AN ARCHITECTURE FOR MULTI-MODAL NATURAL DIALOGUE SYSTEMS

Nils Dahlbäck, Annika Flycht-Eriksson, Arne Jönsson, Pernilla Qvarfordt[*]

Department of Computer and Information Science, Linköping University, Sweden
{nilda, annfl, arnjo, perqv}@ida.liu.se

ABSTRACT

In this paper we present an architecture for multi-modal dialogue systems. It is illustrated from our development of a multi-modal information system for local bus timetable information. The system is based on a natural language interface for typed interaction that is enhanced to handle also multi-modal interaction. The multi-modal user interface was designed based on empirical investigations and some results from these investigations are presented. We also show how information specification forms can be utilised to handle requests typical for timetable information systems and how spatial and temporal information is integrated and used in the system.

1. INTRODUCTION

Today's computer and communication technology creates the opportunity for easy access to many information sources, and the opportunity to support complex information retrieval tasks. For the emerging technology to fulfil its promises it is not enough for the information to be available, it also needs to be easy accessible. This implies that a system must allow the users to formulate their information needs in a naturally intuitive manner. We believe that human dialogues provide the best model candidate for such systems. It is however important to stress that this does not imply that systems should mimic human dialogues in minute detail. Instead the basic principles of such dialogues should be supported; e.g. connectedness between consecutive communicative acts, free choice of expression on semantic and syntactic level, frequent use of abbreviated expressions that rely on the verbal and non-verbal context for their interpretation [6].

Viewed from this perspective, not all multi-modal information systems are dialogue systems, in the strict sense, not even if there is a verbal (spoken or written) input or output channel. For a multi-modal information system to be a dialogue system, all modalities must be integrated and be possible to use for communicative actions as part of an on-going dialogue [5]. Only then can the user freely decide how to interact with the system, e.g. whether to answer a spoken question by pointing in a map or by a verbal answer. From a system architecture point of view, this means that all modalities, i.e. dialogue acts performed utilising any modality, need to be handled.

There are a number of research issues involved in the development of a multi-modal dialogue system. The present paper focus on some of these; the design of an interface where non-verbal commands like pointing or showing a map can be naturally integrated, dialogue management, and the use of domain knowledge in the on-going dialogue. In what follows we will illustrate our architecture for multi-modal dialogue systems from a current prototype development of a system for timetable information for local bus traffic.

2. DESIGN OF THE MULTI-MODAL INTERFACE

As a base for the design of the user interface several investigations were made. Thirty-nine conversations between travellers and timetable informants about timetable requests were recorded in a telephone setting and analysed in order to reveal what kind of information was exchanged in the dialogues. An investigation of usage of paper-based timetables was also conducted in order to get an insight in how tables and maps were used by the travellers.

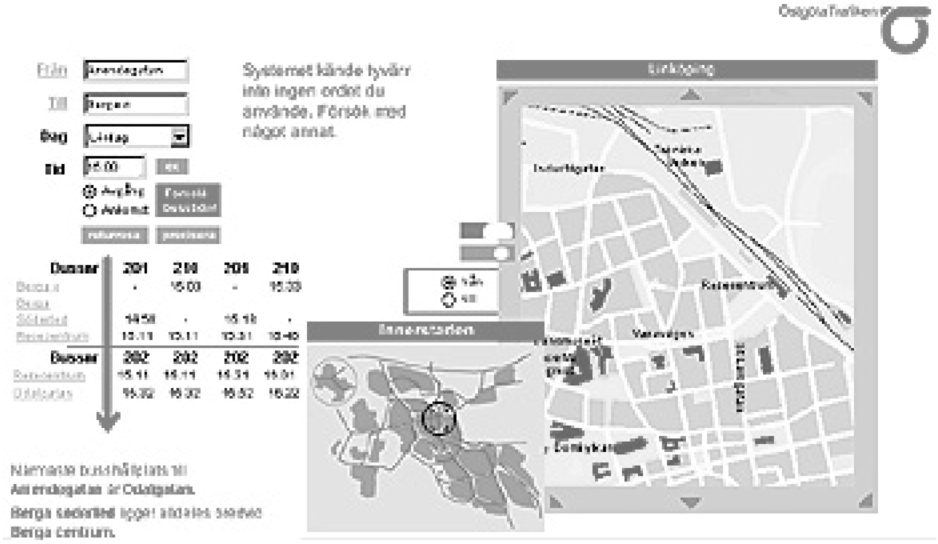
The results of our empirical investigations are reflected in the design of the MALINQF user interface. For instance, travellers often asked for more alternatives than given by the traffic informant. Previous studies on text based user interfaces has also shown that providing more information than required, can give a more effective interaction [1]. This implies that a system should overrule the Gricean maxim "Do not make your contribution more informative than required" which is an often quoted guideline for interactive speech systems (e.g. [4]).

