A Method for Development of Dialogue Managers for Natural Language Interfaces

Arne Jnsson*
Department of Computer and Information Science
Linkping University
S- 58183 Linkping, Sweden
arj@ida.liu.se

Abstract

This paper describes a method for the development of dialogue managers for natural language interfaces. A dialogue manager is presented designed on the basis of both a theoretical investigation of models for dialogue management and an analysis of empirical material. It is argued that for natural language interfaces many of the human interaction phenomena accounted for in, for instance, plan-based models of dialogue do not occur. Instead, for many applications, dialogue in natural language interfaces can be managed from information on the functional role of an utterance as conveyed in the linguistic structure. This is modelled in a dialogue grammar which controls the interaction. Focus structure is handled using dialogue objects recorded in a dialogue tree which can be accessed through a scoreboard by the various modules for interpretation, generation and background system access.

A sublanguage approach is proposed. For each new application the Dialogue Manager is customized to meet the needs of the application. This requires empirical data which are collected through Wizard of Oz simulations. The corpus is used when updating the different knowledge sources involved in the natural language interface. In this paper the customization of the Dialogue Manager for database information retrieval applications is also described.

Introduction

Research on computational models of discourse can be motivated from two different standpoints. One is to develop general models and theories of discourse for all kinds of agents and situations. The other approach is to account for a computational model of discourse for a specific application, say a natural language interface (Dahlbäck and Jönsson, 1992). It is not obvious that the two approaches should present similar computational theories for discourse. Instead the different motivations should be considered when presenting theories of dialogue management for natural language interfaces. Many models for dialogue in natural language interfaces are not only models for dialogue in such interfaces but they also account for general discourse. The focus in this work is on dialogue management for natural language interfaces and not general discourse. Thus, the focus is on efficiency and habitability, i.e. a dialogue manager must correctly and efficiently handle those phenomena that actually occur in typed human-computer interaction so that the user does not feel constrained or restricted when using the interface. This also means that a dialogue manager should be as simple as possible and not waste effort on complex computations in order to handle phenomena not relevant for natural language interfaces. For instance, the system does not necessarily have to be psycholinguistically plausible or able to mimic all aspects of human dialogue behaviour such as surprise or irony, if these do not occur in such dialogues.

Grosz and Sidner (1986) presented a general computational theory of discourse, both spoken and written, where they divide the problem of managing discourse into three parts: linguistic structure, attentional state and intentional state.

The need for a component which records the objects, properties and relations that are in the focus of attention, the attentional state, is not much debated, although the details of focusing need careful examination.

However, the role that is given to the intentional state, i.e. the structure of the discourse purposes, and to the linguistic structure, i.e. the structure of the sequences of utterances in the discourse, provide two competing approaches to dialogue management:

- One approach is the plan-based approach. Here the linguistic structure is used to identify the intentional state in terms of the user's goals and intentions. These are then modelled in plans describing the actions which may possibly be carried out in different situations (cf. Cohen and Perrault, 1979; Allen and Perrault, 1980; Litman, 1985; Carberry, 1990; Pollack, 1990).
- The other approach to dialogue management is to use only the information in the linguistic structure to model the dialogue expectations, i.e. utterances are interpreted on the basis of their functional relation to the previous interaction. The idea is that these constraints on what can be uttered allow us to write a grammar to manage the dialogue (cf. Reichman, 1985; Polanyi and Scha, 1984; Bilange, 1991;

*This research was financed by the Swedish National Board for Technical Development and the Swedish Council for Research in the Humanities and Social Sciences.

Jönsson, 1991).

For the strong AI goal or the computational linguistics goal to mimic human language capabilities the plan recognition approach might be necessary. But, for the task of managing the dialogue in a natural language interface, the less sophisticated approach of using a dialogue grammar will do just as well, as will be argued below.

The work presented in this paper is restricted to studying written human-computer interaction in natural language, and natural language interfaces for different applications which belong to the domain that Hayes and Reddy (1983) called simple service systems. Simple service systems "require in essence only that the customer or client identify certain entities to the person providing the service; these entities are parameters of the service, and once they are identified the service can be provided" (*ibid.* p. 252).

A method for customization

The method presented in this paper proposes a sublanguage approach (Grishman and Kittredge, 1986) to the development of dialogue managers. A dialogue manager should not account for the interaction behaviour utilized in every application, instead it should be designed to facilitate customization to meet the needs of a certain application.

