ICOT Technical Memorandum: TM-0599

TM-0599

Knowledge Representation and Reasoning for Discourse Understanding

by

S. Kinoshita, H. Sano, T. Ukita,
K. Sumita and S. Amano

October, 1988

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1 Chome
Minato ku Tokyo 108 Japan

(03) 456 3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Knowledge Representation and Reasoning for Discourse Understanding*

Satoshi KINOSHITA, Hiroshi SANO, Teruhiko UKITA,
Kazuo SUMITA, and Shin'ya AMANO

Toshiba Corp. R&D center
1, Komukai Toshiba-cho, Saiwai-ku,
Kawasaki 210, Japan

ABSTRACT

Extra-linguistic knowledge is necessary for discourse understanding. In this paper, we classify the knowledge, and present a framework to describe it using frames and rules. With this framework, it is easy to represent an IS-A hierarchy, which is based on a classification by different viewpoints, and to describe functions of objects as declarative knowledge. Furthermore, to treat ambiguities in discourse understanding and to process the utterance based on assumptions, the system has a world mechanism for inference. Lastly, we report the appraisal of this framework through the knowledge representation of a VCR and the conversation experiment by the dialogue system.

1. Introduction

Context processing technology is the most important in building a question-answering system which reads sentences and answers the user's questions, or a machine translation system which translates sentences correctly, based on their contexts. Determining anaphoric reference and complementing ellipses are one of the most fundamental problems in context processing, and they are especially difficult in understanding conversations.

For context understanding, it is obvious that the extra-linguistic knowledge and reasoning based on it are necessary as well as the linguistic one. However, how the extra-linguistic knowledge should be represented and used has not been fully discussed. In this paper, we propose a framework for representing the extra-linguistic knowledge, which is a multi-paradigm knowledge representation system based on frames and rules.

Knowledge representation based on only frames is insufficient in the following points.

(a) In classifying the concepts in a task domain, we can classify them in a different way by changing the view point, but frames cannot express the taxonomy based on such classification simply.

(b) Negative expression and inference concerning negation are insufficient.

(c) We cannot describe the behavior of objects and logical relations between attributes as declarative

* This work is supported by ICOT(Institute for New Generation Computer Technology).

knowledge.

The framework we propose here solves these problems. Moreover, the inference system is based on a *world mechanism*, which offers multiple hypothetical worlds for discourse analysis and problem solving.

We have been developing a consulting system which helps the user to use electrical equipment such as a VCR(video cassette recorder) through a conversation with him/her. We then report the appraisal of the framework, which is obtained through the description of knowledge for this task and experiment of the consulting system.

2. Knowledge and inference for context understanding

2.1 Knowledge for context understanding

Let's discuss "context understanding" before thinking about "knowledge" for it. We use the term "context understanding" in two meanings. One is understanding context itself, that is, understanding the contents of whole sentences or understanding the topic or the structure of the conversation. The other is understanding each sentence correctly in the flow of the context, that is, determining the meaning of a sentence by resolving anaphoric reference and ellipsis using the information of the context. But these two meanings are complementary. If we can't understand the meaning of each constituent sentence, we can't understand the meaning of the whole piece of text, and in some cases, understanding one sentence helps understanding the whole. Furthermore, if an ambiguity exists, in a sentence, that can't be resolved in semantic analysis, the context may be able to give the answer. Since an utterance in conversations depends on its context very much, the latter is more important. In this paper, we will use the term mainly in the latter meaning.

Needless to say, there are two kinds of knowledge for understanding a sentence. One is linguistic knowledge such as a grammar, and the other is extra-linguistic knowledge, knowledge on the world around us. In this paper, we focus on the latter and just call it "knowledge".

As knowledge necessary for context understanding, we classify the knowledge into 4 categories as listed below.

(a) Knowledge of objects

(b) Knowledge of events

(c) Relationships between events

(d) Concept hierarchy (thesaurus)

(a) Knowledge of objects

We use the term "objects" to distinguish entities or things, with, or without, physical existence. Intuitively they correspond to what are denoted by nominals. Knowledge of objects in a task domain is essential for understanding a sentence.

