The Design of a Multi-Domain Mandarin Chinese Spoken Dialogue System

Yi-Chung Lin, Tung-Hui Chiang, Heui-Ming Wang, Chung-Ming Peng, and Chao-Huang Chang

Advanced Technology Center
Computer & Communication Research Laboratories
Industrial Technology Research Institute
Chutung, Taiwan 310, R.O.C.
{lyc,thchiang,hmwang,cmpeng,changch}@atc.ccl.itri.org.tw

ABSTRACT

In some dialogue systems, the design of dialogue strategy is bound tightly to the domain by straightforwardly hard-coding the response actions into the system. Such a paradigm is quite easy to build up a prototype system, but makes it difficult to port the system across different domains. This paper presents a domain-transparent design of dialogue management to increase system portability. The basic idea of this framework is to extract the domain-dependent factors to form an external domain knowledge database, leaving the dialogue management component independent of the tasks. Based on the proposed framework, porting to another domain needs only to replace the domain knowledge database without changing the dialogue management module. This paper also proposed a task description table interface enabling system developers to design the dialogue strategy flexibly. With this approach, the effort of porting a spoken dialogue system across different domains can be relieved.

1. INTRODUCTION

In the age of rapid progress in computer hardware technology, a friendly spoken interface is anticipated to be the next generation man/machine interface. In the past decade, researches have been shifted from the theories and basic technologies of automatic speech recognition to spoken dialogue systems. Many spoken dialogue systems have been built to deal with various applications, such as air traffic information service [1], banking service [2], etc. Although those dialogue systems have been demonstrated successfully in some limited domains, spoken dialogue systems are not yet widespread in our everyday life. The main obstacle is the high cost of porting a system from one domain to another.

Generally, a typical spoken dialogue system may consist of several components of speech recognition, language understanding, dialogue management, language generation and speech synthesis. Among those components, the dialogue management component tends to be the most domain-dependent one. Therefore, the most straightforward design is to hard code the possible interactions between human and machine into routines [3]. This approach can easily construct a prototypical system but make the system hard to port to a different domain. Another approach is to manage the interactive actions based on a finite state network [4,5]. In the network, each state represents a particular status about the internal and external resources available for the system (such as, dialogue history, current user input, the database query results, etc.). The actions to be carried out and the next state to be entered are associated with a state transition arc. In this approach, the task-dependent part of the dialogue is pushed to the state network to make the dialogue manager portable across different tasks. However, since the number of dialogue states is usually very large, it is laborious to associate every dialogue state with corresponding actions and transition arcs. Besides, managing dialogue with state network tends to be system initiative because the dialogue manager conducts the conversation flow by travelling one of the pre-defined paths in the state network and the users are discouraged (or even forbidden) to deviate from the system's plan.

To pursue a mixed initiative dialogue management manner, Goddeau et al. [6] proposed to handle the dialogue status by filling an electronic form (or E-form). The E-form consists of task-relevant slots whose contents are the constraints acquired from user's utterances. The response to user depends on the current state of the E-form. This approach avoids enumerating all possible dialogue states to form a network and allows the user not to follow the system's prompt. However, the E-form approach is limited to a single goal approach, whose goal is trying to fill the empty slots in the E-form. It is difficult to deal with a complex task of multiple goals.

In this paper, a domain-transparent framework is proposed to develop a portable dialogue manager. In this framework, the domain-dependent factors used in decision making are extracted to form an external domain knowledge database. The domain knowledge database consists of a set of dialogue states. Each dialogue state is associated with an action and the condition to apply that action. By this way, the dialogue management process is simplified to be looking up the most appropriate action in the domain knowledge database. Once the domain changes, only the domain knowledge database needs to be replaced, leaving the dialogue manager itself unchanged.

To enable system developers to easily maintain the domain knowledge database and clearly describe the dialogue strategy, a table interface, called *task description table* (TDT), is proposed. With this interface, a system developer can easily specify a dialogue state by giving the conditions of the domain-dependent factors. Each dialogue state in TDT is associated with an action to indicate the response. Using this approach, porting a dialogue manager to another domain needs only to specify a new TDT. Thus, the effort of porting a dialogue system across domains can be relieved.

