Dialogue Management for Telephone Information Systems

Scott McGlashan and **Norman Fraser** and **Nigel Gilbert**
Social and Computer Sciences Research Group
University of Surrey
Guildford, U.K.

Eric Bilange
Cap Gemini Innovation
118 rue de Tocqueville
5017 Paris, France

Paul Heisterkamp
Daimler-Benz AG
Wilhelm-Runge-Strasse 11
D-7900 Ulm, Germany

Nick Youd
Logica Cambridge Ltd
104 Hills Road
Cambridge, U.K.

Abstract

A distributed approach to spoken dialogue management for real-time telephone information systems is outlined.

1 Introduction

The approach to dialogue management described here has been developed as part of the Sundial project (Speech UNderstanding in DIALogue) whose goal is to build real-time integrated computer systems capable of maintaining co-operative dialogues with users over standard telephone lines.[1] Systems have been developed for four languages – French, German, Italian and English – within the task domains of flight reservations and enquiries and train enquiries.

Each system consists of five components: a speech recognition component inputs acoustic speech signals from the telephone and outputs a word lattice; a parser extracts a plausible string from the lattice and assigns it syntactic and semantic representations; a dialogue manager gives these representations an interpretation, decides how the dialogue may continue and, if it is the system's turn to speak, plans a system utterance; a text generator constructs a detailed linguistic representation from the plan; and a text-to-speech system synthesizes the representation for transmission by telephone.[2]

2 Two Problems of Dialogue Management

The SUNDIAL dialogue manager addresses two problems facing spoken dialogue systems. The first is that, for maximum usefulness, dialogue management needs to be **generic**: i.e. the dialogue manager module must be able to interpret and generate utterances in more than one language and across more than one task domain. The second problem is that dialogue management must provide **co-operative interaction** with the user. To achieve this, the system needs to produce utterances which are perceived by the user as natural, coherent and helpful within the context of the dialogue. The purpose and modality of the dialogue forms part of this context: in our case, task-driven spoken dialogues. The system needs to deal with inconsistent task information, such as that obtained when the speech recognizer and parser yield a plausible but incorrect interpretation; for example, the user said *I want to fly to Lannion*, but the system understands *London* rather than *Lannion*.

3 A Distributed Solution

The Sundial solution to these problems lies, in part, in the **distributed architecture** of the dialogue manager. The dialogue manager is responsible for providing a co-operative interaction with the user (Gerlach and Horacek, 1989). From the user's point of view, system utterances must be appropriate and helpful to the user *in the current dialogue context*. To achieve this, both interpretation of user utterances and planning of system utterances must be informed by the past and current states of the interaction. Co-operative dialogue management, therefore, requires the construction and maintenance of an **interactional model**: i.e. a model which specifies the layers of structure which can be distinguished in dialogues. Following Grosz and Sidner (1986) we distinguish linguistic structure, attentional, or belief structure, and intentional structure. Intentional structure is further differentiated into dialogue structure and task structure (Bunt, 1989). However, rather than using a **unitary model** where these layers are given as a single structured representation, we have adopted a **distributed model** where these layers are distributed across a number of modules. Each module consists of a partial model of the interaction, rules for updating the model as well as associated dialogue management functions. Furthermore, modules may need to collaborate with each other in order to determine how to update

[1] SUNDIAL is partly funded by the Commission of the European Communities under the ESPRIT II programme as project P2218.

[2] See Peckham (1991) for an overview of the Sundial system.

their model: their update rules may depend upon information maintained in another module.

The interactional model is *falsification definite* since agents **assume** rather than infer that their belief model corresponds to that of the other agent. This places an obligation on agents to make explicit the state of their belief model so as to give the other agent the opportunity to correct or modify it. For example, if the system utters *You want to. go to London* in response to the user's request, the user has an opportunity to contest the current state of the system's belief model.