When we read a sentence, we sometimes can't understand, nor even imagine, the meaning of the whole sentence even though we can understand the meaning of each word. This sort of situation may be caused by lack of knowledge of what properties the object in the sentence has by nature. We think this is the same reason that Minsky proposed frames to recognize objects[Minsky 75]. The most systematic knowledge of an object is a model. Having a model of an object helps the understanding of sentences about it.

(b) Knowledge of events

"Events" distinguish some kind of action, state or movement of an object. If we talk about something, we can't describe it without a predicate and other objects, i.e. in a form of an event. So we must have knowledge of events. Although this seems to be rather linguistic, knowledge of the kind of object that may be an event agent or object is strongly connected to the world.

(c) Relationships between events

Determining anaphoric referents and complementing elliptic information are the most fundamental problem of context understanding. For this, we have to recognize the relationships between the sentences being processed and the context. For example, Schank proposed the method of using knowledge about the typical sequence of events that is called script [Schank 77], but it cannot analyze sentences which cannot be matched to the scripts, this means it lacks flexibility. In order to solve this problem, Wilensky tried to recognize the relationship between events as a goal and a plan for it [Wilensky 83]. We have this sort of knowledge and utilize it to understand sentences. In our current task, we recognize knowledge of procedures to use a VCR.

(d) Concept hierarchy

Concept hierarchy is necessary in context understanding because we use various expressions to refer the same object in a dialogue. Let's consider the following sentences.

(1) Did you push the play-back switch?

(2) Yes, I pushed the switch.

Except using pronouns, we use expressions that mean the different concepts as in the example above: a 'switch' is a super class of a 'play-back switch'.

2.2 Inference for context understanding

We can see various kinds of inference in the process of context understanding (for example [Rieger 75 and Ishizaki 86]). Here we will discuss an inference framework for treating ambiguities of interpretation.

The most important requirement for an inference system, for context understanding, is a framework to describe the situations according to the various interpretations of the sentence simultaneously, if it has an ambiguity. As an example, let's consider the utterances below.

(3) I put a cassette tape into the VCR, and pushed the play-back switch.

(4) But it didn't work.

Assume that the candidates for 'it' are 'cassette tape 001', 'VCR 001', and 'playback switch 001' ('001' denotes instances). Then the hearer(system) can assume three situations as interpretations shown in Figure 1. We call each situation described in a box a 'hypothetical context world' or simply a 'context world'.

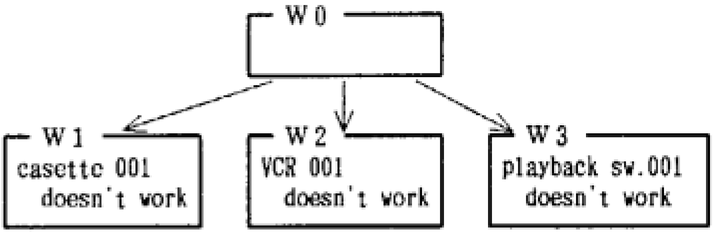

Figure 1 Example of context worlds

The process of context understanding is to select the most appropriate context world. To do so, we must evaluate every possible interpretation. But in some cases, we can't evaluate them unless we infer from the interpretation, which may change the contents of the working memory, and for that, we must be able to create context worlds simultaneously.

The interpretation which is decided by the context analysis is not always correct. If the system finds an inconsistency, it must resolve it. In the field of expert systems, Doyle proposed an inference system called TMS [Doyle 79]. A context understanding system should have this sort of inference system, and the world mechanism is a key to its realization.

Considering human language activity, we can find that we talk about not only objects and events in a real world but also those in an imaginary world. So the system should process the utterance based on conditionals. We think this is possible by utilizing context worlds. For processing such sentences, the system has only to create a new context world and process the following sentences in this world. The context world which corresponds to the imaginary world *per se* differs from the one which is created for context understanding, but can be processed in the same framework.

