INTEGRATING DOMAIN SPECIFIC FOCUSING IN DIALOGUE MODELS

Nils Dahlbäck and Arne Jönsson[1]
Department of Computer and Information Science
Linköping University, S-581 83, LINKÖPING, SWEDEN
nilda@ida.liu.se, arnjo@ida.liu.se

ABSTRACT

Natural language interaction requires dialogue models that allow for efficient and robust human computer interaction. Most systems today use some kind of speech-act based dialogue model. While successful in a number of applications, these models have known limitations, both from linguistic and computational points of view, which has led a number of workers to suggest using the dialogue participants goals/intentions to model the dialogue. In this paper we suggest that amending speech act based models with sophisticated domain knowledge makes it possible to extend their applicability. Two kinds of domain knowledge are identified, one is the Domain Model; a structure of the discourse 'world', and the other is the Conceptual Model which contains domain specific general information about the concepts and their relationships in the domain. These extensions have been utilized in the LINLIN dialogue manager and the paper presents results from customizing the dialogue manager to two different applications.

1 INTRODUCTION

Most current dialogue models for natural language interaction build a dialogue structure from some kind of speech-act based classification of the participants' moves. Examples of this are the SUNDIAL[3] and Verbmobil[2] dialogue systems, as well as our own LINLIN system and others. There are minor differences between these, e.g. in how detailed the classification of the moves is, and to what extent the model is geared towards a particular kind of dialogue (e.g. Verbmobil for negotiation dialogues) or if a more general and domain independent set of moves is used (e.g. the LINLIN model).

Both theoretical analysis and empirical observations suggest that these models can be used in more than one dialogue situation of the same kind, but it is less clear whether they also can be used in other kinds of task domains (for a discussion of related issues, see [5]). It is therefore quite likely that sooner or later revisions or amendments will be needed. The limitation of these kinds of dialogue models has also been known outside the speech and natural language communities (c.f. [9]).

A number of researchers have therefore argued for models based on the intentions underlying the speakers' utterances. It is claimed that these are needed for ellipsis resolution, establishing anaphora resolution, and for providing helpful answers or corrections when there is no direct answer to be found for the speaker's question, e.g. because of faulty presuppositions. We do not deny the general claim that there are cases where this kind of information might be needed for well-functioning systems (though for at least some cases of ellipsis it has been shown that computationally simpler models might do the job as well). However, these models, based on the recognition of the speaker's underlying plans, have also some disadvantages. First, there are no computationally tractable plan recognizes, and, second, the task of identifying the speaker's underlying plan is sometimes very difficult, given the meager data that are supplied in the utterance as such.

The question is, then, whether there instead are possible amendments to the speech-act based models that are simpler from a system development point of view? The argument in the present paper is that using knowledge of the domain is in some cases a viable alternative. Using the terminology of the model developed by Grosz & Sidner [6], we suggest using the attentional state but not the intentional state. In their theory the attentional state is used for representing the entities discussed during the course of interaction, i.e. the focus of attention. Furthermore, the attentional state is part of the discourse segmentation process used for global focusing, i.e. focusing between segments, not local focusing, i.e. focusing within a segment.

2 DIALOGUE MANAGEMENT

The context used in this paper is a general dialogue system shell that can be customized to different applications, though the principled distinctions we make also apply to one-shot designs. To make the discussion more concrete, we will use the LINLIN natural language interface as an example context [7, 8].

LINLIN is controlled by a dialogue manager, which can be viewed as a controller of resources for interpretation, background system access and generation. The Dialogue Manager receives input from the interpretation modules, inspects the result and accesses the background system with information conveyed in the user input. Eventually an answer is returned from the background system access module and the Dialogue Manager then calls the generation modules to generate an answer to the user. If clarification is needed from any of the resources, the Dialogue Manager deals with it.

The Dialogue Manager controls the interaction by means of a dialogue grammar. It also holds the information needed by the modules in the interface, including the Dialogue Manager itself. One important feature of the Dialogue Manager is that the information is

[1] Authors are in alphabetical order.

modeled in dialogue objects, which represent the constituents of the dialogue. During the course of interaction a dialogue tree is built up from instances of dialogue objects. The dialogue tree serves two purposes. First, it serves as a vehicle for monitoring the dialogue, to guide decisions on how to proceed in the dialogue and where a user move fits into the dialogue; if it is to be regarded as a new initiative, a clarification request, or a response to a system initiative. The dialogue tree also records the focus parameters to be used by the referent resolving algorithms of the interpretation and generation modules.