This paper is organized as follows. First, we briefly overview the architecture of our spoken dialogue system in Section 2. Then, the details of the proposed framework are present in

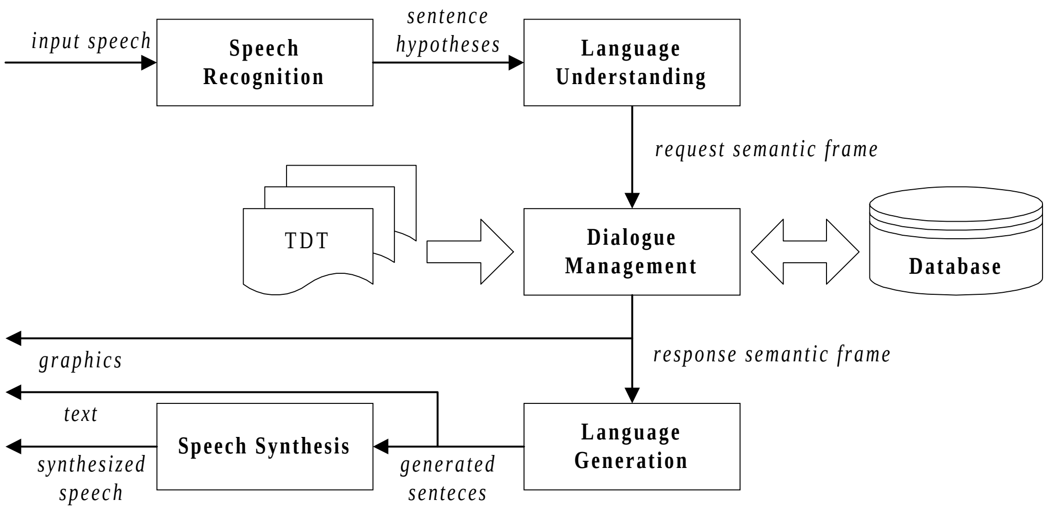

Figure 1: Block diagram of a multi-domain Mandarin Chinese spoken dialogue system.

Section 3. Section 4 demonstrates example TDTs designed for different tasks. Finally, the conclusions are made in Section 5.

2. SYSTEM OVERVIEW

The block diagram of our system is depicted in Figure 1. As the user speaks to the system about their requests, the speech recognition component recognizes the user's utterances and provides N-best sentence hypotheses to the language understanding component. The language understanding component then analyzes the meanings of the sentences, representing the results using a particular data structure called semantic frame[7]. Based on the semantic frame, the dialogue manager determines the most appropriate response by taking the conversation contexts into consideration. The responses, including actions of asking more constraints, confirming user's requests, providing suggestions, etc., are then presented to the user in graphics, text, and voice by using the language generation component and the text-to-speech synthesizer. The details of dialogue management component will be discussed in Section 3, while the other components are briefly described as follows.

2.1. Speech Recognition

The speech recognizer is based on our HMM-based speaker-independent large-vocabulary Mandarin continuous speech recognition system. In this system, a class-based bigram language model is used to decode the input speech into word sequences. Thus, the interface between the speech recognition and the language understanding component is N-best paradigm.

2.2. Language Understanding

The language understanding component comprises of four analysis phases: word segmentation, part-of-speech tagging, robust parsing and semantic interpretatio. At present, we have implemented a robust parser by adopting chart parsing algorithm together with probabilistic context-free grammars. The parser is completely syntax checking, leaving semantic checking performed in the semantic interpreter. The semantic interpreter determines the meanings of the input in term of semantic representation form called semantic frame [7], in which the thematic roles of constituents and the meanings of content words are specified.

2.3. Language Generation

The task of language generation is to generate text from the response semantic frame. In general, there might be more than one possible realization for a given response semantic frame. To allow flexible language generation, a set of message templates, expressing possible ways of realization, and their likelihood values are attached to each dialogue state. At present, the likelihood value, which will be further estimated from the collected corpora, of the realization alternatives are assigned equally to a constant value. That is, the message templates are selected randomly.

2.4. Speech Synthesis

To respond user in voice, the Mandarin text-to-speech system developed by CCL/ITRI is integrated in our system. In this TTS system, a markup language is defined to account for additional information that can make the synthesized speech correct and sound more natural. For example, digit, date and currency are pronounced differently although their surface forms are the same. Understanding the meanings of words, e.g., digit or currency, makes the TTS component able to pronounce them correctly.

