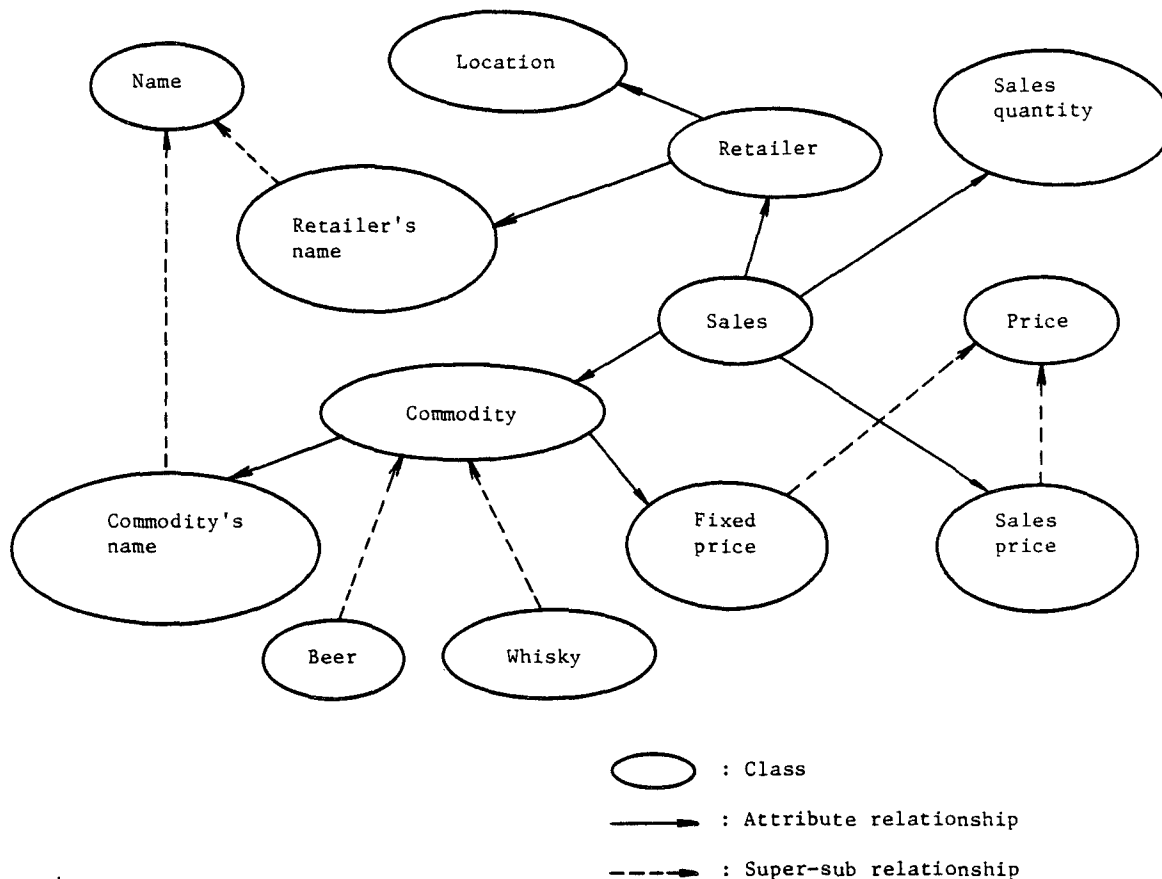


Figure 1. Part of the world model for sales.

A class is represented as an object in the object-oriented programming sense (Bobrow, 81), which describes a thing or event in the domain. There are only two types of relationship; attribute relationship and super-sub relationship. This model matches the user's image and is very simple, so design and editing of the model is easy.

