Acoust. Sci. & Tech. **24**, 6 (2003)

PAPER

A multilingual-supporting dialog system across multiple domains

Yunbiao Xu[1], Masahiro Araki[2] and Yasuhisa Niimi[2]

[1]*College of Computer Science and Information Engineering, Hangzhou University of Commerce, Hangzhou, China*
[2]*Department of Electronics & Information Science, Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto, 606–8585 Japan*

(*Received 4 April 2002, Accepted for publication 22 January 2003*)

Abstract: This paper presents a multilingual-supporting dialog system that was implemented in Chinese and Japanese in tasks of sightseeing, accommodation-seeking and PC assembling guidance. Such a dialog system benefits from three main methods we proposed, including case frame conversion, template-based text generation and topic frame driven dialog control scheme. The former two methods are for improving the portability across languages, and the last one is for improving the portability across domains. The case frame conversion is used for translating a case frame described in a particular language into that described in a pivot language. The template-based text generation is used for generating text responses in a particular language from abstract responses described in the pivot language. The topic frame driven dialog control scheme makes it possible to manage mixed-initiative dialog based on a set of task-dependent topic frames. Both Chinese and Japanese experiments in several domains showed that the three methods proposed could be used to improve the portability of the dialog system across domains and languages.

Keywords: Multilingual, Dialog system, Multiple domains, Dialog controller, Case frame, Topic frame

PACS number: 43.72.Kb [DOI: 10.1250/ast.24.349]

1. INTRODUCTION

With their rapid increase in performance and decrease in cost, computers have fast become a ubiquitous part of our lives, and have been more and more widely used to access, share and exchange scientific, cultural, social and economic resources available not only in the local community but also in the global community. Human-machine dialog systems, as one of communication applications to help people realize above activities, in the past decades, have been being witnessed to be true in some limited domains. In 1989, the research group of MIT developed a prototype of conversational systems [1], and such technologies have been used in some common domains, such as Air Travel Information Service (ATIS) [2]. But not all users could use their own native languages for communication in such applications.

In this paper we present a method for designing dialog systems that support across multi-languages and multi-task domains.

The two techniques have been investigated to support the portability of language processing systems among languages. One is to develop the direct translation of semantic feature structures, for example, of a language into these of another language. This technique was used in the speech-to-speech translation systems such as ASURA [3] and Verbmobil [4]. The other is to use a pivot language. In this technique, the translation of a language into another was attained by first translating a language into an intermediate language (called a pivot language) and then translating the pivot language into another. This technique was used speech-to-speech translate systems such as JANUS [5] and a few multi-lingual spoken dialog systems such as MIT Voyager and Jupiter systems [6,7].

We used the pivot language technique to support the portability of a dialog system across languages for the following reasons.

A spoken dialog system generally consists of several components. However, these components can be grouped into two large parts, a speech interface part including speech recognition and synthesis, syntactic and semantic analysis, and response generation, and a dialog control part including discourse analysis and dialog control. Both parts are dependent on the language and the task.

The first aim of this paper is to make the dialog control part independent of the language. For this purpose we devised to make the discourse analysis of dialogs by using a paradigm of the case frame as a pivot language. This is because it has been used to represent the meaning of sentences, it contained fragments of information necessary for our discourse analysis in the compact forms, and the translation of case frames of a language into case frames of a pivot language is much easier than the translation in other formalisms such as parse trees.

The second aim of this paper is to make the dialog control part independent of the task. For this purpose, in recent years, multi-domain dialog systems such as [8] were reported. [8] used a design strategy separating task-dependent factors to form a Task Description Table (TDT) from the core dialog engine, but tasks dealt with in [8] were simple slot-filling tasks. In this paper, we also tried to separate the algorithm for the dialog control including the discourse analysis from several knowledge sources dependent on the task. The algorithm for the dialog control adopted in this paper is an extension of the frame-driven dialog control scheme that has been used in a text-based dialog system [9]. The method is based on the fact that topics in a goal-oriented dialog tend to move according to a task-dependent structure [9]. For a given task we construct a set of topic frames, each of which is composed of a few related topics that might appear in dialogs on the task. A topic frame also contains the method for how to interact with a user about the topics described therein. As a dialog proceeds, several topic frames are activated corresponding to topics included in user's utterances and forms a dialog history on the topic.

In recent years, VoiceXML-based spoken dialog platforms are widely used for voice portals. In principle, VoiceXML-based platforms can realize multilingual, multiple domain dialog systems. However, because dialog control information must be explicitly defined in VoiceXML files, it is difficult to separate task dependent knowledge and dialog control information. In addition, language dependent information such as texts for speech synthesis is directly described in VoiceXML files. Therefore, domain portability and language portability are relatively low in VoiceXML-based systems. In contrast, in our proposed method, dialog control is represented in task dependent frame structure and there is no need to describe dialog control scripts explicitly. Also, case frame converter separates the language dependent part with the language independent dialog controller. As a result, domain portability and language portability are relatively high in our proposed method.

