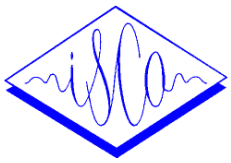

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

DIALOG MANAGEMENT SYSTEM MASCOTS IN SPEECH UNDERSTANDING SYSTEM

Tetsuya Yamamoto † *Yoshikazu Ohta* †
Yoichi Yamashita ‡ *Riichiro Mizoguchi* ‡

†Faculty of Engeneering, KANSAI University
3-3-35 Yamate-cho, Suita-city, Osaka, 564 JAPAN

‡The Institute of Scientific and Industrial Research, OSAKA University
8-1 Mihogaoka, Ibaraki-city, Osaka, 567 JAPAN

Abstract

Many kinds of expert systems have been developed in various fields to date. To realize interaction through spoken Japanese between a user and such an expert system, we are currently developing a dialog management system called MASCOTS in addition to the speech understanding system and the speech synthesizer. MASCOTS manages the complex flow of dialog using two stacks and plan information. It sends useful information tailored in appropriate forms to the expert system and helps the language processing system by predicting the next user utterance. This paper describes the architecture of MASCOTS focusing on exchanges of the information with the language processing system.

1 INTRODUCTION

Recently many kinds of expert systems have been developed in various fields. In order to realize communication through spoken language between a user and such an expert system, we need a speech understanding system for the user utterance and a speech synthesizer for the system utterance.

To this end, we have been developing SPURT-I(SPeech Understanding system with Rule-based and Topic-directed architecture)[1] as the speech understanding system which accepts Japanese utterances spoken in every syntactic unit ("bunsetsu") and outputs a word sequence corresponding to the input speech.

During dialog, however, users do not always return responses expected by the system. For instance, when the system asks a question, the user may ask another question instead of answering it. Furthermore, the representation of the user utterance takes various forms even though it might have the same meaning.

Therefore, it is very useful to find out the kind of the utterance and to standardize its representation form before sending the recognized utterance to the expert system. In addition, SPURT-I can recognize utterances more efficiently if it uses information of the characteristics of dialog.

With this view, we are currently developing a dialog management system MASCOTS(MAnagement System for COnversation using Twin-stacks and Sr-plan)[2] which manages the complex flow of dialog using two stacks and plan information.

MASCOTS has three functions : the first is to find out the kind of the user utterance by identifying the correspondence between utterances made by the user and the system, the second is to send the relevant information contained in the user utterance in a standardized representation to the system and the last is to help the language processing subsystem ASP(ASsociation-based Parser)[3] in SPURT-I by providing it with useful information for its disambiguation process.

This paper describes the architecture of MASCOTS focusing on exchanges of the information with ASP.

2 OUTLINE OF MASCOTS

2.1 Total interface for communication

Block diagram of the total system configuration is shown in Fig.1. The utterance of the user who works with the problem solving system is input to acoustic processing subsystem named SPREX(SPeech Recognition EXpert) and is converted to a sequence of phonemes. Then the phoneme sequence is transformed into a set of conflicting candidates in every "bunsetsu" by BCG(Bunsetsu Candidate Generator). ASP accepts a candidate lattice of "bunsetsu"s and identifies a correct sequence of them. On the other hand, the speech synthesizer generates natural speech from the outputs of the problem solving system and utters it to the user.

MASCOTS locates between these speech processing systems and the problem solving system, and manages the dialog among them. Processing in MASCOTS is tightly connected to ASP, so these two systems share some parts of their knowledge bases.

2.2 Characteristics of dialog

MASCOTS deals only with goal-oriented dialogs such as ones appearing in consultation, information retrieval, CAI and so on.

