Knowledge Sources In Spoken Dialogue Systems

Nils Dahlbäck & Arne Jönsson

Department of Computer and Information Science
Linköping University, S-581 83, LINKÖPING, SWEDEN
nilda@ida.liu.se, arnjo@ida.liu.se

ABSTRACT

We describe how the architecture of the modularized LINLIN dialogue manager can be augmented to handle the cases where the user's initial information request is underspecified, and where therefore the system needs to ask the user about the missing pieces of information. In many existing spoken dialogue systems this has been handled by a so-called task model, which has been embedded in the dialogue move management. The present paper makes two points. First, that the term 'task' is ambiguous, since it can refer to the user's underlying reasons for engaging in the dialogue, to the user's information seeking activity, and to the system's actions when preparing for providing the user with an answer to the posed question. Second, we argue that this aspect of dialogue management should be managed by a separate module of the system, both because our previous work has shown that it is not necessary in all cases, and because it is conceptually different from general interaction management knowledge. For these reasons we advocate using a separate knowledge module called an Information Specification Form for managing these cases.

1. Introduction

Current work on dialogue systems for human computer interaction using spoken dialogues can, with some oversimplification, be seen as originating in two different traditions, rooted in the speech and the computational linguistics research communities. The motivations behind the early approach within the AI oriented computational linguistics community was to develop general computational models for all kinds of dialogues and discourses. A pivotal contribution here was the paper by Grosz and Sidner [9]. That work had a wide scope, and aimed for very general models applicable to all kinds of dialogue situations. The potential drawback on this for developers of particular systems has been the large computational overhead, as well as the risk of catering for aspects of dialogue not occurring in the particular domain the system is built for.

The speech community, on the other hand, has focused more on developing one-shot designs for particular systems. The focus has been more on catering for the particular phenomena occurring in the particular dialogue situation than on general applicability. This has lead to impressive performance in the particular situations chosen, (cf. [2, 1]). The draw-back is in this case instead the lack of generalizability and portability (there is an interesting similarity between this and the semantic grammar approach of the 70'ies [12, 16]).

We have, in our work, strived for finding a middle ground between these two approaches; for generalizability within specific classes of dialogue situations. The general approach has been to develop a general architecture for the class of dialogue situations called 'simple service systems' [10], where the ambition has been to keep the processing and knowledge modules separated. The first step was to develop a dialogue manager [13], which controls the interaction and models the entities under discussion. This was shown to be sufficient for the information seeking dialogues of a relational database system, the CARS system [13]. When looking at other seemingly very similar application domains, it became apparent that the system needed knowledge not only about the structure and process of the dialogue as such, but also of the non-linguistic domain of the dialogue. Examples of this were charter travel information and ordering of HiFi equipment [4]. In a paper at Eurospeech 97 [3] we described the amendment of the dialogue manager with two separate but closely related knowledge sources. First, the conceptual model, which contains general information of the conceptual relationships between objects in the domain, some of which are general and some of which are particular to the domain, and which sometimes even deviate from standard usage. Second, the domain model, which holds a description of the entities and their relationships in the particular domain. It is the first structure that knows that 'beach' and 'distance-to-beach' are important properties of hotels in the charter domain (but not for hotels in general), and it is the second that knows that Heraklion is a city on Crete, and that Crete is an island in the Greek archipelago.

In this paper we discuss an additional aspect of dialogue management necessary in a number of application domains for spoken dialogue systems, namely the so-called 'task-models' of many current systems. This is needed to cater for those cases where the request for information from a user is under-specified, and the system therefore needs to request additional information from the user before being able to provide an answer. Common examples of this are when a user e.g. wants to know when there are train connections between two cities, but does not specify the time of the day, or if there are any constraints on the type of trains. This can be carried out by the dialogue model, as is the case in many of today's best working existing systems, but that would not conform to our modularized approach. Instead we will here suggest a separate module for this aspect of dialogue management. For reasons that will become clear in the next section, we do not call it a task-model, but, an information specification form.