The MALINQF user interface is depicted in figure 1. It has four different parts, all visible at the same time, a fill-in form for expressing questions to the database, a map that can be used for entering points of arrival/departure in timetable questions, an area for showing the results of database queries, e.g. timetables, and finally an area for messages from the system. The map consists of an overview map and a map showing magnified parts of the overview map. The magnified map has two fixed magnification factors, showing different amounts of detail.

It is possible to interact with the prototype by keyboard, mouse, and/or speech. For example, the user can enter a point of departure by typing it, selecting it with the mouse in the map, or saying it. The prototype is fully functional except for the speech-recognition part, which is simulated by a wizard. The reason for using a wizard instead of a speech recogniser was that the main interest in the study was to investigate users interaction with the system, rather than the performance of the system. However, in the future we will integrate a speech recognition component with the user interface.

One issue that we were especially interested in investigating was how speech could support users with different amount of background knowledge. One such type of background knowledge that we believe influences the interaction is the variation in knowledge of the domain of the

[*] Authors are in alphabetical order.



Systemet kände tyvärr inte igen ordet du använde. Försök med något annat.

Bussar	201	210	201	210
[illegible]	-	15.03	-	15.33
[illegible]				
[illegible]	14.58	-	15.18	-
[illegible]	15.11	15.11	15.31	15.40
Bussar	202	202	202	202
[illegible]	15.11	15.11	15.31	15.01
[illegible]	15.32	15.32	15.52	15.22

Närmaste busshållplats till [illegible] är [illegible].
Berga [illegible] ligger [illegible]
Berga centrum.

Figure 1: The MALINQF prototype. The different parts of the user interface is depicted; at the top left the fill-in form, at the bottom left the timetable, at the center top a list of bus stops in an district or an error message from the system, and at the right the overview map and the detailed map. In the overview map the sight is visible, indicating what part of the city is visible in the detailed map.

application, i.e. travelling by bus in a city. In our application the users are all travellers, the main differences are travelling frequency and knowledge of the city where the travel takes place. Users have their own requirements on the interaction and different combinations of interaction modalities addresses different information needs. If the user, for instance, does not know the name of the actual bus stop but only knows that it is in a certain area or near some other place, filling in a form is not of much help. In these cases a map might be more useful. A map on the other hand requires that the user knows the geographic location of a bus stop. This is not always the case, especially if the user is not familiar with the town. In such cases it might be better to enter the name using for example speech input.

To investigate these assumptions we conducted an experiment where we compared traditional interaction, i.e. keyboard and mouse, with multi-modal interaction, i.e. allowing also speech interaction, (cf. [13]). A total of 12 subjects participated in the study. The subjects were divided into three groups, corresponding to their knowledge on local buses in Linköping. Before and after the main study the subjects were asked to answer a questionnaire. The pre-questionnaire recorded the subjects background, and the post-questionnaire emphasised the subjects attitudes towards the system.

In the study, each subject used the user interface in two conditions; one with multi-modal interaction and one with traditional interaction. The subjects were randomly assigned to start with one condition, and then switch to the other. The subjects were first given a short introduction to the prototype and then had to solve three different scenarios in each condition.