Figure 1 shows the part of the world model for a sales domain. The commodity class has two attribute classes, commodity's name and fixed price. The beer and whisky classes are subclasses of the commodity class and inherit its attributes. Figure 2 shows a part of the definition of the sales class. The internal representation of a class object is a frame expression. A slot represents a relationship to another class using a \$class facet and mapping information to the database schema using a \$storage facet. The value of a \$storage facet denotes the class name which has mapping information. The sales class has four attribute classes: RETAILER, COMMODITY, SALES PRICE, and SALES QUANTITY. An object may also include the method for handling data within it.

The system allows the user to define lexical information in the world model. For example, the noun 'commodity' corresponds to the commodity class. The verb 'sell' and the noun 'sale' both correspond to the sales class. The verb 'locate'

```

SALES
  RETAILER $class RETAILER
           $storage SALES_RETAILER_STORAGE
  COMMODITY $class COMMODITY
            $storage SALES_COMMODITY_STORAGE
  PRICE $class SALES_PRICE
        $storage SALES_PRICE_STORAGE
  QUANTITY $class SALES_QUANTITY
           $storage SALES_QUANTITY_STORAGE
  
```

Figure 2. Internal representation of a class.

corresponds to the arc between the relation and location classes. Lexical information is physically stored in the word dictionary. The dictionary is represented as a table of the relational database system. Figure 3 shows part of the dictionary. The dictionary consists of a headword, an identifier, a part of speech, parsing information and other fields. The correspondence to the world model is represented in the OBJECT feature of the PARSE field. The verb also has its case frame information in the PARSE field. All the information relating to a specific domain is stored in the world model, so the user need only create the world model to customize KID to a specific application. This results in transportability of the system.

HEADWORD	IDENTIFIER	POS	PARSE
商品 (commodity)	SHOUHIN	N	(OBJECT COMMODITY)
販売 (sell)	HANBAI-SURU	VB	(OBJECT SALES) CLASS (CASE ((RETAILER *GA *WA (SALES_QUANTITY NP)))
れ	RE-RU	AUX-VB	_____
を	WO	AUX	(P *WO)
の	NO	AUX	(P *NO)

Figure 3. Word dictionary.