3. DOMAIN-TRANSPARENT DIALOGUE MANAGEMENT

In the conversational process, the dialogue manager cooperates with the user to satisfy the user's requests. In some dialogue system, the design of the dialogue manager is bound to the domain by straightforwardly hard-coding the associated response actions into the system. Such a paradigm is quite easy

		Loc.	Date	Topic	
P	S1	–	–	–	Welcome()
G1	S2	–	×	×	Ask(Loc.)
G1	S3	×	–	×	Ask(Date)
G1	S4	×	×	–	Ask(Topic)
	S5	+	+	+	Query()
P	S6	×	> 7	×	Warn(Too_long)

Table 1: TDT of a weather forecast information system

to build a prototype system, but makes it difficult to port the system across different domains.

Motivated by the portability concern, a domain transparent framework is proposed. In this framework, domain-dependent factors used in decision-making are extracted out of the dialogue manager to form an external domain knowledge database. The domain knowledge database is composed of a set of dialogue states, each of them being associated with an action and the conditions to apply that action. The conditions of a dialogue state is specified in terms of the domain-dependent factors. For example, in the weather forecast information service system, the domain-dependent factors would be "date", "location", and "topic" (e.g., temperature, relative humidity, etc.). The state which lacks location information is specified with a "VOID" condition in the "location" factor. Therefore, the response associated to this state would be asking user to clarify the location information. Based on this approach, the dialogue management strategy can be simplified as looking up the most appropriate response by matching the current dialogue status with the dialogue states in the domain knowledge database. Moreover, porting the dialogue manager to a different domain only requires to replace an domain knowledge database without the effort of re-design.

3.1. Task Description Table

To facilitate the porting process, a table interface, called task description table (TDT) is proposed so that system developers can easily maintain the domain knowledge database and clearly describe the dialogue strategies. In a TDT, the system developer can easily specify the dialogue strategies by filling conditions and actions of each dialogue state. Each row of the table represents a dialogue state, while each column corresponds to a domain-dependent factor, called *parameter*. The content of a TDT cell describes the condition of the parameter of the corresponding dialogue state. Table 1 is an example of TDT designed for the weather forecast information service system. Three parameters, *location*, *date* and *topic*, are used and six dialogue states are specified with different conditions. The possible conditions include valid parameter (denoted by "+"), void parameter (denoted by "–"), parameter greater than, less than or equal to a value (denoted by ">", "<" and "=", respectively). If no condition is specified, the system encounters a "don't care" (denoted by "×") situation. The actions of states are listed on the right hand side of the table, including one welcome action, three actions for asking more information, one query and an warning action. On the left side, the marker "P" denotes S1 and S2 have higher priority and the marker "G1" denotes S2, S3 and S4 are grouped together. The details of priority and grouping are discussed in the following section.

3.2. Dialogue State Matching

Given the TDT, the main task of the dialogue manager becomes to look up the most appropriate action by matching current dialogue status with the dialogue states. In some cases, a certain dialogue status perhaps does not match any dialogue state in a TDT because of incomplete TDT specification. In such cases, the designer can still easily modify the TDT iteratively to refine the dialogue strategy.

With the flexibility of allowing "don't care" conditions, a certain dialogue status could match more than one dialogue state. The TDT also provides the alternatives of setting priority and grouping the dialogue states. Once multiple dialogue states are matched, the action of the state with the highest priority will be executed. If the matched states belong to a group, the actions of these states will be executed simultaneously. Taking Table 1 as an example, the initial dialogue status, where no parameters have been specified, would match four states (S1, S2, S3, S4) in the TDT. Since S1 has higher priority (marked by "P"), the dialogue manager will take the "welcome" action. In case the topic is given, the states S2 and S3 will be matched. Since S2 and S3 fall in the same group (marked by "G1"), the actions of asking for location, Ask(Loc.), and asking for date, Ask(Date), will be executed simultaneously.

4. EXPLORING ANOTHER DOMAIN WITH TDT

Currently, the dialogue manager of our spoken dialogue system is working on two different domains with two TDTs. The first application is weather forecast information system which provides the service of inquiring some weather forecast information and satellite images. The second application is railway ticket ordering system which provides train ticket ordering service in Taiwan. The TDT for weather forecast information system is given in Table 1, and the TDT for the railway ticket ordering system is listed in Table 2.