3. Design and realization of the knowledge representation system

As stated earlier, we are developing a consulting system which guides the usage of electrical equipment, such as a VCR, through a conversation with the user. In this chapter we show the concrete framework for knowledge representation of the task domain and the inference mechanism for understanding the user's

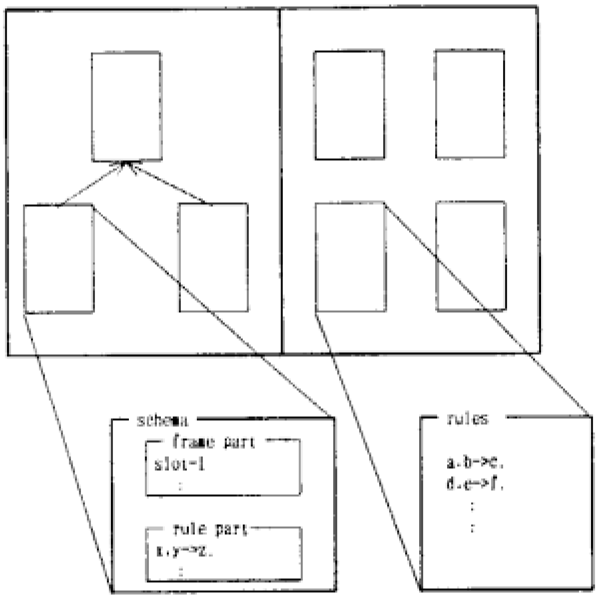

Figure 2 Composition of Knowledge Base

```
schema(cls, vcr_model_750,
        [(superC, type_of_sound_recording, [(value,[hi_fi_vcr])]),
         (superC, type_of_casette, [(value,[vhs_vcr])]),
         (has_part,        [(value,[power_switch, play_back_switch,...])]),
         (status,          [(car_num,1 ),
                          (enumeration,[stand_by,play_back, record, ...])]),
         (power,           [(car_num,1 ), (enumeration,[on,off ])]),
         (channel          [(car_num, 1),
                            (value_class, integer),
                            (value_condition, (_,V,(V>=1,V=<12))) ])
        ],
        [s_rule(power, power_on,VCR,
                          ( kr_schema(VCR,[(power,[off])],true),
                            event(push,1,[(object,[SW])])  ),
                          kr_schema(VCR,[(power,[on])],true),
                          [('$var_constraint',(
                             SW#power_switch :- part_of(SW,VCR) ))]),
        s_rule(power,power_lamp_on,VCR,
                          kr_schema(VCR,[(power,[on])],true),
                          kr_schema(LP,[(lamp_status,[on])],true),
                          [('$var_constraint',(
                               LP#power_lamp :- part_of(LP,VCR))])
 ]).
```

Figure 3   Schema example for a VCR (a part).
Kr_schema is a predicate that extracts or verify slot values.
Words beginning with a capital letter denote variables.

utterance.

There are a variety of knowledge representation paradigms such as a semantic network, and the features of each have been discussed. We use 'frames' and 'rules' because of the ease of describing the concept hierarchy and logical relation and of managing the described knowledge.

Figure 2 illustrates the composition of a knowledge base in the knowledge representation system. Knowledge of objects and events is described in a framework called a 'schema' that is composed of a frame part and a rule part, and knowledge of the relations between events, such as the causal relations, is described as a set of rules.

3.1 Knowledge representation

3.3.1 Knowledge representation using schema

Figure 3 shows a schema example(a part of the full description of a VCR). The schema is composed of a frame part and a rule part. Static knowledge is described in the frame part: such as super/sub class relations which composes an IS-A hierarchy, whole-part relations, and attributes of the object, and in the rule part, actions or the behavior of the object are described.

There are two types of schemata in the knowledge base. One is an instance schema and the other is a class schema. Roughly speaking, an instance schema corresponds to an real object, and the information of the object is described in it. A class schema corresponds to a class of objects, and the restrictions of attributes or possible behavior of the objects are described.

(A) Expression of knowledge by the frame

As stated above, static knowledge such as an IS-A hierarchy is described in the frame part. As KRL [Bobrow 77], a frame is defined as a set of slots, and each slot is a set of facets. The features of the frame system in this paper are described below.