Based on the proposed scheme, we built two dialog systems for Chinese and Japanese, both managing spoken dialogs in several task domains, and conducted dialog experiments. The results showed that the proposed dialog control scheme was promising for improving the portability of a dialog system across languages and task domains.

In the following, we will briefly illustrate our dialog system in section 2. Then the case frame conversion, the topic frame driven dialog control scheme and the template-based text generation will be presented orderly in section 3, section 4 and section 5 respectively. In section 6, we will show experiment evaluation resulted from our current dialog system in three different domains, followed by the conclusion and the future work in section 7.

2. SYSTEM OVERVIEW

Figure 1 shows the structure of our current dialog system, consisting of a speech interface that is obviously language dependent, and a dialog controller that has been extended to be language independent. Among two streams of the speech interface, the speech input stream converts user's utterances into word strings and then into semantic representations described by case frames, while in the speech output stream an inverted process is performed [10].

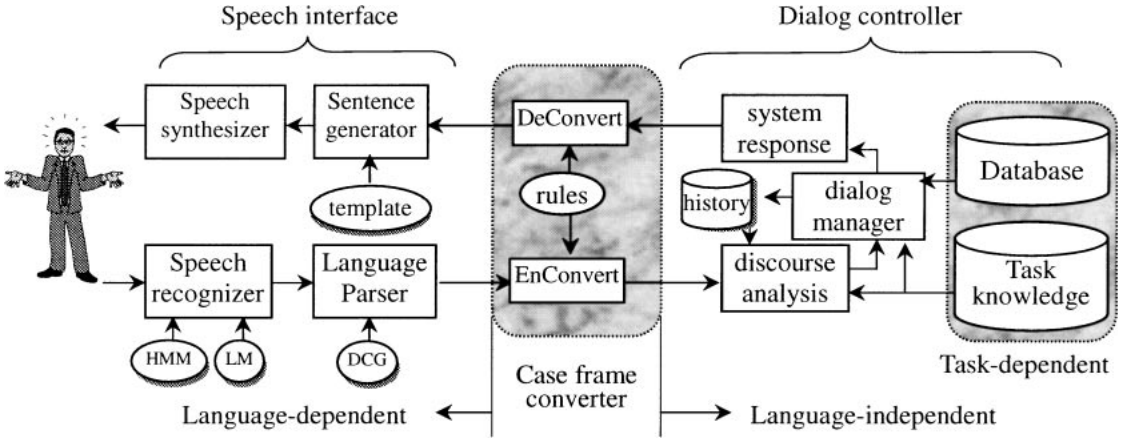

Fig. 1 The block diagram of our dialog system.

The dialog controller, which is a language-independent module, extracts a topic and a dialog act from a semantic representation of an utterance based on a set of task-dependent knowledge bases, performs some necessary actions, such as retrieving information from a task-dependent database and creates an abstract response that contains a template name and necessary arguments to be used to generate text sentence in the succeeding process.

In our system, a semantic representation described by a case frame is used to communicate between the speech interface and the dialog controller. To enable different speech interfaces to share a common dialog controller, we embedded a case frame converter to convert such a case frame described in source language words (hereafter called *a SLW case frame*) into one described in pivot language words (hereafter called *a PLW case frame*) or vice versa.

Now, the text generator could generate Chinese and Japanese sentences based on separate sets of the templates for each distinct language.

3. CASE FRAME CONVERSION

3.1. A Case Frame

Figure 2 shows a paradigm of syntax analysis of Chinese sentence "我需要景点介紹 (I need the sightseeing guide)" resulting from the actual dialog of our dialog system. The second line of Fig. 2 contains a semantic representation that is described by a list of four terms, (1) a word used to reflect whether the utterance is an affirmative answer [Yes/Ok], a negative answer [No], an ambiguous reply [uhm], or empty item [non], we call such word a *Lead-word*, (2) a main verb word, (3) a modality information specifying the tense and the aspect properties of an utterance, and (4) a set of case elements (hereafter called slots). Each slot indicates one of such relations between a main verb word and a noun phrase/word filled in its slot, like 動作主格 (agent), 対格 (object) just as seen in Fig. 2. Our system only conveys the list containing above four terms from the speech interface to the dialog controller. Apart from the *Lead-word*, we call the list containing only the rest three terms a case frame.

```
Sentence: 我需要景点介紹/I need the sightseeing guide.
ss([[non],[普通動詞(需要)],[陳述],[動作主格([人称詞:我]),対格([案内名:景点介紹])]])
+-- s([普通動詞(需要)],[陳述],[動作主格([人称詞:我]),対格([案内名:景点介紹])])
|    +-- 格要素(動作主格([人称詞:我]))
|    |    +-- 名詞句([人称詞:我])
|    |         +-- 代名詞([人称詞],[代名詞(我)])
|    |              +-- 我
|    +-- 述語([普通動詞(需要)],[陳述])
|    |    +-- 普通動詞(普通動詞(需要))
|    |         +-- 需要
|    +-- 格要素(対格([案内名:景点介紹]))
|         +-- 名詞句([案内名:景点介紹])
|              +-- 普通名詞([案内名],[普通名詞(景点介紹)])
|                   +-- 景点介紹
+-- 文末記号([endSent])
     +-- 。
```

Fig. 2 A paradigm of syntax analysis in our system.

