ISCA Archive
http://www.isca-speech.org/archive

International Symposium on Chinese Spoken Language Processing (ISCSLP 2002)
Taipei, Taiwan
August 23-24, 2002

THE DESIGN OF A MULTI-DOMAIN CHINESE DIALOGUE SYSTEM

Wei-Tek HSU, Huei-Ming WANG and Yi-Chun LIN

Advanced Technology Center, Computer and Communications Laboratories,

Industrial Technology Research Institute, Chutung, Hsinchu

whsu@itri.org.tw hmw@itri.org.tw lyc@itri.org.tw

ABSTRACT

For a multi-domain spoken dialogue system, there are two major difficulties in building its dialogue management. One is to interpret users' interested domain correctly and switch between domains smoothly. The other is the high cost of merging the dialogue management of different application domains. To solve these problems, an implicit domain identification and switching mechanism for dialogue management is proposed to make the process of domain identifying and context switching transparent to users. To merge the dialogue management of different application domains efficiently, a merging process applying the goal-oriented table-driven dialogue management model is also proposed in this paper. By integrating two existing single-domain spoken dialogue systems, a multi-domain spoken dialogue system is set up to prove the usability of proposed solutions. Tested on two corpora collected by both single-domain systems, the multi-domain system contributes 5.47% and 2.28% errors comparing with the responses of two single-domain systems respective.

1. INTRODUCTION

With power of natural language, interactive spoken dialogue systems are expected to become the main stream of next-generation man/machine interface. To meet this expectation, dialogue systems must be able to provide services of various applications. However, nowadays, most researches still focus on developing dialogue system on a specific domain and little effort is made to study the issue of multi-domain dialogue system. There are two major difficulties for multi-domain dialogue management. The first one is to interpret users' interested domain correctly. This problem arises from the ambiguous utterances across different domains. The ambiguity comes from that one keyword may carry different concepts in different domains [1]. For example, the Chinese word *"Tai2 Nan2"* could represent either a "company" in a stock quote system or a "city" in the weather forecast system[1]. The second one is the high cost of merging the dialogue management of different application domains [1][2]. Since the dialogue management is the most domain-specific part in a dialogue system, it is the most difficult part to integrate while merging different single-domain systems into one multi-domain system.

To solve the domain identification problem in a multi-domain dialogue system, the simplest approach is to specify the domain explicitly by user [3]. In this approach, user interacts with the dialogue system starting from uttering domain id in the first turn then uttering his/her request in the following turns. Obviously, it does prevent the problem of ambiguity caused from polysemous words; however, the dialogue strategy is not as natural as our daily conversations, moreover, users have to know each domain id in advance. Lin [2] proposed a distributed architecture for cooperative spoken dialogue agents to facilitate building a multi-domain spoken dialogue system. In this architecture, each dialogue agent handles the dialogue of its domain, and cooperates with each other by one user interface agent. Thus, different spoken dialogue agents handling different domains can be developed independently. The domain identification and switching is achieved by first computing the most similarity scores of recognized utterance and the grammar/language model of domains. Then compare the score of current domain and one of the other domain that has the highest score. If the comparison result exceeds a predefined threshold, the domain will switch to the domain with the highest score. It provides implicit domain identification and switching approach. However, the domain ambiguous problem is still not solved because the similarity scores could be almost the same for polysemous words.

In this paper, we propose an implicit domain identification and switching mechanism for dialogue management to make the process of domain identifying and context switching transparent to users. The basic idea of the proposed approach is to use the domain-specific concepts or concept sequences to classify user's dialogue into one of the available service domains. For example, for a multi-domain system providing stock quote and weather forecast service, when a user says a company name, it's obvious that he/she is going to use the stock quote service, not the weather forecast service. If ambiguity occurs, such as *"Tai2 Nan2"* mentioned above, the dialogue history or the discourse of the conversation is taken into account to further decide user's target domain.

