A Cooperative Dialogue Game for Resolving Ontological Discrepancies

Robbert-Jan Beun and Rogier M. van Eijk

Institute of Information and Computing Sciences
Utrecht University, P.O. Box 80.089
3508 TB Utrecht, The Netherlands
{rj,rogier}@cs.uu.nl

Abstract. The goal of this paper is to present a computational framework that enables us to generate elementary speech act sequences in a dialogue between an electronic assistant and a computer user. Since naive users of complex systems often do not think and communicate in terms of domain characteristics, we will concentrate on the conversational process of the understanding of the meaning of a vocabulary shared by two dialogue participants. In order to give meaning to their vocabulary, agents need to translate terms into their private domain ontologies. We consider a dialogue game in which agents produce speech acts or 'moves' to transfer relevant information with respect to a particular agreement about the meaning of the words in the vocabulary. Describing the properties and the dynamics of the cognitive states or cognitive constructs in relation to the various dialogue contributions is an essential part of this work. In particular, we address the following basic questions: What type of cognitive constructs should be included to model the dialogue's basic structural properties? How do the various dialogue contributions change the existing cognitive constructs? How do these changes influence the generation of new contributions?

1 Introduction

When we interact with computers, we often want them to be endowed with characteristics that closely mimic human communication. One of these characteristics is the ability of humans to react in a cooperative manner to the communicative actions of the dialogue partner. In everyday conversation, people effortlessly answer questions, accept or deny assertions, confirm the receipt of a message and provide relevant feedback in case of communication problems. Since the cognitive and communicative abilities of humans are so well adapted to the real-time processing of these various interaction structures, we expect that including natural conversational skills in interfaces may contribute to a more efficient and satisfactory human-computer interaction.

It was only twenty years ago that interaction with computers was for the most part only possible through symbols that could be understood exclusively by expert users. Today we can hardly imagine that the interface once did not include the graphical apparatus of icons, buttons, pictures and diagrams that we have become so accustomed to. Clearly, the visual interactive qualities of interfaces have improved a lot, but they are still unable to generate the basic communication structures in a similarly powerful and

F. Dignum (Ed.): ACL 2003, LNAI 2922, pp. 349–363, 2004.

cooperative way as we find in human-human communication. Today's commercially available systems hardly ever answer questions in a proper way, are unable to argue about particular information and rarely provide relevant or even truthful feedback in case of communication errors.

An important reason for this shortcoming is the lack of fundamental knowledge about the basic concepts and the theoretical principles that drive a conversation. The goal of this paper is to present some of these theoretical principles and a computational framework that enables us to generate elementary speech act sequences in a dialogue between an electronic assistant and a computer user. Since naive users of complex systems often do not think and communicate in terms of domain characteristics, we will concentrate on the conversational process of the understanding of the meaning of a vocabulary shared by two dialogue participants. Users of cars, for instance, often speak in terms of 'safety' or 'comfort', while domain characteristics are, for instance, expressed in terms of 'power brakes', 'presence of airbags' and 'suspension system'. In order to give meaning to the user's vocabulary, his or her vocabulary has to be translated into the domain ontology. In the framework discussed in this paper, we will try to show how parts of the translation process can be simulated in a computational dialogue framework.

In our approach, two electronic agents play a dialogue game (see also [7]) in which speech acts or 'moves' are produced to transfer relevant information with respect to a particular agreement about the meaning of the words in the vocabulary. We will distance ourselves from the idea that conversation can be modelled by a concatenation of speech acts regulated by a set of sequencing rules or a grammar (see also [14] and [9]). In line with [5], agents and their behaviour are modelled, respectively, by cognitive states in terms of various types of beliefs and the rules that generate adequate speech acts and that determine the change of the cognitive states as a result of a particular speech act.

In what follows, a dialogue game and its underlying communication model will be described that enable us to generate cooperative speech act sequences. A particular instance of the model will be chosen in which the agent that simulates the user's behaviour - the user-agent - has no access to the outside world and only receives information based on the exchange of conversational units. On the other hand, the agent that simulates the computer system - the computer-agent - may receive information by both conversational exchanges and domain observations. Describing the properties and the dynamics of the cognitive states or cognitive constructs in relation to the various dialogue contributions is an essential part of this work. In order to develop such a framework, the following questions will be addressed: What type of *cognitive constructs* should be included to model the dialogue's basic structural properties (see [17])? How do the various dialogue contributions *change* the existing cognitive constructs (see e.g. [8] and [6])? How do these changes influence the *generation* of new contributions?

