Enhancing collaboration with conditional responses in information-seeking dialogues*

Ivana Kruijff-Korbayová and **Elena Karagjosova**
University of the Saarland, Saarbrücken, Germany
{korbay,elka}@coli.uni-sb.de

Staffan Larsson
Göteborg University, Sweden
sl@ling.gu.se

Abstract

We aim at improving the collaborative character of the responses in the GoDiS dialogue system participating in information-seeking dialogues. In this paper we describe how the system currently deals with successful and unsuccessful database search and show the need for generating more collaborative responses. We consider conditional yes/no responses as one kind of collaborative responses and discuss issues related to their implementation.

1 Introduction

Our goal is to improve the collaborative character of the responses of a dialogue system participating in information-seeking dialogues (ISDs). We employ the information-state approach to dialogue developed in the TRINDI and SIRIDUS projects (Cooper et al., 1999; Lewin et al., 2000). Within this framework, we propose an implementation of collaborative conditional responses as answers to yes/no-questions. Such responses are illustrated in (2).

(1) Not if you want to fly economy class.

(2) Yes, if you can fly business class.

In (1), the negative response is contingent on flying economy class, whereas in (2) the positive response is contingent on flying business class.

It has been argued that task-oriented dialogues are natural and efficient when they are collaborative (Chu-Carroll and Brown, 1997; Rich et al., 2000). Collaboration means that both the system and the user are contributing to solving the task at hand. ISDs are a particular kind of task-oriented dialogues. For instance in the travel domain, the task is to determine a set of parameters of a possible journey with respect to a database to which only the system has access. However, a typical ISD system does not enable any collaboration in determining the journey parameters: The system does not provide the user with any indications of what journeys are (still) available in the database. The user has to "blindly" specify her desires, and equally blindly revise them, refining (if they are under-constraining) or relaxing them (if they are over-constraining).

In the travel domain, it is not a viable option for the system to guide the user in the initial specification of journey parameters. It is impossible to enumerate all available options for individual parameters since the number of potential journeys in the database is typically large. On the other hand, once the search space is restricted by setting some initial set of parameters, the system could be collaborative in the subsequent phases. Our proposal mainly concerns collaboration by providing responses that help the user to revise or refine the initial parameters.

We show that one useful way of accomplishing collaboration in the parameter revision and refinement phase is by providing a conditional response (CR): a positive or negative response clarifying the condition(s) under which this response holds. It is the making of this condition explicit that makes the response collaborative. For example, a negative CR can be collaborative by mentioning such parameter(s) in the condition, whose relaxation could result in a positive response instead (*economy class* in (1)). A positive CR can be collaborative by mentioning such parameter(s) that are necessary for preserving the positive response (*business class* in (2)).

The paper is structured as follows. In Section 2 we describe the dialogues GoDiS can participate in, and show the need for collaborative responses after both successful and unsuccessful database search. Section 3 discusses the meaning of CRs also taking into account the context in which they occur. Section 4 concerns the implementation. We briefly describe the information-state update approach in

* Work on this paper was supported by SIRIDUS (Specification, Interaction and Reconfiguration in Dialogue Understanding Systems), EC Project IST-1999-10516.

GoDiS and then discuss the implementation of CRs in terms of the conditions under which they can be selected and the effects they have on the information state. We also describe the underlying processing of the database search results. In Section 5 we review some related work on generating cooperative responses in over-constrained situations. Finally, in Section 6 we indicate some issues for future work.

2 Information-seeking dialogue in GoDiS

Our work is set in GoDiS (Gothenburg Dialogue System). We use the information state (IS) approach to dialogue (Larsson and Traum, 2000; Larsson, 2002a) (cf. Section 4.1 for details).

GoDiS has been initially designed to handle information-seeking dialogues in the travel domain, such as (3). It collects from the user a set of parameters constraining a journey, and performs a database search with these constraints. If the search succeeds, GoDiS returns the price of the journey (3e). Otherwise, it indicates the search has failed (3f).

(3)
a. S: Welcome to the travel agency!
b. U: The price for a flight from Malmö to Paris on the first of April please.
c. S: What class did you have in mind?
d. U: Economy.
e. S: The price is 7654 crowns.
f. S': Sorry, there is nothing matching your request about price.

Normally the system returns the value of one parameter, e.g., *price*. But when the user does not specify all the search parameters expected by the system, as in (4c), multiple answers and the corresponding parameter values are returned, as in (4d).