The investigation showed that multi-modal interaction was to some extent more efficient than traditional interaction. The users made fewer errors, completed the task in fewer steps, and found their way in the map easier with multi-modal interaction. However, the task completion time between the two types of interaction did not differ. The investigation also showed that users with weak domain knowledge were better supported by multi-modal interaction, than by traditional interaction, and vice versa for users with good domain knowledge. Since multi-modal user interfaces provide users with several means of interaction, the users can choose the means that are the most efficient for them and for their purpose.

The investigations also provided implications for further refinement of the multi-modal user interface. For example, we could see indications that the users needed support through a more elaborated dialogue with the system. We also claim that efficiency is not the only important feature for multi-modal interaction. The users' subjective experience of the dialogue's co-operativeness and reliability must also be taken into account in the design of the system.

3. THE MALIN SYSTEM

The MALIN system (Multi-modal Application of LINlin) is a prototype system under development, which consists of processing modules for interpretation, generation, dialogue management and knowledge co-ordination, as shown in figure 2. These in turn consult various knowledge sources such as the timetable database, a domain model, dialogue models, lexicon, grammar etc.

The MALIN system is based on the LINLIN system [11]. The kernel of the LINLIN system is a dialogue manager developed for natural language interaction. Its design is based on data from Wizard of Oz investigations on written interaction for information seeking domains, which lead us to the following conclusions: Interaction is quite simple and can be structured in terms of IR-segments, sequences of utterances starting with an initiative and ending with a response. Questions are also fairly simple and the majority can be analysed as asking about properties of given objects, or conversely, asking for objects satisfying a list of properties. Adjacent segments cohere in simple ways, although complexity may vary from one domain to another. The dialogue manager controls the interaction by means of a dialogue grammar, and holds information needed by

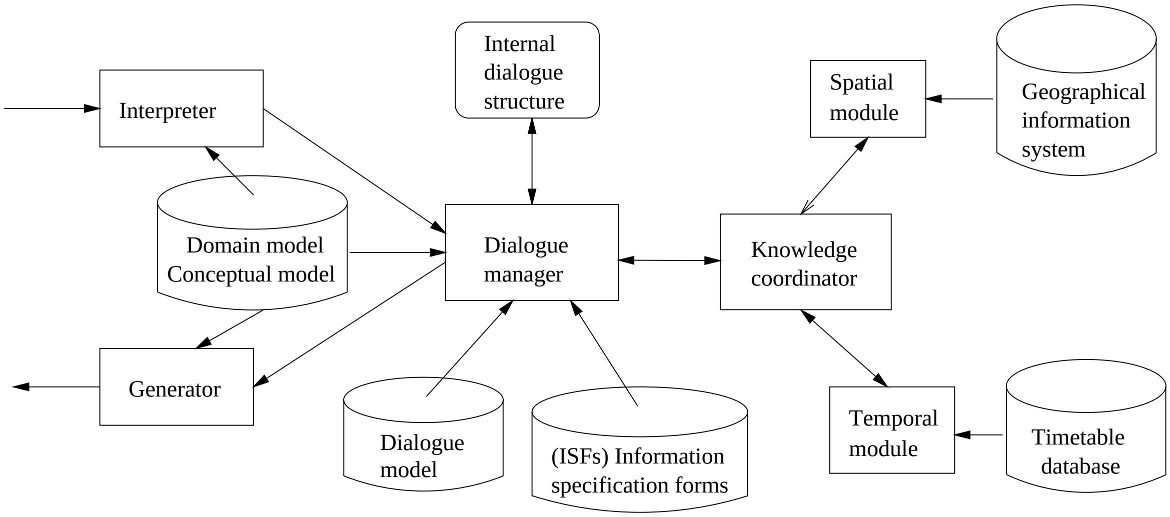

Interpreter
Internal dialogue structure
Spatial module
Geographical information system
Domain model Conceptual model
Dialogue manager
Knowledge coordinator
Generator
Dialogue model
(ISFs) Information specification forms
Temporal module
Timetable database

Figure 2: An overview of the system. The picture shows the different processing modules, interpreter, dialogue manager, knowledge co-ordinator, and generator. The various knowledge sources; conceptual and domain models, dialogue model, information specification forms, and timetable database are also depicted, but not the grammar and lexicon.

the modules in the system, including the dialogue manager itself, in a dialogue tree.