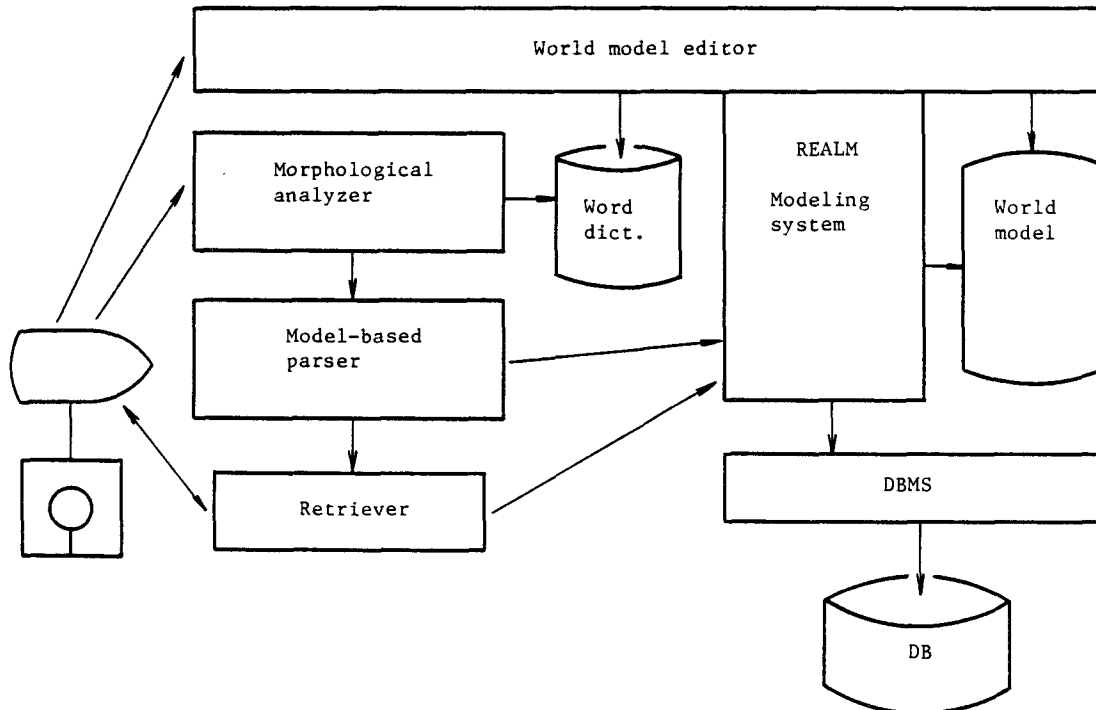


Figure 4. System configuration.

#### SYSTEM CONFIGURATION

KID is the front-end system of the database management system, the configuration being shown in Figure 4. The user enters a query via Japanese word processing terminal. Since a Japanese-language sentence is not separated into words, the morphological analyzer segments the sentence to get the list of words, using the word dictionary. The model-based parser analyzes the word list, and semantically interprets it, using the world model as a basis. The result is the "meaning structure" consisting of the parsed tree and the relevant part of the world model representing the meaning of the input query. The retriever generates the

Japanese-language paraphrase from the meaning structure and outputs it to the user terminal for confirmation. Then, the retriever translates the meaning structure into the query language of the target database management system and executes it. The result is displayed on the user terminal. The world model is managed by the modeling system, REALM (REAL world Modeling system), and is edited by the world model editor.

#### MORPHOLOGICAL ANALYZER

A Japanese-language sentence is not separated into words. The system must segment a sentence into its component words. The morphological

彼の行動は大人気なかったです。  
 His behavior was childish.

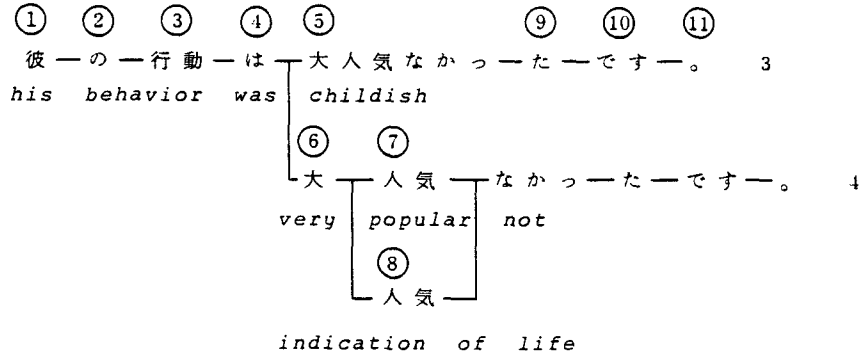


Figure 5. An example of morphological analysis.

analyzer performs this segmentation. KID selects the segmentation candidate with the least number of 'bunsetsu'. We believe this method to be the best method for segmenting a Japanese-language sentence (Yoshimura, 83). This method uses a breadth-first search of a candidate word graph. Since many candidate words are generated by this method, the performance of the segmentation is not so good. We use the optimum graph search algorithm, called A\* (Nilssen, 80), to search the candidate word graph.

Figure 5 shows an example of morphological analysis. This sentence has three possible segmentations. The first line is the correct segmentation, having the least number of 'bunsetsu'. The algorithm A\* estimates the number of bunsetsu in the whole sentence at each node of the candidate word graph, and selects the next search path. This method eliminates useless searching of the candidate graph. In Figure 5, the circled numbers denote the sequence of the graph search.

The morphological analyzer segments a sentence using connection information for each word. The connection information depends on the part of speech. Detailed classification of words leads to correct segmentation. However, it is difficult for an end-user perform this kind of classification. Thus, we classify words into two categories: content words and function words. Content words are nouns, verbs, adjectives, and adverbs, which depend on the application. They are classified roughly. Function words include auxiliaries, conjunctions, and so on, which are independent of the domain. They are classified in detail. It is easy for the user to roughly classify content words. This morphological analyzer segments sentences precisely and efficiently, and generates a word list. This word list is then passed to the model-based parser.