2 Natural Dialogue

In its basic form, a dialogue can be conceived as a linear alternating sequence of symbolic elements between two participants [12]. The various contributions in the dialogue have a meaning and a purpose, i.e. there is a relation between the symbolic elements and particular cognitive constructs that result from the interpretation process of the dialogue contributions, and the sender intends to accomplish through them a particular effect on the cognitive state of the addressee. In general the utterances do not form independent segments of speech, but show a coherent structure of conversational units like words in a single sentence. A criterion for the acceptability of a dialogue is usually hard to give and heavily depends on its contextual characteristics such as the goals and knowledge of the dialogue participants.

In our case the main goal will be determined by an initial question asked by the user-agent. Given a limited number of belief constructs, a restricted number of initial states can be distinguished from which the dialogue may start. For instance, the user-agent by mistake assumes that the meaning of the predicate is a shared belief by both partners, but the computer-agent has no knowledge about the meaning. Or worse, both dialogue partners initially assume a different shared meaning of the predicate, but the discrepancy remains unnoticed. The latter case appears probably quite often in dialogue and may cause serious communication problems.

Before we start the description of the dialogue game, we will first discuss some examples. In all example dialogues the user-agent U asks the first question to the computer-agent C whether the predicate 'safe' is applicable to a particular car.

Dialogue 1:
U: Is it a safe car?
C: Yes, it is.
U: OK

Dialogue 2:
U: Is it a safe car?
C: Yes, it has air bags and a good crash test.
U: OK

In these two cases both participants know the term 'safe' and may believe that they have shared knowledge about its meaning, although discrepancies may exist between the two belief states. In dialogue 2, the reason for the extra information given by C may be that he is aware of possible misunderstandings and therefore the meaning of the term is verified by explicitly stating the definition in the response.

In Dialogue 3 neither of the two participants accepts the other as an expert on the meaning of the term and neither contributes extra information on which a decision may be forced, therefore the dialogue ends in an infinite loop of disagreement.

Dialogue 3:
U: Is it a safe car?
C: Yes, it has air bags and a good crash test.
U: To my opinion, a safe car also has automatic screen wipers.
C: To my opinion, a safe car does not need automatic screen wipers.
U: Well, it does.
C: No, it doesn't. . . .

Clearly, the property of infinity has to be avoided in the framework. C's strategy could be, for instance, to accept temporarily the definition stated by U and drop it after the answer has been given (Dialogue 4). This implies that C has to make a distinction between his own private beliefs and his beliefs about U's private beliefs.

Dialogue 4:
U: Is it a safe car?
C: Yes, it has air bags and a good crash test.
U: To my opinion, a safe car also has automatic screen wipers.
C: It is my opinion that a safe car does not need automatic screen wipers, but this car has them, so in your terms, it would be safe.
U: OK

Agent C also has various strategies in the second turn. If a common understanding is doubted, he may simply ask for the meaning without manifesting his own interpretation.

Dialogue 5:
U: Is it a safe car?
C: What do you mean by 'safe'?
U: A safe car is a car that has air bags and a good crash test.
C: In that case, it is safe.

A problem in Dialogue 5 is that, depending on C's tenacity, U may incorrectly come to believe that U's interpretation of the term is a shared belief and the dialogue suggests that C has no interpretation. Not giving an interpretation in these cases will be considered as a violation of the Gricean quantity maxim, which roughly states: 'Say enough, but not too much' [10]. A possible solution is given in Dialogue 6 where C immediately provides the extra information after U has manifested her translation of the term:

Dialogue 6:
U: Is it a safe car?
C: What do you mean by safe?
U: To my opinion, a safe car is a car that has air bags and a good crash test.
C: Well, I think that a safe car also has automatic screen wipers.
U: I don't care whether the car has automatic screen wipers.
C: In that case, it is safe.

In Dialogue 6, C gives another cue by stressing the pronomen 'you', which already implicates that there may be other translations as well, in particular C's translation. In the framework below, we will avoid the extra turns and in case of doubt of a shared translation, C will explicitly state his translation of the term.

A rather bizarre case is given in Dialogue 7 where both participants do not know a translation of term.

Dialogue 7
U: Is it a safe car?
C: What do you mean by safe?
U: I don't know what it means.
C: In that case I cannot say whether it is a safe car.

