Mixed-Initiative Retrieval Dialogues Using Abductive Reasoning

Jon Atle Gulla*
Norwegian University of Science and Technology
Department of Computer Science
(NTNU-IDT)
N-7034 Trondheim, Norway
jag@idt.ntnu.no

Adelheit Stein
German National Research Center for Information Technology, Integrated Publication and Information Systems Institute
(GMD-IPSI)
Dolivostr. 15, D-64293 Darmstadt, Germany
stein@darmstadt.gmd.de

Abstract

Using intelligent dialogue components, we can relate the isolated functions of information retrieval systems to larger tasks and help the user satisfy her information needs. Such dialogue components monitor the interaction between user and system and suggest what the user should do on the basis of her goals and the dialogue history. However, for these components to have the necessary flexibility, they must allow both user and system to take the initiative and control the direction of the dialogue. In our retrieval system, certain dialogue scripts decide who is in charge of the interaction and what are the recommended user acts in a goal-directed dialogue. The user can take the initiative at any point of the dialogue by issuing an act not recommended to her. The system uses abduction to interpret this act in light of the dialogue history and initiates a new dialogue addressing the user's choice of interpretation.

Introduction

To offer effective interaction facilities, any advanced system must rely on an—implicit or explicit—method for allocating initiative to the interactants. Modeling mixed initiative (MI) interaction requires that the system apply strategies and mechanisms for deciding *when to take/pass initiative* depending on the current dialogue situation and *how to adapt its behavior* accordingly. Hence, the system uses knowledge representation and reasoning techniques to plan its own strategy and co-operate with the user. MI also means that the roles of the agents are not predetermined and that the agents must be enabled to negotiate the control of initiative (Allen, 1994). The agents not only interact to solve domain problems, but also collaborate at the level of "interaction management" (Bunt, 1996) and negotiate the problem-solving strategy, e.g., to build up "SharedPlans" (Grosz & Sidner, 1990; Sidner, 1994).

Issues of initiative and dialogue control are related to some basic issues of *intelligent human-computer collaboration* (cf. Terveen, 1995). Research relevant to the topic concentrates on AI-oriented approaches to collaborative activity/planning, discourse modeling/planning, and adaptive user modeling. Focusing on the agents' beliefs, goals, and intentions, most computational models of collaborative discourse have been developed for natural language applications, for example, text planning systems for advisory/explanation dialogues (e.g., Moore & Paris, 1993; Haller, 1996) and other task-oriented systems such as many spoken dialogue systems (see e.g., Dalsgaard *et al.*, 1995 for a collection). In addition to goals and plans, some existing computational dialogue models explicitly take initiative and other social factors such as conversational conventions, expectations and obligations into account (e.g., Traum & Allen, 1994; McRoy & Hirst, 1995; Jokinen, 1996).

This paper presents a theoretically motivated approach to the management of mixed-initiative dialogues in *intelligent information retrieval systems*. Most of the above mentioned approaches and systems presuppose well-defined (planning) tasks, e.g., transportation planning or repair of technical devices. Information retrieval in large databases is a highly interactive—and possibly very complex—task, but the information needs of users of such systems are usually less well defined and users often opportunistically change their goals and strategies during interaction. Hence, system support should be situation and context dependent without relying on a strict plan–goal hierarchy.

State-of-the-art retrieval systems provide many user-support mechanisms, such as query construction aids or access to thesauri, but these are mostly treated as isolated extra-dialogic functions or tools offered to the user. It has been suggested (e.g., Belkin *et al.*, 1995;

* Jon Atle Gulla has been involved in the research reported here during his employment at GMD-IPSI.