In such dialogs, an utterance, which of course does not always consist of one sentence, contains at least either of a stimulus(request, order and so on) to the opponent or a response to the stimulus given by the opponent explicitly or

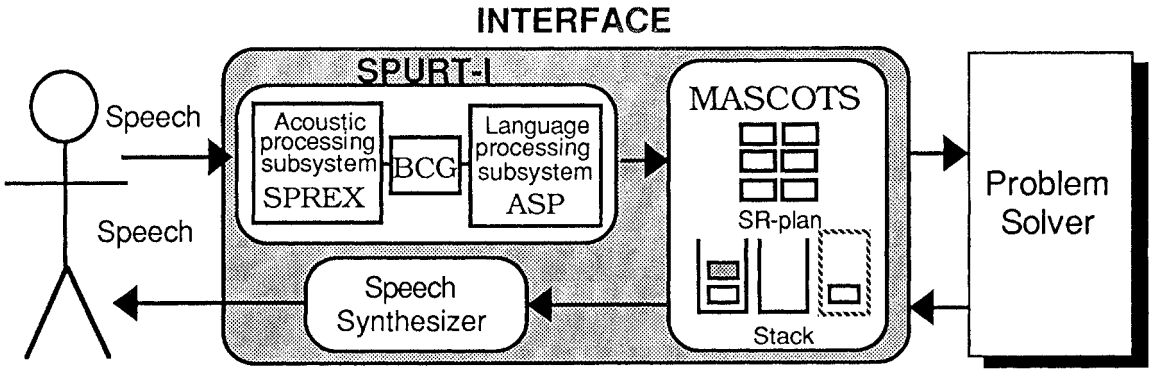

Fig.1 *Total system configuration.*

implicitly. Communication is thus performed by giving stimuli or responses to each other and some additional information such as confirmation, explanation, conditions to restrict the contents and so on. And intertwined interactions does make the dialog complex. In general, however, the last interaction completes first.

2.3 SR-plan

MASCOTS has a knowledge base for the plausible pattern of interaction between the user and the problem solving system. We call this knowledge SR-plan. An SR-plan is composed of some data and a sequence of computer procedures for interaction based on *Stimulus* and *Response*.

MASCOTS has nine user plans and eight system plans. The user SR-plans are for dealing with the interaction constructed by stimuli from the user and responses to them from the system and each user SR-plan has stimulus templates corresponding to the utterance forms of the user stimuli. Similarly, the system SR-plans are for the interaction by the system stimuli and the user responses to them and each system SR-plan has response templates for the user responses. Some examples of the system SR-plans are shown in Fig.2.

- SP-ASK-COMPONENT
  to ask objects or locations of an action
- SP-ASK-REASON
  to ask reason of an action
- SP-ASK-FACT
  to ask whether a fact is true or not
- SP-COMMAND
  to order the user to do some tasks

Fig.2 *Some of the system SR-plans.*

Fig.3 shows some of the templates defined in a system SR-plan SP-ASK-FACT as an example.

System: 中井先生は, 懇親会に出席されますか?
(Is Prof. Nakai going to attend the social evening?)

type-1 [affirmative, negative]
　はい, いいえ, 等 (Yes., No. etc.)
type-2 [QUESTION-FRAMES]
　出席されます, 等 (He is going to attend it. etc.)
type-3 [guessing of QUESTION-FRAMES]
　出席されると思います, 等 (I guess he is. etc.)
　　:
　　:

Fig.3 *Response templates in SP-ASK-FACT.*

These templates are used for identifying the correspondence between a stimulus and a response.

Each template is constructed by two kinds of generic keywords. One is registered in advance. For instance, a generic keyword *why* includes the keywords of "why", "how come", etc. The other is instantiated when the stimulus of the system is given, such as QUESTION-FRAMES which will be instantiated to include the words used in the system utterance. QUESTION-FRAMES in type-2 in Fig.3, for instance, will include the keywords of "Prof. Nakai", "attend" and "the social evening" after the system asked a question that "Is Prof. Nakai going to attend the social evening?". This generic keyword will also include the keywords defined under the same category in thesaurus, which are "be present at", "make an appearance", "conference", "assemblage" and so on. This means that the templates written in an SR-plan are instantiated when the latter kind of generic keywords are instantiated according to the last system utterance.

