Integrating Flexibility into a Structured Dialogue Model: Some Design Considerations

Michael F. McTear, Susan Allen, Laura Clatworthy, Noelle Ellison, Colin Lavelle and Helen McCaffery

Faculty of Informatics, University of Ulster at Jordanstown, United Kingdom

ABSTRACT

Structured dialogue models are the most commonly used dialogue models in commercial systems, particularly as they are relatively easy to design and re-use. The current paper reports on a study that examined the feasibility of combining more flexible dialogue control with a structured dialogue model. Several systems were built using the RAD (Rapid Application Developer) component of the CSLU toolkit, augmented with the Phoenix natural language parsing system and a dialogue manager that used a representation of the system’s information state to determine the system’s next question or action. Results indicated that with an optimized continuous speech recognizer a dialogue permitting flexible input can be concluded efficiently and successfully, while in cases of degraded recognition the recovery strategies and more structured dialogue control enhance the likelihood of a successful transaction. The paper discusses a number of design issues that support developers in making structured dialogue models more flexible.

1. INTRODUCTION

Structured dialogue models, in which the user’s input is generally restricted to single words or phrases that provide responses to carefully designed system prompts, are the most commonly supported dialogue models in current commercial spoken dialogue toolkits. Examples include the Nuance Developers Toolkit, the Unisys Natural Language Speech Assistant, and evolving standards for speech-enabled web applications such as VoiceXML [10, 13, 14]. Structured dialogue models have the advantage that they are relatively easy to design and re-use. The required vocabulary and grammar for each dialogue state can be specified in advance, resulting in more constrained and consequently more reliable speech recognition and language understanding [4, 6, 9]. However, the disadvantage with these models is that they are inflexible and that, as the dialogue becomes more complex, they can lead to a combinatorial explosion of dialogue states [1]. Attempts have been made to maintain the advantages of structured dialogue models while providing greater flexibility in terms of user input and dialogue flow [4, 6]. User input can be more flexible if the system includes a natural language understanding component. This enables the user of a flight information system, for example, to speak all the required information in one utterance, or in different combinations of the information as required, instead of having to provide each of the required parameters one at a time. To handle more flexible input such as this, the system needs to keep track of what information the user has provided and what the system still needs to know.

This is achieved in various systems using a frame-like data structure that maintains a record of the dialogue history [2, 3, 5, 11]. Finally there needs to be a dialogue control algorithm that decides on the basis of the dialogue history what questions the system needs to ask, whether and when information should be verified, and at what point the database or other external information sources can be consulted.

The current paper reports on a study that examined the issues involved in introducing more flexible dialogue control into a structured dialogue model with a view to evolving a set of design guidelines for spoken dialogue systems. The next section describes a number of systems that were built using the CSLU toolkit, which is available free-of-charge under a license agreement for educational, research, personal, or evaluation purposes from the Center for Spoken Language Understanding (CSLU) at the Oregon Graduate Institute of Science and Technology [12]. An evaluation of the performance of these systems is followed by a discussion of the issues arising from this study, including some guidelines for future work. The final section summarizes the paper and draws some conclusions.

2. SYSTEM ARCHITECTURE

Several systems were built using the RAD (Rapid Application Developer) component of the CSLU toolkit. In its basic form RAD provides the developer with a graphically-based authoring environment for designing and implementing simple spoken dialogue systems using a state-based dialogue model. Speech input can take the form of isolated words that are recognized using a tree-based recognizer, or relatively short strings of words whose structure has been specified by a finite-state recognition grammar. There is no language understanding component provided in the basic system, and for the purposes of this study a version of the Phoenix parser [15], ported to the CSLU toolkit, was used. One of the systems used the Profer parser, which has similar functionality to Phoenix and is currently under development at CSLU [8]. Three systems were built for travel information (flights and trains), a fourth system was built for movie information, and a fifth system for directory enquiries. The systems were designed in such a way as to permit flexible user input to the initial system prompt. In the optimal case of good speech recognition and understanding the system could retrieve the required information from its database and complete the transaction with a minimum number of two system turns and one user turn. Where this strategy was unsuccessful, the system could enter a more constrained dialogue to recover from problems such as misrecognitions and incomplete user input. Figure 1 presents an abstract view of the data flow in the implemented systems.
2.1. Speech Recognition and Understanding