In practice we do not expect these cases, since it implies that U does not really know what she is talking about. As we will see, however, the dialogue rules enable C to give the responses that are provided in this dialogue.

From the previous discussion we conclude that there may be four basic initial settings:

a. both agents have the same interpretation and so agree about the meaning of the term (Dialogue 1 and 2),
b. both agents have different interpretations (Dialogue 3, 4 and 6),
c. only one of the agents has an interpretation (Dialogue 5),
d. none of the agents has an interpretation (Dialogue 7).

So, in general, in response to the question "$P(c)$?" by U, where P refers to the term that has to be translated and which is applicable to the object c, C has three basic moves in the second turn:

a. C may simply answer the question (e.g. "Yes, it is", "No, it is not")
b. C may give information about his translation (e.g. "Do you mean that it has?", "No, because it has no ...")
c. C may simply ask a counter-question (e.g. "What do you mean by 'safe'?")

Note that, depending on C's knowledge state, the initial setting in b. may lead to different responses by C. For instance, suppose that C incorrectly believes that a particular interpretation is shared, then he may simply answer the question without providing extra information. As a result, a possible deviation in U's and C's interpretation will not be manifested in the dialogue. If, on the other hand, C does not believe that U shares the information, C may add extra information about his interpretation of the term. In conclusion, the content of C's turn not only depends on his knowledge about the answer to the question, but also on his knowledge about his dialogue partner.

Another important observation is that C's response in b. enables the participants to initiate an *argumentation* about the interpretation of the term, since his interpretation or a part of the interpretation is manifested in the dialogue and U may notice inconsistencies with her own interpretation. Examples were already given in Dialogue 3 and Dialogue 6. Below we will assume, however, that the computer-agent gives priority to the translation of the user-agent.

3 The Basic Model

The dialogue framework presented in this paper is based on a simple model employed in human-computer interaction [13, 1, 15]. Underlying this model is the recognition that humans interact naturally with their environment in two ways: symbolically and physically. On the one hand, if there is an intermediary interpreter, humans can interact symbolically and use language to give commands, ask for or provide information, etcetera. On the other hand, physically, one manipulates objects, for instance, by moving or fastening them, or observes them by seeing, feeling, hearing, tasting or smelling. The essential difference between the two types of interaction is that actions of the first type (e.g. illocutionary acts and their semantic content [4, 16]) need an interpreter who can bridge the gap between the symbols and their actual meaning and purpose, while actions of the second type are related in a more direct manner to human perception and action.

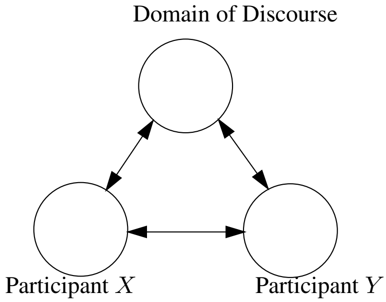

Fig. 1. The triangle metaphor

In parallel with the distinction symbolical vs. physical, humans engaged in dialogues can perform two types of *external* actions: a. *communicative actions* intended to cause some cognitive effect in the recipient, and b. *non-communicative* actions to observe or change particular properties of the domain. Obviously, the two types of action can be considerably interrelated (c.f. [11, 2]). In addition, we will include an action type that is neither communicative nor external, namely *inference* – i.e. the process of adjusting the cognitive states of participants solely based on their previous states. In short, the basic model includes perception, action, communication and thinking in an extremely rudimentary form.

The distinctive interaction channels are represented in the so-called triangle metaphor (Figure 1), where the corners represent the domain of discourse (or the external world) and the two participants, and the arrows the flow of information between the participants themselves and between the participants and the domain. The external actions can be expressed in terms of the flow of information between the corners of the triangle. A communicative act performed by participant X towards participant Y is a flow of information from X to Y; observation of the domain is a flow of information from the domain towards the observer and an action carried out in the domain is a flow of information from the actor to the domain. Below, the term *communication* will be used exclusively in reference to an information flow between the participants; *interaction* will be conceived in a broader sense and includes flows of information between the participants and the domain.

In practice, the communication channel between the two participants may cause messages to be delayed (as in letters) or disturbed by, for instance, noise. Also, the channel can be duplex, where both participants can speak at a time, or half-duplex, where only one participant can speak at the time. Here, we will consider the channel between the participants and between the participants and the domain of discourse as an ideal half-duplex channel, which means that no information is delayed or lost during transfer and that information can flow only in one direction at a time. Time is unimportant, but the order of communicative and non-communicative acts is important, since the acts change the cognitive states of the dialogue agents.