From Fig. 2, we can find semantic markers such as "人称詞 (personal pronoun)" and "案内名 (functional name)" for the noun words of "我 (I)" and "景点介紹 (sight-seeing guide)" respectively. All noun phrases/words included in an utterance are assigned to some slots of the case frame based on semantic markers of the noun phrases/words.

Here, we show the model of a case frame by (V, M, C) where V denotes the main verb word, M denotes the modality information that contains one to several modality elements, i.e.,
$M = [M_0, M_1, \ldots, M_n]$.
C denotes one to several case elements, i.e.,
$C = case\_id_1(case_1), \ldots, case\_id_k(case_k)$.
It also can be wrote as $C = [C_0, C_1, \ldots, C_k]$.

This case frame model would be used to perform syntactic analysis by the language parser.

3.2. Syntactic Analysis

In our system, a simple thesaurus, a set of case frames and a set of syntactic rules for acceptable sentences, such three kind of knowledge source are used to parse an utterance. All words that our system can accept are clustered into several groups in POS firstly at shallow level, and then are semantically grouped and organized as a tree structure like a thesaurus at deep level. The concepts of the upper nodes closing to the root node in this structure play roles of semantic markers in interpretation of utterances. Just as seen in Fig. 2, a slot of the case frame is always filled by a single pair-value with the formalism like as [A:B], or by a compound pair-value with the formalism like as [A:B:[C:D]], where A and C are semantic markers in the thesaurus, B and D are detail noun words/phrases contained in user's utterance. More complex recursions can be used to represent relation between two adjacent hierarchical pair-values. It is easy to find such compound pair-values in the experiments showed in section 6, such as U203. In the following sections, we simplified the pair formalism from [A:B:[C:D]] into [A:[C:D]] if A is the same as B.

The syntactic analysis of utterances is performed based on the case grammar in which the meaning of a sentence is represented by a case frame associated with a main verb of that sentence. In our system, case frames and syntactic rules are integrated in a framework of the definite clause grammar (DCG) [11]. Comparing with the context free grammar (CFG), one of the differences is that DCG is able to use enhancement items, just as shown in Fig. 3.

In Fig. 3, the second line means that the value of the syntactic constituent "Lead" is a member of the list "yes, no, uhm, ok" or null (non). The 6th and the 7th lines mean that the syntactic constituent C_0 is just the same as the case constituent C_0.

```
ss(Lead, V, M,C)→
   Lead,{ member(Lead,[yes,no,uhm,ok,non]) },
   s(V,M,C),
    EndSymbol.
s( V, [ M0,…,Mn], [C0,…,Ck] )→
    case_element[ C0 ],
    { case_frame( V,[M0,…,Mn],[C0,…,Ck] ) },
    verb[ V,M0,…,Mn ],
    …
     case_element[ Ck ].
```

Fig. 3 A paradigm of DCG grammar.

Just as seen in Fig. 2, the language parser outputs a hierarchical data structure in which a case frame was attached.

3.3. Implementation of Case Frame Conversion

Our previous Japanese dialog system, SDSKIT-3 [12], conveyed directly whole the syntactic hierarchical structure of an utterance to its dialog controller to make the discourse analysis. To promote SDSKIT-3 to be a multi-lingual-supporting one, we first improved the DCG grammar rules to enable each case frame to contain complete information, and then improved the analysis strategies of the discourse to enable to determine the topic, and the dialog act of user's utterance depending only on the case frame and the *Lead-word* without any additional information.

A case frame converter was adopted in our proposed dialog system. The converter translates a SLW case frame into a PLW one through two kinds of processes orderly and recursively. The first one is called case constituent ordering/pruning process based on a set of rules, and the second one is called noun word replacing process based on a translation table. Figure 4 shows such a conversion example for Chinese case frame into a PLW one.

The same semantic meaning can be represented in different case frames in different languages, such as case constituent and case element's ordering and so on. The case constituent ordering/pruning process treats this problem. For example, as shown in Fig. 4, the case elements of the Chinese sentence "我想看園林 (I want to visit gardens)" are different from that one of the Japanese sentence "庭園が見たいです" although they are equivalent in the intention. So the converter has to have a set of language-dependent rules that perform conversion of main verb words, modality information, case constituents and resorting their ordering. These rules are manually predefined.

Fig. 4 An example of case frame conversion.

As mentioned in section 3.2, all words that our system can accept are organized in a thesaurus. So, for all noun words to fill in the slots of a case frame, we can create quickly noun word translation tables to map a word in a source language into an appropriate noun word in the pivot language. The noun word replacing process replaces noun words by looking for this translation table to change a SLW case frame into a PLW one before being conveyed to the dialog controller, or make an inverse conversion before the arguments are conveyed to the text generator.

