Dialogue actions for natural language interfaces

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

Abstract

This paper presents an action scheme for dialogue management for natural language interfaces. The scheme guides a dialogue manager which directs the interface's dialogue with the user, communicates with the background system, and assists the interpretation and generation modules. The dialogue manager was designed on the basis of an investigation of empirical material collected in Wizard of Oz-experiments. The empirical investigations revealed that in dialogues with database systems users specify an object, or a set of objects, and ask for domain concept information, e.g. the value of a property of that object or set of objects. The interface responds by performing the appropriate action, e.g. providing the requested information or initiating a clarification subdialogue. The action to be carried out by the interface can be determined based on how objects and properties are specified from information in the user utterance, the dialogue context, and the response from the background system and its domain model.

1 Introduction

Users of natural language interfaces should conveniently be able to express the commands and queries that the background system can deal with, and the system should react quickly and accurately to all user input. Among other things this means that the interface must be able to cope with connected dialogue. However, it does not mean that the interface must be able to mimic human interaction. On the contrary, it is erroneous to assume that humans would like to interact with computers the same way as they communicate with humans (cf. [Dahlbäck, 1991b; 1991a; Dahlbäck and Jönsson, 1992; Dahlbäck *et al.*, 1993; Krause, 1993]). Human computer interactions have their own sublanguages (cf. [Grishman and Kittredge, 1986]) whose characteristics often allow a much simpler dialogue model than models capturing human interaction.

To illustrate some properties of such human computer interaction consider figure 1. In information retrieval systems a common user initiative is a request for domain concept information of a specified object, or set of objects. Utterance U11 illustrates this. The requested domain concept information is the value of the property shape and the domain object is the Ford Fiesta costing 26 800 crowns. Unfortunately the system could not answer the question as the property (shape) is not utilized in the domain, instead, in utterance S12, the system provides information about its capabilities. In U13 a new request for information on another property of the same domain object is presented. This time the pronoun *it* replaces the rephrasing of the specification of the object, i.e. the Ford Fiesta costing 26 800 crowns. In utterance U15 the user asks for the same concept information but related to another object, while in U17 the object stays the same but the property is altered. In U19 the property remains the same but this time the user utilizes a definite description to specify an object discussed previously, and originally specified in utterance U11.

The dialogue model presented in this paper does not intend to mimic human conversation. It is based on the observation that for information retrieval applications a common user initiative is a request for domain concept information of a specified object, or set of objects (cf. [Ahrenberg, 1987]). A dialogue manager utilizing this information when deciding which action to perform for user initiatives concerned with accessing the application will provide efficient and robust user-friendly human-computer natural language interaction.

2 The Dialogue Manager

A dialogue manager directs a natural language interface and holds information needed by the modules in the interface, including the dialogue manager itself. The Dialogue Manager considered in this paper was designed from an analysis of a corpus of 21 dialogues, using five different background systems [Ahrenberg *et al.*, 1990; Jönsson, 1991], collected in Wizard Oz-experiments [Dahlbäck *et al.*, 1993]. The Dialogue Manager need to be customized to account for the sublanguage carried out in a specific application. Customization allows us to adapt the behaviour of the interface to the requirements of the application (see Jönsson [1993a; 1993b] for details).

The results presented here are based on the customiza-

U11: What is the shape of Ford Fiesta costing 26 800 crowns?
S12: Wait...
Cars cannot answer questions concerning the shape of car models.
U13: Is it rusty?
S14: Wait...
Checking...

Manufacturer	Model	Year	Rust
Ford	Fiesta	1982	2

U15: Does the Mercedes from 1982 have any rust damage?
S16: Wait...
Checking...

Manufacturer	Model	Year	Rust
Mercedes	200	1982	5

U17: How fast is a Mercedes 200?
S18: Wait...
Checking...

Manufacturer	Model	Year	Rust	Top Speed
Mercedes	200	1982	5	160

U19: How fast is the Fiesta?
S20: Wait...
Checking...