In the model, we clearly distinguish between the world and knowledge (or beliefs) about the world. The world is represented by a set of concrete objects (cars, clothes, buildings, . . .) that have particular characteristics (colour, weight, . . .) with a particular value (green, red, heavy, ...). Also, the objects may have particular relations between them (next, heavier, ...). The knowledge about the world is a *representation* of the objects with their characteristics and their relations. In the latter case we will also use the term *ontology*. In general, the ontology abstracts the essence of the domain of discourse and helps to catalogue and distinguish various types of objects in the domain, their properties and relationships. To indicate the subjective nature of the agents' information state, we will often use the term 'belief' instead of knowledge.

Agents may assign characteristics to certain aspects of the world that are not directly perceivable. A particular agent may find the colour red ugly, but ugliness is not a direct perceivable feature or does not even have to exist in the actual world. Red will therefore be called a *perceivable* feature, while ugliness represents a *non-perceivable* feature. We will assume that non-perceivable features are always based on or expressible in one or more perceivable features. So, ugliness may be a combination of, for instance, the features red and big.

4 A Cooperative Dialogue Game

In the dialogue game described in this paper, we will make some simplifications with respect to the model described in the previous section. Firstly, we assume that the user-agent U of the question has no access to the domain of discourse and that her partner C can observe the perceivable features whenever necessary. This implies that C is always able to answer the question whenever a translation of the term is available. Secondly, it will be assumed that both agents know the meaning of perceivable predicates; in other words, the predicates are part of the agents' shared beliefs, have the same interpretation for both agents and, consequently, the meaning of perceivable predicates is never part of the discussion. Thirdly, it will be assumed that the communication channel is ideal, i.e. the partner will always receive correctly the information sent by one of the participants. Also, observations made by C will always be in agreement with the actual state of the world.

Facts about the world will be described as one-place predicates, such as $Red(car1)$ or $Big(car2)$. It will be assumed that all facts in the world are static and perceivable. The first assumption implies that facts in the world do not change during the interaction and therefore do not become false. The second assumption implies that all facts may become part of the belief state of the computer-agent.

We will distinguish the following sets:

1. Ω, i.e. a set of perceivable features in the world common to both U and C,
2. Ψ, i.e. a set of non-perceivable features, and
3. $\Psi \leftrightarrow \Omega$, i.e. a set of translations from non-perceivable features to perceivable features.

Since non-perceivable features and their translations are subjective by their very nature, we will add the agent's cognitive state in the description. The agents' cognitive state consists of the following constructs:

– Private information of an agent m about the domain of discourse (B_m)
– Private information of an agent m about what the other agent believes ($B_m B_n$)

- Private information of an agent m about shared beliefs of both agents m and n ($B_m MB$)
- A pending stack that contains in order of appearance the speech acts that have to be processed.

Furthermore, we will introduce some abbreviations:

- Below we use capital letters for predicates $P, Q, R, \ldots$ and small letters for the accompanying propositions, $p, q, r, \ldots$ For instance, P applied to a particular object c, notated as $P(c)$, is abbreviated to the proposition p.
- $P(c) \leftrightarrow Q(c)$: The translation of the non-perceivable predicate P is Q, where Q is a single perceivable predicate or a conjunction of perceivable predicates. Below we will also write $p \leftrightarrow q$ in case of propositions.
- $B_m z$: The proposition or translation z is part of the private beliefs of agent m; if z is a proposition, the predicate Z can either be a perceivable predicate or a non-perceivable predicate.
- $B_m MBz$: The proposition or translation z is part of the shared beliefs of agent m; if z is a proposition, the predicate Z can either be a perceivable predicate or a non-perceivable predicate.
- $\neg\exists q B_m(p \leftrightarrow q)$: Agent m has no translation of the non-perceivable predicate P in his private belief .
- $\neg\exists q B_m MB(p \leftrightarrow q)$: Agent m has no translation of the non-perceivable predicate P in his shared belief.

Note that an agent's cognitive state not only contains propositions, but also translations, which can be considered as a special type of propositions.