**MODEL BASED PARSER**

In its first phase the parser generates 'bunsetsu' from the word list. The parser syntactically analyzes the relationship between these 'bunsetsu'. At the same time, the parser

semantically checks and interprets the relationships, based on the world model. 'Bunsetsu' sequences of a Japanese-language sentence are relatively arbitrary. And conversational sentences may include errors and ellipses, therefore the parser must be robust, in order to deal with these ill-formed sentences. These factors suggest that semantic interpretation should play an important role in the parser.

The basic rules of semantic interpretation are the identification rule and the connection rule. These rules check the relationship between the classes which correspond to the 'bunsetsu' and interpret the meaning of the unified 'bunsetsu'. The identification rule corresponds to a super-sub relationship. If two classes, corresponding to

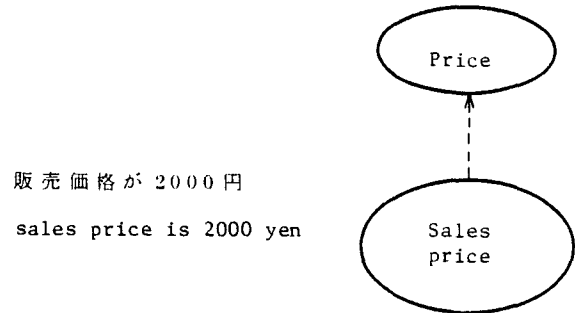


Figure 6. An example of the identification rule.

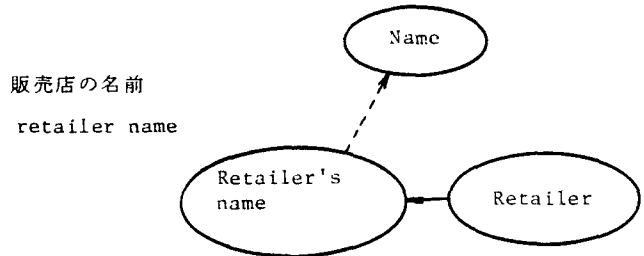


Figure 7. An example of the connection rule.

two phrases, are connected by a super-sub relationship, this rule selects the subclass as the meaning of the unified phrase, because the subclass has a more specific and restricted meaning than the super class. Figure 6 shows an example of the identification rule. In this example, the phrase 'sales price' corresponds to the sales price class, and '2000 yen' corresponds to the price class. The identification rule selects the sales price class as the unified meaning. The connection rule corresponds to an attribute relationship. If two classes are connected by an attribute relationship, this rule selects the root class of the relation as the meaning of the unified class, because the root class clarifies the semantic position of the leaf class in the world model. Figure 7 shows an example of the connection rule. In this example, the phrase 'retailer' corresponds to the retailer class, and 'name' corresponds to the name class. The connection rule selects the retailer class as the unified meaning.

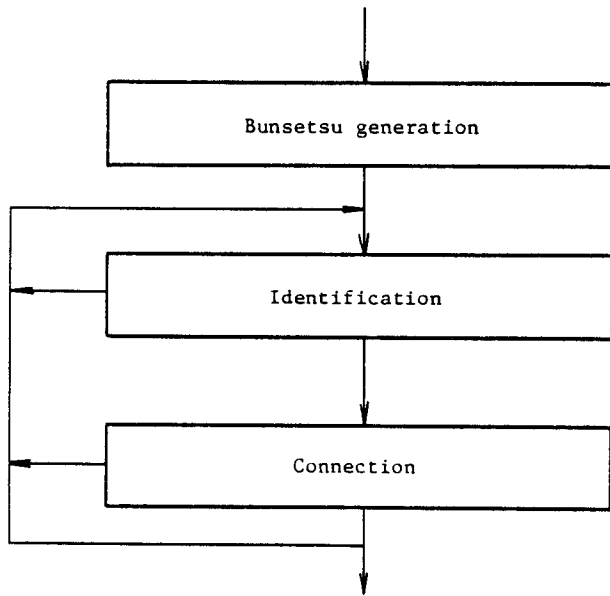


Figure 8. Parsing process.