Manufacturer	Model	Year	Rust	Top Speed
Ford	Fiesta	1982	2	145

Figure 1: Example of human computer dialogue using the CARS system. From a corpus of dialogues collected in Wizard of Oz-experiments. U denotes user utterances and S utterances from the system. The corpus examples are translated from Swedish.

tion of the dialogue manager for three applications, other than those utilized in the design, using a set of 30 new dialogues. One of the applications, CARS, allows users to retrieve information from a consumers guide on properties of used cars. In another application, TRAVEL, the application domain was charter trips to the Greek archipelago. The TRAVEL application not only utilized information retrieval but also, in one scenario, allowed users to order a specified charter trip.

Dialogue management information is modeled in dialogue objects. These represent the constituents of the dialogue. A dialogue object has two components. One component contains static information describing the properties and relations of the dialogue object. Another is a process description of a prototypical use of the dialogue object, an action plan [Jönsson, 1991]. During the course of interaction a dialogue tree is built up from instances of dialogue objects (For more details on the Dialogue Manager see Jönsson [1993a]).

Two types of static parameters are distinguished: situation parameters and content parameters. The situation parameters specify the situation in which the dialogue object occurs, such as Initiator, Responder and contextual information. The content parameters are needed for focus structure and dialogue structure. Focus structure concerns the recording of entities mentioned in the discourse to allow a user to refer to them in the course of the interaction. Dialogue structure involves handling the relationships between the segments in the 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:

	4/7	11/7	18/7	25/7
1 week	2725	2725	2675	2625
2 weeks	3272	3275	3225	3175

U31: what is next cheapest
S32: Wait...
Price Village Suites:

	4/7	11/7	18/7	25/7
1 week	3150	3150	3100	3050
2 weeks	4025	4025	3975	3925

U33: what service can these hotels provide
S34: Wait...
The basic price at Villa Margarita includes shared apartment.
Cleaning 3 days/week, no dish-washing. No meals.
The basic price at Village Suites includes shared apartment.
Cleaning 3 days/week, no dish-washing. No meals.

Figure 2: Example of dialogue using the TRAVEL system.

2.1 Focus structure parameters

As discussed above, users of information retrieval systems request database information by specifying a database object, or a set of objects, and ask for the value of a property of that object or set of objects. The dialogue objects model database objects using a parameter termed Objects and the domain concept information in a parameter termed Properties. The values to these parameters depend on the background system, and the natural language interface needs to be customized to account for the demands from each application [Jönsson, 1993b]. For the CARS application a relational database is used and the objects are cars described by the sub-parameters (Manufacturer, Model, Year). The TRAVEL, application utilizes a hierarchically structured database with the Greek archipelago on top, then the resorts and finally the hotels at each resort. However, 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 figure 2. 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.

The value to the Objects parameter can be explicitly provided as, for instance, it is in *show saab 900 of 1985 model*. However, this is not often the case. Instead, the user provides only partial information, or a new set of objects by specifying properties, e.g. *Show all medium size cars with a safety factor larger than 4*. It is also possible to describe new objects by way of other objects, as for example in U27 in figure 2. The Objects parameter will achieve values from such intensionally specified object descriptions by the extensional specification provided from the database access system.

The Properties parameter models the domain concept in a sub-parameter termed Aspect which can be specified in another sub-parameter termed Value. For instance, utterance U17 in figure 1 *How fast is a Mercedes 200?* provides Aspect information on the domain concept, *speed* which is specified by the database manager to *160*, i.e. the Value of the Aspect *speed* is *160*.

For some applications a third focal parameter is needed, termed Secondary Objects. Its purpose is to restrict the search in the database to allow the user to investigate objects from a subset of objects one at a time as exemplified in figure 2. The user picks out the set of hotels at the resort but is only interested in a subset of them. If we apply the principle that hotels are appended to the Objects parameter if the resort remains the same, the Objects parameter will hold the subset requested in U33. However, to restrict the database search in U31 to the set specified in S28, Secondary Objects is needed to hold the subset from which individual objects are investigated.

The focus parameters are properties of discourse segments (cf. [Zancanaro *et al.*, 1993]), not moves. Focus is maintained using a simple copying principle where each new dialogue object is instantiated with a copy of the focus parameters from the previous dialogue object (cf. [Seneff, 1992]). This forms the initial context for the dialogue object and is updated with new information from the user initiative and the response from the background system.