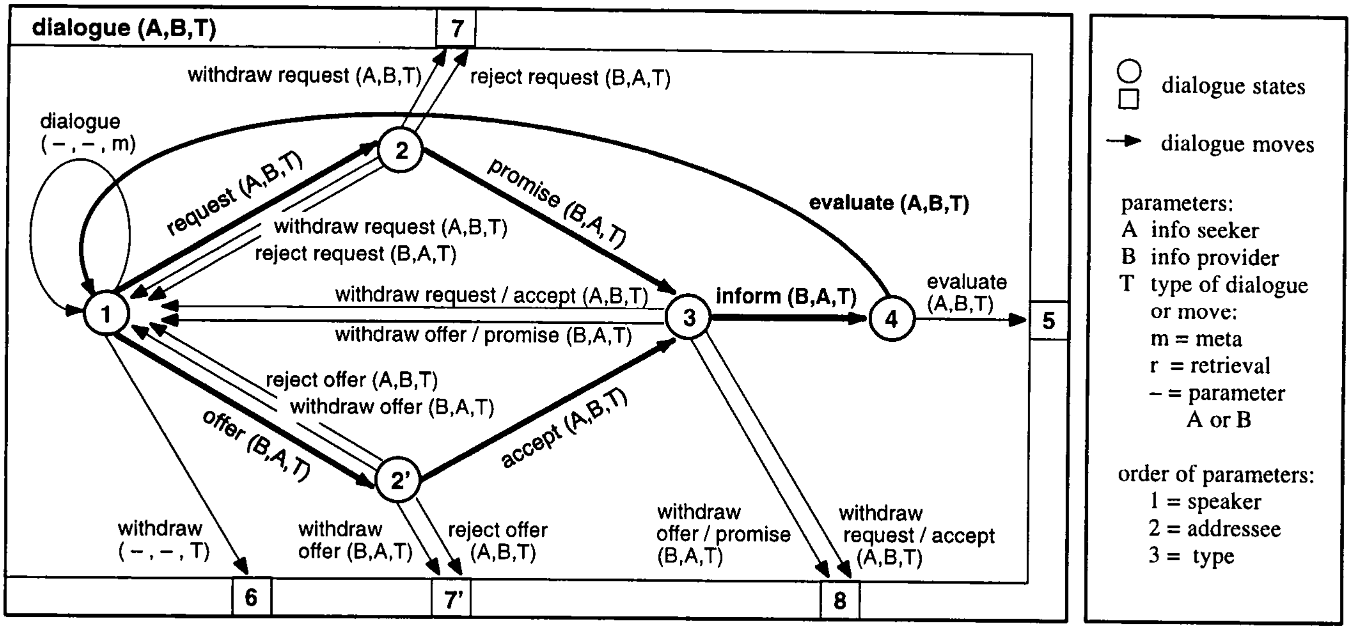

Figure 1: Top level of COR

Stein & Maier, 1995) that the *entire* retrieval interaction should be represented as a complex web of "conversational acts" based on a multi-layered model of dialogue tactics and strategies. Our current prototype retrieval system MIRACLE mainly supports graphical interaction with a few natural language extensions. However, all of the graphical and linguistic actions of user and system are seen as conversational dialogue acts and represented in a complex dialogue history.

MIRACLE (Multimedia Information RetrievAl of Concepts in a Logical Environment) integrates three active components (see Stein *et al.*, 1997): (1) a retrieval engine which uses abductive reasoning to generate reformulations (or expansions) of a given ambiguous user query and offers the reformulations to the users as possible interpretations of their information need; (2) a probabilistic multimedia indexer, called MAGIC; and (3) the Abductive Dialogue Component (ADC) introduced in the remainder of this paper. When an ambiguous user input that departs from the expected course of action is recognized, the dialogue component takes the initiative negotiating hypothetical explanations of the act with the user.

In the rest of the paper we first outline our theoretical framework for modeling dialogues and then discuss how this model is employed by the ADC to control and plan changes in initiative between user and retrieval system. The conclusions of the paper sum up the lessons learned and point at some open issues for future research.

Modeling Information-Seeking Dialogue

A good dialogue component offers both *user guidance* and *dialogue flexibility.* Drawing on knowledge of the user's intentions or goals, our component recommends acts to the user that guide her in the interaction with the system and helps her find the desired information. Since user intentions are often vague and unstable, we have an additional mechanism that allows the user to overrule the recommendations made and take the initiative herself. To achieve this combination of flexibility and guidance, we make use of the conversational model COR and a battery of dialogue scripts.

- COR describes the interaction options available at any state of an exchange allowing for *symmetric changes of initiative* between the agents.
- A dialogue script implements a goal-directed information-seeking strategy and offers the *recommended dialogue acts* at the various stages of a retrieval session.