Kelley (1983) presents a method for developing a natural language interface in six steps. The first two steps are mainly concerned with determining and implementing essential features of the application. In the third step, known as the first Wizard of Oz-step, the subject interacts with what they believe is a natural language interface but which in fact is a human simulating such an interface (cf. Dahlbäck *et al.*, 1993; Fraser and Gilbert, 1991). This provides data that are used to build a first version of the interface (step four). Kelley starts without grammar or lexicon. The rules and lexical entries are those used by the users during the simulation. In step five, Kelley improves his interface by conducting new Wizard of Oz simulations, this time with the interface running. However, when the user/subject enters a query that the system cannot handle, the wizard takes over and produces an appropriate response. The advantage is that the user's interaction is not interrupted and a more realistic dialogue is thus obtained. This interaction is logged and in step six the system is updated to be able to handle the situations where the wizard responded.

The method used by Kelley of running a simulation in parallel with the interface was also used by Good *et al.* (1984). They developed a command language interface to an e-mail system using this iterative design method, UDI (User-Derived Interface). Kelley and Good *et al.* focus on updating the lexical and grammatical knowledge and are not concerned with dialogue behaviour.

The Dialogue Manager presented in this paper is customized to a specific application using a process inspired by the method of User-Derived Interfaces. The starting point is a corpus of dialogues collected in Wizard of Oz-experiments. From this corpus the knowledge structures used by the Dialogue Manager are customized.

The Dialogue Manager

The Dialogue Manager was initially designed from an analysis of a corpus of 21 dialogues, other than the 30 used for customization (see below) collected in Wizard of Oz-experiments using 5 different background systems[1]. It can be viewed as a controller of resources for interpretation, database 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 it is dealt with by the Dialogue Manager.

The Dialogue Manager uses information from dialogue objects which model the dialogue segments and moves and information associated with them. The dialogue objects represent the constituents of the dialogue and the Dialogue Manager records instances of dialogue objects in a dialogue tree as the interaction proceeds. The dialogue objects are divided into three main classes on the basis of structural complexity. There is one class corresponding to the size of a dialogue, another class corresponding to the size of a discourse segment (cf. Grosz and Sidner, 1986) and a third class corresponding to the size of a single speech act, or dialogue move. Thus, a dialogue is structured in terms of discourse segments, and a discourse segment in terms of moves and embedded segments. Utterances are not analysed as dialogue objects, but as linguistic objects which function as vehicles of one or more moves.[2]

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. From the perspective of dialogue management the dialogue objects modelling the discourse segment are the most interesting. An initiative-response (IR) structure is assumed (cf. adjacency-pairs, Schegloff and Sacks, 1973) where an initiative opens a segment by introducing a new goal and the response closes the segment (Dahlbäck, 1991). 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 a set of parameters for specifying the initiator, responder, context etc. needed

[1]For further details of the Dialogue Manager, see (Ahrenberg *et al.*, 1990); (Jönsson, 1991) and (Jönsson, 1993).

[2]The use of three categories for hierarchically structuring the dialogue is motivated from the analysis of the corpora. However, there is no claim that they are applicable to all types of dialogue, and even less so, to any type of discourse. When a different number of categories are utilized, the Dialogue Manager can then be customized to capture these other categories.

in most applications. Another set of parameters specify 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 (cf. Ahrenberg, 1987). These are focal parameters in the sense that they can be in focus over a sequence of IR-units.

Two principles for maintaining the focus structure are utilized. A general heuristic principle is that everything not changed in an utterance is copied from one IR-node in the dialogue tree to the newly created IR-node. Another principle is that the value for Objects will be updated with the value from the module accessing the database, if provided.

The dialogue objects are used to specify the behaviour of the Dialogue Manager and thus the specification of the dialogue objects must include information on what actions to take in certain situations. This is modelled in two non-focal content parameters, Type and Topic.

Type corresponds to the illocutionary type of the move. Hayes and Reddy (1983, p 266) identify two sub-goals in simple service systems: 1) "specify a parameter to the system" and 2) "obtain the specification of a parameter". Initiatives are categorized accordingly as being of two different types 1) update, U, where users provide information to the system and 2) question, Q, where users obtain information from the system. Responses are categorized as answer, A, for database answers from the system or answers to clarification requests. Other Type categories are Greeting, Farewell and Discourse Continuation (DC) (Dahlbäck, 1991) the latter of which is used for utterances from the system whose purpose is to keep the conversation going.