The details on how to update the focal parameters vary and need to be considered when customizing the dialogue objects for a specific application. For instance, consider the system response S18 in figure 1. This response does not only contain the requested information on the Aspect sub-parameter *top speed*. It also provides information on the Aspect sub-parameter *rust* specified in the previous user initiative. If the value to the Objects parameter remains the same (or is a subset of the previous value), the value to the Properties parameter will be the conjunction of the previous value and the new values provided in the new move. This principle is appropriate when information is presented in tables allowing additional information to be presented conveniently [Ahrenberg *et al.*, 1993].

2.2 Dialogue structure parameters

The dialogue is 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 and a third class corresponding to the size of a single speech act, or dialogue move. Utterances are not analyzed as dialogue objects, but as linguistic objects which function as vehicles of one or more moves. There are various other proposals as to the number of categories needed. They differ mainly on the modeling of complex units that consist of sequences of discourse segments, but do not comprise the whole dialogue. For instance, LOKI [Wachtel, 1986] and SUNDIAL [Bilange, 1991] use four. In LOKI the levels are: conversation, dialogue, exchange and move. SUNDIAL uses the categories Transaction level, Exchange level, Intervention level and Dialogue Acts. The feature characterizing the intermediate level (i.e. the Dialogue and Exchange levels respectively in Wachtel's and Bilange's models) is that of having a common topic, i.e. an object whose properties are discussed over a sequence of exchanges. However, as illustrated in figure 1, a sequence of segments may hang together in a number of different ways; e.g. by being about one object for which different properties are at issue. But it may also be the other way around, so that the same property is topical, while different objects are talked about (cf. [Ahrenberg *et al.*, 1990]). Thus, only one discourse segment category is distinguished and 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, 1991b].

To specify the functional role of a move we use the parameters Type and Topic.

Type corresponds to the illocutionary type of the move. For so-called simple service systems[1] two subgoals can be identified [Hayes and Reddy, 1983, p. 266]: 1) specifying a parameter to the system and 2) obtaining 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. The Dialogue Manager utilizes other Type categories such as Greeting, Farewell and Discourse Continuation (DC) [Dahlbäck, 1991b] the latter being used for utterances from the system whose purpose is to keep the conversation going, but they will not be further considered in this paper.

Topic describes which knowledge source to consult. For information retrieval applications three different knowledge sources are utilized: the database for solving a task (T), acquiring information about the database, system-related, (S) or, finally, the ongoing dialogue (D). If the background system allows ordering of a specified item a fourth category is needed to account for such utterances.

The Type/Topic parameters can be used to describe the dialogue structure, i.e. which action to be carried out by the interface. This in turn can be modeled in a dialogue grammar [Jönsson, 1993a].

3 Actions for task-related initiatives

Normally a natural language interface to database information retrieval applications is user-directed, i.e. the user initiates a request for information from the background system and the interface responds with the requested information. The interface only takes the initiative to begin a clarification request under three

[1] 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" [Hayes and Reddy, 1983, p. 252].

Objects	Properties	Action(s)
Correct Partly Correct Not Provided	Correct Partly Correct Aspect	A_T
Correct Partly correct Not provided Incompatible	Erroneous Value Ambiguous Aspect	$Q_D/A_D\ A_T$ (A_D)
Correct	Not provided	$Q_D/A_D\ A_T$
Erroneous	-	A_S
-	Erroneous Aspect	A_S
Incompatible		A_S
(Too large to print)		$Q_D/A_D\ A_T$

Table 1: A summary of the Dialogue Manager's actions to task-related initiatives.

circumstances[2]:

- a difficulty arises when interpreting an utterance, e.g. unknown words or questions outside the domain of the database.
- a difficulty arises when accessing the database, e.g. when the user needs to provide a parameter for correct access.
- a difficulty arises in the presentation of the result from the database access, e.g. the answer is too large to print on one screen.