3.1. Information Specification Forms

The principles for dialogue management used in the LINLIN system worked well for information retrieval applications where a user initiative normally specifies object(s) and/or properties in enough detail for background system access. This is not often the case for timetable information requests. In order to handle timetable information requests correctly, a variety of parameters, such as departure and/or arrival time and day, must be specified before the background system can be accessed.

Furthermore, from the empirical investigations on local bus timetable information requests, we have identified a number of different user information needs [12]. 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 user wants information on which bus routes that go from one point to another.

Since we need to handle various information needs, we cannot follow the usual approach of having the information specification task integrated in the dialogue management. Instead the LINLIN model is extended with Information Specification Forms, ISF, to model the information pieces needed to access the background system for various user tasks. The ISF assumes 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[1] used in many spoken dialogue systems to model a set of information pieces necessary to perform a task such as providing timetable information (cf. [2]). 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. [10] for another view on this.

Based on information from the Interpreter and the current dialogue, as modelled in the dialogue tree, an instance of an ISF, corresponding to the user task, is associated with the current node in the dialogue tree. The ISF is used to see what information is missing and the dialogue manager generates meaningful follow-up questions to underspecified information requests. These sub-dialogues are generated by inspecting the ISF and asking for the additional information required to fulfil the task. This is controlled by the dialogue grammar which is enhanced to also consider the information in the ISF.

The ISFs are only one of the knowledge sources utilised by the dialogue manager when controlling the interaction; conceptual, domain and dialogue models are also consulted when needed [7]. The domain model is a structure of the world the dialogue is about while the conceptual model contains general information about the concepts and their relationships in the particular domain.

3.2. Representation and use of domain knowledge

Requests for information are passed by the dialogue manager to the knowledge co-ordinator when an ISF for a trip or route is fully specified. The task of the knowledge coordinator is to decide what knowledge sources to consult, integrate the information received from these, and return it to the dialogue manager.

One important, but easily over-looked, difference between the domain of local bus traffic and the commonly worked on rail information systems [14, 2] is that users' natural way of expressing departure and arrival locations rarely makes use of the official names of the bus stops. These locations are instead described using street or area names, locative expressions like *close to the library*, or by pointing and clicking in a map. For this reason a representation of and reasoning about the geographical domain becomes a necessity.

[1] The notion of task is confusing as the term is used to describe different tasks such as user tasks and system tasks. We therefore use the term information specification form (ISF). This is further discussed in [8].

Rail-traffic information systems often have an explicit model of the temporal domain (cf. [3, 14]), but the knowledge of the geographical domain is implicit and lies in the lexicon/grammar. In a multi-modal dialogue system for local bus traffic information a more elaborated domain model, which supports both spatial and temporal reasoning is needed. The input to the system, e.g. temporal and spatial descriptions, can be ambiguous and vague. Such vague qualitative descriptions and concepts must be mapped to quantitative and exact information, which is needed when searching the timetable database. This is accomplished by utilising domain knowledge. Our approach is to represent the domain knowledge in two modules, the spatial reasoning module and the temporal reasoning module, see figure 2. The spatial module utilises a geographical information system and is further described in [9].

Apart from mapping vague information onto precise descriptions the domain model is also utilised to provide the dialogue manager with information about clarification requests, such as inconsistencies in the input or missing information, when needed. This is illustrated in the following example.

> U: I would like to know how I can travel from the hospital down to IKEA in Linköping.

The utterance is recognised as a request for route information with the phrase "the hospital" as a departure location and "IKEA in Linköping" as an arrival location. The dialogue manager fills the slots in an ISF and then passes the request to the knowledge co-ordinator. The knowledge co-ordinator consults the spatial module, which tries to map the locations to two sets of bus stops. When the spatial reasoner discovers that "the hospital" is an ambiguous reference to a location, it tries to disambiguate the information. Since no more spatial information about the departure location is given, a clarification is needed. The knowledge co-ordinator passes this information to the dialogue manager which poses a question to the user.