The Sundial dialogue manager consists of five modules. The *linguistic interface module* interfaces the dialogue manager with the parser and is responsible for maintaining a linguistic model of system and user utterances. The *dialogue module* is responsible for maintaining a model of dialogue context, building an interpretation of user utterances and determining how the dialogue should continue. Dialogue structure is based on the framework of Roulet and Moeschler in which dialogue is hierarchically structured into exchanges, interventions and dialogue acts (Moeschler, 1989). In order to determine the appropriate contextual interpretation of user utterances, the dialogue module interacts with the *belief module* which maintains a model of belief containing not just concepts created directly as a result of user utterances, but also inferential extensions. For example, if the system initiated an exchange to determine the departure date of a flight, this exchange can be closed if the belief model can interpret the user's utterance as referencing a 'date' concept. The belief module requires context information from the dialogue module in order to guide the interpretation process. For example, *the second* can reference date or flight concepts (*the second of May* or *the second flight*) depending on the context. The belief module additionally cooperates with the *message planning module* in order to provide semantic descriptions of concepts referenced in the plan of system utterances.

Finally, the *task module* is responsible for maintaining a model of the task structure of the dialogue, consulting domain-specific application databases and informing the dialogue module of the current state of the task. Typically, this involves deciding whether sufficient task information has been provided by the user and, if not, which additional parameters are required. Less frequent are interactions which arise when a user provides sufficient, but incorrect, information: for example, there is no flight to the stated destination city. Rather than simply informing the dialogue module that the task cannot be satisfied, the task module attempts to relax task parameters and suggest alternative values for them (Guyomard and Siroux, 1988). For example, if there are no flights available in the morning of the given departure day, flights in the afternoon may be suggested. In such cases, system utterances are selected and formulated so as to make explicit to the user the fact that an alternative is being offered.

4 Implementation

The Sundial dialogue manager has been implemented in Quintus Prolog and tested on a variety of different hardware platforms. It has been integrated with the rest of the Sundial system and successfully manages a range of English, French, and German spoken (public network) telephone dialogues relating to flight enquiries, flight reservations, and train enquiries. Language and task can be changed in the dialogue manager simply by resetting switches which govern the choice of static knowledge bases consulted during dialogue management.

Current average response times for the entire Sundial system (including speech recognition and synthesis) are around 10 seconds, with the dialogue manager taking 1-2 seconds. However, a significant amount of optimization has still to be carried out, and it is expected that response times will approach real time within the next six months. The lexicon currently contains around 300 entries for each language.

5 Conclusion

Two problems facing dialogue management in telephone information systems — co-operative interaction and portability across languages and task domains — have been addressed by the dialogue manager in the Sundial system. Our solution has been to distribute the interactional model, as well as functions dealing with co-operative interaction, across a number of semi-independent modules. Each module is an expert on a local part of the interactional model. The language of spoken interaction and the application task can both be changed by instructing the relevant modules to consult different knowledge bases. In this way the generic Sundial dialogue manager can be customized for specific languages and applications.

References

Harry C. Bunt. Towards a dynamic interopretation theory of utterances in dialogue. In Ben A.G. Elsendoorn and Herman Bouma, editors, *Working models of human perception*. Academic Press, 1989.

Michael Gerlach and Helmut Horacek. Dialog control in a natural language system. In *Proceeding of the 4th Conference of the European Chapter of the Association for Computational Linguistics*, 1989. Manchester.

Barbara J. Grosz and Candace L. Sidner. Attentions, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, July-September 1986.

Marc Guyomard and Jaques Siroux. An approach to co-operation in man machine oral dialogue. In *ERGO-IA*, 1988.

Jacques Moeschler. *Modélisation du dialogue, représentation de l'inférence argumentative*. Hermes, 1989.

Jeremy Peckham. Speech understanding and dialogue over the telephone: an overview of progress in the Sundial project. In *Proceedings of the 2nd European Conference on Speech Communication and Technology*, pages 1469–1472, 1991.