For cost-effectively merging the dialogue management of different application domains, we use the goal-oriented table-driven approach, proposed by Wang [4], to manage the man/machine interaction in a multi-domain system. In this approach, the domain-dependent factors in managing dialogues are extracted to form an external domain knowledge description. Add a new domain is just to incorporate a new domain knowledge description. And a root table is used to link different

[1] Company *"Tai2 Nan2"* is the Tainan Spinning CO., Taiwan. City *"Tai2 Nan2"* is city Tainan, located in southern Taiwan.

domain knowledge description and indicate the way to invoke the interested domain.

To prove the usability of the proposed approach, a multi-domain spoken dialogue system is set up to provide stock quote service and weather forecast service by integrating two existing single-domain dialogue systems, *eStock* and *eWeather*. *eStock* is an online stock quote system and *eWeather* is an online weather forecast system. To test the multi-domain system, two corpora collected by *eStock* and *eWeather* are used. Compared with the responses of the two single-domain systems, the multi-domain dialogue system makes 5.47% and 2.28% errors respectively. These low error rates indicate that the multi-domain system is almost as good as its two original single-domain systems.

This paper is organized as follows. First, the goal-oriented table-driven approach of dialogue management is overviewed in section 2. Then the design of the multi-domain dialogue management is introduced in section 3. Section 4 shows the system architecture of the experimental multi-domain dialogue system. Section 5 shows the experiment results and makes some discussions. The conclusions are made in section 6.

2. GOAL-ORIENTED TABLE-DRIVEN DIALOGUE MANAGEMENT

In our previous work, a domain-transparent framework is proposed to meet the portability requirements of dialogue management. In this framework, domain-dependent factors in managing dialogues are extracted to form an external domain knowledge description. The dialogue management kernel is simplified to be domain-independent, and becomes a standard and portable procedure. Thus, porting to another domain only needs the external domain knowledge description to be replaced, leaving the dialogue manager unchanged.

The domain knowledge description is modeled by the goal-oriented table-driven model and represented in the form of *Hierarchical Task Description Table Set*, HTDTS in brief. The basic idea of the model is to divide-and-conquer the complicated dialogue. In our daily conversation, it is natural for both sides to decompose a complex goal into several simple sub-goals so that each of them can be more easily satisfied by sub-dialogue. For example, to satisfy the goal of booking train tickets, the system has to get the user's itinerary (sub-goal) and find the proper scheduled train run, then, the seat needs to be arranged (sub-goal), and finally, confirm (sub-goal) with the user whether he/she booked the right ticket.

Based on the goal-oriented table-driven model, HTDTS is composed of a set of tables and each table handles the dialogue to achieve its own sub-goal. Take the example of booking train tickets mentioned above, the HTDTS is composed a root table where the dialogue starts, and tables with sub-goals of *get_itinerary*, *arrange_seat* and *confirm*. The tables are organized in a hierarchical structure according to a predefined dialogue flow to achieve a particular goal. Traveling from one table to another according to the hierarchy of HTDTS is like the behavior of opening a sub-dialogue to satisfy a sub-goal.

To allow the mixed-initiative dialogue, a user-initiated table switching mechanism is provided in the goal-oriented table-driven model. This mechanism makes users not obliged to follow predefined dialogue flow to complete a particular goal. The approach is to monitor user's intention by checking intention concept of the input concept sequence parsed from recognized utterance. The intention concept is critical to identify user's intention of current utterance. Once the value of intention concept changed, the user-initiated table switching mechanism is triggered to do the context switching of pushing the data of unfinished goal into stack, and backtrack to find the table that handles the intention concept of the new value, then switch to the table found. The backtracking is tracing the path of the HTDTS hierarchy from the interrupted table. The dialogue of the interrupted and unfinished goal will proceed from the interrupted point after the new goal is accomplished or user changes his/her intention back to the old goal.