Topic describes which knowledge source to consult. In information retrieval applications three different topics are used: the database for solving a task (T), acquiring information about the database, system-related, (S) or, finally, the ongoing dialogue (D).

The empirical basis for customization

The Dialogue Manager is customized on the basis of a corpus of 30 dialogues collected in Wizard of Oz-experiments using the actual applications. Three different applications were used and each application utilized 10 dialogues for customization. The simulations were carefully designed and carried out using a powerful simulation environment, (Dahlbäck *et al.*, 1993).

In the experiments there were 14 female and 16 male subjects with varying familiarity with computers. Most subjects were computer novices. The average age was 26 (min. 15, max. 55). Most of the subjects were students but there were also others with varying backgrounds, such as cleaning staff and administrative assistants. The subjects did not realize that it was a simulation and they all, in post-experimental interviews, said that they felt very comfortable with the "system".

In the simulations a scenario is presented to the subjects. In one of the simulations, CARS, the scenario presents a situation where the subject, and his/her accompanying person, have just got the message that their old favourite Mercedes had broken down beyond repair and that they would have to consider buying a new car. They had a certain amount of money available and using the computerized CARS system were asked to select three cars, and also to provide a brief motivation for their choice.

The CARS database is implemented in INGRES, and output from the database can be presented directly to the subjects. Thus, answers from the system, after successful requests, are tables with information on properties of used cars. The users/subjects found this type of output very convenient as they could view a particular car in the context of other similar cars. This can be seen as an argument favouring an approach to natural language interfaces where complex reasoning is replaced with fast output of structured information. Possibly more information than asked for is provided, but as long as it can be presented on one screen, it is convenient.

The dialogues in the other domain, TRAVEL, were collected using two scenarios, one where the subjects were asked to gather information on charter trips to the Greek Archipelago and another where they have a certain amount of money available and were asked to use the TRAVEL system to order such a charter trip. In TRAVEL it is also possible to provide graphical information to the subjects, i.e., maps of the various islands.

The use of empirical material

An important question is how to use empirical material on the one hand and common sense and prior knowledge on human-computer interaction and natural language dialogue on the other. Dahlbäck (1991) claims that this partly depends on the purpose of the study, whether it is aimed at theory development or system development. In the latter case, one always has the possibility to design the system to overcome certain problems encountered in the corpus.

In this work empirical material is used for system development from two different perspectives. The first is to develop a dialogue manager for a natural language interface which can be used in various applications. Here the empirical material needs to be analysed with the aim of designing a dialogue manager general enough to cover all the dialogue phenomena that can occur in realistic human-computer dialogues using various background systems. Thus, phenomena which occur in the empirical material must be accounted for and also certain generalizations must be made so that the Dialogue Manager can later be customized to cover phenomena that are not actually present in the corpus but are likely to occur for other applications.

Empirical material is also used for customizing the Dialogue Manager to actual applications. Here generalization is less emphasized, instead many details of how to efficiently deal with the phenomena in the implementation are more interesting.

How can empirical material be used for customization? One can take the conservative standpoint and say that only those phenomena actually occurring in the dialogues are to be handled by the Dialogue Manager, (cf. Kelley, 1983). This has the advantage that a minimal model is developed which is empirically well motivated and which does not waste time on handling phenomena not occurring in the corpus. The drawback is that a very large corpus is needed for coverage of the possible actions taken by a potential user. This was also pointed out by Ogden (1988, p 296), who claims that "The performance of the system will depend on the availability of representative users prior to actual use, and it will depend on the abilities of the installer to collect and integrate the relevant information".

The other extreme standpoint is to only use the linguistic knowledge available. One problem with this approach is that it is plausible that much effort is spent on handling phenomena which will never occur in the dialogue, while at the same time not account for actually occurring phenomena. However, as pointed out by Brown and Yule (1983, p 21) "A dangerously extreme view of 'relevant data' for a discourse analyst would involve denying the admissibility of a constructed sentence as linguistic data".

For the purpose of customization, two kinds of information can be obtained from a corpus:

- First, it can be used as a source of phenomena which the designer of the natural language interface was not aware of from the beginning.
- Second, it can be used to rule out certain interesting phenomena which are complicated but which do not occur in the corpus.