(4)
a. U: I want to fly from Malmö to Paris on the first of April.
b. S: What class did you have in mind?
c. U: It doesn't matter.
d. S: The price is 7654 crowns. Business class. The price is 456 crowns. Economy class.

The result of a database search is currently one of the following: (i) the proposition *fail(q)* indicating that no answer to the question was found (realized as in (3f)), (ii) a proposition specifying a unique value of a requested parameter (realized as in (3e)), (iii) a set of alternative answers (realized as in (4d)).

However, a dialogue like (3) often does not stop at the result of the database search, be it negative or positive. The user may revise or refine some parameter(s) and initiate a new search. For example, after (3f), the user may continue by changing the departure day (5a). Then, after (5b), the user may continue by trying to further constrain the search by specifying an additional parameter, for example the airline as in (5c). However, such continuations can become dull as the user is trying to find out which combinations of parameters succeed, as in (5c)-(5f).

(5)
a. U: Can I fly on the second?
b. S: Yes.
c. U: Can I fly with Ryanair?
d. S: Sorry, there is nothing matching your request.
e. U: What about Lufthansa?
f. S: Sorry, there is nothing matching your request.

We propose that the system should be collaborative and help the user find a satisfiable set of parameters. This involves the system being able to indicate a parameter to relax upon failed database search, or to indicate a parameter to keep in cases where some hits are found but they are too many to enumerate, and thus the search criteria need to be refined.

There are of course various ways to realize this collaborative system behavior. In this paper, we concentrate on the realization by generating conditional responses, such as those in (6b), (6c) and (7b).

(6)
a. U: Can I fly on the second?
b. S: Not if you want to fly economy class.
c. S': Yes, if you can fly business class.

(7)
a. U: Can I fly on the second?
b. S: Yes, if you can fly with SAS.

In (6a) (as a continuation of (3f)), the user changes the departure day to April 2^{nd}. In (6b) the system not only gives a negative answer, but also indicates that the failure of the database search with the changed parameter is conditional on the parameter *economy class* which the user has specified earlier. In an alternative response (6c) in this context, the system suggests *economy class* as an alternative for which the database query would be successful.

The result of the database search, and therefore the answer to a user's question, can also be contingent on a parameter the user has not specified. In

Figure 1: Patterns of conditional responses

	Negative CR	*Positive* CR
Question	$?p$	$?p$
Response	Not if c	Yes if c
Assertion	If c, then not-p	If c, then p
Implicature	If not-c, then p	If not-c, then not-p

this case, too, it makes sense to indicate this contingency to the user: in (7b) the system gives a positive answer to the question and also indicates that the database search is successful (and thus the answer to the question is positive) as long as an additional parameter, namely the SAS airline, is assumed.

CRs are briefly described in the next section.

3 Conditional responses

The CRs we currently consider have the form *Not if c / Yes if c*. As an answer to a yes/no question $?p$, a CR not only answers the question (positively or negatively), but it also implicates that if c does not hold, the answer would have the opposite polarity.[1] For example, (6b) suggests that the negative answer concerns only the case where the parameter *class* is set to *economy* whereas for a different value of this parameter the answer may be positive. Figure 1 summarizes the patterns of CRs.

Conditional responses are discussed in (Green and Carberry, 1999), and characterized in terms of the speaker's motivation to provide information "about conditions that could affect the veracity of the response". However, they consider only the cases in which the speaker does not know whether the condition holds, while utterances in which the condition is already specified are left unnoticed.

In (Karagjosova and Kruijff-Korbayová, 2002b; Karagjosova and Kruijff-Korbayová, 2002a), we proposed an analysis distinguishing two types of CRs with respect to the contextual status the parameter in the condition, on which the CR is contingent:[2] (i) CRs contingent on a contextually-determined parameter (**CDCR**s), as in (6b, 6c). (ii) CRs contingent on a contextually non-determined parameter (**NDCR**s), as in (7b). We also argued that this distinction is important for the *dialogue move* that such responses perform: A CDCR makes a proposal for revising the contextually determined parameter, and a NDCR raises the question whether the parameter holds. In fact, CRs perform multiple dialogue acts: they are a response (backward-looking function) and a question or a proposal (forward-looking function) at the same time. Both these functions of CRs are reflected in the implementation we are developing.