Figure 8 shows the parsing process of the model-based parser. In each process, input sentences are scanned from left to right. In the first phase, 'bunsetsu' are generated from the word list. At the same time the parser attaches the object which is instantiated from the corresponding class to each 'bunsetsu'. The following identification and connection phases perform semantic interpretation using these instance objects, and determines the relationship between phrases. The identification process and connection process are executed repeatedly until all the relationship between phrases have been determined. The identification process has priority over the connection process, because a super-sub relationship represents a same concept generalization hierarchy and has stronger connectivity than an attribute relationship, the latter representing a property relation between different concepts. This parsing mechanism is very simple, allowing the user to expand each process easily. Each process consists of a number of production rules, which are grouped into packets according to the relevant syntactic patterns. Each packet has an execution priority according to the syntactic connectivity of each pattern. Thus the identification or addition of the rules are localized in the packet concerned with the modification. This simple parsing mechanism and the modular construction of the parsing rules contribute to the expandability of the parser.

Figures 9 and 10 show an example of parsing. This query means 'What is the name of the retailer in Geneva who sells commodity A?'. The morphological analyzer segments the sentence, and the model-based parser generates the phrases in the parentheses. The identification process is not applied to these phrases, because there is no super-sub relationship between them. Next, the model-based parser applies the connection process. The phrase 'Geneva' can modify the phrase 'commodity A' syntactically, but not semantically, because the corresponding classes, "Location" and "Commodity", do not have an attribute relationship. The phrase 'commodity A' can modify the phrase 'to sell' both semantically and

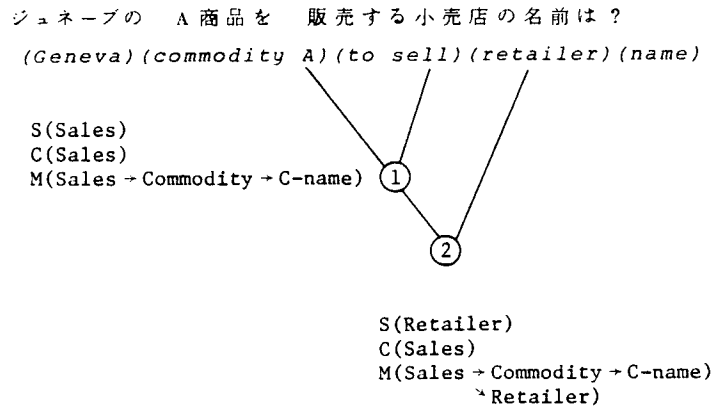


Figure 9. An example of parsing (1).

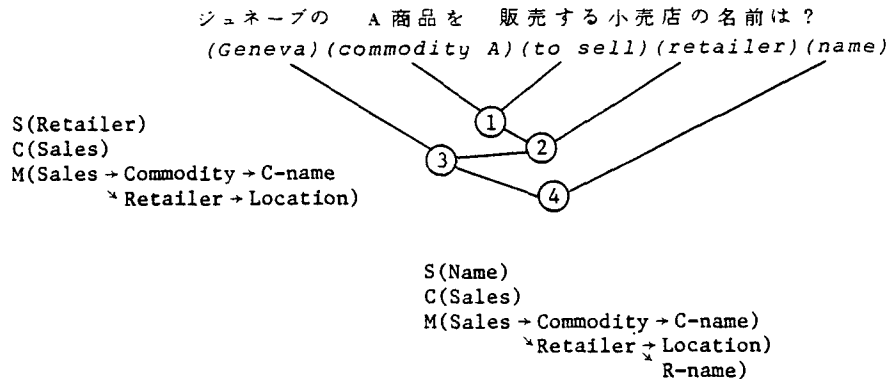


Figure 10. An example of parsing (2).