The first point also includes the use of the corpus to make the system behaviour more accurate. This can be illustrated by the use of clarification subdialogues. In the CARS dialogues, when the user initiative is too vague and the system needs a clarification, it first explicitly states the alternatives available and then asks for a clarification. Subjects using the CARS system follow up such a clarification subdialogue as intended. However, in the TRAVEL system there are certain system clarification requests which are less explicit, and which do not state any alternatives. These clarifications do not always result in a follow up answer from the user.

To illustrate the second point, consider the use of singular pronouns. Singular pronouns can be used in various ways to refer to a previously mentioned item. One could argue that if a user utters something like *What is the price of a Toyota Corolla?*, and the answer is a table with two types of cars of different years, then the user may form a conceptualization of Toyota as a generic car and can therefore utter something like *How fast is it?* referring to properties of a Toyota Corolla of any year.

In the work on developing the Dialogue Manager, the use of pronouns in the corpus in various situations motivates the need for designing the Dialogue Manager to capture both uses of singular pronouns. However, when customizing the Dialogue Manager the situation is different. For instance, in the CARS dialogues the users restrict their use of singular pronouns. Thus, the customized Dialogue Manager for the CARS database is not provided with specific means for managing the use of singular pronouns if presented in the context above. If they occur they will result in a clarification subdialogue. However, the "normal" use of singular pronouns is allowed. There is another motivation for this position. Excluding the generic use of a singular pronoun leads to a simpler Dialogue Manager. On the other hand including the normal use of singular pronouns will not increase the complexity of the Dialogue Manager.

The principle utilized in the customization of the Dialogue Manager is obviously very pragmatic. If the phenomenon is present in the corpus then it should be included. If it is not present, but if it is present in other Wizard of Oz-studies using similar background systems and scenarios and implementation is straightforward, the Dialogue Manager should be customized to deal with it. Otherwise, if it is not present and it would increase the complexity of the Dialogue Manager, then it is not included.

This does not prevent the use of knowledge from other sources (cf. Grishman *et al.*, 1986). In the customization of the Dialogue Manager for the CARS and TRAVEL systems, knowledge on how the database is organised and also how users retrieve information from databases is used in the customization.

Customizing the Dialogue Manager

Customization of the Dialogue Manager involves two major tasks: 1) Defining the focal parameters of the dialogue objects in more detail and customizing the heuristic principles for changing the values of these parameters. 2) Constructing a dialogue grammar for controlling the dialogue.

The focus structure

In the CARS application, task-related questions are about cars which means that the Objects parameter holds various instances of sets of cars and Properties, are various properties of cars. In TRAVEL, on the other hand, users switch their attention between objects of different kinds: hotels, resorts and trips. This requires a slightly modified Objects parameter. It can be either a hotel or a resort. However, in TRAVEL the appropriate resort can be found from a hotel description by following the relation in the domain model from hotel to resort. Finding the hotel from a resort can be accomplished by a backwards search in the dialogue tree. Therefore, one single focused object – a hotel or a resort – will suffice. The value need not be a single object, it can be a set of hotels or resorts.

The general focusing principles need to be slightly modified to apply to the CARS and TRAVEL applications. For the CARS application the heuristic principles apply well to the Objects parameter. An intensionally specified object description provided in a user initiative will be replaced by the extensional specification provided by the module accessing the database, which means that erroneous objects will be removed, as they will not be part of the response from the database manager. For the TRAVEL application the principles for providing information to the Objects parameter are modified to allow hotels to be added if the resort remains the same.

The heuristic principles for the Properties parameter for the CARS application need to be modified. The principle is that if the user does not change Objects to a set of cars which is not a subset of Objects, then the attributes provided in the new user initiative are added to the old set of attributes. This is based on the observation that users often start with a rather large set, in this case a set of cars, and then gradually specify a smaller set by adding restrictions (cf. Kaplan 1983), for instance in CARS using utterances like *remove all small size cars*. For the TRAVEL application the copy principle holds without exception.

The modifications of the general principles are minor and are carried out during the customization.

The results from the customizations showed that the heuristic principles applied well. In CARS 52% of the user initiatives were fully specified, i.e. they did not need any information from the context to be interpreted. 43% could be interpreted from information found in the current segment as copied from the previous segment. Thus, only 5% required a search in the dialogue tree. For the TRAVEL application without ordering, 44% of the user initiatives were fully specified and 50% required local context, while in the ordering dialogues 59% were fully specified and 39% needed local context.