[1]Also corpus examples reported in (Karagjosova and Kruijff-Korbayová, 2002b; Karagjosova and Kruijff-Korbayová, 2002a) justify this treatment which distinguish berween the asssertion and implicature of the CR.

[2]This distinction is also supported by data from the Verbmobil appointment-scheduling corpus and the SRI American Express (AMEX) travel agency data (cf. (Karagjosova and Kruijff-Korbayová, 2002a)).

4 Generating CRs in GoDiS

In this section we describe our implementation of CRs as *answer moves* in GoDiS. Dialogue moves in GoDiS are modelled in terms of information state *update rules*. Specifying such rules involves defining preconditions for selecting a particular move (*move selection*) and the effects of the move on the information state (*move integration*). In § 4.1 we describe the GoDiS information state. In § 4.2 and § 4.3 we discuss the selection and integration of a CR as an answer move, respectively.

4.1 GoDiS information state

GoDiS is an experimental dialogue system built using the TrindiKit (Larsson et al., 2000) and using the Information State (IS) approach to dialogue modeling (Larsson and Traum, 2000; Larsson, 2002a). The type of record assumed for the GoDiS IS is a version of Ginzburg's Dialogue Game Board, (Ginzburg, 1996), and is shown in Figure 2.

The IS is divided into a PRIVATE and a SHARED part, the latter containing information that the system assumes to be shared by the user. The SHARED part contains information about the latest utterance (speaker and move(s)), the shared commitments (a set of propositions) and the stack of questions under discussion (QUD, (Ginzburg, 1996)).

In the PRIVATE part, the plan contains the system's long-term goals (a list of actions), whereas the agenda contains more immediate actions. For example, the action to ask the user where she wants to go, is represented on the agenda as $findout(\lambda x.dest(x))$, which will then be converted into the move $ask(\lambda x.dest(x))$.

Current versions of GoDiS use keyword and keyphrase spotting, and a simplified semantics. A user's utterance *I'd like to go to London* will be recognized as a move giving a destination through the word 'to' followed by a city, and its contents will be represented in the shared commitments as the

Figure 2: The GoDiS Information State

$$
\begin{bmatrix}
\textsc{private} & : & \begin{bmatrix} \textsc{agenda} & : & \textsc{Stack(Action)} \\ \textsc{plan} & : & \textsc{StackSet(Action)} \\ \textsc{bel} & : & \textsc{Set(Proposition)} \end{bmatrix} \\
\textsc{shared} & : & \begin{bmatrix} \textsc{com} & : & \textsc{Set(Proposition)} \\ \textsc{qud} & : & \textsc{Stack(Question)} \\ \textsc{lu} & : & \begin{bmatrix} \textsc{speaker} & : & \textsc{Participant} \\ \textsc{moves} & : & \textsc{AssocSet(Move,Bool)} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

Figure 3: CR selection algorithm

Given a set of search parameters $P = \{p_1, \ldots, p_n\}$, where each $p_i = \alpha_i(v_i)$ is a proposition specifying the value v_i of an attribute α_i.
Given a (possibly empty) set of database search solutions $\textsc{sol} = \{S_1, \ldots, S_m\}\ s.t.\ \forall S_i \in \textsc{sol} : P \subseteq S_i$.

If responding to a yes/no question q which specifies search parameters $Q = \{q_1, \ldots, q_k\}$

- **if** $\textsc{sol} = \emptyset$ (database search failed)
  then
  - **if** $\exists p_i \in P/Q$ s.t. $\exists \textsc{sol}' = \{S_1, \ldots, S_k\}\ s.t.\ \forall S_i \in \textsc{sol}' : P/p_i \subset S_i\}$ (some parameter identified as responsible for search failure; if this parameter is relaxed, search succeeds)
  - **then** answer q with a conditional negative response with condition p_i
  - **else** answer q with an unconditional negative response
- **else** (database search succeeded)
  - **if** $\forall S_i \in \textsc{sol} : S_i/P \neq \emptyset$ (there are other user-unspecified parameters in search result)
  - **then**
    - **if** the size of $\textsc{sol}$ is less than Max (the results are enumerable)
    - **then** answer q by enumerating $\textsc{sol}$
    - **else** (the results cannot be enumerated)
      - **if** $\exists p_j\ s.t.\ \forall S_i \in \textsc{sol} : p_j \in S_i/P$ (all results share some parameter p_j)
      - **then** answer q with a conditional positive response with condition p_j
  - **else** answer q with a positive response

predicate-logic proposition $dest(london)$. The corresponding question, where does the user want to go, will be represented on the QUD as $?\lambda x.dest(x)$.