We assume that the agents can reason about beliefs by Modus Ponens and that the following dependencies exist between the cognitive constructs of an agent:

$$(R1)\ B_m MBz \rightarrow B_m z \ \&\ B_m B_n z$$

Hence, if a proposition or a translation is part of the mutual belief of agent m, it is also part of the private belief and of the beliefs about the other. It is important to note that the opposite does not hold.

In R2 it is expressed that the user-agent takes over the perceivable propositions of the computer-agent.

$$(R2)\ B_m B_n p \ \&\ p \in \Omega \rightarrow B_m p,$$

provided that m denotes the user-agent.

In fact, the rule establishes particular roles in the dialogue, since the computer-agent is considered as an expert on perceivable propositions. A problem could be that the information state of the user-agent becomes inconsistent. This has to be prevented by an update function, which will not be discussed in this paper.

5 Rules of the Dialogue

Moves are fully determined by the cognitive state of the participant who performs the move and the rules that are applicable to this state. A double arrow '$\Rightarrow$' links the preconditions of the move to the move itself. The left side of the arrow is of type proposition and represents the preconditions in terms of the cognitive state of an agent; the right side is of type action and represents the generated move. We will use the expression TOP_m to indicate the speech act that is on top of the stack of agent m. The following speech acts are part of the framework.

We let m denote the performing agent and z and z' range over (negated) propositions and translations. Agent m tells that z holds: $tell(m, z)$, it asks whether z holds: $tell(m, z?)$, it tells that is does not know whether z holds: $tell(m, z*)$, it tells that z holds under condition that z' holds: $tell(m, z \mid z')$ and it tells that it stops the dialogue: $stop(m)$.

The precise meaning of the speech acts is determined by the generation and update rules given below.

We will assume that the initial question by the user-agent has been asked in the first turn. For reasons of legibility, we will describe the rules in the order of the various turns. It should be stressed, however, that the applicability of the rules depends on the preconditions of a particular move and is not determined by the turn. So, the variables m and n below may refer to the user-agent or the computer-agent.

5.1 The Second Turn

After the user-agent has asked the initial question, the computer-agent has three possibilities to continue the dialogue:

1. The computer-agent knows that a translation is shared by his partner and so gives the answer.
2. The computer-agent has a translation, but he believes that A does not share it.
3. The computer-agent has no translation.

Generation rules G1a and G1b express that if m believes that if p has previously been asked by the partner n - i.e. n's question is on top of the m's stack - and m believes that there is a shared translation of p and m does not believe that the partner has a different translation and m believes p (G1a) or m believes not p (G1b), then an answer will be provided without extra information:

$$(G1a)\ TOP_m = tell(n, p?)\ \&\ \exists q B_m MB(p \leftrightarrow q)\ \&\ B_m p$$

$$\Rightarrow tell(m, p)$$

$$(G1b)\ TOP_m = tell(n, p?)\ \&\ \exists q B_m MB(p \leftrightarrow q)\ \&$$

$$B_m(\neg p) \Rightarrow tell(m, \neg p)$$

Note that we do not consider the case where m does not know the answer, since we have assumed that m was always able to find an answer to the question as long as the proposition is build up from perceivable predicates and a translation is available.

In the rules G2a and G2b, m does not believe that his partner shares the meaning of the term, but a translation is available. As a result, a conditional answer is generated and extra information about the translation is added:

$$(G2a)\ TOP_m = tell(n, p?)\ \&\ \neg\exists q(B_m MB(p \leftrightarrow q)\vee$$
$$B_m B_n(p \leftrightarrow q))\ \&\ B_m(p \leftrightarrow r)\ \&\ B_m(p)$$
$$\Rightarrow tell(m, p \mid p \leftrightarrow r)$$
$$(G2b)\ TOP_m = tell(n, p?)\ \&\ \neg\exists q(B_m MB(p \leftrightarrow q)\vee$$
$$B_m B_n(p \leftrightarrow q))\ \&\ B_m(p \leftrightarrow r)\ \&\ B_m(\neg p)$$
$$\Rightarrow tell(m, \neg p \mid p \leftrightarrow r)$$

In both rules we have added the extra precondition that m may have no belief about the belief of the other with reference to the translation ($\neg\exists q B_m B_n(p \leftrightarrow q)$). In those cases, m should give priority to n's belief (G2c and G2d; see also the fourth turn):

$$(G2c)\ TOP_m = tell(n, p?)\ \&\ B_m B_n(p \leftrightarrow q))\ \&\ B_m(q)$$
$$\Rightarrow tell(m, p \mid p \leftrightarrow q)$$
$$(G2d)\ TOP_m = tell(n, p?)\ \&\ B_m B_n(p \leftrightarrow q))\ \&\ B_m(\neg q)$$
$$\Rightarrow tell(m, \neg p \mid p \leftrightarrow q)$$