3. THE MULTI-DOMAIN DIALOGUE MANAGEMENT

In this section, we discuss the issues of merging the dialogue management of different application domains efficiently by using the goal-oriented table-driven approach and the proposed implicit domain identification and switching mechanism that makes dialogue switching across different domains correctly and smoothly.

3.1 Domain Merging

Based on the goal-oriented table-driven paradigm, merging domains is just to incorporate different domain HTDTS and design a new root table to indicate the way to invoke the target domain HTDTS. Figure 1 depicts a simplified HTDTS of multi-domain dialogue management that handles dialogues of stock quote domain and weather forecast domain.

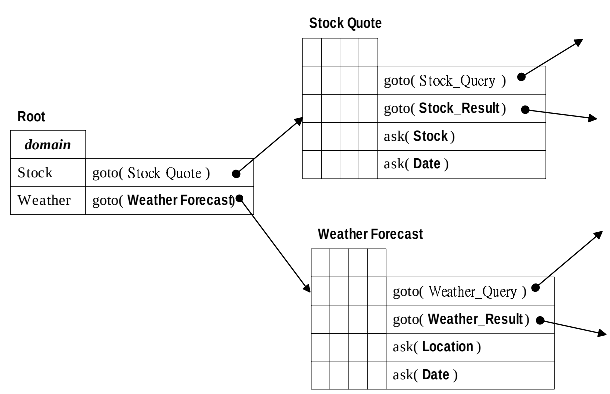

Figure 1: Simplified HTDTS of Multi-Domain Dialogue Management

As depicted in Figure 1, the table *StockQuote* is the root table of HTDTS of stock quote domain, and the table *WeatherForecast* is the root table of HTDTS of weather forecast domain. In the merged HTDTS, the table *Root* takes place of both original root tables and defines the conditions to invoke the HTDTS of domains. If the value of domain concept of the input concept sequence equals to *Stock*, the HTDTS of stock quote domain will be invoked by *goto* its original root table *StockQuote*. The similar process for the value of domain concept equals to *Weather*. The domain switching could happen when the dialogue proceeds not in the root table. For example, the domain concept is changed to *Weather* while the dialogue proceeds in the table *StockQuote*. The domain switching mechanism will be discussed in details in section 3.2.

3.2 Domain Identification and Switching

The basic idea of domain identification is to use the domain-specific concepts or concept sequences to classify user's dialogue into one of the available service domains. For example, two utterances of "*Please tell me the weather in Taipei tomorrow*" and "*Please tell me the price of TSMC*[2]" will be parsed to the concept sequences of "tell_me weather_topic location date" and "tell_me stock_topic stock" respectively. For a multi-domain dialogue system that provides stock quote service and weather forecast service, it's obvious that the "tell_me weather_topic location date" is the domain-specific concept sequence of weather forecast service, and the "tell_me stock_topic stock" is the domain-specific concept sequence of stock quote service. Similarly, if the user uttered a fragment in conversation, such as "*weather*" or "*price*", the corresponding concepts of weather_topic and stock_topic are domain-specific concepts of weather and stock domain respectively.

To make use the domain information provided from input concepts of dialogue management, a domain identification table is used to define the identification rules. Table 1 shows an example of domain identification table. Each row in the table defines an identification rule and each column specifies the input concept that is critical to identify domain. For the limited space of the column, Loc. stands for location, W_topic stands for weather_topic and S_topic stands for stock_topic. The possible conditions on concept include valid concept (denoted by "+"), void concept (denoted by "–") and "don't care" (denoted by "×") the concept no matter what it's valid or void.

concept				Domain
Loc.	W_topic	Stock	S_topic	
×	+	–	–	*WeatherForecast*
–	–	×	+	*StockQuote*
+	–	–	–	as previous domain default is *WeatherForecast*
–	–	+	–	as previous domain default is *StockQuote*