The action to be carried out for task-related questions depends on how the information in the user initiative together with the information copied from the previous IR-unit and context information from the dialogue tree and the answer from the database system specify the values to the focal parameters Objects and Properties. This contrasts with other structural based approaches, such as Sitter and Stein [1992], where the user's purpose is considered primary when deciding which action to carry out. An object or property description can be either: correct, partly correct, incompatible, ambiguous, erroneous, or not provided. Erroneous means that the user has specified an object which is not in the database. Partly correct means that the description contains at least one correct object or property description, but also one or more erroneous descriptions. Incompatible descriptions utilize elements which do not belong together, e.g. Volvo Camry.

The relation between the values to the Objects and Properties parameters and the resulting action described in terms of Type and Topic is summarized in table 1[3]. Any combination of Objects and Properties in a cell in a row results in the action to the right. From the table we

[2]The system also takes the initiative to collect ordering information.

[3]When presenting the dialogue actions, Topic type will be indicated with a subscript to the Type, e.g. A_T denotes a task-related answer. IR-units are presented as a TypeTopic-pair with the Initiative separated from the Response by a slash (/).

U17: which 10 car models between 60 000 and 70 000 crowns are most spacious
S18: Wait...
Checking...
Information on space is either coupe or boot.
Please be more specific.
U19: best coupe

Figure 3: Example of ambiguous Aspect resulting in a clarification request.

can identify three basic actions to task-related IR-units depending on the values of the parameters Objects and Properties: A_T, A_S, and $Q_D/A_D\ A_T$.

- A_T is the normal action following a Q_T. This describes a successful task-related user initiative followed by a successful system answer with information taken from the database. This requires correct values for both Objects and Properties. The values for these parameters can be taken either from the preceding dialogue or they could be provided in the user input. What is important is that the initiative in context provides enough information so that it can be used to access the background system and that the answer from the background system is in some sense correct. A special case is when no explicit Objects description is provided but the Properties are fully specified and can be used to access the database, e.g. *show all medium class cars costing less than 70 000 crowns.*

  If the parameters Objects or Properties are partly correct, i.e. contain one or more erroneous items, then an answer is presented on the correctly specified items together with information about what was erroneous, if possible.

- $Q_D/A_D\ A_T$ is to be considered as a special case of the normal A_T-action as specified above. This category is concerned with cases where the system initiates a clarification subdialogue to achieve more information from the user in order to get fully and correctly specified values to Objects or Properties. If the user decides not to answer the clarification request, then the values from the initiating IR-unit are copied to the new IR-unit and interaction proceeds from there. The treatment of multiple sequential clarifications follows the same pattern as that for one clarification subdialogue.

  A clarification subdialogue can be initiated when the Objects are correctly specified but the values of the Value slot to the Properties are erroneous or under-specified. For instance, in *remove all cars with low operational safety* the expression low is too vague. Another case is where no Aspect is provided or the provided Aspect is ambiguous. The latter is illustrated in utterance U17 in figure 3.

  Such cases are handled by a system initiated clarification subdialogue, a Q_D/A_D, directed from the IR-unit which started the interaction, normally a Q_T, with the under-specified or ambiguous property copied from the initiating IR-unit. The Aspect slot is used to hold the parameter for which the system wants an answer and the Value slot is used for the user's answer. If the user answers correctly, as in U19 in figure 3, the values for Properties in the initiating IR-unit are updated. A Q_D/A_D-unit is identified from the type information, i.e. the Type of the response from the user is A. Otherwise the user move is regarded not to be an answer to the systems clarification request. A clarification subdialogue is not initiated unless the system is able to explicitly provide alternatives to the user.

A special case of clarification request occurs when a correct specification of the parameters Objects and Properties is provided, but the answer is too large to print on the screen. In such cases the system initiates a clarification subdialogue asking the user to restrict the number of items to be printed, for example, *S2: Wait... There are 76 car models which satisfy your requirements.* CARS *normally only shows 25 cars at a time. Do you want to see them all?*. The answer can be either a number, a restriction such as *U3: remove cars costing less than 40 000 crowns*, or Yes or No. It is used to restrict the number of objects to output on the screen and also in some cases affect the values of the Objects parameter.