4. TOPIC FRAME DRIVEN CONTROL SCHEME

4.1. Topic Frame and Dialog Act

Our spoken dialog system has a set of 'topic frames' as a knowledge source on topics. A topic frame forms mutually related topics into a frame that might appear in a task domain. For example in sightseeing dialogs, the name of a hotel, the room charge, the location and so forth form a hotel frame. Figures 5 and 6 show the data structure

```
typedef struct slot_message
{
    char *SlotName;          // slot name
    char *Value;             // value to fill in
    char *ValueType;         // semantic marker
    char *PreCommand;
    char *PostCommand;
    int     Priority;
    struct slot_message *prior, *next;
} SlotMessage;
```

Fig. 5 The data structure of a slot in a topic frame.

```
typedef struct Frame
{
    char *FrameName; // name of the topic frame
    char *status;        // suspend, closed, ongoing
    int FrameId;
    SlotMessage *SlotMessagePointer;
    struct Frame *prior, *next;
} FRAME;
```

Fig. 6 The data structure of a topic frame.

of a topic frame as a data structure in programming language C.

As seen in Fig. 6, a topic frame consists of a topic frame name, a status id, and one to several slots whose data structure is showed in Fig. 5. Each slot consists of a slot name, a value field (usually empty), a pre-command and a post-command that were triggered before and after the slot has been filled respectively. The priority item reflects the priority of the slot as to being filled.

The value field is filled by one of a word, a numerical value and a pointer to other topic frame. The ValueType field describes this information. Especially the value field is filled by a word, the semantic marker of the word is given to the ValueType field. The current system has four methods to fill an unfilled slot: (1) to ask a user, (2) to retrieve instances from a database, (3) to use the default value attached to the slot, and (4) to link one of other topic frames. The PreCommand field is described by one of these methods. When a slot has been filled, the dialog controller first takes one of three actions: (1) to move the control to the slot with the next highest priority, (2) to move the control to the topic frame just linked, and (3) to present the information required by a user. The PostCommand field is described by one of these actions.

For a task domain, a set of task-dependent topic frame is predefined. Since some slots are allowed to be filled with some other topic frames, the set of the topic frames forms implicitly a set of a tree structure, which we call a static topic tree simply. We assume that topics in a dialog move along this tree structure as a dialog proceeds. Thus a dialog forms a subtree of the static topic tree. We call this tree *a dynamic topic tree* which represents the history of a dialog on the topic.

We are aware that each utterance in a dialog has its own purpose, that is, an intention a user wants to convey to the dialog system. Hereafter we call this a dialog act. Table 1 lists a set of upper dialog act categories in our dialog system.

The spoken dialog system has a knowledge source on dialog acts, a state transition network in which a state corresponds to a dialog act. This network describes possible transitions of dialog acts through dialogs. Thus, the discourse history on dialog acts is represented by a state in this network.

4.2. The Discourse Analysis

Now we are ready to explain the discourse analysis through which the topic and the dialog act of an utterance are identified. Since we have reported the method for the discourse analysis in detail in [13], we will describe the outline of it here. The discourse analysis in our system consists of the bottom-up and the top-down analysis.

We use two tables in the bottom-up analysis on the topic. One is called a topic_slot table which describes relations among the name of topic frame, a slot name in the topic frame which should be filled with a word, the semantic marker of that word. For example, a topic frame of "temple," its slot of "builder," and the semantic marker "person." The other is called a verb_focus table which is formed from case frames of verbs. In task domains we treat, a slot filler of a verb with a specific case marker tends to be focused as a topic, that is a filler of some slot of a topic frame. The verb_focus table describes this relation. For example, a verb "build," a slot filler of "agent" a topic frame of "temple," and its slot of "builder." Theses two tables totally contain about 300 records. It is worth to notice that these tables can be constructed automatically from a set of the topic frames and a dictionary on case frames of verbs in which necessary case markers are labeled.

Bottom-up candidates for the topic are decided by looking for each slot filler of the case frame of an utterance up in these two tables.

When the bottom-up analysis has produced more than a candidate, the top-down analysis on the topic is invoked. The top-down analysis uses the dynamic topic tree which represents the discourse history on the topic, and the nodes of which are slots of topic frames on which have been mentioned in the dialog. A certain node in the dynamic topic tree is specified as the current node, which means a subdialog is currently held on the slot corresponding to the node. The top-down analysis searches for some of the bottom-up candidates in the dynamic topic tree equidistantly from the current node. The distance between two nodes is calculated by repeatedly applying the following rules; the distance is equal to 1 if the two nodes are sibling, and 2 if they are a parent and a child.

The node that has the shortest distance from the current node is decided as the topic of the utterance under consideration. When more than a condidate remain, the dialog controller makes a confirmation to the user.

The dialog system has a table for the bottom-up analysis on the dialog act. The table describes relations of a dialog act to the predicate and the modality information of an utterance. In this table predicates are grouped into four types: for example, in the sightseeing guide task. (1) verbs to express sightseeing actions of a user like 'visit' and 'stay,' (2) verbs to demand information like 'tell me,' (3) verbs to express user's actions to sightseeing plans like 'add . . . to my plan' and (4) predicates to express response like 'I see' and 'That is fine.' Modality information is grouped into question and non-question. The dialog acts listed in Table 1 are classified as shown in Table 2.