Rule G3 expresses that if no translation is available, m asks for a translation to her partner:

$$(G3)\ TOP_m = tell(n, p?)\ \&\ \neg\exists q B_m(p \leftrightarrow q)$$
$$\Rightarrow tell(m, p \leftrightarrow r?)$$

In rule G3 the question by m refers to the perceivable predicates, not the whole translation. In natural language this can be expressed by a WH-question, indicating that the variable r has to be instantiated (e.g. 'What is the translation of p?').

5.2 The Third Turn

So far we have described the rules that regulate the second turn in the dialogue. In the third turn, the response of the computer-agent is on top of the stack of the user-agent. Depending on this response and the cognitive state of the user-agent, the user-agent has four possible reactions:

4. the computer-agent's response may be accepted,
5. the response may be rejected and a translation may be provided,
6. the user-agent may indicate that he has a translation available or
7. the computer-agent may indicate that he does not have a translation.

As we already discussed, the last of these cases is rather bizarre.

In G4a and G4b, n accepts the statement by m that p or that $\neg p$ as long as there is no proof for the contrary and stops the dialogue.

$$(G4a)\ TOP_n = tell(m, p)\ \&\ \neg B_n(\neg p) \Rightarrow stop(n)$$

$$(G4b)\ TOP_n = tell(m, \neg p)\ \&\ \neg B_n(p) \Rightarrow stop(n)$$

In G4c, n accepts a translation if there is no other translation available, and therefore also accepts the truth value of p:

$$(G4c)\ TOP_n = tell(m, p \mid p \leftrightarrow q)\ \&\ \neg\exists r B_n(p \leftrightarrow r\ \&$$

$$r \neq q) \Rightarrow stop(n)$$

In G5 the translation is rejected because n has found a translation that does not correspond to his own translation. In a rejection, the agent tells the grounds for his rejection ($p \leftrightarrow r$), so that m has knowledge about the reason of the discrepancy.

$$(G5)\ TOP_n = tell(m, p \mid p \leftrightarrow q)\ \&\ B_n(p \leftrightarrow r)$$

$$\Rightarrow tell(n, p \leftrightarrow r)$$

G6 expresses that if a question has been asked by m about the translation, n will manifest his translation if he has one.

$$(G6)\ TOP_n = tell(m, p \leftrightarrow q?)\ \&\ B_n(p \leftrightarrow q)$$

$$\Rightarrow tell(n, p \leftrightarrow q)$$

If there is no translation, the agent will say so (G7).

$$(G7)\ TOP_n = tell(m, p \leftrightarrow q?)\ \&\ \neg\exists r B_n(p \leftrightarrow r)$$

$$\Rightarrow tell(n, p \leftrightarrow q*))$$

5.3 The Fourth Turn

Depending on the cognitive state of the computer-agent, he may apply one of the previous rules, or a rule that stops the dialogue. If, for instance, the user-agent has manifested a translation and the computer-agent had no translation available, the translation will be used by m to provide an answer to the initial question. This is expressed in rules G2c and G2d.

If n has manifested that he does not know a translation, the dialogue ends, as expressed by rule G8:

$$(G8)\ TOP_m = tell(n, p \leftrightarrow q*) \Rightarrow stop(m)$$

6 The Update of Cognitive States

The update function yields a new cognitive state depending on the old state and the move just performed. To represent the consequences (or postconditions) of a particular move, we introduce '$\gg$'. The left side is of type action and represents the performed move; the right side represents the postconditions and denotes how the cognitive states should be updated. POP_m means that the top of the stack of m will be removed, $PUSH_m$ indicates that the just performed speech act has to be put on top of m's stack .

We will not be concerned with the full details of the update mechanism and assume that the cognitive states will be updated in accordance with the principles expressed in the rules R1 and R2. In the postconditions we will represent always the weakest conditions. For instance, if the shared beliefs are represented in the postcondition, the private beliefs and beliefs about the other are automatically updated in accordance with rule R1.