4.2 CR Selection

The selection rules specify the conditions for selecting particular dialogue moves by the system. The conditions relevant for selecting a CR are specified in the selection algorithm in Figure 3.

The first step we make towards enhancing collaboration with CRs is to enable them under certain circumstances as answers to yes/no questions. This requires extending the treatment of user's questions in the GoDiS system for the travel agency domain beyond dealing only with questions concerning flight prices or visa requirements of various destination countries. We need to add new domain plans for dealing with other questions. Since questions like (5a) concern the availability of flights, we deal with them by defining a domain plan which consists of searching the database for flights satisfying a set of parameters. The parameters used for the search consist of those provided in the question plus other parameters provided ealier (if any), retrieved from the shared-commitments part of the information state.[3]

If the question revises a parameter that has already been specified earlier, the parameters provided in the question itself take precedence; for example in answering (8e) the search needs to be for a flight on April 2^{nd} even though the user earlier specified the departure day as April 1^{st}.

(8)
a. S: Welcome to the travel agency!
b. U: The price for an economy flight from Frankfurt to Paris on April first please.
c. S: Sorry, there is nothing matching your request about price.
d. S′: The price is 200 Euro.
e. U: Can I fly on the second?
f. S: Not if you want to fly economy class.
g. U: Can I fly from Luxembourg?

[3] In a more cautious version of the plan, we should also ensure that the search only be triggered if some minimal amount of information about the flight has already been provided by the user (e.g., the departure or destination city).

In GoDiS, the parameters specified in a follow-up question replace the earlier specified parameters. (This is implemented as downdate of the previous parameter and an immediate update with the new one.) For example, before (8e), the IS contains $dep\_day(first)$, and after them it contains $dep\_day(second)$.

It is an open question whether such immediate revision of the parameter is warranted. (8f) is an example where the immediate revision could lead to trouble: Intutively, what the user is asking about in (8g) are economy flights from Luxembourg to Paris on April 1^{st}. However, the immediate revision strategy would lead to a search for economy flights from Luxembourg to Paris on April 2^{nd}. This arises as follows: In (8b), the user specifies the parameters as $dept\_city(frankfurt)$ & $dest\_city(paris)$ & $month(april)$ & $dept\_day(first)$. No matter whether the search result is negative (8c) or positive (8d), the user may want to explore other possibilities by asking (8e). If the system uses the immediate revision strategy, the departure day parameter will now be set to $dept\_day(first)$, and remain that also when (8g) is being interpreted, which is wrong.

The revision should therefore be at least conditioned upon success of the database search and possibly also upon the user's subsequent acknowledgement. Another solution is to treat such follow-up questions as introducing alternatives. Presently we leave this is an issue for future work.

If the system is indeed responding to a yes/no question, the next decision is based on the result of the database search, and on the subsequent processing of the results. Our version of the database search retrieves a (possibly empty) set of records that satisfy the search criteria. This differs from the previous versions of GoDiS, where the search retrieves the value of one parameter, e.g., *price*.

Currently we are dealing with two of the possible four cases of CRs: (i) a negative CDCR instead of a plain negative response, as a collaborative recovery from a failed database search; (ii) a positive NDCR instead of a plain positive response, as a collaborative continuation after a successful database search. We discuss the respective portions of the selection algorithm below.

Negative CDCR. When the database search with a set of user-specified parameters fails, the system attempts to collaborate by suggesting which (if any) parameter the user might relax to get a successful search instead. To find out, the system performs additional database searches with the parameters relaxed one at a time. If the relaxation of some parameter leads to a successful search the system produces a negative NDCR contingent on this parameter.

Figure 4: A toy database

dep	**dest**	**month**	**dep_day**	**class**	**airline**
London	Paris	April	2^{nd}	busin	BA
London	Paris	April	3^{rd}	econ	BA
Malmö	Paris	April	2^{nd}	econ	SAS
Malmö	Paris	April	2^{nd}	busin	SAS

For illustration, consider (6a) and the database records in figure 4. A search with the parameter set (9) fails. However, a search with the modified parameter set (10), where the *class* parameter is relaxed, succeeds, so (6b) can be generated, indicating that the negative answer depends on *class*.