- A_S is used for task-related user initiatives resulting in a system answer which provides information about the database system. Information can be provided on various aspects of what type of information there is in the database and what type of questions that can be used to elicit this information. A typical example is *Cars cannot answer questions concerning the shape of car models.* An A_S is utilized for any utterance with erroneous Objects or Aspect. Incompatible Properties and Objects also result in an A_S, this means that although both Properties and Objects are correct, they cannot be used together.

To illustrate the action scheme consider utterance U11 *What is the shape of Ford Fiesta costing 26 800 crowns?* in figure 1. This will be interpreted as a task-related question, a Q_T, with correctly specified Objects parameter. However, the Aspect sub-parameter is erroneous, as there is no information in the database on the concept *shape*. Furthermore, the system can not provide alternatives to the user. Thus, the resulting action is an A_S, S12. The next user utterance, U13, is a Q_T with both correct Objects, as copied from the previous IR-unit, and correct Aspect sub-parameter, *rust*. Thus, the resulting action is an A_T, S14.

It is not always possible to directly use the values in the Objects and Properties slots, even if correctly specified. For applications such as TRAVEL, with hierarchically structured databases the Dialogue Manager sometimes needs to search the domain base or the dialogue tree to find an applicable object or property. For instance, if the user in the dialogue in figure 2 asks for concept information on properties associated with resorts, such as climate, when the hotels are in focus, the domain model is utilized to find the appropriate resort.

There are user initiatives which do not depend on the values of Objects and Properties, such as system-related questions, Q_S, i.e. the user requests information about the system. These are recognized on the grounds of linguistic information provided by the syntactic/semantic analyzer [Ahrenberg, 1988].

If ordering is allowed it is important to know which task is currently being performed, exploring the database or ordering. This problem has been discussed by, for instance, Ramshaw [1991], and Lambert and Carberry [1991]. They present models using three different, but interacting, levels of plans to know when users stop exploring different plans and instead commit themselves to one plan. However, a result emerging from the analysis of our dialogues [Jönsson, 1993a] is that the subjects clearly signal when they change plan, using utterances such as *I would like to order a trip for two to Lefkada.* Thus, retrieval of ordering information from the users can be collected in a formalized fashion controlled by the system, (cf. [Hoeppner *et al.*, 1986]).

4 Results

Dialogue objects has been customized to meet the demands of the three systems discussed above: CARS and TRAVEL with and without ordering. The customized dialogue objects for the CARS system has also been integrated with an INGRES database and interpreting modules using a grammar and a lexicon covering a subset of the utterances found in the corpus. A context free grammar with less than 20 rules can accurately model the dialogue structure utilized in the corpus. The principle of copying information from one dialogue object to the other provides the correct context for most referring expressions. For CARS only 5% required a search in the dialogue tree. The corresponding numbers for TRAVEL were 6% for information retrieval and 2% if ordering is utilized (For more details on the results from customizing the dialogue and focus structures, see Jönsson [1993a] and Ahrenberg *et al.* [1993]).

The action scheme presented in table 1 covers all task-related user initiatives utilized in the corpus. In the CARS application 85% of the user initiatives are task-related questions. In the TRAVEL application without ordering the number of task-related user initiatives account for 93% of the user utterances and finally when ordering is allowed 90% of the user utterances are task-related. The other user initiatives are system related questions, farewells, greetings, etc which are interpreted from linguistic information. Thus, a majority of the users' initiatives are task-related and will be handled efficiently and accurately using the action scheme.

5 Discussion

The Dialogue Manager presented in this paper is restricted to written human-computer interaction in natural language. However, when communicating with a natural language interface, a user should not be limited to typed keyboard input and screen output. The possibilities of using various modalities must be addressed to further improve the interaction. Examples of systems which use a variety of modalities for both interpretation and generation include AlFresco [Stock, 1991], XTRA [Wahlster, 1991], Voyager [Zue, 1994] and CUBRICON [Neal and Shapiro, 1991].