Representation of the concept hierarchy

We can classify one concept in a different way by changing the viewpoint, that is on what feature we see

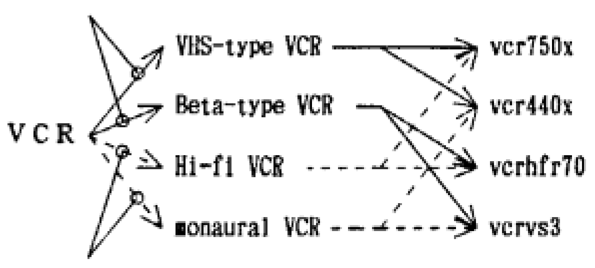

Figure 4 Example of classification by different viewpoint

it for discrimination. Figure 4 is a part of an IS-A hierarchy of our current task domain. If we classify a VCR, it can be classified in two different ways. Considering the type of cassette tape, we can classify it into 'VHS-type VCR' and 'Beta-type VCR', but by the type of sound recording, it is classified into 'hi-fi VCR' and 'monaural VCR'. So far, it is difficult to describe such a hierarchy naturally, using conventional frame systems. In our system, it is described quite easily and naturally by attaching the viewpoint of the classification as additional information for a slot (See superC slot of Figure 3).

This attachment of viewpoints is used in judging anaphoric referents. In a conversation, we use different expressions to refer to the same object, and sometimes they mean different concepts. In such a case, we can use the hierarchical relation between the concepts for the judgement.

Let's consider checking of identification of two objects; one is in the context and the other is in the sentence being processed. Assume that B is a concept of the former, and A is of the latter. If A and B are different, there are four possible relations between A and B.

(a) A is a sub-class of B.

Example: A:VHS VCR and B:VCR

(b) A and B are both sub-classes of a common concept C in the same view point.

Example: A:television, B:VCR, and C:electrical equipment

(c) A and B are both sub-classes of a common concept C in the different view point.

Example: A:VHS VCR, B:hi-fi VCR and C:VCR

(d) B is a sub-class of A (opposite case of a).

Example: A:VCR and B:VHS VCR

Figure 5 illustrates the above classification. An arrow between concepts is a subC(sub-class) link, and a dotted line around arrows is the viewpoint for the classification. The case (a) is the most general case. If two objects don't have any inconsistent property, we can judge that they are same. In case (b), since these two objects essentially have inconsistent properties, we cannot recognize that they are the same object, but in the case of (c), it is possible to assume they are same. The case (d) is the opposite case of (a) . In this case, by judging that they are same, the information about the preceding object will increase.

The above example illustrates the case that a sub-class relation between two classes is direct. The same procedure should be applied for indirect relations among concepts.

Negative expression and the open/closed assumption

For a system which performs problem solving through a dialogue with a user, most data necessary for problem solving are unknown at the begining of inference. Furthermore an input sentence by the user may be a negative sentence. Although many systems based on Prolog treat negation on a closed world assumption, this is not always appropriate.

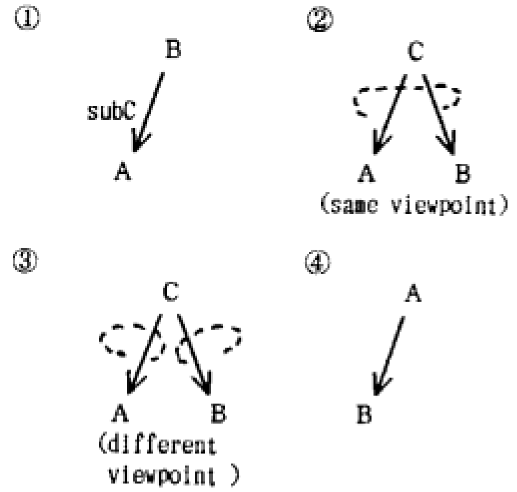

Figure 5 Relations between concepts

```
schema(cls, record,
  [(superC,         [(value, [action])])]),
   (agent,          [(value_class, human_being)]),
   (object,         [(value_class, tv_program)]),
   (tool,           [(value_class, vtr)]),
   (obligatory_case,
                    [(value, [object, tool])] )   ], []).
```