2.4 Stack

MASCOTS employs two stacks for manipulating SR-plans : One is for the system SR-plans and is referred to as *system stack* and the other is for the user SR-plans and is referred to as *user stack*.

A user SR-plan is pushed down into the user stack when the user provides a stimulus. Later, that plan is basically popped up and stored in the history data base when the system responds to it and the user confirms it explicitly or implicitly. Similarly, the system stack is used for the system SR-plans. When the user provides another stimulus before responding to the last stimulus given from the system, a new plan is pushed down into the user stack. Then the process of the previous plan is suspended.

3 PREDICTION OF THE NEXT USER UTTERANCE

3.1 Top-down inference

Unlike one way speech such as telling a story, the user utterances in the dialog can often be predicted since major parts of them are responses to the stimuli of the system. Fig.4 shows expected utterances of the users.

- After system stimulus
  1. Response to it
  2. New stimulus
- After system response to user stimulus
  1. Its confirmation
  2. Response to the suspended stimulus
  3. New stimulus

Fig.4 *Expectation of user utterances.*

MASCOTS predicts the user utterances according to the information above. Then, it sends templates corresponding to the expectation to ASP. ASP can perform top-down analysis of the input lattice utilizing this information.

3.2 Bottom-up inference

Top-down processing has at least two shortcomings. One is that there is no expectation which stimulus(user SR-plan) comes next. Templates of stimulus are in one of nine user SR-plans and we cannot predict which SR-plan the user will use unlike the case of response templates.(Templates of responses are in the active system SR-plan, that is the current plan, which have been activated by the last system stimulus.) The other is that there is no ordering among multiple templates in one SR-plan. An SR-plan has a lot of templates. However, we cannot say which template is the best even if we have known the appropriate SR-plan for the utterance of interest.

In order to overcome these difficulties, we introduce bottom-up analysis. When a word lattice is given to ASP, ASP scans all the candidate words to find out some keywords which are registered in advance or generated from the last system stimulus. Then MASCOTS selects all the templates composed of only the generic keywords found by ASP. These templates will be ordered according to the top-down inference and some other heuristics described below.

3.3 Ordering of templates

The selected templates are scored by four kinds of heuristics as follows.

1. by the top-down inference (Top-Down point)
2. by recognition score in acoustic processing for each keyword (Recognition point)
3. by the kind of keywords (Key-Word point)
4. by frequency of the keywords in the history of the dialog (Context point)

The template of the highest score is send to ASP. The success of the ASP analysis using this template indicates that the utterance is of type directly inferred from the template. When the analysis fails, the next probable template is sent and the process is repeated until it succeeds. If all the expectation fails, then complete bottom-up processing using the dependencies between the words is performed.

4 EXAMPLE OF THE SYSTEM BEHAVIOR

Let us take an example of a dialog at the registration desk shown in Fig.5.

S1: 懇親会へは出席されますか?
(Are you going to attend the social evening?)
U2: 中井先生は出席されますか?
(Is Prof. Nakai going to attend it?)
S3: はい, 出席されます.
(Yes, he is.)
(S:System, U:User)

Fig.5 *An example of dialog.*

Suppose that MASCOTS is requested by the problem solving system, that is the registration desk system in this case, to ask a question if the user will attend the social evening. This is what we call a stimulus asking a fact, so system SR-plan SP-ASK-FACT is activated and pushed down into the system stack. Then the generic keyword QUESTION-FRAMES is instantiated from the system utterance as mentioned in Section2.3. Simultaneously, the templates of the expected responses in SP-ASK-FACT are instantiated using this generic keyword. In this case, templates of affirmative, negative, assertion of a fact and guess of a fact are expected. After that, the content of utterance is sent to the speech synthesizer and S1 is uttered.