Conversational Roles model (COR)

Focusing on social and illocutionary aspects (Sitter & Stein, 1996) COR models information-seeking dialogue as a co-operative negotiation based on general conversational rules and expectations, similar to Winograd and Flores' "Conversation for Action" model (1986). The two agents enter commitments and expect that pending commitments of the other agent are met, or—if this turns out not to be possible—are retracted in a way that can be understood and accepted by both partners.

Figure 1 displays the top level of the COR model as a recursive transition network (RTN) where the transitions are generic *dialogue moves*, which are also structured and represented as RTNs (for details on the structure of moves see Sitter & Stein, 1996; Stein *et al.*, 1997). Moves consist of the respective atomic *dialogue act* (e.g., a request in the request move) and optional transitions such as *clarification subdialogues.* Note that even the entire move may be omitted in certain situations— hence, remain implicit (e.g., promise and evaluate are often skipped in retrieval dialogues).

A COR dialogue cycle is either initiated by the information seeker (A) with a request (A wants B to do some action, e.g., search the database and present the information) or by the information provider (B) with an offer (B intends to do some action, e.g., search and later give the information). Thus, within these *initiative* moves the conditions of action—and topic of the whole dialogue cycle—are formulated, whereas *responding* moves adopt these conditions (positive: promise and accept; negative: rejections and withdrawals). Inform and evaluate are responding moves in that they neither define nor adopt such conditions of action. In retrieval dialogues request, offer and inform clearly have a *task-oriented function,* as they change the semantic context, whereas the other moves have *dialogue control functions* (Bunt, 1996) such as positive/negative feedback or interaction management.

Applying the COR model allows us to represent the interchange of moves and acts in a structured history. All future possible actions in any state are also accounted for, hence, we achieve much of the flexibility needed. However, COR does not provide us with the means to decide which of the possible sequences (or paths) through the dialogue should be preferred in a particular dialogue context. We therefore employ dialogue scripts to define such paths and to provide the necessary user guidance.

Dialogue Scripts

A dialogue script is an executable representation of an information-seeking strategy (cf. Belkin *et al.*, 1995). Such a strategy describes a particular way of using an information retrieval system to do a specific task and shows the sequences of function calls involved in larger tasks and activities. There are usually many strategies available for a specified task, e.g., finding an item either by browsing or specifying some search terms, and the user has to choose an appropriate one at the beginning of her retrieval session.

The dialogue script describes all possible system actions and all recommended user actions at the various stages of the retrieval process. A recursive transition network formalism with preconditions and postconditions is used to represent the scripts, and the transitions contain references to dialogue acts found in the COR model. The preconditions decide when an act is available, whereas the posconditions make sure that the necessary actions are executed by the system. A script may also call other sub-scripts to deal with smaller tasks.