Table 1. Domain Identification Table

The first two rows in Table 1 handle the input concept sequence with domain-specific concepts, weather_topic, stock_topic, and decide the domain immediately. The last two rows handle the ambiguity caused by polysemous words without sufficient information given in the utterance. The dialogue history will be referenced to decide user's interested domain. Take the following dialogue as example,

[2] Taiwan Semiconductor Manufacturing Company

<u>example dialogue 1</u>
U1: *Please tell me the price of TSMC.*
S1: *The price of TSMC is eighty-three dollars. Anything else?*
U2: *How about "Tai2 Nan2"?*

<u>example dialogue 2</u>
U1: *Please tell me the weather in Taipei tomorrow.*
S1: *The weather in Taipei is partly cloudy… tomorrow. Anything else?*
U2: *How about "Tai2 Nan2"?*

Both "*Tai2 Nan2"* in user's second utterance U2 could be parsed as concept location or stock. If the input concept is location, the third rule in Table 1 will be applied. The domain will be decided as the domain of previous turn, or set to the default domain if it's the first turn. Thus, in example dialogue 1, the domain is set to *StockQuote* and in example dialogue 2, the domain is set to *WeatherForecast*. If the input concept is stock, the forth rule in Table 1 will be applied, and the domain decision process is similar to the location input.

After the domain of input concept sequence is identified by domain identification mechanism, a domain concept with the value of domain id will be added to the input concept sequence. The following is an example for "*Please tell me the price of TSMC*".

concept	Domain	tell_me	stock_topic	stock
value	*StockQuote*	*tell me*	*the price*	*TSMC*

The domain concept is similar to the intention concept mentioned in section 2. Thus, once the value of domain concept is updated by the domain identification mechanism, the user-initiated table switching mechanism is triggered. Then, backtracking according to the hierarchical path from the interrupted table, and the root table will be traversed and found to handle domain concept of the identified domain. Finally, the dialogue control is switched to the root table and the interested domain is invoked by the way defined in the root table. If the old domain is unfinished, the dialogue will proceed from the interrupted point when the new domain is finished or user intentionally goes back to the old domain.

4. SYSTEM ARCHITECTURE

We have setup a multi-domain spoken dialogue system to provide stock quote service and weather forecast service by integrating two existing single-domain dialogue systems, *eStock* and *eWeather*. *eStock* is an online stock quote system and *eWeather* is an online weather forecast system. In this section we will briefly introduce the architecture of our multi-domain dialogue system. The block diagram of the system is depicted in the Figure 2.

Generally speaking, for a spoken dialogue system, the most domain-dependent component is dialogue management. The grammar and the generation rules for language understanding and language generation are also domain-dependent [2]. In our multi-domain spoken dialogue system, the grammar of *eStock* and *eWeather* is integrated to a unified multi-domain grammar. The HTDTS used here is a merged one of two HTDTS of the single-domain system by the merging process mentioned above.

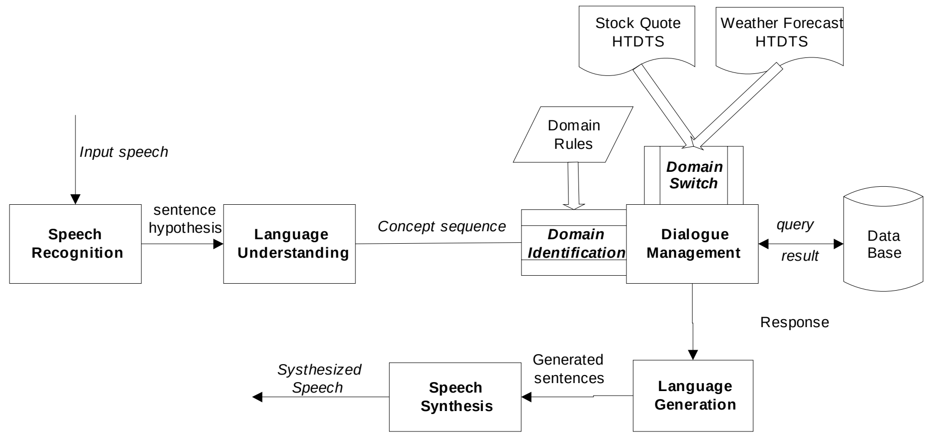

Figure 2 : Architecture of Multi-Domain Dialogue System

The language generation rules remain two single-domain rule sets because we put no effort on this part now.