syntactically, because the classes "Commodity" and "Sales" have an attribute relationship. In this case, the predicate connection rule is applied, generating the unified phrase, node 1. The parser uses these three kinds of objects to check the connectivity. The syntactic object S represents the syntactic center of the unified phrase. In the Japanese-language the last phrase of the unified phrase is syntactically dominant. The conceptual object represents the semantic center of the unified phrase, and is determined by the identification and connection rule. The meaning objects M represent the meaning of the unified phrase using the sub-network of the world model. The predicate connection rule determines the sales class to be the conceptual object of node 1, because the sales class is the root class of the attribute relationship. The meaning objects are Sales --> Commodity --> Commodity name. The predicate connection rule also generates noun phrase node 2 and the S, C, and M of the node is determined as described in Figure 9. Next, the noun phrase connection rule is applied. This rule is applied to a syntactic pattern such as a noun phrase with a postposition 'no' followed by a noun phrase with any kind of postposition. The phrase 'Geneva' and the unified phrase 3 are unified to node 3 by the noun phrase connection rule (see Figure 10). This rule also generates node 4. The meaning of this sentence is that of node 4.

Errors or ellipses of postposition, such as 'no' or 'ga', are handled by packets which deal with the syntactic pattern. On the other hand, ellipses are handled by the special packets which deal with non-unified phrases based on the world model. These special packets have a lower priority than the standard packets. Different levels of robustness can be achieved by using the suitable packet for dealing with errors or ellipses.

#### CUSTOMIZATION

To customize the KID system to a specific application domain, the user has to perform several domain-dependent tasks. First, the user

makes a class network for the domain either from queries, which we call a top-down approach, or from the database schema, a bottom-up approach. Then, the user assigns words to the classes or attributes of the class network. Lastly, the user describes mapping information between classes and the database schema within the classes.

The world model editor supports these customization processes. The world model editor has three levels of user interface, in order to assist various users in editing the world model (see Figure 11). The first level is a construction description level, in which the user makes a structure of a class network. The second level is a word assignment level, in which the user assigns words to classes or attributes. These two levels are provided for end-users. The third level is a class- or word-contents description level. This level is provided for more sophisticated users, who understand the internal knowledge representation. The world model editor enables users to navigate any of the interface levels. Various users can edit the knowledge, according to their own particular view. Thus, knowledge base editing is made easier.

#### EVALUATION

We have applied KID to three different applications; housing, sales, and new drug tests. Figure 12 shows a result of an evaluation of KID. The target domain is a new drug test. We prepared 400 sentences for the evaluation. In a little less than a month, 91% of the sentences had been accepted. We decided a sentence is accepted, if the sentence is correctly analyzed and the correct data is retrieved from the database. We divided the 400 sentences into four groups and performed a blind test and a full capability test for each group, in stages. In the blind test, sentences are tested without changing any knowledge of the system. In the full capability test, we make all possible extensions or modifications to accept the sentences. The acceptance ratio of the blind test is improving, so we believe KID will soon become available for practical use.