U1a and U1b express that a question is pushed on top of the stack of the recipient. The speech act has no further consequences for the cognitive state of the dialogue partners.

$$(U1a)\ tell(m, p?) \gg PUSH_n$$

$$(U1b)\ tell(m, p \leftrightarrow q?) \gg PUSH_n$$

U2 expresses that a proposition is simply added to the mutual beliefs of the dialogue participants and pushed on the stack of the partner:

$$(U2)\ tell(m, z) \gg B_n MBz\ \&\ B_m MBz\ \&$$

$$PUSH_n\ \&\ POP_m$$

where z denotes a (negated) proposition.

In case the statement contains a translation, the translation is added to the belief state of the partner about the other and the stack of the performer of the speech act is popped.

$$(U3)\ tell(n, p \leftrightarrow q) \gg B_m B_n (p \leftrightarrow q)\ \&\ POP_n$$

$$(U4)\ tell(m, p \mid p \leftrightarrow q) \gg PUSH_n\ \&\ B_n B_m q$$

$$(U5)\ tell(m, \neg p \mid p \leftrightarrow q) \gg PUSH_n\ \&\ B_n B_m \neg q$$

$$(U6)\ tell(n, p \leftrightarrow q*) \gg POP_n\ \&\ POP_m$$

We will now turn to an example where the computer agent C and the user-agent U play the co-operative dialogue game based on the previously introduced cognitive constructs and the generation and update rules.

In Figure 2 we have depicted the game-board, i.e. the cognitive states of C and U, the communicative acts (MOVE) that occur as a result of the dialogue rules, and, in addition, a reference to the applied update and generation rules (RULE); empty states are indicated by 'ϵ'. We have only depicted the content of a cognitive state in case of a change. We have omitted beliefs on mutual beliefs because we assume them to be initially empty and moreover, in this particular dialogue they do not change.

In this example, the computer-agent has a translation that does not correspond to the user-agent's translation. It can be observed how the various dialogue rules regulate the behaviour so that the computer-agent uses the translation of his partner to provide an answer, but does not transfer the translation to his own beliefs.

In natural language, this dialogue looks as follows:

B_C	$B_C B_U$	Stack$_C$	MOVE	B_U	$B_U B_C$	Stack$_U$	RULE
$p \leftrightarrow q_1 \wedge q_2$ $q_1, \neg q_2, q_3, \neg p$	ϵ	ϵ		$p \leftrightarrow q_1 \wedge q_3$	ϵ	ϵ	
			$tell(U, p?)$				
		$tell(U, p?)$					$U1a$
			$tell(C, \neg p \mid p \leftrightarrow q_1 \wedge q_2)$				$G2b$
				$p \leftrightarrow q_1 \wedge q_3$ $\neg(q_1 \wedge q_2)$	$\neg(q_1 \wedge q_2)$	$tell(C, \neg p \mid \ldots)$	$U5$ $R2$
			$tell(U, p \leftrightarrow q_1 \wedge q_3)$				$G5$
	$p \leftrightarrow q_1 \wedge q_3$					ϵ	$U3$
			$tell(C, p \mid p \leftrightarrow q_1 \wedge q_3)$				$G2c$
				$p \leftrightarrow q_1 \wedge q_3$ $\neg(q_1 \wedge q_2)$ q_1, q_3, p	$\neg(q_1 \wedge q_2)$ q_1, q_3	$tell(C, p \mid \ldots)$	$U4$ $R2$
			$stop(U)$				$G4c$

Fig. 2. Dialogue in which there is no shared understanding of a non-perceivable predicate

U: Is this car a safe car?
C: No, it is not safe in case you mean that it has air bags and a good crash test.
U: Well, with 'safe' I mean: does it have airbags and automatic screen wipers?
C: Oh, well, in that case the car is safe.
U: Thank you.

7 Discussion

In the previous sections, we have sketched a dialogue framework that enables a computer system to generate particular feedback sequences in interaction with a user of the system. The framework is comparable to approaches in dialogue game theory (see also [7]) and consists mainly of two parts: a. a game-board that contains information about a particular state of the game (i.e. the mental states of the participants) and b. the dialogue rules that control the behavior of the participants (generation rules) and that prescribe how the game-board changes (update rules). The framework is based on an explicit modeling of mental states in terms of the beliefs of the dialogue participants and their goals. Parts of these mental states function as preconditions for the generation of feedback contributions. In this paper we have applied the dialogue game to problems that may arise as a result of conceptual disparities about a particular domain of discourse between a user-agent and a computer-agent and we have shown how the framework enables the system to generate feedback either to resolve the disparity or to accept it and respond in an adequate manner.