The main difference between multi-modal interfaces to simple service systems and conventional natural language interfaces to such applications is their ability to utilize a combination of input and output modalities such as speech, graphics, pointing and video output. Thus, more advanced interpretation and generation modules are required and principles for determining how to utilize each media are needed [Arens *et al.*, 1993].

However, the dialogue and focus structures need not necessarily be more complicated. For instance, Voyager [Zue, 1994] successfully utilizes the approach presented here of copying the focus parameters from one segment to the other [Seneff, 1992]. Sitter and Stein [1992] present a model for dialogue management to information-seeking dialogues. The model assumes that conversation is based on possible sequences of dialogue acts which are modeled in a transition network. In Stein and Thiel [1993] the model is extended to handle multi-modal interaction as utilized in the MERIT system [Stein *et al.*, 1992].

Thus, it seems that for simple service systems, the dialogue model presented here will be sufficient not only for natural language interfaces but also interfaces utilizing various other modalities. However, for task-oriented dialogues, where the user's task directs the dialogue [Loo and Bego, 1993], a model of this and the user's goals need to be consulted in order to provide user-friendly interaction (cf. [Burger and Marshall, 1993]). This does not imply the necessity of a sophisticated model based on the user's intentions. Utilizing a hierarchical structure of plans based on the various tasks possible to carry out in the domain might do just as well (cf. [Wahlster *et al.*, 1993]).

6 Summary

Natural language interaction will be more robust and habitable if the users can participate in a coherent dialogue with the system. For natural language interfaces to information retrieval applications the necessary dialogue actions can be determined using a straightforward solution. Users specify a database object, or set of objects, and ask for domain concept information of that object or objects. This is modeled in two parameters, one associated with the objects and another with the requested properties of that object. The parameters are specified from information in the user initiative, the discourse and the background system and its domain model. The action to be carried out by the interface can be determined from the specification of these objects and properties parameters.

Acknowledgments

This work results from a project on Dynamic Natural-Language Understanding supported by The Swedish Council of Research in the Humanities and Social Sciences (HSFR) and The Swedish National Board for Industrial and Technical Development (NUTEK) in the joint Research Program for Language Technology. The work has been carried out with the members of the Natural Language Processing Laboratory at Linköping University, Sweden, and I am especially indebted to Lars Ahrenberg, Nils Dahlbäck and Åke Thurée.

References

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

[Ahrenberg *et al.*, 1993] Lars Ahrenberg, Arne Jönsson, and Åke Thurée. Customizing interaction for natural language interfaces. In *Workshop on Pragmatics in Dialogue, The XIV:th Scandinavian Conference of Linguistics and the VIII:th Conference of Nordic and General Linguistics, Göteborg, Sweden*, 1993.

[Ahrenberg, 1987] Lars Ahrenberg. *Interrogative Structures of Swedish. Aspects of the Relation between grammar and speech acts.* PhD thesis, Uppsala University, 1987.

[Ahrenberg, 1988] Lars Ahrenberg. Functional constraints in knowledge-based natural language understanding. In *Proceedings of the 12th International Conference on Computational Linguistics, Budapest*, pages 13–18, 1988.

[Arens *et al.*, 1993] Yigal Arens, Eduard Hovy, and Mira Vossers. On the knowledge underlying multimedia presentations. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 280–306. MITPress, 1993.

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

[Burger and Marshall, 1993] John D. Burger and Ralph J. Marshall. The application of natural language models to intelligent multimedia. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 174 – 196. MITPress, 1993.

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

[Dahlbäck *et al.*, 1993] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of oz studies – why and how. *Knowledge-Based Systems*, 6(4):258–266, 1993.

[Dahlbäck, 1991a] Nils Dahlbäck. Empirical analysis of a discourse model for natural language interfaces. In *Proceedings of the Thirteenth Annual Meeting of The Cognitive Science Society, Chicago, Illinois*, pages 1–6, 1991.

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

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

[Hayes and Reddy, 1983] 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.

[Hoeppner *et al.*, 1986] Wolfgang Hoeppner, Katharina Morik, and Heinz Marburger. Talking it over: The natural language dialog system ham-ans. In Leonard Bolc and Matthias Jarke, editors, *Cooperative Interfaces to Information Systems.* Springer-Verlag, Berlin Heidelberg, 1986.