The case frame of an utterance and some clue words, if any, are used to resolve ambiguities in Table 2.

The top-down analysis on the dialog act is simple. Top-

Table 1 Upper dialog act categories.

GR:	Greeting
AI:	Ask-Infomation
GI:	Get-Infomation
CN:	For a user to confirm his/her knowledge or belief, or for the system to confirm its inference
RC:	Responce-to-Confirmation
AK:	Acknowledgement

Table 2 Classification of dialog acts.

Predicate types	Modality	
	Question	Non-question
(1)	AI, CN	GI
(2)	AI, CN	AI
(3)	AI, CN	GI
(4)		AI, GI, RC, AK

down candidates are all the dialog acts which can be reached from the current state in the state transition network on the dialog act. We observed these were the case where two dialog acts AI and CN could be separated. In such cases the dialog act AI has the priority.

4.3. Handling of Topic Frame

The topic frame driven control scheme is based on the fact that topics in a goal-oriented dialog tend to move according to a task-dependent tree structure. So, for a given task, we can construct a few related topics as a topic frame, which might appear in dialogs on the task domain under consideration. Figure 7 shows an actual slot entry with slot name "期間 (period)" of topic frame "観光 (sightseeing)" that was applied in the dialog controller of our dialog system in sightseeing domain.

The default value of the period slot is set to 1-day, from Fig. 7, to obtain this slot, the dialog controller will execute orderly the commands described in the PreCommand field, that is, "to ask a user" which means to generate a sentence to inquire a user a period of sightseeing, to synthesize it, and to wait for user's response. After the slot has been filled, the commands which was described in the Post-

```
期間(
  [ default(1,[ ]) ],
  [ to ask a user ],
  [ next_priority ],
  1, [ ], [ 日数, 期間, 予定, 程度 ]
  )
```

Fig. 7 An actual slot entry of the topic frame "観光."

Command field will be triggered; in this case, the dialog control will move to the slot with the next priority.

Generally speaking, the behavior of the dialog controller is quite simple when it is to speak to a user. It searches for unfilled slot in the dynamic topic tree in the depth-first way from the current node, and tries to fill that slot according to the method described in its PreCommand field. When the slot has been filled, the action described in its PostCommand field is conducted. This action may change the current node. For example, it is the case the action is "to move the control with the next highest priority." In such a case, the dialog controller restarts the depth-first search from the new current node.

When the dialog controller is interpreting user's utterances, that is, conducting the discourse analysis, it searches equi-distantly in the dynamic topic tree as mentioned in the previous section. This may causes to change the current node. In this case the depth-first search also restarts from the new current node.

5. TEMPLATE-BASED TEXT GENERATION

Contrasting with a machine translation system, a dialog system works in limited domains, and its response sentences are relatively predictable. One effective approach to the sentence generation for such a dialog system is to concatenate templates after filling slots by applying recursive rules along with appropriate constraints (person, gender, number, etc.) [1]. Based on this idea, we designed a set of templates for Chinese and Japanese text generators respectively. Each one consists of a sequence of word strings and/or slots to be filled by the arguments resulted from preceding process. Table 3 shows a set of selected templates for examples listed in section 6. In Table 3, the lowercases "x" and "y" are replaced by words which are decided from uppercases "X" and "Y" respectively according to the conversion rules of distinct languages, i.e., the lowercases "x" and "y" specify source language words, and the uppercases "X" and "Y" specify pivot language words.

To design such templates, we analyzed 170 response sentences of SDSKIT-3 for five different domains, and defined four category of template with total number of 48 templates, in which each category contains one to several sub-categories to reflect different syntactic situations, as shown in Table 4. Each template could generate a text sentence by selecting one randomly from one to four candidates after their slots have been filled by the input arguments to enable the output be more flexible in perception.

An argument could be one of following three terms, (1) a retrieve value from a database, (2) a semantic marker word according to the thesaurus described in section 3.2, (3) a frame topic name according to the topic frame tree,

Table 3 Several selected Chinese/Japanese templates for the experiments listed in section 6.2.

Template name	Input	Text sentence (for different languages)
promInit2PlsSelect	X	Chinese: 這是 x 系統。y 中、汝需要那一項服務？ Japanese: こちらは x システムです。y について、何なりとお尋ね下さい。
promDayItinerary	X	Chinese: 現在讓我門来確定第 x 天的行程計画。 Japanese: x 日目のコースを決めましょう。
promAlternative	X, Y	Chinese: x 和 y 中、汝期望那一個? Japanese: x と y と、どちらがよろしいですか？
inquireDays	non	Chinese: 汝打算化幾天時間? Japanese: 何日の御予定ですか？
inquireWhichIsSuit	X, Y	Chinese: 請問、汝喜歡 x 的 y? Japanese: x の y がよろしいですか？

Table 4 Four categories of response template.

categoly	*meanings*
Prompt	1. greetings to user. 2. prompt valid candidates to select. 3. prompt the selected information and the system dialog status.
Confirm	1. confirm current plan. 2. confirm current dialog intention. 3. confirm a previous selected action.
Inquire	ask user to fill an unfilled topic frame slot.
Answer	answer to a user the result retrieved from a database.

just as the response sentence S101 described in section 6 that is generated from the template "promInit2PlsSelect" listed in Table 3 with arguments "京都観光案内," the Y of that template will be filled with the slot names of topic frame "京都観光案内," and (4) a constant.