Figure 6 Event expression by schema.

```
rule( operation, play_back,
   seq( event( put_on, 1, [( object,[PW])]),
         event( insert, 1, [( object, [K]), ( goal, [VTR])]) ,
         event( push, 1, [( object, [SW] )]) ),
   event( play_back, 1, [( object, [K]), ( tool,[VTR])] ),
 [('$var_constraint',(
     VTR # vtr;
     K # cassette;
     PW # power :- attribute_of ( PW, VTR) ;
     SW # play_back_switch :- part_of ( SW, VTR) ) )]).
```

Figure 7 Operation procedure example by rule.

Solving this problem, we must have a framework to describe negation explicitly and also to describe whether the data used for reasoning is 'open' or 'closed'. 'Open' and 'closed' indicate the assumption for inference. If the system holds all the possible data about an item, we say the data is closed for the item, and if there is a possibility of another data other than the current one, we say the data is open.

In our knowledge representation, we can set a negative value as a slot value and add the information that the data for the slot is 'open' or 'closed'. With this framework, we can use the open/closed world assumption properly in comparison of slot values and reasoning based on them.

Knowledge of events

We use case representation to describe an event. Using a schema, a case representation is described quite naturally by representing each case as a slot. Figure 6 shows an example description for a verb "rokuga-suru(to record something on a video tape)", where slots (agent, object and tool) have a special facet describing constraints on fillers of these cases. In this example, the constraint is described that the agent, object and tool must be a human, a TV program and a VCR respectively. The requirement for slot fulfillment, in implementing a task, is described in 'obligatory-case' slot. These constraints are used to reduce search space in interpreting an input sentence.

(B) Knowledge representation using schema rule

In addition to static knowledge, there is knowledge of functions of an object or logical relations between attributes. We use a rule in a schema called a 'schema rule' to describe this sort of knowledge.

In the case of a VCR, we have to represent how the internal state of a VCR changes by its operation. Figure 3 shows two schema rule examples. The first one describes that, if the power-switch of a VCR is pushed when its power is 'off', then the power will be 'on'.

To describe such knowledge using only a frame, we must describe it using attached procedures like if-added demons. Since there are many functions for usual devices, the clearness of the knowledge will be lost. By describing such knowledge using a schema rule, we can describe it as declarative knowledge, and the relation between the elements of knowledge will be kept clear.

Describing such knowledge of an object bit by bit, we get its model, and by forward reasoning with the knowledge, we can simulate its behavior. Lastly, these schema rules are inherited by lower classes like the slots of a frame and, therefore, economical description of functions is possible.

3.1.2 Knowledge representation using rule

In addition to knowledge described in a schema, we have to represent the relations between events such as a causal relation. In our framework, we use 'if-then' rules to describes knowledge of this sort.

For the current task, rules are used to represent procedures to operate a VCR and are prepared for every operational procedure, such as play-back and recording. Figure 7 is an example of a rule that describes how to play back a video cassette. The conclusion part holds the goal(to play back), and the conditional part holds the actions to accomplish it. Furthermore, every rule has some constraints for the variables. In the example, the variable 'SW' has the constraint such that its content must be an instance of 'play-back switch' and be part of a VCR.

3.1.3 Representation of time

In a task which has only to treat the present state of the apparatus, such as trouble diagnosis, there is little necessity to represent time. However, in guiding the usage of an apparatus, the order of user's operations is quite important. Since we have a mechanism to describe context worlds, it is possible to represent the states of objects at the various points of time by creating worlds for every point of time in consideration, but this representation makes it difficult to manage the context worlds. For this reason, we extend the data structure of a slot value of our frame system to be the pairing of its value and a time description. So every slot may have various values according to time.

3.2 Inference function

As a basic function, the system has an inference mechanism, for frames, to inherit slots or to check the restrictions of a slot, and a mechanism of forward/backward reasoning using rules. As we described, we have two kinds of rules; a schema rule in a schema and a rule in the rule base. In consultation of an apparatus, it is important that the system holds the state of the machine and predicts a possible state based on a simulation. In our system, this is performed by forward reasoning using schema rules.