In the absence of a large vocabulary speech recognizer, the currently available facilities in the CSLU toolkit were exploited to permit the user to input an unconstrained string at the first prompt. A finite state recognition grammar was constructed that could accept, within a structure consisting of optional and obligatory words, any word in the vocabulary in combination with optional silence and garbage words. More constrained recognition grammars were constructed for other dialogue states that required more structured input, such as dates, times, or single items of information such as a departure location or a name. The recognized string was then passed to the parser, which produced a bracketed parse containing the relevant concepts and their values, as shown in Figure 2.

2.2. Information Verification

This process involved extracting the attribute-value pairs from the bracketed parse and updating the system’s dialogue history, a frame-like structure consisting of a set of attribute-value pairs, with any values that had been extracted — for instance, in the case of this example, with the departure and arrival locations and the arrival time range.

2.3. Additional questions

This process involved confirmation of any acquired values and elicitation of missing values based on the information in the system’s dialogue history. The process was implemented as a sort of production system with a series of ordered condition-action statements, e.g.:

- If Destination is unknown, then ask <destination question>
- If Destination and Origin are unknown, then ask <route question>
- If Destination is unconfirmed, then ask <destination-confirmation-question>

Various combinations of questions were explored to optimise transaction times. For example: if both destination and origin were unknown, the system would ask a question about the route, although the system reverted to single item questions in the case of subsequent misrecognitions. Sub-dialogues were used for the ‘additional questions’ process to support modularity and re-use of recurring dialogues for items such as dates and times. More constrained recognition and language understanding grammars were written to parse the more restricted input anticipated in these sub-dialogues.

3. EVALUATION

The results of user evaluations with the systems showed that when the initial input was correctly recognised, the dialogue could be concluded efficiently with a minimum of exchanges. However, as expected, there was a high error rate for the recognition of the longer unconstrained strings spoken at the initial system prompt, with word accuracy rates falling to between 10% to 20%. In this case, however, the dialogue control mechanisms directed the dialogue into more constrained states in which the information could be elicited more reliably, with high word accuracy rates in these states, reaching almost 100% in some cases. The speech understanding and dialogue control components were also tested off-line with keyboard input, resulting in transaction accuracy rates of almost 100%. These results suggest that with an optimised continuous speech recogniser a structured dialogue permitting flexible input can be concluded efficiently and successfully in a minimum of turns.
4. DISCUSSION

The purpose of the current study was to examine the issues involved in introducing more flexible dialogue control into a structured dialogue model. These issues can be grouped under two main headings.

4.1. Knowledge integration

It is often claimed that the ability to participate in a spoken dialogue involves the integration of a number of different knowledge sources. In the current systems integration between the dialogue manager and the speech recognition and understanding modules was built into the dialogue design using sub-dialogues with special vocabularies and grammars. A more sophisticated approach is to load vocabularies and grammars dynamically at run-time, depending on current state of the dialogue.

Two of the systems involved a combination of dynamic recognition and the use of the system’s database to determine the subsequent dialogue flow. In the ‘movie’ system the user was able to pose a range of different queries, each requiring different outputs. For example: a query specifying a movie title, a day and a time period required a location as output, while a query with location, day and time period required a list of movie titles. Once the values for these queries had been elicited and verified, the system searched the database for a set of results to output to the user. Other queries that were less constrained required the system to retrieve information from the database and then use this information to offer further choices to the user to constrain the query so that a reasonable set of results could be found and output. For example: the query ‘where’s <movie-title> showing?’ involved a database search for locations. These would be presented to the user using dynamic recognition, and on receiving the user’s choice of location, the system searched for and returned the times for the movie at that location. Similar database lookup and dialogue paths were implemented for several other unconstrained queries.