6. EXPERIMENTAL EVALUATION

It is not very clear how to design an evaluation metric for above methods of our dialog system. However, we thought that it would be possible to evaluate both of the speech recognition and correct responses resulting from the dialog system, and furthermore, the bookkeeping list of the dialogs represented in case frames, in themselves, could be used to demonstrate the feasibility of the proposed methods. So, in the following, we will report experiment condition and evaluation in section 6.1, and then the bookkeeping lists in section 6.2.

6.1. Experiment Condition and Evaluation

Since the dialog controller is an extension of SDSKIT-3, and has been extended into a language-independent one, we took the language adopted by such a dialog controller as pivot one. To test the dialog system in a general lab room by speaking to a microphone, we invited two groups of subjects who are students studying in the disciplinary of computer speech application but not familiar with any spoken dialog system before the experiments, one group consists of 15 Chinese native subjects (10 males and 5 females) to use Chinese speech interface, and another group consists of 15 Japanese native subjects to use Japanese speech interface respectively.

The experiments are performed in three task domains respectively including the sightseeing guidance, the accommodation-seeking guidance of Kyoto City, and PC-assembling guidance.

Before starting the experiments, one of the authors makes a short demo of this spoken dialog system to all of the subjects. Then they were asked to talk with the dialog system as natural as possible just as to talk with a familiar person.

For the spoken sentences that are uttered by the subjects, to judge whether the speech recognition and the response of the dialog controller are correct or not, we designed a trace interface to show the text sentences resulting from the speech recognizer and the response text generator respectively at real time.

Once the speech recognizer fails in recognizing and the dialog controller also fails in providing a suitable answer, one of the authors will ask the subject to speak again, or to make a slight change of the style of the spoken sentences if necessary. The sentences that were uttered by the Japanese subjects and the Chinese subjects are collected automatically to form Japanese and Chinese test-set sentences, and then, the OOVs (Out-Of-Vocabulary) and the perplexities of these two sets of test-set sentences are calculated. The OOVs and the perplexities are 8.2% and 28.8 for the Japanese test-set sentences, 9.5% and 28.1 for the Chinese test-set sentences respectively.

For Chinese test, based on (1) a set of monophone acoustical HMM models whose syllable accuracy is 69.2% evaluated by HTK V3.0, (2) 95 CFG rules with total of 68 POS categories and 320 words, among 30 dialogs with 698 utterances whose average length is 9.3 syllables, 558 utterances, i.e. about 80.0% of 698 utterances, are recognized correctly, and 471 utterances of the 558 correctly recognized utterances, i.e. about 84.4%, got fitting responses from the dialog controller.

For Japanese test, based on (1) a set of tied-mixture monophone acoustical models, (2) 52 CFG rules with total of 70 POS categories and 229 words, among 30 dialogs with 630 utterances, 537 utterances, i.e. about 85.2% of 630 utterances, are recognized correctly, and 463 utterances of the 537 correctly recognized utterances, i.e. about 86.2%, got fitting responses from the dialog controller.

6.2. Bookkeeping Lists of Several Actual Dialogs

In the following, several actual dialogs in three domains in Chinese and Japanese respectively will be showed in Figs. 8, 9 and 10 respectively. For the purpose of understanding, the user's utterances labeled with "U_{xxx}" and the generated sentences labeled with "S_{xxx}" will be annotated with corresponding English sentences. We listed the response codes that are resulted from the dialog controller and are language independent, and in which "RetX" and "RetY" are returned arguments that contain generated text sentences, and then followed by corresponding generated sentences for the output stream of Chinese/Japanese speech interfaces. For the input stream of speech interfaces, we listed user's utterances followed by corresponding *LeadWord* and its SLW case frames, and then followed by corresponding interpretation results of the utterances that consist of three terms, including dialog act type, dialog topic and the focus case element. To read easily, in the following, we use the symbol "→" to identify the beginning of new lines. All case frames and the interpretation results in the following are described in Prolog lists.