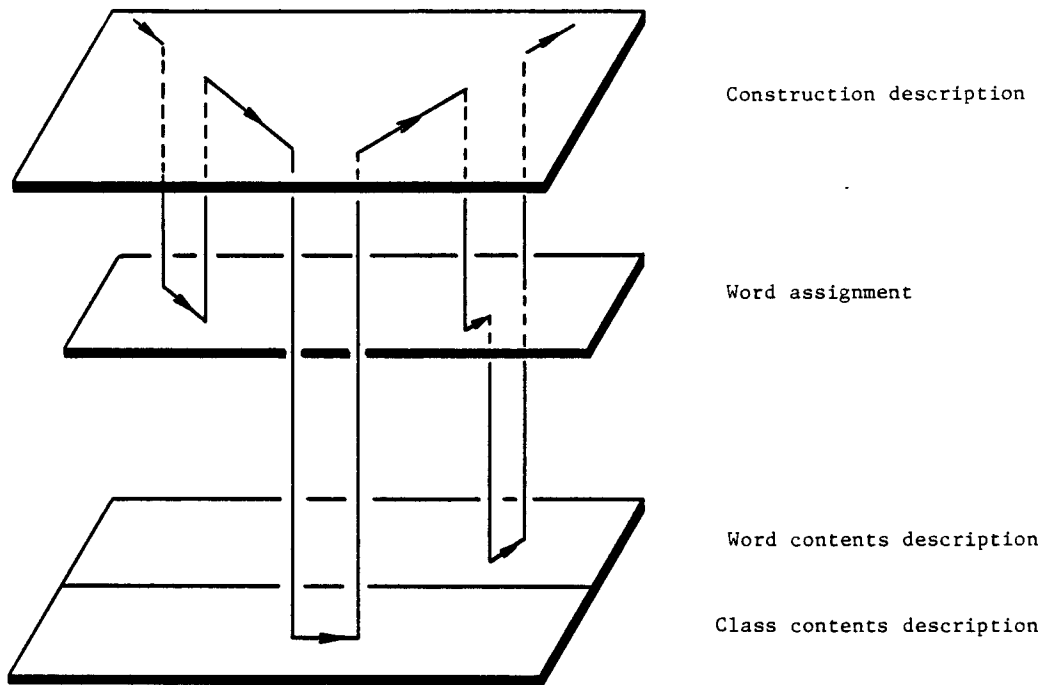


Figure 11. The world model editor.