(9) $\{dept\_city(malmoe), dest\_city(paris), month(april), dept\_day(second), class(econ)\}$

(10) $\{dept\_city(malmoe), dest\_city(paris), month(april), dept\_day(second)\}$

At the moment, we leave the order in which the parameters are tested arbitrary. Moreover, we currently only model the relaxation of one parameter p_j, but it is conceivable to look for a combination of parameters. More sophisticated techiques for conflict resolution are proposed for example in (Qu and Beale, 1999). It is therefore one possible strand of future work for us to incorporate such techniques.

We do not allow a NDCR contingent on parameters mentioned in the question, because of the oddity of answering a question such as (11a) with (11b).[4]

(11) a. S: Can I fly from Malmö to Paris on April 1^{st}?
b. U: Not if you want to fly on April 1^{st}.

Positive NDCR. Our system is collaborative also when the database search with the user-specified parameters succeeds, but there are more results than can be sensibly conveyed to the user at once.[5] In this case, the system attempts to use a CR to indicate when the success of the search depends on any

[4] As an anonymous reviewer pointed out, other follow-up responses are possible, such as: *You cannot fly on April* 1^{st}. *But April* 2^{nd} *would be possible.* We are aware of this, but leave the use of other forms for future work.

[5] The maximum number of results conveyed at once depends on the system's output modality: a graphical interface enables to output more than spoken mode or text mode on a small display. GoDiS uses either spoken or textual mode.

Figure 5: Effect of CRs on information state

	-CDCR *Not if c*	*+CDCR* *Yes if c'*	*-NDCR* *Not if c*	*+NDCR* *Yes if c'*
IS before:				
QUD	$?p$	$?p$	$?p$	$?p$
Shared	c	c'		
IS after:				
Shared	$\{c \wedge \neg p, c' \wedge p\}$	$\{c' \wedge p, c \wedge \neg p\}$	$\{c \wedge \neg p, c' \wedge p\}$	$\{c' \wedge p, c \wedge \neg p\}$
QUD	$?c$	$?c'$	$?c$	$?c'$

paramater as yet unspecified by the user. This helps to avoid future failed search as in (5c)-(5f).

This part of the algorithm relies on the database search returning all the records that satisfy the search criteria. We implement a simple subsequent processing which determines whether all results have any other parameter(s) in common. When this is the case, a positive NDCR can be generated.

For illustration, consider (7a) again. Given the database in Figure 4, a search with the parameter set (12) returns more than one hit. They all share the parameter $airline(sas)$, so (7b) can be generated, indicating this parameter as one on which the positive answer is contingent.

(12) $\{dept\_city(london), dest\_city(paris), month(april)\}$

4.3 CR integration

The integration rules specify the effects of particular dialogue moves on the IS. The way we model the various effects of a CR on the information state is shown schematically in Figure 5.[6]

In GoDiS, the question $?p$ that a CR can be used to answer is represented as the question under discussion (QUD, (Ginzburg, 1996)). The answer provided by a CR is added to the shared commitments. The propositional content of a CR is a conditional. Since GoDiS does not support implication, the proper encoding of CRs as conditionals requires an extension of the semantic representation used in GoDiS. Presently, we use an approximation instead: We define the propositional content of a CR as a combination of its assertion and its implicature, and encode it as a set of two alternatives (e.g. $\{c \wedge \neg p, \neg c \wedge p\}$ for negative CDCR). This approximation is made possible by the fact that we only alow the generation of a CR in situations where the implicature is known to hold, and thus the conditional can be turned into a bi-conditional.

In addition to answering the question $?p$, the effect of a CR is that it puts a question corresponding to the condition on QUD. The effects of CDCRs and NDCRs on the IS differ, because they occur in different contexts: A NDCR contingent upon c as an answer to question $?p$ has the effect of raising the question whether this condition c, which had not been previously mentioned, should hold. In contrast, asserting a CDCR contingent upon c as an answer to question $?p$ cannot simply raise the question whether the condition c should hold, because c has been already determined and is part of the shared commitments. A CDCR therefore has the effect of *re*raising the question whether c should hold (Larsson, 2002a). This is consistent with the claim that CDCRs and NDCRs perform different dialogue moves (Section 3).