S101: promInit2PlsSelect(京都観光案内,RetX)
→ 這是京都市導遊系統。景点介紹、交通指南、住宿介紹中、汝需要那一項服務？ (This is a guide system in Kyoto city, may I help you in sightseeing guide, traffic guide, or accommodation seeking?)
U101: 我需要景点介紹。(I need the sightseeing guide.)
→ [non], [普通動詞(需要)], [陳述], [動作主格([人称詞:我]), 対格([案内名:景点介紹])]
→ [伝達], [観光], [案内:[観光:[]]]
S102: inquireDays(RetX)
→ 汝打算化幾天時間？(How many days do you plan to spend?)
U102: 我打算化両天時間。(I will spend two days.)
→ [non], [普通動詞(化)], [希望], [動作主格([人称詞:我]), 期間格([期間:両天時間])]
→ [伝達], [期間], [観光:[期間:[2]]]
S103: promDayItinerary(1,RetX), promAlternative(社寺,庭園,RetY)
→ 現在讓我門来確定第一天的行程計画。寺廟和園林中、汝期望那一個？(Ok, let's plan the first day's itinerary. What are you interested in temples or gardens?)
U103: 我想看庭園。(I prefer to visit gardens.)
→ [non], [普通動詞(看)], [希望], [動作主格([人称詞:我]), 対格([観光興味:庭園])]
→ [伝達], [観光興味], [行程:[観光興味:[庭園]]]

Fig. 8 Chinese experiment in sightseeing domain.

S201: promInit2PlsSelect(京都観光案内,RetX)
→こちらは、京都観光案内システムです。観光、交通、宿泊について、何なりとお尋ね下さい. (This is a guide system in Kyoto city, may I help you in sightseeing guide, traffic guide, or accommodation seeking?)
U201: 宿泊案内をお願いします. (I need the accommodation guide.)
→ [non], [普通動詞([願う])], [丁寧], [対格([宿泊案内])]
→ [伝達], [宿泊], [案内:[宿泊:[]]]
S202: promAlternative(ホテル,旅館,RetX)
→ ホテルと旅館と、どちらがよろしいですか？(Which do you like to stay at a hotel or a hostelry?)
U202: 旅館をお願いします. (I'd like to stay at a hostelry.)
→ [non], [普通動詞([願う])], [丁寧], [対格([旅館])]
→ [伝達], [宿泊希望], [宿泊:[宿泊希望:[旅館]]]
S203: inquireWhichIsSuit(どの辺り, 旅館,RetX)
→ どの辺りの旅館がよろしいですか？(Which area do you like to stay in?)
U203: 洛北方面をお願いします。(I hope to find it in Rakuhoku-area.)
→ [non], [普通動詞([願う])], [丁寧], [対格([方面:[地域名:洛北]])]
→ [伝達], [希望場所], [旅館プラン:[希望場所:[洛北]]]

Fig. 9 Japanese experiment in accommodation-seeking domain.

S301: promInit2Why(パソコン構築案内, RetX)
→這是電脳組装指南系統。請告知組装電脳的用途。(This is a guide system in PC-assembling. Please tell me the purpose to use a PC.)
U301: 我想做計算機通訊。(I want to do it for the Internet.)
→ [non], [普通動詞(做)], [希望], [動作主格([人称詞:我]), 対格([利用目的:計算機通訊])]
→ [伝達], [利用目的], [構築:[利用目的:[パソコン通信]]]
S302: inquireMorePurpose(RetX)
→還有其他用途吗？(Do you use it for other purpose?)
U302: 我還想玩遊戯。(I want to use it to play game too.)
→ [non], [普通動詞(玩)], [希望], [動作主格([人称詞:我]), 対格([利用目的:遊戯])]
→ [伝達], [利用目的], [構築:[利用目的:[ゲーム]]]
S303: inquireMorePurpose(RetX)
→還有其他用途吗？(Do you use it for other purpose?)
U303: 没有了。(No, No more.)
→ [没有了], [non], [non], [non]
→ [否定応答], [non], [non]
S304: promPurposeParts([パソコン通信,ゲーム],[ハード,メモリ,ハードデスク,ディスプレイ],RetX), PromDecideParts(ハード,RetY), confirmDecidePart(ハードB,RetZ)
→計算機通訊、遊戯所需的配件包括 CPU、内存、硬盤、顕示器。現在讓我門来選購 CPU。請問、汝対 CPU-B 感覚如何？(For the purpose of the Internet and the game, The CPU, memory, harddisk and monitor are necessary. Now, lets decide a CPU, How do you think about CPU-B?)
U304: 価格是多少。(How much money is it?)
→ [non], [普通動詞(是)], [疑問], [主格([price:価格]), 断定格([価格疑問:多少])]
→ [要求], [値段], [パーツ:[値段:[query]]]
S305: answerPrice(一万五千,RetX)
→価格為一万五千日元。(It is fifteen thousand JPY.)

Fig. 10 Chinese experiment in PC-assembling domain.

7. CONCLUSION AND FUTURE WORK

This paper described a multilingual-supporting dialog system implemented in Chinese and Japanese for several domains. Such a dialog system adopted three kernel methods including case frame conversion and template-based text generation to promote the portability across languages, topic frame driven dialog control scheme to promote the portability across domains. We are aware that through separating language-dependent and task-dependent knowledge sources from the system to construct external knowledge bases, the cost to port a dialog system across languages and domains will be confined only in replacing such knowledge bases without changing the dialog controller module. The Chinese and Japanese experiments in several domains demonstrated that; (1) The case frame conversion method is one of rapid and effective methods to promote a dialog system into a multilingual one. (2) The proposed dialog control scheme is able to treat different domains. It is a task-independent dialog control scheme. (3) The proposed template-based text generation method is able to generate very natural sentences, and it is easy to utilize. Therefore, these methods are feasible and effective to improve the portability across domains and languages.