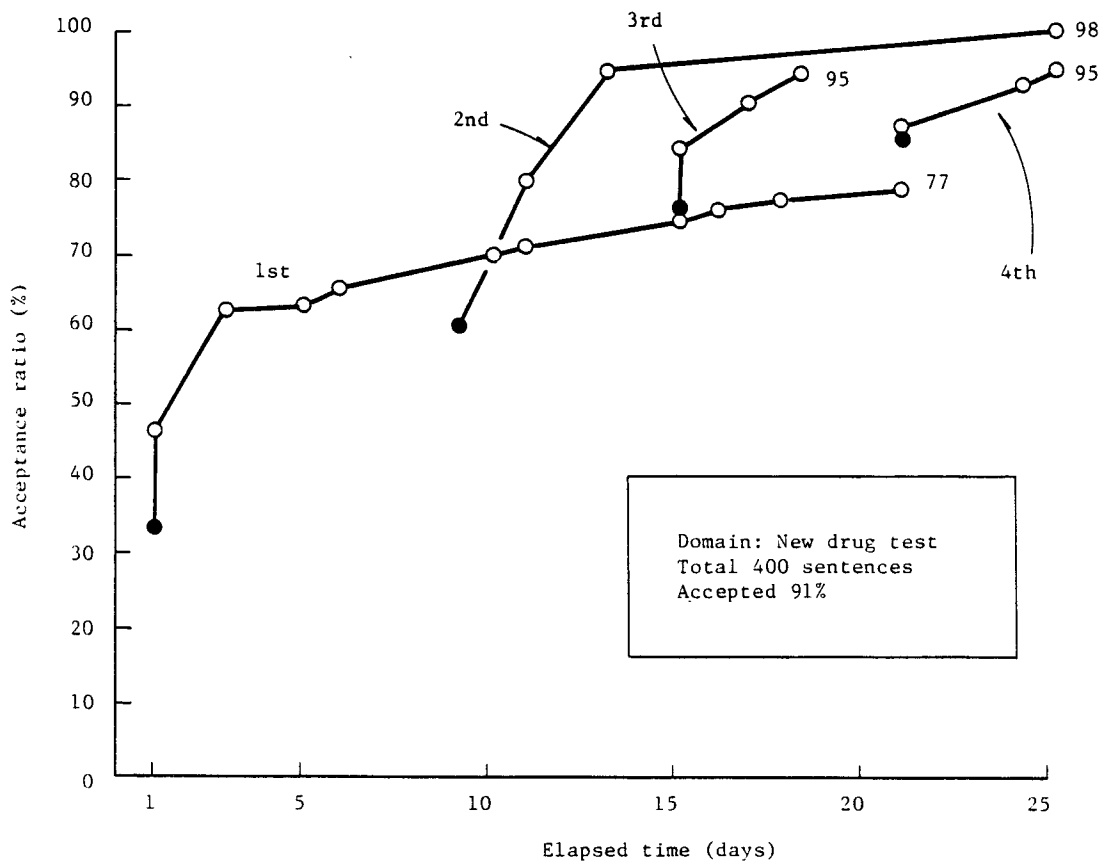


Figure 12. Evaluation of KID.

## CONCLUSIONS

In this paper, the three features of the Japanese-language interface KID were described. KID has both a simple mechanism of parsing and modularized grammar rules, so the parser is highly extendable. The semantic interpretation has clear principles based on the structure of the world model and syntactic information of the input sentence. Thus, the different levels of robustness are achieved by the adequate portion of the parser for dealing with the errors or ellipses. The world model integrates the domain-dependent knowledge separately. The user only has to customize the world model to a specific application. This customization is supported by the world model editor which provides various levels of user interfaces to make the world model editing easy for various users.

KID is now implemented as a front-end system for the relational database system (Makinouchi, 83). KID is implemented in Utilisp (Chikayama, 81), a dialect of Lisp. The morphological analyzer is 0.7 ksteps, the model-based parser is 2.3 ksteps, and retriever is 2.2 ksteps. The grammatical rule is 2.7 ksteps written in a rule description language, and is made up of 70 packets. KID uses several tools and utilities. The modeling system REALM is 2 ksteps, the world model editor is 1.3 ksteps, the window system is 1.7 ksteps, and the knowledge-based programming system, Minerva, is 3.5 ksteps.

We have several plans for future development. We will expand the system to accept not only retrieval sentences but also insertion, deletion, update, statistical analysis, and graphic operation. The parser coverage will be extended to accept wider expressions, including parallel phrases and sentences.

## ACKNOWLEDGEMENTS

To Mr. Tatsuya Hayashi, Manager of the Software Laboratory we express our gratitude for giving us an opportunity to make these studies.

## REFERENCES

- Bobrow, D. G., Stefik, M., The LOOPS Manual: A Data Oriented and Object Oriented Programming System for Interlisp, Xerox Knowledge-Based VLSI Design Group Memo, 1981, KB-VLSI-81-13.
- Chikayama, T., Utilisp Manual, METR 81-6, Mathematical Engineering Section, University of Tokyo, 1981.
- Ginsparg, J. M., A Robust Portable Natural Language Data Base Interface, Proc. Conf. Applied Natural Language Processing, 1983, pp.25-30.
- Grosz, B. J., TEAM: A Transportable Natural-Language Interface System, Proc. Conf. Applied Natural Language Processing, 1983, pp.39-45.
- Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., Slocum, J., Developing a Natural Language Interface to Complex Data, ACM TODS, Vol. 3, No. 2, 1978, pp.105-147.
- Izumida, Y. et al., A Database Retrieval System Using a World Model, Symposium on Database System, 43-2, Information Processing Society of Japan, 1984 [in Japanese].
- Makinouchi, A. et al., Relational Database Management System RDB/V1, Transactions of Information Processing Society of Japan, Vol. 24, No. 1, 1983, pp.47-55 [in Japanese].
- Nilssen, N. J., Principles of Artificial Intelligence, Tioga, 1980.
- Waltz, D. L., An English Language Question Answering System for a Large Relational Database, Communication of the ACM, 1978, 27(7), pp.526-539.
- Yoshimura, K., Hitaka, T., Yoshida, S., Morphological Analysis of Non-marked-off Japanese Sentences by the Least BUNSETSU'S Number Method, Transactions of Information Processing Society of Japan, Vol. 24, No. 1, 1983, pp.40-46 [in Japanese].