2. Task models

The notions of task and task model are used in more than one sense in work on dialogue systems. Grosz' [8] early work on so-called 'task oriented dialogues' was concerned with cases where two people work cooperatively on a task, where a 'task' is some real-world non-linguistic activity, that is directed towards achieving a particular goal, and that can be broken down into small steps, each having its own goal (ibid, p 12). These dialogues were distinguished from 'question-answering' dialogues, where the important difference is that in the latter kinds of dialogues the answerer cannot be viewed as sharing a goal in common with the questioner. Task-oriented dialogues have a structure that closely parallels the structure of the task being performed. Therefore the system needs some kind of representation of the task and its domain, in order to be able to interpret the utterances in the dialogue, as well as for being able to segment the dialogue into sub-dialogues.

In contradistinction to this kind of task model, the term 'task' is often used by developers of present-day spoken dialogue systems to describe the sequence of information that needs to be collected for by the information providing system to answer the users' initial question, e.g. departure time, arrival time, etc., for example [1]. There are two important differences between these two cases. In the first, the task model includes also the goals of the user, but not in the latter. Furthermore, in the former case this knowledge is a separate structure, whereas in the latter it is intertwined with other aspects of the dialogue model. This is a reflection of the fact that in the information providing dialogue case, there is less need for understanding the non-linguistic task for which the information is sought, than in the task-oriented dialogues that was Grosz' concern. Except for a few and rare cases, there is no need for the information provider to know why the caller wants to know the departure and arrival times for trains from Paris to Lyon. The answer will be the same, regardless of whether the caller is planning a trip or writing a university paper on travel times between major cities in France. This is an example of how the 'dialogue-task distance' [6] can affect the kinds of knowledge necessary to build into the dialogue system.

As we have seen, the term 'task' is ambiguous, since it can refer to the user's underlying reasons for engaging in the dialogue, to the user's information seeking activity, and to the system's actions when preparing for providing the user with an answer to the posed question. For these reasons we advocate using a separate knowledge module called an Information Specification Form (ISF) for managing these cases. An additional advantage of the name chosen is that is illustrates which kind of data structure is used , i.e. a slot-and-filler structure.

By treating this as a structure separate from the dialogue management, we allow for a more generic design of the system, and opens up for the possibilities of making customization to other domains a less laborious task.

3. Integrating dialogue knowledge

In this section we will exemplify how an information specification form, an ISF, can be incorporated in a dialogue system. We will illustrate this from our current development of a local bus timetable information system [15, 7]. Our approach is to extend an existing dialogue system the LINLIN-system [13], to handle also such information request specification sub-dialogues. We will do this by assuming a slot-and-filler structure with attributes reflecting the information needed to properly access the background system. This is hence similar to the so-called task models used in many spoken dialogue systems to model a set of information necessary to perform a task such as providing time-table information cf. [1].

LINLIN is a modularized dialogue system consisting of processing modules for Interpretation, Generation, Background system access and Dialogue management, figure 1. We will here assume that the interpretation module handles the actual interpretation, including the ability to interpret sentence fragments, multi-sentential, extragrammatical utterances and anaphora resolution. Similarly the generation module produce a suitable response to the user, and the knowledge coordinator reasons with the background system in order to find an answer to the user query [7]. The dialogue manager controls the interaction and can be viewed as a controller of the other modules.

For information retrieval applications where information on a variety of parameters must be specified in order to access the background system, such as departure and/or arrival time and day, dialogue grammar information on objects and properties [13], is not sufficient. We also need knowledge of what information that is provided and what information that is further required in order to access the background system.

From empirical investigations on local bus timetable information requests, we have identified a number of different user information needs [14]. The most common, called trip information, occurs when the user needs to know how and when on a particular day, most often the present day, one can travel from one point to another in town by bus. Another common information need, called route information, is when the caller wants information on which bus routes