One issue that needs to be considered in relation to this is whether a CDCR involves removing the previously established proposition c (downdate). We believe that the examples (13c) and (13d) as alternative continuations of (13a) (as a continuation of (3f) above) show that the previously established proposition should not be downdated as a direct result of the CR, because the proposed revision can still be either accepted or rejected by the user.

(13) a. U: Can I fly on the second?
b. S: Not if you want to fly economy class.
c. U: Ok, I'll fly business class.
d. U': What about the third?

In (13a), the user asks about economy flights from Malmö to Paris on April 1^{st}, and the system gives a negative CDCR suggesting the parameter $class(economy)$ as responsible for the failed database search. In (13c), the user accepts the alternative of flying in business class. In this case, the proposition $class(economy)$ can be deleted from the shared commitments, and replaced by $class(business)$ as a result of (13c).

(13d), on the other hand, should be interpreted as asking about economy flights from Malmö to Paris on April 3^{rd}, indicating (in an indirect way) that the user does not want to revise the parameter $class(economy)$, but rather tries the parameter $dept\_day(third)$ instead of $dept\_day(second)$. In this case it would have been wrong to remove the proposition $class(economy)$ from the shared commitments as a result of the CDCR.

[6]The signs '-' and '+' abbreviate "positive" and "negative". c and c' are contextual alternatives.

Therefore the established proposition c should be kept in the shared commitments, while the alternative c' proposed in the CDCR is being considered. This is a modification of the way reraising is handled in GoDiS that we have to consider in more detail in future work.

5 Related work

Some of the problems we are dealing with in providing alternative suggestions to the user in information-seeking dialogues in over-constrained sitations, have been addressed in work concerned with conflict resolution (Qu and Beale, 1999), (Chu-Carroll and Carberry, 1994) and others (see (Qu and Beale, 1999) for further references).[7]

(Qu and Beale, 1999) propose a constraint-based model for cooperative response generation aiming at detecting and resolving situations in which the user's information needs have been over-constrained. In contrast to earlier work using heuristics for identifying relaxation candidates (e.g., based on constraint weights, (Abella et al., 1996; Pieraccini et al., 1997)), (Qu and Beale, 1999) employ AI techniques like constraint satisfaction, solution synthesis and constraint hierarchy.

(Chu-Carroll and Carberry, 1994)'s work on detecting invalid beliefs or plans and suggesting alternative solutions by relaxing over-constrained queries and proposing relaxation modification is also related to the work presented in this paper.

In comparison, our technique to identify a relaxation candidate is much simpler. We are currently exploring the possibility of employing the more elaborate techniques proposed in (Qu and Beale, 1999) also in the GoDiS system.

In contrast to the works cited above, we also deal with under-constrained situations where a positive CR indicates additional parameters in order to prevent failed future search. The only other work addressing also this issue we are aware of is described in (Hochberg et al., 2002), but their paper not provide much details in this respect.

6 Conclusions and future work

We proposed to improve the collaborative character of the responses of the GoDiS dialogue system participating in information-seeking dialogues. We described the way the system currently deals with successful and unsucessful database search and argued for the need to generate more collaborative responses that indicate sensible further search options to the user. We showed that CRs are suitable for this purpose, because they provide a positive or negative response contingent on a particular parameter. We then defined CRs in terms of the conditions on selecting them and their effect on the dialogue context.

We are currently implementing our proposal in the GoDiS system. We have preliminary versions of the update rules for CRs which are subsequently tested for improvement. We have implemented parts of the selection algorithm and a preliminary version of the database search and the subsequent processing of the results. Besides completing the implementation, we are presently considering several future extensions of our work.

First of all, we intend to implement a more sophisticated version of the further processing of the database search results by employing smarter techniques for identifying relaxation candidates and additional parameters.

Another extension concerns the integration rule for CDCRs. Currently, we are implementing the effect of a CDCR on the IS as reraising a question. However, reraising the question whether a proposition should hold can be seen as opening a negotiation whether the already specified answer to that question should be preserved or revised (because it has already once been determined). It would therefore be natural to provide an account of CDCRs as proposals opening negotiation. We plan to do this using the issue-based account of collaborative negotiative dialogues proposed in (Larsson, 2002b), which is based on (Sidner, 1994).

Another possible extension is to consider uses of CRs in other cases than as answers to yes/no-questions. That is, CRs can be used as answers to wh-questions, as in (14), or as grounding confirmations, as in (15).[8]

(14) a. U: Which day can I fly?
b. S: Monday, if you take business class.