Moreover, to realize the context world, we designed the system to create a working memory for reasoning according to each context world. Hereafter we call each working memory simply the 'world'. Using this mechanism, we can create context worlds for each possible interpretation in discourse analysis, and evaluate their properties simultaneously. Also in problem solving, we can process a sentence based on a conditional using this framework. This will be described in 4.2.

3.3 Implementation

We implemented the knowledge representation system on a PSI(Personal Sequential Inference machine) developed by ICOT. The programming language is ESP(Extended Self-contained Prolog). The size of the program is about 7000 lines including comment lines.

Since ESP is a kind of object oriented language, it is possible to translate all knowledge written in a schema to an object of ESP. However, in the current system, for ease of debugging and loading speed, we transform a set of schemata in one file to an object of ESP, and load it into the main memory.

Talking of execution speed, it is not satisfactory because the system processes the frame data by interpretation. By compiling frame data to a Prolog program like DCKR(Definite Clause Knowledge Representation)[Tanaka 86], we can make the system faster.

4. Application and appraisal

4.1 Application in a dialogue system

We are developing an experimental question-answering system called ISAC(Information Service System by Analyzing Conversational Context) whose current task is consultation for operating a VCR, and we use the knowledge representation system as the knowledge module of the system.

Starting with SHRDLU[Winograd 72], many dialogue systems have been developed. The biggest difference from SHRDLU is, in SHRDLU, the world for conversation is closed and the system holds all information for inference, but in our system, the system can get information about the world only through the user's input. This corresponds to the situation whereby an expert could consult via telephone conversation.

discourse analysis

Here we briefly introduce a procedure of discourse analysis for interpreting an input sentence[Ukita 88]. The procedure consists of four major steps; anaphora detection, referent candidate extraction, non-contradiction testing, and optimal candidate selection.

First the procedure detects anaphoric indicators for a sentence, which are pronouns or nouns, with or without definite articles. Furthermore the omitted obligatory cases of the events(predicates) are detected as ellipses.

Next, for every anaphoric indicator, the procedure searches for referent candidates that have already emerged in the context. In the system, they are usually defined as instance schemata. If the procedure can't find an appropriate candidate for an indicator, it creates a new instance schema and treat it as the candidate. The inter-object relationships such as whole-part or attribute-value are also checked here.

When more than two possible interpretations remain after the non-contradiction test, the procedure tries to find causal links between the interpretations and the context, and decides the most informative interpretation. This is a step of optimal candidate selection. In inspecting causal connectivities, the procedure searches for rules that describe the usage of a VCR. Let's consider an example sequence of sentences, whose preceding context contains a cassette tape and a practical image on a television set.

(5) I pushed the eject button. It didn't come out.

Assume that the referent candidates of "it" are the cassette tape and the image. To decide the appropriate referent, the procedure finds a rule such as follows:

If (a cassette is inserted in a VCR),

and (the eject button is pushed),

then (the cassette comes out).

Using this rule, the procedure takes preference of the interpretation that the cassette didn't come out.

Currently we don't use schema rules which are described in a schema for this step. In referent candidate extraction, if an anaphoric indicator is a noun, the procedure searches for an object that is an instance of a concept meant by the noun. This is just the case in (a) and (c) described in Figure 5. We have to improve the discourse analyzing program to treat other cases.

Problem solving

The current task of the consulting system is categorized into 3 sub-tasks as follows.

(a) Answering inquiry about an operating procedure

(example: Would you tell me how to play back this tape?)

(b) Solution of fault which originates in an operation mistake

(example: I pushed the play-back switch, but it didn't work.)

(c) Prediction of an action of the VCR

(example: What happens if I push the play-back switch now?)

An input sentence is classified into one of these types, and the problem solving module for each type is activated. Finally the result of the task is output as a question or an answer to the user.

Dialogue example

Figure 8 shows a conversation example using the experimental system. Before this conversation, the system was given the situation that there is a VCR and a television set, and that the initial state of the devices is such that the power is off. Unless such an initial state is given, the system asks the user about them or his/her actions, and continues the consultation.