[Jönsson, 1991] Arne Jönsson. 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*, 1991.

[Jönsson, 1993a] Arne Jönsson. *Dialogue Management for Natural Language Interfaces – An Empirical Approach.* PhD thesis, Linköping University, 1993.

[Jönsson, 1993b] Arne Jönsson. A method for development of dialogue managers for natural language interfaces. In *Proceedings of the Eleventh National Conference of Artificial Intelligence, Washington DC*, pages 190–195, 1993.

[Krause, 1993] Jürgen Krause. A multilayered empirical approach to multimodality: Towards mixed soultions of natural language and graphical interfaces. In Mark T. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 328 – 352. MITPress, 1993.

[Lambert and Carberry, 1991] Lynn Lambert and Sandra Carberry. A tripartite plan-based model of dialogue. In *Proceedings of the 29th Annual Meeting of the ACL, Berkeley*, pages 193–200, 1991.

[Loo and Bego, 1993] W. Van Loo and H. Bego. Agent tasks and dialogue management. In *Workshop on Pragmatics in Dialogue, The XIV:th Scandinavian Conference of Linguistics and the VIII:th Conference of Nordic and General Linguistics, Göteborg, Sweden*, 1993.

[Neal and Shapiro, 1991] Jeannette G. Neal and Stuart C. Shapiro. Intelligent multi-media interface technology. In Joseph W. Sullivan and Sherman W. Tyler, editors, *Intelligent User Interfaces.* ACM Press, Addison-Wesley, 1991.

[Ramshaw, 1991] Lance A. Ramshaw. A three-level model for plan exploration. In *Proceedings of the 29th Annual Meeting of the ACL, Berkeley*, pages 39–46, 1991.

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

[Seneff, 1992] Stephanie Seneff. A relaxation method for understanding spontaneous speech utterances. In *Paper presented at the Fifth DARPA Workshop on Speech and Natural Language*, 1992.

[Sitter and Stein, 1992] Stefan Sitter and Adelheit Stein. Modeling the illocutionary aspects of information-seeking dialogues. *Information Processing & Management*, 28(2):165–180, 1992.

[Stein and Thiel, 1993] Adelheit Stein and Ulrich Thiel. A conversational model of multimodal interaction in information systems. In *Proceedings of the Eleventh National Conference of Artificial Intelligence, Washington DC*, pages 283 – 288, 1993.

[Stein *et al.*, 1992] Adelheit Stein, Ulrich Thiel, and Anne Tißen. Knowledge based control of visual dialogues in information systems. In *Proceedings of the 1st International Workshop on Advanced Visual Interfaces, Rome, Italy*, 1992.

[Stock, 1991] Oliviero Stock. Natural language exploration of an information space: the alfresco interactive system. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, Sydney, Australia*, pages 972–978, 1991.

[Wachtel, 1986] Tom Wachtel. Pragmatic sensitivity in nl interfaces and the structure of conversation. In *Proceedings of the 11th International Conference of Computational Linguistics, University of Bonn*, pages 35–42, 1986.

[Wahlster *et al.*, 1993] Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans-Jürgen Profitlich, and Thomas Rist. Plan-based integration of natural language and graphics generation. *Artificial Intelligence*, 63:387 – 427, 1993.

[Wahlster, 1991] Wolfgang Wahlster. User and discourse models for multimodal communication. In Joseph W. Sullivan and Sherman W. Tyler, editors, *Intelligent User Interfaces.* ACM Press, Addison-Wesley, 1991.

[Zancanaro *et al.*, 1993] Massimo Zancanaro, Oliviero Stock, and Carlo Strapparava. Dialogue cohesion sharing and adjusting in an enhanced multimodal environment. In *Proceedings of the International Joint Conference of Artificial Intelligence, Chambery, France*, pages 1230– 1236, 1993.

[Zue, 1994] Victor W. Zue. Toward systems that understand spoken language. *IEEE Expert*, 9:51–59, 1994.