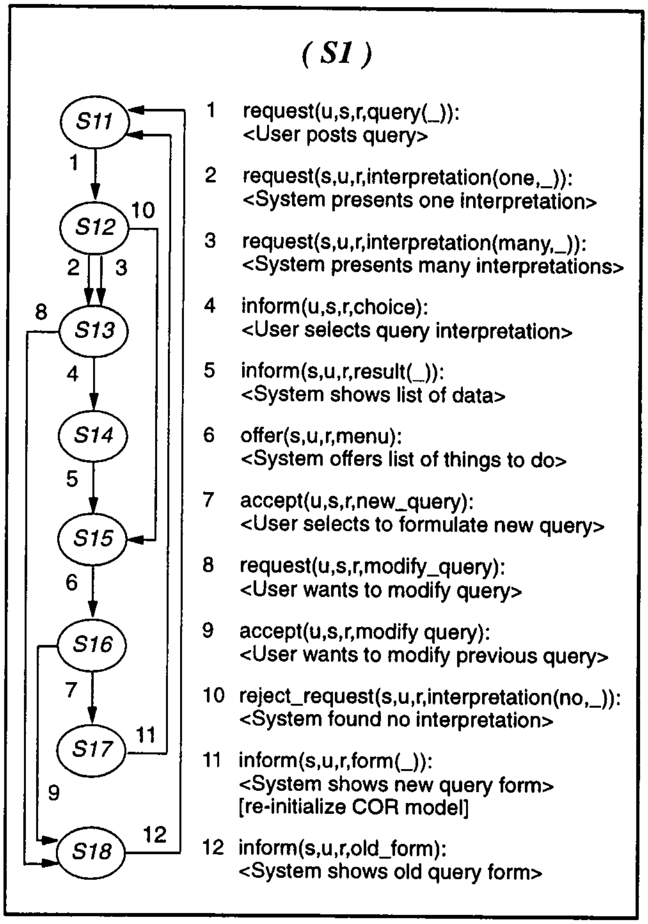

Figure 2: Structure of script *S1*

Script *S1* in Figure 2 helps the user retrieve information from the MIRACLE information retrieval system (see (Stein *et al.*, 1997) for examples). The user first fills out a query form that is analyzed and interpreted by the system. Note that even though abduction is used also for interpreting queries, this abduction process has nothing to do with the abductive dialogue component (ADC) described in this paper. ADC deals with ambiguous *dialogue control acts*, whereas another component of the system—the retrieval engine—takes care of ambiguous search terms/queries.

If the user query can be given different interpretations, the system initiates a subdialogue to decide on the correct one (transition (2) and (3)). After inspecting the interpretations, the user can either choose to modify the query form (transition (8)) or use one of the presented interpretations when accessing the underlying database (transition (4)). If (4) is preferred by the user, the system retrieves the data from the database and afterwards tells her what she can do next. The user can either continue with her previous query (transition (9)) or start all over again with an empty query form (transition (7)). The postcondition `re-initialize COR model` in transition (11) deletes the dialogue history and activates the COR model's starting state if the user chooses to formulate a completely new query.

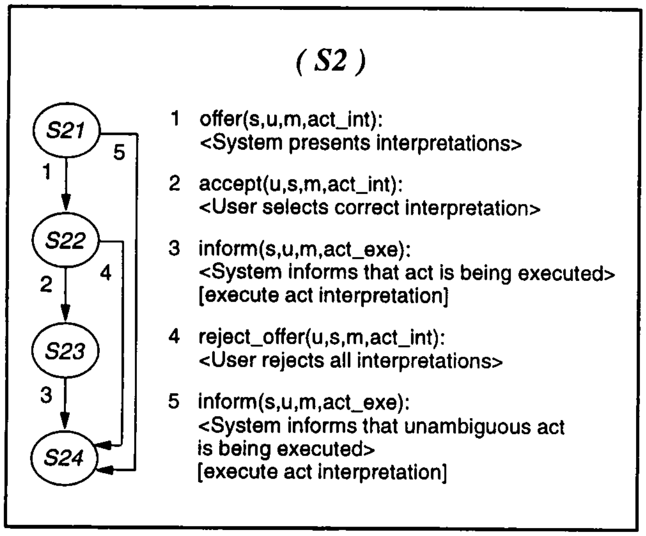

Figure 3: Structure of script *S2*

Another simple script is *S2* in Figure 3, which is triggered when the user has selected an ambiguous function from the user interface (e.g., an 'undo'). The system presents possible interpretations of the user's act in terms of concrete actions to be executed. If the user does not like any of the interpretations presented to her, she can go directly to the end state of the script and her ambiguous act will be ignored by the system.

Abductive Dialogue Component (ADC)

At the beginning of a dialogue, the Abductive Dialogue Component initializes the COR model and a selected script. As the dialogue develops, transitions in the script are fired, and their associated COR acts are used to change the state of the COR model. By collecting all possible transitions leading out from the active state of the script, ADC can present a list of recommended dialogue acts to the user. The COR model describes the general conversational state of the dialogue and gradually builds up a structured history of dialogue acts performed by system and user. As long as the user follows the recommendations in the script, the transitions in the script decide who has the initiative in the dialogue.

However, the user is not forced to choose among the acts included in the script. She can suspend the script and take the initiative herself by selecting a dialogue control act directly from the COR model. Since these acts are not directly concerned with the retrieval task, they have to be interpreted in light of the dialogue context. The 'withdraw' acts in the COR model, for example, are rarely included in any script, but can be selected by the user to change or cancel previous acts.