4.2 Appraisal of the framework of knowledge representation

We report the appraisal of our framework of the knowledge representation, which is got by describing the knowledge about a kind of home hi-fi VCR. Using its operation manual, we described the knowledge of the VCR, but neither knowledge concerning spatial relations, such as positions of switches, nor knowledge, which requires adjectival or adverbial expression, is described.

4.2.1 Representation using schema

The number of concepts which are described in schemata is about 300. 120 schemata are concepts of user: ビデオテープを再生したい (I want to playback the video-tape)*

system: ビデオの電源を入れて、ビデオにカセットテープを入れて、再生ボタンを押してください
(Please turn on the VTR power, insert the cassette into the VTR, and push the play-back button.)

user: ビデオにカセットテープを入れて、再生ボタンを押した
(I inserted the cassette in the VTR and pushed the play-back button)*

system: はい、わかりました (OK)

user: 動かない ("It doesn't operate" or "It doesn't move")**

system: ビデオの電源はオンですか (Is the VTR power 'ON'?)

user: はい (Yes)

system: 再生ランプは点灯していますか (Is the play-back lamp lit?)

user: 点灯しています (It's lit)*

system: VHF出力端子はテレビに接続されていますか
(Is the VHF output terminal connected to the TV set?)

user: いいえ (No)

system:VHF出力端子をテレビに接続してください
(Please connect the VHF output terminal to the TV set.)

Figure 8 Dialogue example using the experimental system

* In Japanese, the subject is omitted.

** Japanese verb '動く (ugoku)' has multi-meanings.

events (verb relevant to an operation of the VCR), and 80 schemata are concepts of the VCR itself and its constituents, such as switches and lamps, and the rest are concepts of the attributes, such as channel or an immaterial object, such as a television program or a picture.

Representation of concept hierarchy

In describing an IS-A hierarchy, we could represent a classification based on different viewpoints easily and naturally. In the current hierarchy, there are only two sub-hierarchical structures: one is shown in Figure 4, and the other is a sub-hierarchy whose root concept is a 'terminal'. However we found the framework is quite valid.

Negative expression and the open/closed world assumption

Since, in the current task, each slot can only have one value, we are unable to use the open/closed assumption to its full advantage. Speaking of negation, when the knowledge representation system gets a command to fill a slot with a negative value, it will try, first, by *reductio ad absurdum* to infer a positive one. If this fails, the negative one will be used.

For example, since the slot 'power' in Figure 3 may only have the value 'on' or 'off' at one time, the slot will be set to 'off' if the negation of 'on' is provided. In a consultation, the user may answer the system's question with a negative sentence. So this reasoning is very useful to process such a sentence.

Representation using schema rules

The functions of the VCR are described in about 170 schema rules: how it behaves according to an operation and how the state of the lamps is affected. Since these rules describe its model, we can simulate its behaivior, and this reasoning is quite useful for problem solving.

Currently we can't describe the events which happen continuously or iteratively under a certain state. For example, "If a VCR is in playback state, the cassette tape in it goes around" or "the display of the VCR's clock changes every minute". To describe this kind of knowledge, we have to extend our framework.

4.2.2 Representation using rules

The operation procedures of the VCR are described in about 30 rules. In describing the knowledge, we used the expressions in the operation manual au much as possible. However, we can't describe the following expressions as they are.

(a) The case that the purpose or the condition for an operation is described.

(example: "Please push the video button so as to put on the video lamp.")

In such a case, we neglected the purpose or the condition and wrote only the operation .

(b) The case that the goal of an operation is expressed as two events.

(example: "To watch a TV program while you record another program, ...")

In this example, we used a word that means 'a program on a different channel', but this is not always possible.

4.2.3 Inference in the world mechanism

In processing a conditional sentence (we will call it a 'hypothetical sentence'.), we found inference within the world mechanism to be useful. To process a hypothetical sentence, the system first creates a new world, evaluates it's potential, and continues the dialogue. Through the experiment, we found that our choice of world is important for processing the next sentence. As an example, let's consider a sequence of the user's input during a consultation.

