

The Design Of A Chinese Spoken Dialogue System Engine

XianFang Wang, LiMin Du

{wangxf, dulm}@iis.ac.cn

Center for Speech Interactive Information Technology

Institute of Acoustics, Chinese Academy of Science, Beijing PRC 100080

ABSTRACT

In this paper, we propose the design principle of a Chinese Spoken Dialogue System Engine. The principle of the design is classifying the actions of a dialogue system into two types: the interactive actions and the task transaction actions. The purpose of the interactive actions is to know the users' intention and the information items that are necessary for task transaction and the task transaction actions are those actions that provide the data and the services according to the requirement of the user. According to this principle, one dialogue system is divided into two parts: Dialogue Manager (DM) and Task Transact (TT). It is easy to port a dialogue system to another domain using this principle.

1. INTRODUCTION

With the development of spoken language technology, a friendly spoken interface will be the man/machine interface of the next generation [1]. Many spoken dialogue systems have been demonstrated to be effective in various applications, such as travel information service, stock information service. However, there are still many obstacles in using the spoken interface in our everyday life. One of the obstacles because of which spoken dialogue systems are not yet widespread is the difficulty in porting a dialogue system across different domains [2,3]. Many dialogue systems should be modified much if they are ported to be another domain and it is difficult to debug and test during the porting process.

To increase the portability of the system means to decrease the dependency of the system on a specified domain. But, if we analyse the process of a dialogue, we will find that the independency of the system on the domain and the performance of the system are conflicting. In this paper, we don't touch the questions in the field of speech recognition, language understanding and speech synthesis. We just focus on the problem of dialogue management. The aim of dialogue management is to control the dialogue flow so as to retrieve the information items that are necessary for task transaction from the user, and to answer the question of user according to the result of task transaction. In many situations, the relation between the information items is very complicated. Those items are depending on each other. For example, in a railway ticket information system, the depart station and the destination station and the train number are not independent. In a specified dialogue system, the dialogue flow maybe have been predefined or is determined by the system according to the information items that the user has provided and the status of the system.

If we pay all our attention to the problems above, we would find that it is difficult to develop the dialogue system independent on a specified domain. Let's analyse the fundamental behaviour of a dialogue system. A dialogue system is responsible for two tasks: comprehending the users' intention and solving the problem according to the users' requirement, that is, to interact to the user and to complete the task. Of course, the task transaction is task-dependent; it is different among different domains. But the interactive behaviour has common rules in different domains. For example, if the user has not provided an information item that is necessary for task transaction, the system should query the user; if the system cannot be sure which is correct in some possible choices because of the difficulty of speech recognition or language understanding, the system should ask the user to clarify it. The rules of dialogue above are the same in all dialogue systems. So we can develop a portable dialogue system engine used in many domains and develop the domain-dependent component for the specified task.

In this paper, we introduce a Chinese Spoken Dialogue System Engine that adopts a domain-transparent design of dialogue management to increase the system portability. The engine applies a mix-initiative dialogue management strategy. The system is currently implemented in two different domains: a train ticket information system and a stock information system. These two systems both take the telephone speech as input.

In the following sections, the principle of designing a portable dialogue system is discussed first. Then we will propose the structure of our portable dialogue system. In the fourth section, we will illustrate how to port a dialogue system to another domain. At the end of this paper, some discussions are given and some problems that the author met during the process of developing are presented.

2. THE PRINCIPLE OF DESIGNING A PORTABLE DIALOGUE SYSTEM

The principle of the design is to classify the actions of a dialogue system into two types: the interactive actions and the task transaction actions. The interactive actions are those used to know the user's intention and the information that is necessary for task transaction, for instance, whether the user wants to book tickets or query information, the train number that the user wants to book, etc. The task transaction actions are those actions that provide the data and the service according to the requirement of the user, for instance, querying the database, modifying the database, operating devices, etc.

This principle can be used to determine the role of the Dialogue Manager (DM) and the Task Transact (TT). Although the content of the interactive actions are task-dependent, the methods of the interactive action are similar in different domains. That is, if the system wants to know an information item, it should ask the user; if it has doubt as to the reliability of the information items, it should ask the user to clarify it. A task-independent algorithm adopted by the DM could handle them with the help of the task-dependent information of dialogue management. The design of the DM doesn't need to touch any details about the given task. Thus, the portability of the system is improved. The DM that is the same in different domains performs the interactive actions. The TT that is different in different domains performs the task transaction actions. A configure file is used to define the interfaces between DM and TT and the necessary information for language

understanding and dialogue management such as semantic network, response template. An initialise component reads the configure file to initialise the DM and the TT when the dialogue system is started.

3. THE STRUCTURE OF A PORTABLE DIALOGUE SYSTEM

In the following sections, the diagram of our dialogue system is proposed first. The functions of those modules are illustrated. Then, a table named Task Control Table (TCT) are given and we will demonstrate how the Task Assigner use this table manages the dialogue flow and call the Task Transact.

3.1 The diagram of our dialogue system

The DM plays a central role in the Spoken Dialogue System Engine. It manages a dialogue history and uses it to guide the Linguistic Processor (LP) and the Automatic Speech Recogniser (ASR) by transferring the discourse context and the system's expectation as parameters. An Intention Estimator (IE) is included in the DM to extract the complete meaning for further processing. A Task Assigner (TA) included in the DM is to call the TT according to the users' intention. All the task-dependent information that the DM need in the processing above comes from the configure file. The response templates generated by the Dialogue Manager or by the Task Transact are passed to the Language Generator (LG) to generate response. The Text to Speech component (TTS) transforms it to voice and send it to the user.