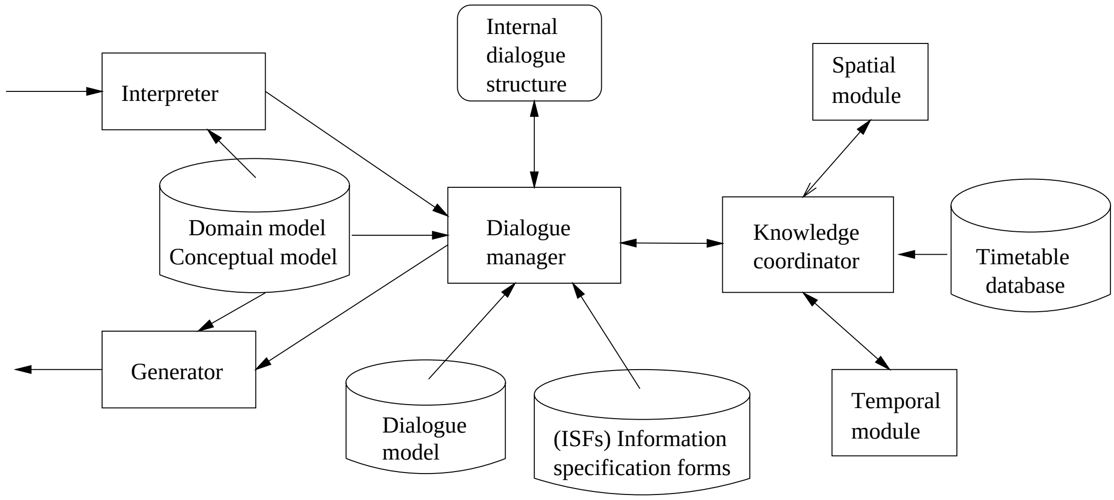

Figure 1: An overview of the linlin system

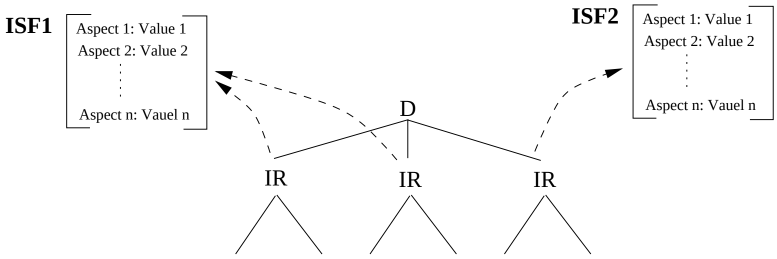

Figure 2: The dialogue tree and the information specification forms

that go from one point to another.

Since we need to handle (at least) two information needs, we cannot follow the usual approach of having the information specification task integrated in the dialogue management. Instead we separate out the ISFs from the general dialogue manager. Thus, the previous knowledge sources, i.e. conceptual model, domain model and dialogue model, is now enhanced with information specification forms, cf. figure 1. Furthermore, as users' can, and often will, provide any piece of information at more or less any point in the discourse, it is important to allow for such user behaviour, cf. [11] for another view on this.

The ISFs hold descriptions on the information needed for various user information needs that can be provided for by the system. When the dialogue manager resolved which ISF to pursue, based on information from the Interpreter and the current dialogue, an instance of that ISF is associated with the current node in the dialogue managers internal dialogue structure; the dialogue tree [13], cf. figure 2[1] to see what information is missing. This is used by the dialogue manager to generate for the user meaningful follow-up questions to underspecified information requests, simply by inspecting the ISF and asking for the additional information required to fulfill the task.

The information specification forms are only one of the knowledge sources utilized by the dialogue manager when controlling the interaction; the conceptual, domain and dialogue models are also utilized when needed [3]. However, adding this new knowledge source only requires an update of the dialogue grammar to also consider the information in the instances of the ISF's.

4. Conclusions

Dialogue systems need to consult information from various knowledge sources in order to control the interaction and interpret user requests appropriately. We consider it important to clearly state what information that is needed for various purposes and to divide this information into knowledge sources that could be useful in any dialogue

[1]The ISF is illustrated as a separate frame connected to the IR-node, but can just as well be part of the IR-node. It is separated here only to clarify its role in the process

system.