In the TRAVEL system there is one more object; the order form. A holiday trip is not fully defined by specifying a hotel at a resort. It also requires information concerning the actual trip: Travel length, Departure date and Number of persons. This information is needed to answer questions on the price of a holiday trip. The order form also contains all the information necessary when ordering a charter trip. In addition to the information on Resort, Hotel, Departure date, etc. the order form includes information about the name, address and telephone number of the user. Furthermore, information on travel insurance, cancellation insurance, departure airport etc. is found in the order form. The order form is filled with user information during a system controlled phase of the dialogue.

The dialogue structure

The dialogue structure parameters Type and Topic also require customization. In the CARS system the users never update the database with new information, but in the TRAVEL system where ordering is allowed the users update the order form. Here another Type is needed, CONF, which is used to close an ordering session by summarizing the order and implicitly prompt for confirmation. For the ordering phase the Topic parameter O for order is added, which means that the utterance affects the order form.

The dialogue structure can be modelled in a dialogue grammar. The resulting grammar from the customizations of both CARS and TRAVEL is context free, in fact, it is very simple and consists merely of sequences of task-related initiatives followed by database responses, Q_T/A_T[3], sometimes with an embedded clarification sequence, Q_D/A_D. In CARS 60% of the initiatives are of this type. For TRAVEL 83% of the initiatives in the non-ordering dialogues and 70% of the ordering dialogues are of this type. Other task related initiatives result in a response providing system information, Q_T/A_S, or a response stating that the intitiative was too vague, Q_T/A_D. There are also a number of explicit calls for system information, Q_S/A_S. The grammar rules discussed here only show two of the parameters of the dialogue objects. In fact, a number of parameters describing speaker, hearer, objects, properties, etc are used. These descriptors provide additional information for deciding which actions to carry out. However, the complexity of the dialogue is constrained by the grammar.

The dialogue grammar is developed by first constructing a minimal dialogue grammar from an analysis of dialogues from the application, or an application of the same type, e.g. information retrieval from a database. This grammar is generalized and extended, using general knowledge on human-computer natural language interaction, with new rules to cover "obvious" additions not found in the initial grammar. In the CARS dialogues it included, for instance, Greetings and Farewells, which did not appear in the analysis of the dialogues. In the TRAVEL system it involved, among other things, allowing for multiple clarification requests and clarification requests not answered by the user. Some extensions not found in any of the dialogues were also added, for instance, a rule for having the system prompt the user with a discourse continuation if (s)he becomes unsure who has the initiative. However, if a phenomenon requires sophisticated and complex mechanisms, it will be necessary to consider what will happen if the grammar is used without that addition. This also includes considering how probable it is that a certain phenomenon may occur.

For each new application, new simulations are needed to determine which phenomena are specific for that application. This is illustrated in the TRAVEL system dialogues where ordering is not allowed. In these dialogues some users try to state an order although it is not possible. This resulted in a new rule, U_O/A_S, informing the users that ordering is not possible.

In the work by Kelley (1983) and Good *et al.* (1984), on lexical and grammatical acquisition, the customization process was saturated after a certain number of dialogues. The results presented here indicate that this is also the case for the dialogue structure. From a rather limited number of dialogues, a context free grammar can be constructed which, with a few generalizations, will cover the interaction patterns occurring in the actual application (Jönsson, 1993).

Summary

This paper has presented a method for the development of dialogue managers for natural language interfaces for various applications. The method uses a general dialogue manager which is customized from a corpus of dialogues, with users interacting with the actual application, collected in Wizard of Oz-experiments. The corpus is used when customizing dialogue objects with parameters and heuristic principles for maintaining focus structure. It is also used when constructing a dialogue grammar which controls the dialogue.

The customization of the Dialogue Manager for two

[3] For brevity, when presenting the dialogue grammar, Topic type will be indicated with a subscript to the Type. The Initiative is the first TypeTopic-pair while the Response is the second separated by a slash (/).

different applications – database information retrieval and database information retrieval plus ordering – was also presented. Customization was carried out for two different domains: properties of used cars and information on holiday trips. For both domains questions can be described as queries on specifications of domain concepts about objects in the database and simple heuristic principles are sufficient for modelling the focus structure. A context free dialogue grammar can accurately control the dialogue for both applications. The results on customization are very promising for the approach to dialogue management presented in this paper. They show that the use of dialogue objects which can be customized for various applications in combination with a dialogue grammar is a fruitful way to build application-specific dialogue managers.