As user speaks to the system, 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 by parsing it and represents in a concept sequence. Based on the input concept sequence, the dialogue manager first determines the most appropriate domain by using domain identification mechanism and adds the domain information to the input concept sequence. If the domain changes, the domain switching mechanism will be triggered to switch to the specified domain. Then, the dialogue management determines the most appropriate response to user or queries the database. The responses are presented in spoken natural language generated by the language generation component and text-to-speech synthesizer.

5. EXPERIMENT AND DISCUSSIONS

To prove the usability of the proposed approach, two corpora are used to test the multi-domain spoken dialogue system. One corpus consists of 1,042 utterances collected by *eStock*, the other corpus consists of 972 utterances collected by *eWeather*. Compared with the responses of the two single-domain systems, the multi-domain dialogue system makes 5.47% and 2.28% errors respectively. Most errors in the multi-domain system are caused by the erroneous speech recognition and parsing results that were occurred in the original domain, and are thus unavoidable.

In this work, the proposed approach could fast construct an integrated service of domains without redesigning the dialogue management. The easy coded domain identification table makes use of the domain information provided from input concepts. By referencing the domain identification table, the domain identification and switching mechanism identifies and switches domains automatically without interfering users. The goal-oriented table-driven dialogue management model also provides an efficient way to merge the dialogue management of different domains. The low error rates encourage us that the proposed approach facilitates building a multi-domain spoken dialogue system.

6. CONCLUSIONS

To build a multi-domain spoken dialogue system, two major difficulties for the dialogue management are encountered but little touched. One is to interpret users' interested domain correctly and switch among domains smoothly. The other is the high cost of merging the dialogue management of different application domains. Based on the goal-oriented table-driven dialogue management model in our previous work, an efficient merging process and an implicit domain identification and switching mechanism are proposed in this paper. Using the proposed approach, without redesigning the dialogue management, a multi-domain system is built by integrating two existing single-domain spoken dialogue systems, one for stock quote domain, the other for weather forecast domain. Tested on two corpora collected by both single-domain systems, the multi-domain system contributes 5.47% and 2.28% errors comparing with the responses of two single-domain systems respective. The low error rates indicate that the multi-domain system is almost as good as two single-domain systems.

7. ACKNOWLEDGEMENTS

This paper is a partial result of Project A311XS1211 conducted by ITRI under sponsorship of the Ministry of Economic Affairs, R.O.C. Especially thanks to Mr. Jyh-Shing Hsu, Dr. Shih-Chieh Chien, and Dr. Jau-Hung Chen for their technical support on system integration, speech recognition and speech synthesis.

8. REFERENCES

[1] Seneff S. et al., "Multimodal Discourse Modelling In A Multi-User Multi-Domain Environment", *Proceeding of ICSLP'96*, USA, 1996.

[2] Lin B.-S. et al., "A Distributed Architecture For Cooperative Spoken Dialogue Agents With Coherent Dialogue State and History", ASRU'99, 1999.

[3] Hsu J -S. et al., " ePortal - Voice Portal System Developed in ITRI ", pp. 229-232, Vol 1, *International Conference on Speech Processing*, Aug. 2001, ETRI, Daejeon, Korea.

[4] Wang H.-M. and Lin Y.-C. , "Goal-Oriented Table-driven Design for Dialogue Manager", *Proceeding of ICSLP 2000*, Beijing, 2000.