Acts not included in the script are referred to as *unexpected acts* and are analyzed by ADC using abduction and the dialogue history.

Abductive Reasoning

In an abductive system there is a conclusion available, and the task of the system is to find potential facts that would explain this conclusion. The conclusion comes in the form of an observation, whereas the potential facts form a hypothesis that—if true—logically implies the observation. A domain theory defines the concepts found in the observation and establishes logical relationships with other concepts. Concepts that are not defined by the theory are referred to as abducibles, and these can be included in the hypothesis explaining the observation.

In ADC the observation is the unexpected act of the user, and the rules of the domain theory are called *dialogue control rules*. The general idea is to offer the user a simple, but general set of context-dependent dialogue options taken from the COR model. These options are indetermined when isolated, but can be given an interpretation in a particular context. The context is here the dialogue history, and the dialogue control rules map from concrete system actions and properties of the dialogue history to general COR acts like 'withdraw' and 'quit'. When an unexpected act is encountered, we look for a hypothesis that together with the history and the dialogue control rules imply the unexpected act:

$$History \cup Dialogue\_rules \cup Hypothesis \Rightarrow Unexpected\_act$$

The hypothesis is a set of concrete actions that—if the user really wanted to do these actions—explains why the user chose that particular unexpected act

(1) $request(X,Y,Z,W) \wedge change\_act(request(X,Y,Z,W) \rightarrow withdraw(request(X,Y,Z,\_))$
[If the user has posted a request and would like to change this request, she hits 'withdraw'.]

(2) $suspend\_script \wedge no\_results \rightarrow withdraw(request(u,s,r,\_))$
[If the user wants to suspend the script and has no results, she withdraws her request.]

(3) $redo(X) \rightarrow change\_act(X)$
[If the user redoes an act, she changes the original act.]

(4) $extend(request(X,Y,Z,query(Q))) \rightarrow change\_act(request(X,Y,Z,query(Q)))$
[If the user extends a previous query, she changes the original query.]

(5) $\neg inform(X,Y,Z,interpretation(\_,\_)) \wedge \neg inform(X,Y,Z,result(\_)) \rightarrow no\_results$
[If no interpretations of queries or database items have been presented, there are no results.]

Table 1: Some dialogue control rules

in this context. In most cases, there are several hypotheses that imply the unexpected act from the user. Each hypothesis forms an *interpretation of the act* (cf. Hobbs *et al.*, 1993), and the user has to decide afterwards which one she finds closest to her intentions.

As seen in rules (1) and (2) in Table 1, for example, a 'withdraw' act can be interpreted as a withdrawal of the previous query or as discontentment with the active script. Table 1 shows 5 of the dialogue control rules defined so far in the system. The abducibles here are concrete actions that might have been intended when the user submitted the COR withdraw act (in Table 1 the predicates *redo/1, extend/1* and *no_results/1* are abducibles).

Interpreting Unexpected User Acts

Consider a situation, in which script *S1* is used to retrieve some information from the target database. The user fills out the query form provided, and the system tries to map the query terms onto the structures of the underlying domain database. The user suddenly realizes that she has made some mistake (e.g., used a wrong search term) and interrupts the system's search by hitting the 'withdraw' button. As can be seen seen from Figure 2, there is no 'withdraw' act recommended in state S12 of the script. The act is made available by the general COR model and allows the user to get out of the guidance offered by script *S1*.

Now, at this point the dialogue history only contains the act requesting some information from the database (for simplicity, we do not show the content of the $query/1$ predicate):

History = {request(u,s,r,query(_))}.

The final $withdraw(request(u,s,r,query(\_)))$ is the unexpected user act—the observation—which is to be analyzed by the abduction engine. The dialogue component uses the dialogue control rules in the reverse direction to find a possible proof for the user's 'withdraw'. A proof tree is complete when all concepts that are not deduced, are either abducibles or confirmed by the dialogue history. Using the rules in Table 1 above, three proof trees are constructed (see Figure 4).