> [The system shows a list of the hospitals]
> S: There are many hospitals. Where are you?
> U: Here. [points at the university hospital in the list]

The new information is integrated with the old by the dialogue manager, which extends the ISF with the new information on departure location. A new request is sent to the knowledge co-ordinator that once more consults the spatial module. This time the spatial module succeeds when it tries to disambiguate the location of the place "the hospital". The place referred to by the user is mapped onto the bus stops near the place. "IKEA" is not ambiguous and is therefore mapped to the nearby bus stops. The two sets of bus stops are returned to the knowledge co-ordinator which turn to the temporal module for a timetable database access. The resulting route information is then returned by the knowledge co-ordinator to the dialogue manager which consults the Generator to present the timetable to the user.

4. CONCLUSION

A multi-modal *dialogue* system is something different from a system that uses more than one input and/or output modality. It also requires careful examination of the communicative acts and the different modalities to form a coherent dialogue, where the interpretation of any such act in any modality is based on the previous dialogue acts in the different modalities. In this paper we described the basic architecture of a multi-modal dialogue system that satisfies these requirements. We have also described in some detail two aspects of this system, the design of the interface and the use of domain knowledge sources for spatial and temporal reasoning. For both areas empirical investigations have been utilised to reveal the necessary requirements, and their impact on the design and implementation of the system has been discussed.

5. REFERENCES

1. Lars Ahrenberg, Nils Dahlbäck, Arne Jönsson, and Åke Thurée. Customizing interaction for natural language interfaces. *Linköping Electronic articles in Computer and Information Science*, 1(1), October, 1 1996. http://www.ep.liu.se/ea/cis/1996/001/.
2. 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.
3. S. Bennacef, L. Devillers, S. Rosset, and L. Lamel. Dialog in the RAILTEL telephone-based system. In *Proceedings of International Conference on Spoken Language Processing, ICSLP'96*, volume 1, pages 550–553, Philadelphia, USA, October 1996.
4. Nils Ole Bernsen, Hans Dybkjær, and Laila Dybkjær. *Designing Interactive Speech Systems: From First Ideas to User Testing.* London, Springer, 1998.
5. Nils Dahlbäck. Some suggestions for expanding the conceptual framework of hc. In *Paper presented at the Basic Research in HCI Symposium at CHI'99, Pittsburgh, PA*, 1999.
6. Nils Dahlbäck and Arne Jönsson. Empirical studies of discourse representations for natural language interfaces. In *Proceedings from the Fourth Conference of the European Chapter of the association for Computational Linguistics, Manchester*, 1989.
7. Nils Dahlbäck and Arne Jönsson. Integrating domain specific focusing in dialogue models. In *Proceedings of Eurospeech'97, Rhodes, Greece.* European Speech Communication Association, 1997.
8. Nils Dahlbäck and Arne Jönsson. Knowledge sources in spoken dialogue systems. In *Proceedings of Eurospeech'99, Budapest, Hungary*, 1999.
9. Annika Flycht-Eriksson and Arne Jönsson. A spoken dialogue system utilizing spatial information. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.
10. Peter A. Heeman, Micahel Johnston, Justin Denney, and Edward Kaiser. Beyond structured dialogues: Factoring out grounding. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.
11. Arne Jönsson. A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering*, 3(2/3):103–122, 1997.
12. Pernilla Qvarfordt. Usability of multimodal timetables: Effects of different levels of domain knowledge on usability. Master's thesis, Linköping University, 1998.
13. Pernilla Qvarfordt and Arne Jönsson. Effects of using speech in timetable information systems for www. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.
14. Albert Russel, Ingmar Herberts, Els den Os, and Louis Boves. Dialog management issues in the localization of a train time table information system. In *Proceedings of ISSD'96, Philadelphia*, pages 81–84, 1996.