The railway ticket ordering system is more complicate than the weather forecast information system. As shown in Table 2, the parameters include departure station (D_ST), departure time (D_TIME), arrival station (A_ST), arrival time (A_TIME), departure date (DATE) and the number of tickets to be ordered (NUM). According to the first dialogue state, S1, in Table 2, the system will initially send greeting message to welcome the user. Then, the grouped states, from S2 to S6, correspond to the process of getting the necessary information for ordering tickets. Once the necessary information is provided, the system will order tickets by executing one of the actions associated to S7 and S8. Additionally, the last two dialogue states are specified to deal with improper ordering. If the user try to order more than 4 tickets or the departure date is a week later, the system will immediately notify the user the limitations on the ticket number or the departure date with warning messages.

		D_ST	D_TIME	A_ST	A_TIME	DATE	NUM	
P	S1	–	–	–	–	–	–	Welcome()
G1	S2	–	×	×	×	×	×	Ask(D_ST)
G1	S3	×	–	×	–	×	×	Ask(D_TIME)
G1	S4	×	×	–	×	×	×	Ask(A_ST)
G1	S5	×	×	×	×	–	×	Ask(DATE)
G1	S6	×	×	×	×	×	–	Ask(NUM)
	S7	+	+	+	×	+	+	Order()
	S8	+	×	+	+	+	+	Order()
P	S9	×	×	×	×	×	> 4	Warn(Too_many)
P	S10	×	×	×	×	> 7	×	Warn(Too_long)

Table 2: TDT of a railway ticket ordering system

The TDT in Table 2 also presents a mixed-initiative dialogue strategy. The system tries to determine which train the user wants to take by asking for the departure time, as indicated by state S3. However, the user is allowed to take the initiative to talk to the system when he/she wants to arrive. Then, the system will not ask for the departure time any more because which train the user wants to take can also be determined by the arrival time and arrival station.

5. SUMMARY AND FUTURE WORK

To design a portable dialogue system, a domain-transparent framework of the dialogue management is proposed. In this framework, the domain-dependent factors are extracted to build an external domain knowledge database. A task description table (TDT) interface is also proposed to let the system developer easily design and maintain the domain knowledge database. As a result, porting a dialogue manager to another domain is simplified to the work of filling a new TDT. With the proposed approach, the dialogue manager of our spoken dialogue system can support two applications based on two different TDTs.

Although the proposed TDT can meet the requirements of our current applications, there is plenty of room for improvement. In the future, we will extend the condition operators to user defined functions, allowing more flexibility in dialogue strategy design.

6. ACKNOWLEDGEMENTS

This paper is a partial result of Project 3P11200 conducted by ITRI under sponsorship of the Ministry of Economic Affairs, R.O.C. Especially thanks to the team members, C.-M. Pengwu, S.-C. Chang, S.-C. Chien and C.-C. Kuo, who give the technical support of speech recognition and speech synthesis.

7. REFERENCES

1. Goddeau, D., Brill, E., Glass, J., Pao, C., Phillips, M., Polifroni, J., Seneff, S., and Zue, V., “Galaxy: A Human-Language Interface to On-Line Travel Information,” *Proceedings of ICSLP’94*, Yokohama, Japan, pp. 707-710, 1994.

2. Larsen, L. B., “A Strategy for Mixed-Initiative Dialogue Control,” *Proceedings of EUROSEEECH’97*, pp. 1331–1334, Patras, Greece, 1997.

3. Glass, J., Flammia, G., Goodine, D., Phillips, M., Polifroni, J., Sakai, S., Seneff, S., and Zue, V., “Multilingual Spoken-language Understanding in the MIT Voyager System,” *Speech Communication*, Vol. 17, No. 1-2, pp. 1-18, 1995.

4. Kita, K., Fukui, Y., Nagata, M., and Morimoto, T., “Automatic Acquisition of Probabilistic Dialogue Models,” *Proceedings of ICSLP’96*, pp. 196-199, Philadelphia, USA, 1996.

5. Levin, E., Pieraccini, R., and Eckert, W., “Using Markov Decision Process for Learning Dialogue Strategies,” *Proceedings of ICASSP’98*, pp. 201-203, Seattle, USA, 1998.

6. Goddeau, D., Meng, H., Polifroni, J., Seneff, S., and Busayapongchai, S., “A Form-Based Dialogue Manager for Spoken Language Applications,” *Proceedings of ICSLP’96*, pp. 701-704, Philadelphia, USA, 1996.

7. Chiang, T.-H. and Su, K-Y., “Statistical Models for Deep-Structure Disambiguation,” *Proceedings of Fourth workshop on Very Large Corpora*, pp. 113-124. 1996.