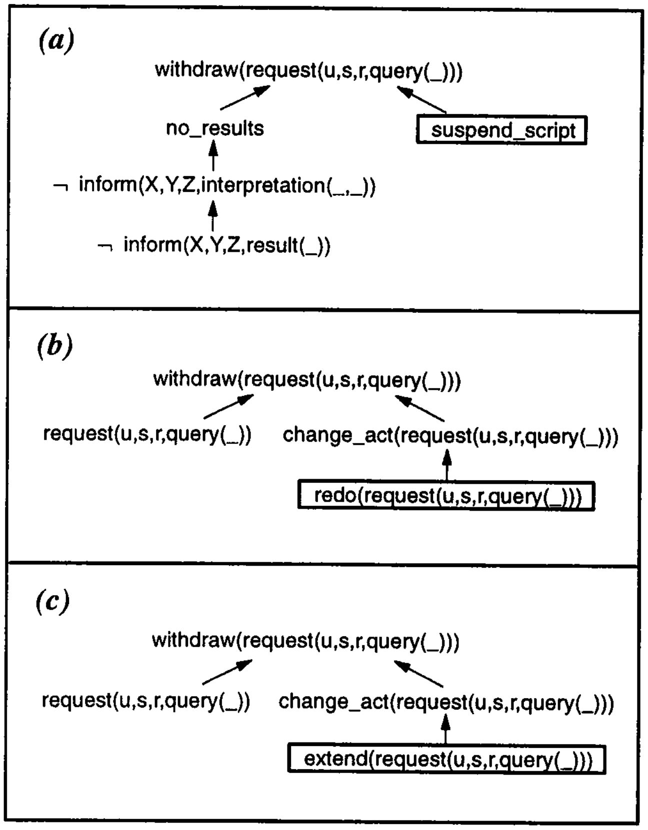

Figure 4: Three interpretations of the 'withdraw' act

These three proof trees correspond to the following interpretations presented to the user:

(a) *Forget the query and discard the script*
(b) *Fill out a completely new query*
(c) *Extend the previous query*

The system has taken the initiative back and asks the user which interpretation she intended. Script S2 is used to negotiate with the user, who can either choose one of the interpretations or go back to the original script again. If the user meant to discard the script, she chooses (a) above and the system initiates a new dialogue based on an introductory script.

Of course, most dialogues are far more complex than the dialogue discussed here. If the user has already retrieved some data, for example, the abduction engine also has to reason about what to do with these results. And if a retrieval session included many interconnected sub-queries, it has to analyze whether the user might need these query terms later on. The good thing about this approach, though, is that it is all handled by a firm logical system that works just as well for complex dialogues as for more simple ones.

Evaluation and Conclusions

ADC is implemented in SWI-Prolog using an abduction engine from Flach (1994). It is developed in the context of the MIRACLE information retrieval system, whose retrieval engine also uses abduction to interpret ambiguous user queries. MIRACLE is available on World Wide Web and offers access to multimedia documents using a basically graphical interface with menus and buttons. The underlying retrieval mechanisms of the system either come from the INQUERY retrieval system (Callan, Croft, & Harding, 1992) or from MIRACLE's own multimedia indexing component MAGIC (Müller & Kutschekmanesch, 1996).

A fundamental problem in ADC is to formulate good dialogue control rules that have the necessary generality. The rules depend on the functionality of the retrieval system, but are also related to previous dialogue acts and retrieval tasks. It might also be useful to refer to extra-dialogic user properties in the rules, so that the system can interpret things differently depending on who the user is (e.g., considering her background knowledge, domain interests, or preferences for certain retrieval strategies).

Another problem is the treatment of very complex dialogue histories. In longer retrieval sessions, these histories grow rather large and not all of the acts in the history are relevant for the interpretation of later unexpected acts. Specifically, there are cycles of acts in the history that are more or less independent of what is happening afterwards and should not be included in later abduction processes.

Still, our experiments so far with the dialogue component are promising. Whereas the dialogue scripts are used for user guidance, the COR model together with some dialogue control rules make it possible for the user to take the initiative and change the direction of the dialogue. Vice versa, the dialogue system may take the initiative interpreting unexpected or ambiguous user acts and telling the user how these acts may relate to concrete dialogue strategies and functions available in the retrieval system.