The dialogue object descriptions are domain dependent and can be modified for each new application. The Dialogue Manager is customized by specifying the dialogue objects; which parameters to use and what values they can take. The parameters specified in the dialogue objects reflect the information needed by the various processes accessing information stored in the dialogue tree.

A dialogue object consists of one set of parameters for specifying the initiator, responder, context etc. needed in most applications. Another set of parameters specifies content. Two of these, termed Objects and Properties, account for the information structure of a move (query), where Objects identify a set of primary referents, and Properties identify a complex predicate ascribed to this set. This is based on the observation that a common user initiative is a request for domain concept information from the database; users specify a database object, or a set of objects, and ask for the value of a property of that object or set of objects (cf. [1]). The Objects and Properties parameters are focal in the sense that they can be in focus over a sequence of segments.

Important for a dialogue manager is not only focus tracking, but also dialogue control, i.e. deciding which action to carry out in a certain situation. We have found that the parameters modeling the focus can also be used to direct the dialogue[see 7, 8 for details].

Our claim in this paper is that in order to correctly identify the correct Objects and Properties parameters, the system's knowledge will in many cases need to be amended with domain specific knowledge.

3 DOMAIN KNOWLEDGE

There are two closely related but theoretically separate knowledge structures that are potentially needed for the successful maintenance of the focus structure. One is the domain model (DM), the other the Conceptual Model (CM). This distinction resembles the well-known between episodic and semantic memory[10].

To illustrate the need for two different structures, consider the dialogue below, where the user (U) is trying to book a two-week charter holiday to the Greek archipelago using a computerized travel agency (S).[2]

U: How far is it to the beach from the Divani Hotel?
S: 1.1 km
U: Is there a hotel closer to the beach?
S: Royal Hotel is 150 m.

The expression 'the beach' is used twice in this segment. But it does not refer to the same beach. In fact, it is not obvious that it even refers to beaches on the same island in the Greek archipelago. A dialogue model that tries to resolve co-reference relations based simply on the co-occurrence of similar or identical expressions would not work here. What seems to be needed is some kind of knowledge of the geography of Greece (DM), but also general knowledge of the domain of charter holidays, where 'beach' is an important attribute of a hotel (CM).

3.1 The Domain Model

The domain model is a static structure of the 'world' the discourse is about. Imagine e.g. negotiating about the location where to meet for cease-fire negotiations. It seems an impossible task without a rather detailed knowledge of the geographical domain. Or imagine trying to decide where to go and which hotel to select on your next holiday to Greece. Also in this case it seems likely that a detailed domain model is needed to keep track of the intended referents of definite NPs and anaphoric expressions. Thus, it is in the Domain Model that we store knowledge of the Greek archipelago, e.g. the fact that Heraklion is a town on Crete etc.

The kind of domain knowledge necessary to maintain a focus structure will probably not vary only in its specific content, but also in the kind of knowledge representation used. In the travel case, we have found a hierarchical tree structure useful. But in other, seemingly similar applications, other kinds of knowledge representations are needed. One example of this is the HCRC map task domain.[3] In this scenario two interlocutors are to find and mark a route on a map. For this task two maps with different landmarks are presented to each participant and the goal is to negotiate the two maps to mark the route. In order to do this, it is necessary to identify common landmarks on the maps. For instance:

12 Chris: Do I cross the bridge?
13 Neil: I don't have a bridge. I would cross a bridge.
14 Chris: Okay.
15 Neil: Right. Do you have a stream? Forked stream?
16 Chris: yes
17 Neil: Right. Follow the stream... Follow the path of the stream right down.

In these dialogues the domain model acts as the common ground and references to various objects reflect the results of an action carried out in a particular situation. For instance, once the *stream* has been identified it is used as the focused object and references are grounded from this object. But, in contrast to the Greek holiday example, in this case we seem to need to represent not only the objects and entities in the 'world', but also its Euclidean 2D space, as well as the fact that it is a map,

[2] This dialogue is from a coprus of Wizard of Oz-dialogues collected in Linköping available at http://www.ida.liu.se/~nlplab/dialogues/corpora.html.

[3] The HCRC map task dialogues are from http://www.cogsci.ed.ac.uk/hcrc/wgs/dialogue/public_maptask.

and hence has corners and sides as well.

Our observations suggest that knowledge of the conceptual domain is needed not only for geographical information but also in other cases. As an illustration we provide the example below, from a dialogue where the user (U) is ordering a hi-fi set from a computerized sales person (S).