Although the experiments are successful, however, our system is by no means complete. The main reason why a correctly recognized utterance failed to get a fitting response from the dialog controller is either out of the set of case frame definition or out of search range of topic frames. So, in the future, we will extend the vocabulary size and the set of case frame definition, improve the design of the dialog controller.

ACKNOWLEDGEMENTS

We would like to thank Nishimoto Takuya, Oku Tomoki for their many warmly assistant, valuable discussion/suggestion in various phases of this research.

REFERENCES

[1] V. Zue, "Conversational interfaces: Advances and challenges," *Proc. 5th Eur. Conf. Speech Communication and Technology*, pp. KN-9–KN-18, 22–25 September, Rhodes, Greece (1997).

[2] P. Price, "Evaluation of spoken language systems: The ATIS domain," *Proc. DARPA Speech and Natural Language Workshop*, pp. 91–95 (1990).

[3] T. Morimoto, T. Takezawa, F. Yato, T. Tashiro, S. Sagayama, M. Nagata and A. Kurematsu, "ATR's speech translation system: ASURA," *Proc. 3rd Eur. Conf. Speech Communication and Technology*, pp. 1291–1294. September 21–23, Berlin, Germany (1993).

[4] `http://verbmobil.dfki.de/overview-us.html`

[5] M. Woszczyna, N. Coccaro, A. Eisele, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. P. Rose, T. Sloboda, M. Tomita, J. Tsutsumi, N. Aoki-Waibel, A. Waibel and W. Ward, "Recent advances in JANUS: A speech translation system," *Proc. 3rd Eur. Conf. Speech Communication and Technology*, pp. 1295–1298. September 21–23, Berlin, Germany (1993).

[6] J. Glass, G. Flammia, D. Goodine and M. Phillips, "Multilingual spoken-language understanding in the MIT Voyager system," *Speech Commun.*, **17**, 1–18 (1995).

[7] J. Glass, D. Goodine, M. Phillips, S. Sakai, S. Seneff and V. Zue, "A bilingual voyager system," *Proc. 3rd Eur. Conf. Speech Communication and Technology*, pp. 2063–2066, 21–23 September, Berlin, Germany (1993).

[8] Y.-C. Lin, T.-H. Chiang, H.-M. Wang, C.-M. Peng and C.-H. Chang, "The design of a multi-domain Mandarin Chinese spoken dialog system," *Proc. 5th Int. Conf. Spoken Language Proceing* (*ICSLP98*), Vol. 2, pp. 41–44. November 30 – December 4, Sydney, Australia (1998).

[9] J. G. Bobrow, R. Kaplan, M. Kay, D. Morman, H. Thompson and T. Winograd, "GUS: A frame-driven dialog system," *Artif. Intell.*, **8**, 155–173 (1977).

[10] Y. Niimi, T. Oku, T. Nishimoto and M. Araki, "A rule based approach to extraction of topics and dialog acts in a spoken dialog system," *Proc. EuroSpeech 2001*, pp. 2185–2188, Sept. 3–7, Aalborg, Denmark (2001).

[11] G. Gazdar and C. Mellish, *Natural Language Processing in Prolog — An Introduction to Computational Linguistics* (Addison-Wesley, Wokingham, England, 1989).

[12] Y. Niimi, N. Takinaga and T. Nishimoto, "Dialog management in a spoken dialog system, SDSKIT-3," *Proc. SPECOM '98*, pp. 91–96, Russia (1998).

[13] Y. Niimi, N. Takinaga and T. Nishimoto, "Extraction of the dialog act and the topic from utterances in a spoken dialog system," *Proc. ICSLP '98*, pp. 2079–2082, Australia (1998).

Yunbiao Xu graduated from Zhejiang Institute of Science and Technology, and the Graduate School of Jilin University of Technology, Changchun, China, in 1986 and 1988 respectively. In 2002, he received the Ph. D degree in information and production science from Kyoto Institute of Technology, Japan. He is now an associate professor at the Department of Information and Computer Science, Hangzhou University of Commerce, Hangzhou, China. His current interests are speech recognition, speech synthesis and spoken dialog system.

Masahiro Araki received B. E., M. E. and Ph. D. degrees in information science from Kyoto University, Kyoto, Japan, in 1988 and 1990, 1998 respectively. He is now an associate professor at Department of Electronics and Information Science, Kyoto Institute of Technology. His current interests are spoken dialogue processing and artificial intelligence. He is a member of IPSJ, IEICE, JSAI and ANLP.

Yasuhisa Niimi received the B.E., the M.E. and the Ph.D. degrees from Kyoto University, Japan in 1962, 1964 and 1969, respectively. From 1964 to 1969 he was working at Kyoto University. Since 1970 he has been working at the Department of Electronics and Information Science of Kyoto Institute of Technology, where he is now a honorary professor. His current interests include speech information processing, natural language and artificial intelligence. He published the book "Speech Recognition" (in Japanese) in 1979. He is a member of the IPSJ, IEICE and JSAI.