Next, U2 is uttered by the user. The input of ASP is actually a word lattice, so the lattice includes not only correct candidate words but also many incorrect candidate

words. ASP scans all the candidate words and picks up the generic keywords QUESTION-FRAMES, *do?*, *zizitu* and 2 other generic keywords from them in this case. MASCOTS tries to find the templates which consist of only these generic keywords from the response templates in SP-ASK-FACT(current-plan) and the stimulus templates in all user SR-plans. MASCOTS selects four templates in this case : RESPONSE7, UP-ASK-FACT3 and 2 others. RESPONSE7 is a response template in the current plan SP-ASK-FACT and consists of QUESTION-FRAMES. UP-ASK-FACT3 is a stimulus template in the user SR-plan UP-ASK-FACT and consists of *zizitu* and *do?*. Next, these selected templates are scored using four kinds of heuristics. Since U2 is uttered after the system stimulus, top-down inference gives higher T-D-point to the response templates than to the stimulus templates. Other kinds of points are given in similar manner. The scoring results of the above two templates are shown in Fig.6.

RESPONSE7	$\Longleftarrow$	T-D-point	10
		Res-point	1.7
		K-W-point	0
		Con-point	0
		total	11.7
UP-ASK-FACT3	$\Longleftarrow$	T-D-point	6
		Res-point	3.2
		K-W-point	3
		Con-point	0
		total	12.2

Fig.6 *Scoring results of templates.*

After all, the stimulus template UP-ASK-FACT3 gets the highest point and is sent to ASP. ASP analyzes U2 with this template successfully in this case. And U2 is recognized as a new stimulus and the user SR-plan UP-ASK-FACT is activated. This plan is pushed down into the user stack and the previous plan SP-ASK-FACT is suspended. Then the kind (result of recognition, that is *stimulus*) and the content of utterance which is extracted using information in the template are sent to the problem solving system. However, if ASP cannot analyze the input with the first template successfully, the second template is sent to ASP and this process is repeated until the analysis succeeds.

After S3 answers to U2, the current plan (corresponding to the interaction by U2 and S3) waits a confirmation from the user. If MASCOTS judges that the user confirms it explicitly or implicitly, then the current-plan is popped up from the user stack. Then the suspended plan SP-ASK-FACT becomes active again and waits a response from the user.

5 PERFORMANCE EVALUATION

We made an experiment to evaluate the performance of MASCOTS with a 400-word vocabulary. In the experiment, seven user utterances are picked up from the conversation between one male adult and the registration desk system. Each sentence consists of 1.9 syntactic units on the average. As for ASP input, the average number of candidates in a unit is 37. And the average order of correct candidates is 11th.

Table 1 shows the result of the evaluation. This shows that MASCOTS selected 5.3 templates on the average from all 79 templates and the ordering mechanism put the correct templates at 1.9th place on the average. The circle in "Result" means the perfect success of words identification in the sentence, the triangle means the success of meaning analysis including mismatch of the dependent words, say "to me" for "for me", and the saltire means the failure.

Table 1 *Performance evaluation.*
(1 male adult, a 400-word vocabulary)

	Number of selected templates	Order of correct template	Result
Response	4	2	○
Response	10	3	○
Stimulus	8	1	△
Confirmation	6	1	△
Stimulus	3	1	△
Response	2	2	○
Response	4	3	×
Average	5.3	1.9	6/7

6 CONCLUSIONS

We have discussed a dialog management system MASCOTS which helps a speech understanding system. The basic mechanism of MASCOTS is twin-stack architecture based on SR-plans. The whole system is implemented in Common Lisp and Flavors on Symbolics 3620.

References

[1] M.Hori, K.Tsujino, R.Mizoguchi and O.Kakusho: *A speech understanding system SPURT-I — Dynamic clustering method and and performance evaluation —*, Trans. of IEICE of Japan, J72-D-II, 8, 1291-1298(1989).

[2] R.Mizoguchi, T.Yamamoto and Y.Yamashita: *Dialog management system based on stack structure*, Research Report No.PASL 1-6-4(1989).

[3] M.Hori, R.Mizoguchi, M.Kawachi, K.Uehara, J.Toyoda and O.Kakusho: *Association-based parser for speech understanding system — Framework design based on Cognitive exploration —*, Trans. of IEICE of Japan, J71-D, 5, 774-781(1988).