Database information in the directory assistance system was used in a different way to support the speech recognition and language understanding processes and to determine dialogue control. In this system the possibility of misrecognitions and misunderstandings by the system as well as unknown or inaccurate information given by the user was handled through a process of database lookup involving constraint relaxation. This was achieved using database queries containing declining levels of filled slots, as follows:

1. All information is given and matched (title, forename, surname, department).
2. Department removed, as callers often provided inaccurate department names.
3. Title and surname only, which eradicates the use of incorrect forenames.
4. Forename and surname only.

5. Forename and department, as it was found that, within certain departments, employees are only known by their forename and their department e.g. ‘John from Estates’.

Surname is last because it is the most likely to produce multiple entries. However, if a surname is correct, this will guarantee that a record is retrieved that can then be output to the user. The following example clarifies this process.

| Output from Speech Recognition and Language Understanding: Mr, Jane, Finlay, Administration |
| Database query with all four items: FAIL |
| Database query with department removed: FAIL |
| Database query with title and surname: SUCCEED and RETURN Mr, Leslie, Finlay, Information Technology |

This sequence does not guarantee that the correct employee will be found, especially if all of the recognized words or the critical ones are inaccurate. It does, however, ensure that employee details that are returned to the user are accurate, as they will refer to an actual employee that exists in the database and not one created from the recognized input. If the user rejects this record, the system enters a repair sub-dialogue and prompts the user for the surname only, using further repair mechanisms such as spelling, if required. Often the surname will be sufficient to retrieve a unique record. In cases of multiple entries with the same surname, the system detects that more than one record has been retrieved and uses other information such as forenames and department to disambiguate. Taking these processes together, the effect is that the system will provide results to the user path based on information in the database, which is preferred over the less reliable information provided by the speech recognition process. Moreover, this nature of this information determines the dialogue path choices and repair mechanisms that the system will subsequently follow.

4.2. Design issues

The present study has highlighted a number of design considerations that arise from the combination of structured and flexible dialogue management. The systems begin with a flexible dialogue strategy that, in the best case of accurate recognition and complete user input, will result in a short and successful transaction. Where problems arise, there has to be a trade-off of efficiency against accuracy by making the dialogue progressively more structured until the required items of information have been successfully elicited. Similar methods have been applied in a number of systems [4, 7]. In the current systems, however, an analysis of the types of information to be negotiated in the dialogue played an important part in decisions concerning these methods. Determining priorities in the information to be elicited was based on an analysis of the domain. In the directory enquiry system this involved ranking the items of information in order of importance while, in the case of the movie system,
queries were analysed to determine whether they were sufficiently constrained to provide the basis for a reasonably small number of database matches. Similar analysis was carried out in the flight and train timetable information systems. Once the information requirements of the system have been suitably analysed, decisions can then be taken about dialogue control — for example, whether structured or flexible control is recommended, what types of repair mechanisms can be used, and which items require verification. The use of external information, such as the system’s database, contributes to this process by supplying values that can be used in supplementary system questions whenever the user’s initial queries are insufficiently constrained.

5. CONCLUSIONS

The current paper has examined some ways of extending structured dialogue models. Users do not have to be restricted to input of single words or carefully chosen phrases if the system includes a natural language understanding module that can extract the relevant concepts from the user’s utterances. A dialogue history that maintains a record of these concepts enables the dialogue control module to decide what questions to ask next — whether to acquire new values or to verify already elicited values. In the best case scenario a more flexible system will lead to a reduced transaction time. However, the drawback of this flexibility is that greater emphasis is put on the speech recognition component, which has to deal with the recognition of longer, less constrained strings of continuous speech. This paper has illustrated how a system can cope with less than optimal results from the speech recognizer by reverting to more structured dialogue control mechanisms. Information from external sources such as the system’s database can be used to provide further support for the speech recognition and dialogue control processes. These various factors will require further consideration for dialogues involving Web-based information using evolving standards for interactive speech systems such as VoiceXML. While these dialogues will need to be carefully structured to ensure accurate recognition of the user’s speech, they will also need to provide sufficient flexibility to enable expert users to conclude transactions quickly and efficiently, without becoming involved in lengthy menu-based interactions.

6. REFERENCES