S: Searching ... Record players Oracle Delphi Mark II, Cassette decks Nakamichi Dragon, Turners Revox B 261, Loudspeakers Audio-Tronic Megatrend
U: Price of this combination

In this case it seems necessary for the system to have knowledge of the domain of hi-fi equipment, including that a combination comprises a set of elements e.g. record player, cassette deck, CD-player etc. (For a discussion on these issues, see [4].)

3.2 The Conceptual Model

The Conceptual Model contains general information about the relationships between the objects in the particular domain. It is here we store the fact that in the travel domain, ‘shower’ is a concept closely related to ‘room’, whereas ‘ceiling’ is not.

This kind of knowledge is required also in other cases studied by us. In [4] we present an analysis of five different dialogue domains, and in all cases we found the required concepts and relations to be of a rather small order. Furthermore, in many cases we only need to amend the generic world knowledge with some non-standard concepts and relations. For instance, in a library system (PUB) containing information of all the books, and reports bought at our department and stored in our offices, the relevant knowledge seems to boil down to the following. ‘Book’ ISA ‘publication’, ‘report’ ISA ‘publication’. Furthermore ‘title’, ‘author’, ‘publisher’, ‘topic area’, ‘year-of-publication’, ‘place-of-publication’, and ‘owner’ are all ASPECTS-OF ‘publication’. Note that there are three layers of generality of these conceptual relations. First, there are those belonging to general world knowledge of books, e.g. that they are written by authors and have titles. Second there are those belonging to general library knowledge of books, e.g. that they have a year of publication and a place of publication. And third, there are those relations that are specific to this particular distributed library, namely that each publication has an owner. (One could of course argue that publication ISA something like ‘human-made-artifact’ and that all of these have an owner. But when developing a system tailored for library services, the introduction of such conceptual spaces seems to create unnecessary complications in both structure and in search processes.)

An important question is of course to which extent the knowledge necessary for a particular system is a subset of the general world knowledge. From an engineering point of view, one can take a pragmatic view, and simply check whether the general knowledge modules available before customizing the shell contain the required knowledge. In our empirical work we have instead defined standard knowledge as that found in ordinary dictionaries. In the studies mentioned above we used Webster’s dictionary (1987 edition). ‘Hotel’ is there defined as ”a large and superior kind of inn”. ‘Inn’ is then defined as ”a house that provides lodging accommodation for travelers; a hotel, restaurant or tavern”. But there is no mention of ‘beech’ here. So we can use the general world knowledge to understand that in the context of discussing a hotel, if someone mentions ”the travelers”, we can assume that this defNP refers to travelers at this hotel. But we cannot use this kind of general knowledge to understand the relation between the beech and the hotel discussed above. And likewise, general world knowledge can be used to understand the relation between the concepts ‘room’ and ‘window’, but not between ‘room’ and ‘shower’ when a customer discussing a potential accommodation alternative says ”I need a shower”.

One could perhaps call this kind of knowledge cases of domain specific generic information. Note that the knowledge in the Conceptual Model is more generic than the Domain Model. The conceptual relationships mentioned for the travel domain above do not only apply when talking about trips to Greece, but to any dialogue about charter holidays to these kinds of resorts. As illustrated by these examples, in some cases this knowledge is just an addition to general world knowledge, but in other cases the relations can deviate more or less from this general world.

4 APPLYING THE FRAMEWORK

Taken together, examples such as these suggest to us that a domain specific conceptual representation working together with a dialogue act based dialogue manager will make it possible to extend current technology to a larger set of dialogue situations, and to managing a larger set of linguistic phenomena, than has hitherto been the case. Our results also suggest that for a number of different applications, these domain models will be of limited size and need only a limited set of domain specific conceptual relations.

Customization of the Dialogue Manager involves defining the focal parameters of the dialogue objects in more detail. Consider the travel application discussed above. This application utilizes a three-layered domain model: the Greek archipelago, the various resorts and finally the hotels at each resort. This structure is needed, for instance, by the modules responsible for instantiating an indexical utterance. However, for focus tracking it turns out that there is no need to explicitly represent the various levels in the hierarchy. Instead one single sub-parameter holding any of these object types is sufficient. To illustrate this, consider the following dialogue:

U27: what hotels are there on Crete
S28: Wait...
The hotels in Platanias on Crete are:
Kronos, Agrimia, Village Suites and Villa Margarita.
U29: which one is the cheapest
S30: Wait...
Price Villa Margarita:
<price list>