(15) a. S: There is a flight from Philadelphia to Frankfurt on Sunday.
b. U: So I would arrive on Monday.

[7] An anonymous reviewer noted that the general issue of negotiation strategies after failed database search is presently being addressed by various teams developing commercial dialogue systems, e.g., the Soliloquy system. Another such system we have recently seen demonstrated is developed at IBM (Hochberg et al., 2002).

[8] In (Karagjosova and Kruijff-Korbayová, 2002a) we provide a corpus example illustrating this use of CRs.

c. S: Not if you leave in the morning.
d. S': Yes, if you leave in the evening.

Furthermore, CRs can also be realized by surface forms other than the ones we have considered so far (for example with *unless*). We plan to integrate different forms into our analysis and implementation.

Finally, collaborative responses in over- and under-constrained situation can be achieved in other ways than by using CRs, and it would be intereseting to investigate how the different ways relate and differ, in order to be able to capture systematic choices among them.

References

A. Abella, M.K. Brown, and B. Buntschuh. 1996. Development principles for dialogue-based interfaces. In *Proceedings of ECAI'96 Workshop on Dialogue Processing in Spoken Language Systems*, pages 1–7.

Jennifer Chu-Carroll and Michael K. Brown. 1997. An evidential model for tracking initiative in collaborative dialogue interactions. *Meeting of the Association for Computational Linguistics*, pages 262–270.

Jennifer Chu-Carroll and Sandra Carberry. 1994. A plan-based model for response generation in collaborative task-oriented dialogues. In *Proceedings of AAAI-94*, pages 799–805.

Robin Cooper, Staffan Larsson, Colin Matheson, Massimo Poesio, and David Traum. 1999. Coding Instructional Dialogue for Information States. http://www.ling.gu.se/projekt/trindi/.

Jonathan Ginzburg. 1996. Interrogatives: Questions, Facts and Dialogue. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*, pages 385–422. Blackwell, Oxford, UK/Cambridge, USA.

Nancy Green and Sandra Carberry. 1999. A Computational Mechanism for Initiative in Answer Generation. *User Modeling and User-Adapted Interaction*, 9(1/2):93–132.

Judith Hochberg, Nanda Kambhatla, and Salim Roukos. 2002. A flexible framework for developing mixed-initiative dialogue systems. In *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*, Philadelphia, PA, USA. Universtity of Pennsylvania.

Elena Karagjosova and Ivana Kruijff-Korbayová. 2002a. An Analysis of Conditional Responses in Dialogue. In *Proceedings of the 5th International Conference on TEXT, SPEECH and DIALOGUE*, Brno, Czech Republic.

Elena Karagjosova and Ivana Kruijff-Korbayová. 2002b. Conditional Responses in Information Seeking Dialogues. In *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*, Philadelphia, PA, USA. University of Pennsylvania.

Staffan Larsson and R. Traum, David. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. http://www.ling.gu.se/publications/GPCL.html.

Staffan Larsson, Alexander Berman, Johan Bos, Leif Grönqvist, Peter Ljunglöf, and David Traum. 2000. Trindikit 2.0 manual. Technical Report Deliverable D5.3 - Manual, Trindi.

Staffan Larsson. 2002a. *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University.

Staffan Larsson. 2002b. Issues under negotiation. In *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*, Philadelphia, PA, USA. Universtity of Pennsylvania.

Ian Lewin, C.J.Rupp, Jim Hieronymus, David Milward, Staffan Larsson, and Alexander Berman. 2000. Siridus System Architecture and Interface Report (Baseline). http://www.ling.gu.se/projekt/siridus/.

R. Pieraccini, E. Levin, and W. Eckert. 1997. AMICA: The AT&T mixed initiative conversational arcitecture. In *Proceedings of EUROSPEECH*.

Yan Qu and Steve Beale. 1999. A constraint-based model for cooperative response generation in information dialogues. In *Proceedings of AAAI-99*, Orlando, FL.

Charles Rich, Candance L. Sidner, and Neal Lesh. 2000. COLLAGEN: Applying collaborative discourse theory to human-computer interaction. Technical Report TR-2000-38, Mitsubishi Electric Research Laboratories.

Candance Sidner. 1994. Negotiation in collaborative activity: A discourse analysis. *Knoweldge-based systems*, 7(4):265–267.