We have in our previous work on seemingly similar situations found that the kinds of knowledge required to participate in a dialogue may differ substantially. Let us mention two examples of this. Time-table information for local bus transport seems to require a much more sophisticated geographic/spatial knowledge and reasoning abilities than railway time table information. The reason for this is that in the former case the users of the system use a much wider range of referring expressions to describe departure and arrival locations [7]. A second example is that there was no need for an information request specification form in a system providing consumer report information on used cars, but such knowledge is required in the time-table information systems.

We draw two conclusions from this. First, that more work is required to establish a useful taxonomy of dialogue types, making it possible in advance to specify which kinds of knowledge is required to be implemented in a particular system under development. A first stab on this is done in [5], but more work is clearly required here. The second conclusion is that a modular approach such as the one advocated here has the advantage of making it possible to re-use the core of the system in porting it to another application domain, without having to incorporate aspects of dialogue management not required in the new situation. So the argument of the present paper is not that an ISF model such as the one described here should always be included. On the contrary, we are well aware of the fact that in many cases it is not needed, and that therefore in some situations it can and should be excluded.

While we have emphasized the conceptual difference between the dialogue management and the ISF models, it does not necessarily follow from this that they always have to be separated out in a particular implementation. However, system developers must know what implications different choices have for the system and especially what implications a reduced model will have, i.e. being computationally efficient vs general and conceptually more transparent.

Acknowledgments

This work is supported by The Swedish Transport & Communications Research Board (KFB) and the joint Research Program for Language Technology (HSFR/NUTEK).

5. REFERENCES

1. S. Bennacef, H. Bonneau-Maynard, J. L. Gauvin, L. Lamel, and W. Minker. A spoken language system for information retrieval. In *Proceedings of ICLSP'94*, 1994.

2. Mats Blomberg, Rolf Carlson, Kjell Elenius, Björn Granström, Joakim Gustafson, Sheri Hunnicutt, Roger Lindell, Lennart Neovius, and Lennart Nord. An experimental dialogue system: Waxholm. In *Speech Transmission Laboratory, Quartely Progress and Status Report*, number STL-QPSR 2-3/1993, pages 15–20. Department of Speech Communication and Music Acoustics, Royal Institute of Technology, Stockholm, Sweden, 1993.

3. Nils Dahlbäck and Arne Jönsson. Integrating domain specific focusing in dialogue models. In *Proceedings of Eurospeech'97, Rhodes, Greece*. European Speech Communication Association, 1997.

4. Nils Dahlbäck. *Representations of Discourse, Cognitive and Computational Aspects.* PhD thesis, Linköping University, 1991.

5. Nils Dahlbäck. Kinds of agents and types of dialogues. In J. A. Andernach, S. P. van de Burgt, and G. F. van der Hoeven, editors, *Proceedings of the Ninth Twente Workshop on Language Technology, TWLT 9, Enschede*, pages 1–11, 1995.

6. Nils Dahlbäck. Towards a dialogue taxonomy. In Elisabeth Maier, Marion Mast, and Susann LuperFoy, editors, *Dialogue Processing in Spoken Language Systems.* Springer Verlag Series LNAI-Lecture Notes in Artificial Intelligence, 1997.

7. Annika Flycht-Eriksson and Arne Jönsson. A spoken dialogue system utilizing spatial information. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.

8. Barbara J. Grosz. *The representation and use of focus in dialogue understanding.* PhD thesis, University of California, Berkeley, 1977.

9. Barbara J. Grosz and Candace L. Sidner. Attention, intention and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

10. Philip J. Hayes and D. Raj Reddy. Steps toward graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies*, 19:231–284, 1983.

11. Peter A. Heeman, Micahel Johnston, Justin Denney, and Edward Kaiser. Beyond structured dialogues: Factoring out grounding. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.

12. G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database System*, 3, 1978.

13. Arne Jönsson. A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering*, 3(2/3):103–122, 1997.

14. Pernilla Qvarfordt. Usability of multimodal timetables: Effects of different levels of domain knowledge on usability. Master's thesis, Linköping University, 1998.

15. Pernilla Qvarfordt and Arne Jönsson. Effects of using speech in timetable information systems for www. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.

16. E. D. Sacerdoti. Language access to distributed data with error recovery. In *IJCAI'77*, 1977.