References

Allen, J. F. 1994. Mixed initiative planning: Position paper. Paper presented at ARPA/Rome Labs Planning Initiative Workshop. Also available from http://www.cs.rochester.edu/research/trains/mip/home.html.

Belkin, N. J.; Cool, C.; Stein, A.; and Thiel, U. 1995. Cases, scripts, and information seeking strategies: On the design of interactive information retrieval systems. *Expert Systems and Application* 9(3):379–395.

Bunt, H. C. 1996. Interaction management functions and context representation requirements. In LuperFoy, S.; Nijholt, A.; and van Zanten, G., eds., *Dialogue Management in Natural Language Systems. Proceedings of the Eleventh Twente Workshop on Language Technology*, 187–198. Enschede, NL: Universiteit Twente.

Callan, J. P.; Croft, W. B.; and Harding, S. M. 1992. The INQUERY retrieval system. In *Proceedings of the 3rd International Conference on Database and Expert Systems Application.* Berlin and New York: Springer-Verlag. 78–83.

Dalsgaard, P.; Larsen, L. B.; Boves, L.; and Thomsen, I., eds. 1995. *Proceedings of the ESCA Workshop on Spoken Dialogue Systems—Theories and Applications (ETRW '95), Vigsø, Denmark.* Aalborg, Denmark: ESCA/Aalborg University.

Flach, P. 1994. *Simply Logical: Intelligent Reasoning by Example.* Chichester: John Wiley & Sons.

Grosz, B. J., and Sidner, C. L. 1990. Plans for discourse. In Cohen, P. R.; Morgan, J.; and Pollack, M. E., eds., *Intentions in Communication.* Cambridge, MA: MIT Press. 417–444.

Haller, S. 1996. Planning text about plans interactively. *International Journal of Expert Systems. Special Issue on Knowledge Representation and Inference for Natural Language Processing* 9(1).

Hobbs, J. R.; Stickel, M. E.; Appelt, D. E.; and Martin, P. 1993. Interpretation as abduction. *Artificial Intelligence* 63(1-2):69–142.

Jokinen, K. 1996. Goal formulation based on communicative principles. In *Proceedings of the International Conference on Computational Linguistics (COLING '96)*, 598–603.

McRoy, S. W., and Hirst, G. 1995. The repair of speech act misunderstandings by abductive inference. *Computational Linguistics* 21(4):435–478.

Moore, J. D., and Paris, C. L. 1993. Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics* 19(4):651–694.

Müller, A., and Kutschekmanesch, S. 1996. Using abductive inference and dynamic indexing to retrieve multimedia SGML documents. In Ruthven, I., ed., *MIRO 95. Proceedings of the Final Workshop on Multimedia Information Retrieval (MIRO '95).* Berlin and New York: Springer-Verlag (eWiC, electronic Workshops in Computing series).

Sidner, C. L. 1994. An artificial discourse language for collaborative negotiation. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI '94).* Menlo Park, CA: AAAI Press/The MIT Press. 814–819.

Sitter, S., and Stein, A. 1996. Modeling information-seeking dialogues: The conversational roles (COR) model. *RIS: Review of Information Science* 1(1, Pilot Issue). Online Journal. Available from http://www.inf-wiss.uni-konstanz.de/RIS/.

Stein, A., and Maier, E. A. 1995. Structuring collaborative information-seeking dialogues. *Knowledge-Based Systems. Special Issue on Human-Computer Collaboration* 8(2-3):82–93.

Stein, A.; Gulla, J. A.; Müller, A.; and Thiel, U. 1997. Conversational interaction for semantic access to multimedia information. In Maybury, M. T., ed., *Intelligent Multimedia Information Retrieval.* Menlo Park, CA: AAAI/The MIT Press. chapter 20. (in press).

Terveen, L. G. 1995. Overview of human-computer collaboration. *Knowledge-Based Systems. Special Issue on Human-Computer Collaboration* 8(2-3):67–81.

Traum, D. R., and Allen, J. F. 1994. Discourse obligations in dialogue processing. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL '94)*, 1–8.

Winograd, T., and Flores, F. 1986. *Understanding Computers and Cognition.* Norwood, NJ: Ablex.