(6) What happens if I push the recording switch now?

(7) What happens if I push the play-back switch instead?

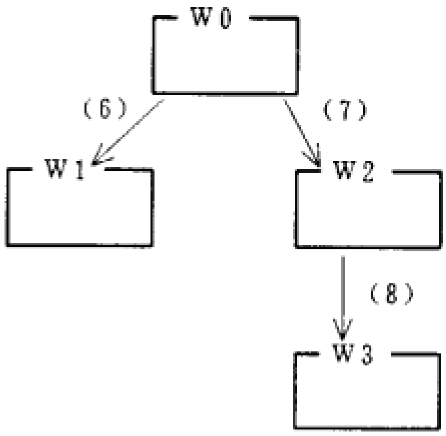

Figure 9 Example of world composition

(8) What happens if I push the fast forward switch after that?

Figure 9 illustrates an example of world composition in processing the sentences above. The world W0 is a world at which sentence (6) was input, and W1 is a world which is created in order to process (6). In processing (7), the supposition (pushing the play-back switch) must be evaluated instead of the supposition of (6), that is, we have to create a new world W2 whose parent is also world W0. Furthermore, in processing (8), we must evaluate it in the light of the supposition of (7), therefore we have to create W3 as a child of W2. For this, it is very important to manage relations between the context and the worlds.

5. Conclusions

In this paper, we discussed a classification of knowledge for context understanding, and a framework for it's representation and reasoning based on this knowledge. We first classified the knowledge into 4 categories; knowledge of objects, knowledge of events, concept hierarchy, relationships between events. The first three are described in a framework called a schema, and the last one is described by rules. Using schemata, we can naturally represent an IS-A hierarchy based on a different viewpoint, and also describe a negative slot value, and use the open/closed assumption properly. Furthermore, by using schema rules, we can describe functions of an object and logical relations between attributes declaratively. As for inference, we pointed out the necessity of a context world for resolving an ambiguity in context understanding, and it's validity in problem solving.

Next we reported the knowledge representation system which is implemented on a PSI, and through the knowledge representation of a VCR, we found that these points to be very valid. At present, there is some knowledge that can't be written in the current framework, but the framework reported here is thought basically applicable to context understanding in other task domains.

References

[Bobrow 77] Bobrow,D.G. and Winograd,T., "An Overview of KRL, a Knowledge Representation Language", Cognitive Science, Vol.1, 1977, pp.3-46.

[Doyle 79] Doyle,J., "A Truth Maintenance System", Artificial Intelligence 12, 1979, pp.231-272.

[Ishizaki 86] Ishizaki,S. and Isahara, H., "Contextual Processing Technology", Vol.27, No.8, 1986, pp.897-905(in Japanese).

[Minsky 75] Minsky,M., "A framework for representing knowledge", in Winston,P. (ed.), The psychology of computer vision, McGraw- Hill, 1975, pp.211-277.

[Rieger 75] Reiger,C.J., "CONCEPTUAL MEMORY AND INFERENCE", in Schank,R. (ed.), Conceptual Infomation Processing, NORTH-HOLLAND, 1975, pp.157-288.

[Schank 77] Schank,R. and Abelson,R., "Scripts,Plans,Goals and Understanding", Lawrence Erlbaum Associates, 1977.

[Tanaka 86] Tanaka,H., "DCKR - Knowledge Represetation in Prolog and Its Application to Natural Language Understanding", in Nagao,M.(ed.), Language and Artificial Inteligence, Else Ver Science Publishers, B.V. (North Holland), 1987, pp.353-366.

[Ukita 88] Ukita,T., Sumita,K., Kinoshita,S., Sano,H., and Amano,S., "Preference Judgement in Comprehending Conversational Sentences Using Multi-Paradigm World Knowledge", Proc. of FGCS'88, 1988(to appear).

[Wilensky 83] Wilensky,R., "Planning and Understanding", Addison-Wesley, 1983.

[Winograd 72] Winograd,T., "Understanding Natural Language", Academic Press, 1972.