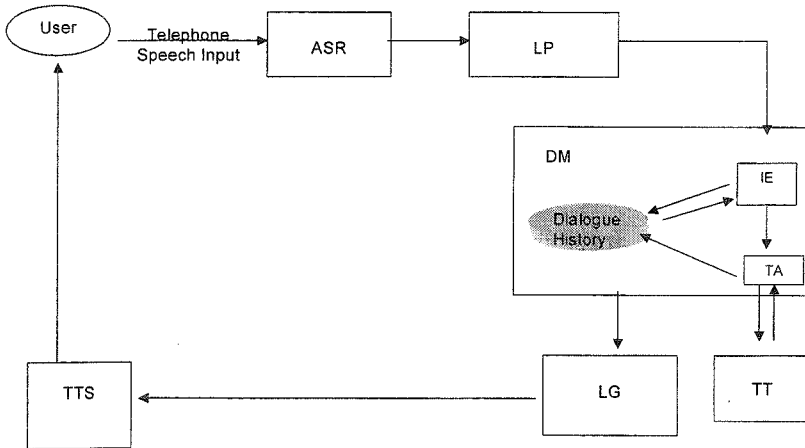


Figure 1. Diagram of our dialogue system

3.2 Task control table

The task assigner must judge whether the information items that have been provided by the user are enough and call the Task Transact to achieve the task. If the information items are not enough, it controls the dialogue flow to ask the user to answer or clarify it. The task control table is used to define how to control the dialogue flow for a given task. For each service that offered by the dialogue system, for instance, purchasing the railway ticket, a task control table item is appointed to control the dialogue flow when the user requires the service. The content of the task control table item contains the information items that are necessary for system to offer this service, for instance, the depart station, the arrive station, the train number, the depart date. Those items constitute an information item table. For each item included in information item table, a query template that used to response to the user when the item is not provided by the user and a confirm template that used to response to the user when the reliability of the information item is not high enough should be provided. The designer should also specify how to call the module of the Task Transact for a given service. This is often the name or address of function included in the TT.

3.3 The steps of dialogue management

The Task Assigner controls the dialogue flow according to the content of the task control table. After the Intention Estimator has been sure of the user's intention and the value of the information items, the Task Assigner selects the appropriate task control table item to perform the consequent processing. The steps are the following:

1. TA checks whether the user has appointed the service that he or she need. If not, query the user's intention. If yes, go to the next.
2. TA checks whether there are information items that are not checked. If no, go to step 5. If yes, select next information item unchecked to check, go to the next.
3. TA checks whether the value of the information item have been provided by the user. If yes, go to the next. If no, TA responses to the user directly, calls the Language Generator with a query template provided by the designer.
4. TA checks whether the reliability of the value of the information item is high enough. If yes, go to step 2. If no, TA responses the user directly, calls the Language Generator with a clarify template provided by the designer.
5. TA calls the function of the Task Transact and responses the result that returned by the TT.

4. PORT A DIALOGUE SYSTEM TO ANOTHER DOMAIN

For a dialogue system in a specified domain, it includes a domain-independent component – Dialogue Manager, a domain-dependent component – Task Transactor and a task configure file that contains task control table. When porting a dialogue system to a new domain, the analysis of the information about the semantic extraction of the users' input as well as the service afforded by the new system should be carried out first. We have assumed that a user is willing to get a service that offered by the system. So the goal of the interactive action of the system is to know which service the user need and the information items that are

necessary for offering the service. The design of the task-dependent is based on the analysis. Then, the TT of the new domain is developed and the interfaces between the DM and the TT are depicted based on the services. Finally, the task control table is written according to the analysis and design above. The DM does not need to be changed during the porting process.

5. DISCUSSION

During the developing process, we met some difficulties when the values of the information items cannot be used for task transaction directly. For instance, in a railway ticket booking system, the depart date maybe have a value "tomorrow" or a value "5th, OCT". That is, the date can be an absolute date or a relative date. We can only apply one of them in Task Transact. Thus, we must translate one format to another. This work is task-dependent. In the current version of our system, we add an information item value check function for each domain to translate the value of synonymous information items.

Another shortcoming of this method is that it lacks the ability of reasoning. Some methods, for instance, Semantic Network, have this ability. In the task control table, we have assumed that all information items are independent. There are no hierarchy among those information items. This is not true.

6. CONCLUSION AND FUTURE WORK

In this paper, we advocate a domain transparent design of a Chinese Spoken Dialogue System Engine. The model of dialogue control is designed using the strategy of machine-human mix-initiative. We argue that the method has many advantages concerning the portability of the system, the easiness of developing a system. However, some shortcomings still exist in this method. There are many things to improve.

This method is currently applied in two applications – a train ticket information system and a stock information system. The result confirms the claims above. In the future, we will focus on the reasoning ability of the system to make the system suitable for more complex application.

ACKNOWLEDGEMENTS

This work was partially funded by the National Key Fundamental Research Project – Image, Speech, Nature Language Understanding and Information Mining. It is a part of its sub-project G1998030505: The Research of the Platform for Chinese Spoken Dialogue System.

REFERENCES

- [1]. Yi-Chung Lin, et al. "The Design of a Multi-Domain Mandarin Chinese Spoken Dialogue System", Proceedings of ICSLP98, pp 41-44, Sidney, Australia, 1998.
- [2]. Michael F McTear, "Modeling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit", Proceedings of ICSLP98, pp 1223-1226, Sidney, Australia, 1998.
- [3]. Peter A. Heeman, et al. "Beyond structured dialogues: factoring out grounding", Proceedings of ICSLP98, pp863-866, Sidney, Australia, 1998.