After utterance U27 the value of the Objects parameter is the resort *Crete*. This will be changed to a set of hotels when the response from the background system is generated, S28. It is easy to traverse the hierarchical domain model upwards from a hotel to a resort, but not down from a resort to a previously specified hotel. For such utterances the dialogue tree need to be consulted. One could argue that users frequently jump between discussing hotels and resorts that would motivate parameters for both resort and hotel, instead of searching the dialogue tree. However, the structure of the travel domain is probably also in a sense present in the users' conceptual model of the domain, i.e. users view a resort as an object having multiple hotels. Furthermore, the hotels are presented as a list of hotels available at the resort; a hotel is always selected from a resort. Consequently, one can assume that users are aware of the ambiguity and seldom use anaphoric expressions for referring back to a hotel when they are discussing a particular resort. This is also corroborated in our empirical investigations. Thus, no extra parameters for resort and hotel are specified, instead the dialogue tree is searched when a user anaphorically refers to a hotel when the resort is in focus.

This contrasts to an application utilizing a relational database. We have customized the Dialogue Manager for such an application in the domain of consumer information on used cars. In this scenario users request information stored in a database on properties of second hand cars. The customized dialogue objects record specifications of cars as Objects and related domain concepts as Properties. However, for this application the Conceptual Model (CM) is essential. Users utilize various expressions for describing properties, for instance the concept *speed* can be used to express either top speed or acceleration. But there is no need to consult a domain model; i.e. users do not reason about how objects relate to each other. Objects discussed during the course of interaction are referred to, but they are accessed from the dialogue objects as recorded in the dialogue tree.

These two cases illustrate one important point, namely that while both kinds of domain knowledge discussed here are required in principle, there are possible cases where it will be necessary to invoke only one of them. Which of these will be required in a particular case need to be decided in the customization of the shell (or in the design of the system). In our view, this decision cannot be made without empirical studies of the sub-language used in the particular task and situation under consideration. As illustrated in the comparison of the MapTask and the Holiday dialogues, superficial similarities do not guarantee the usefulness of similar design solutions.

5 CONCLUSIONS

We argue that current speech act based dialogue models can achieve an increased flexibility and be useful for a larger set of domains if they are amended with domain/task specific knowledge of the static properties of the discourse domain, and with knowledge of the specific concepts and conceptual relations in the system's domain/task space. It is our claim that this has a number of potential advantages over trying to model the users' underlying intentions or plans, at least for the applications and systems developed or studied by us. The major advantage for pure speech (or typed dialogue) systems seems to be that these solutions are computationally simpler. But when moving from pure linguistic interaction to multi-modal systems, the need for developing conceptual models for the pictorial/graphical displays often means that the kind of work on domain modeling proposed in this paper, will be required also for other reasons, and hence in a sense become less costly in terms of invested resources.

REFERENCES

[1] L. Ahrenberg, Interrogative Structures of Swedish. Aspects of the Relation between Grammar and Speech Acts, Ph.D. thesis, Uppsala University, 1987.

[2] J. Alexandersson and N. Reithinger, Designing the dialogue component in a speech translation system. Proceedings of the Ninth Twente Workshop on Language Technology (TWLT-9), pp: 35--43, 1995

[3] E. Bilange, A task independent oral dialogue Proceedings *of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin*, 1991.

[4] N. Dahlbäck, Representations of Discourse, Cognitive and Computational Aspects, Ph.D. thesis, Linköping University, 1991.

[5] N. Dahlbäck, Towards a dialogue taxonomy. In E. Maier, M. Mast, and Susann LuperFoy (eds.) *Dialogue Processing in Spoken Language Systems*. Springer Verlag, Series LNAI-Lecture Notes in Artificial Intelligence, 1977.

[6] B. Grosz and C. Sidner, Attention, Intention and the Structure of Discourse, *Computational Linguistics*, 12(3), pp: 175-204, 1986.

[7] A. Jönsson, *Dialogue Management for Natural Language Interfaces - An Empirical Approach*, Ph.D. thesis, Linköping University, 1993

[8] A. Jönsson, Dialogue Actions for Natural Language Interfaces, *Proceedings of IJCAI'95, Montreal, Canada,* 1995.

[9] S. C. Levinson, Some Pre-Observations on the Modelling of Dialogue, *Discourse Processes*, Vol: 4, pp: 93-116, 1981.

[10] E. Tulving, Episodic and Semantic Memory, In E. Tulving and W. Donaldson (eds) *Organization of memory,* New York: Academic Press, 1972.