Acknowledgements

I am indebted to Lars Ahrenberg and Nils Dahlbäck for many valuable discussions. I will also thank Brant Cheikes, Jalal Maleki, Magnus Merkel and Ivan Rankin for commenting on previous versions of the paper. Åke Thurée did most of the implementation of the Dialogue Manager for the CARS system.

References

Ahrenberg, Lars; Jönsson, Arne; and Dahlbäck, Nils 1990. Discourse representation and discourse management for natural language interfaces. In *Proceedings of the Second Nordic Conference on Text Comprehension in Man and machine, Täby.*

Ahrenberg, Lars 1987. *Interrogative Structures of Swedish. Aspects of the Relation between grammar and speech acts.* Ph.D. Dissertation, Uppsala University.

Allen, James F. and Perrault, C. Raymond 1980. Analysing intention in utterances. *Artificial Intelligence* 15:143–178.

Bilange, Eric 1991. A task independent oral dialogue model. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin.*

Brown, Gillian and Yule, George 1983. *Discourse Analysis.* Cambridge University Press.

Carberry, Sandra 1990. *Plan Recognition in Natural Language Dialogue.* MIT Press, Cambridge, MA.

Cohen, Philip. R. and Perrault, C. Raymond 1979. Elements of a plan-based theory of speech acts. *Cognitive Science* 3:177–212.

Dahlbäck, Nils and Jönsson, Arne 1992. An empirically based computationally tractable dialogue model. In *Proceedings of the Fourteenth Annual Meeting of The Cognitive Science Society, Bloomington, Indiana.*

Dahlbäck, Nils; Jönsson, Arne; and Ahrenberg, Lars 1993. Wizard of Oz studies - why and how. In *Proceedings from the 1993 International Workshop on Intelligent User Interfaces, Orlando, Florida.*

Dahlbäck, Nils 1991. *Representations of Discourse, Cognitive and Computational Aspects.* Ph.D. Dissertation, Linköping University.

Fraser, Norman and Gilbert, Nigel S. 1991. Simulating speech systems. *Computer Speech and Language* 5:81–99.

Good, Michael D.; Whiteside, John A.; Wixon, Dennis R.; and Jones, Sandra J. 1984. Building a user-derived interface. *Communications of the ACM* 27(10):1032–1043.

Grishman, Ralph and Kittredge, Richard I. 1986. *Analysing language in restricted domains.* Lawrence Erlbaum.

Grishman, Ralph; Hirshman, Lynette; and Nhan, Ngo Thanh 1986. Discovery procedures for sublanguage selectional patterns: Initial experiments. *Computational Linguistics* 12(3):205–215.

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

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

Jönsson, Arne 1991. A dialogue manager using initiative-response units and distributed control. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin.*

Jönsson, Arne 1993. *Dialogue Management for Natural Language Interfaces – An Empirical Approach.* Ph.D. Dissertation, Linköping University.

Kaplan, S. Jerrold 1983. Cooperative responses from a portable natural language database query system. In *Computational Aspects of Discourse.* MIT Press. 167–208.

Kelley, John F. 1983. *Natural Language and Computers: Six Empirical Steps for Writing an Easy-to-Use Computer Application.* Ph.D. Dissertation, The Johns Hopkins University.

Litman, Diane J. 1985. *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues.* Ph.D. Dissertation, University of Rochester.

Ogden, William C. 1988. Using natural language interfaces. In Helander, M., editor 1988, *Handbook of Human-Computer Interaction.* Elsevier Science Publishers B. V. (North Holland).

Polanyi, Livia and Scha, Remko 1984. A syntactic approach to discourse semantics. In *Proceedings of the 10th International Conference on Computational Linguistics, Stanford.*

Pollack, Martha E. 1990. Plans as complex mental attitudes. In Cohen, Philip R.; Morgan, Jerry; and Pollack, Martha E., editors 1990, *Intentions in Communication.* MIT Press.

Reichman, Rachel 1985. *Getting Computers to Talk Like You and Me.* MIT Press, Cambridge, MA.

Schegloff, Emanuel A. and Sacks, Harvey 1973. Opening up closings. *Semiotica* 7:289–327.