In the dialogue model that we presented, the computer interface is considered to be a cooperative agent. This view has important consequences for the design of the interface, because the designer has to include the mechanisms that drive a natural human dialogue. In this paper, we have tried to show a small part of the machinery needed for modeling such a dialogue. In order to behave cooperatively, the agent has to be equipped with various mental constructs so that information about a particular domain of discourse (private beliefs) and about its dialogue partner (beliefs about beliefs) can be separated. Moreover, we distinguished between beliefs about the dialogue partner in 'beliefs about the partners private beliefs' and 'beliefs about shared beliefs' (the propositions and interpretations that were already agreed on). Including these types of mental constructs enables the computer agent to adapt its feedback in a number of ways. Information that is part of the shared beliefs can be considered as presupposed and should not be stated explicitly; this can be viewed in rule $G1$, where shared information is not discussed. Beliefs by the agent about private beliefs of the user influence feedback contributions in another way. In rule $G2$, extra information is added because the computer agent has no knowledge about the beliefs of the user (c and d) or because the agent believes that the user has a distinct belief (a and b).

Evidently, the framework is still rudimentary and extensions can be developed along many different lines. One of these lines is, for instance, the use of more complex ontologies. Concepts in real life can be defined in an almost infinite number of different terms and subterms with complex interrelationships and constraints, with different degrees of certainty and relevance. Since the dialogue rules in this paper are based on the structure of the ontology, adapting the rules to the meta-properties of the ontology (e.g. structure, complexity) seems inevitable.

Another simplification is the treatment of goals. Here we have presented goals as a simple stack with the operations 'push' and 'pop'. In these simple cases, it seems that neither a planning approach (see e.g. [3]), nor a speech act grammar approach is needed (or wanted) to build coherent structures of conversation and that feedback generation can be based on the immediately preceding conversational unit. Note that in general the consequences of the speech act ask is that goals are added to the stack of the receiver and that the speech act tell deletes goals from the stack of the sender. An important shortcoming of this approach is that, once the goals are deleted, the agents 'forget' what has been discussed before, so a 'rule designer' has to be careful in popping goals from the stack. An advantage is that the framework does not suffer from the same computational complexity as in most planning approaches where agents are not only able to reason about the discourse domain in the future, but also about their own and their partner's beliefs and intentions. We do not expect, however, that nested beliefs have to be modeled beyond the third level (A believes that B believes that A believes), since they simply seem to be unnecessary to model the basic properties of a cooperative dialogue (see also [17]).

It seems that the general framework of a dialogue game, in terms of the defined mental states and the generation and update rules applicable to these states, is an interesting and fundamental framework for adequate feedback generation. The framework does not suffer from the problems that we have in speech act grammars, such as a lack of situational dependency, and those that we have in planning approaches, such as computational complexity. In the long run, a planning approach is inevitable, but it remains to be seen which dialogue phenomena have to be modeled with a planning approach and which phenomena can be modeled without planning. It seems reasonable not to include complex methods as long as we can solve the same problems in a computational more simple and, therefore, more attractive way.

References

1. R.M.C. Ahn, R.J. Beun, T. Borghuis, H.C. Bunt, and C.W.A.M. van Overveld. The DenK-architecture: A fundamental approach to user-interfaces. *Artificial Intelligence Review*, 8:431–445, 1995.
2. G. Airenti, B.G. Bara, and M. Colombetti. Failures, exploitations and deceits in communication. *Journal of Pragmatics*, 20:303–326, 1993.
3. J.F. Allen and C.R. Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.
4. J.L. Austin. *How to do Things with Words*. Oxford University Press, Oxford, 1962.
5. R.J. Beun. On the generation of coherent dialogue: A computational approach. *Pragmatics and Cognition*, 9(1):37–68, 2001.
6. H.C. Bunt. Information dialogues as communicative action in relation to partner modelling and information processing. In Taylor et al. [18], pages 47–73.
7. L. Carlson. *Dialogue Games. An Approach to Discourse Analysis*. D. Reidel Publishing Company, Dordrecht, 1985.
8. G. Gazdar. Elements of discourse understanding. In A.K. Joshi, B.L. Webber, and I.A. Sag, editors, *Speech act assignment*. Cambridge University Press, Cambridge, 1981.
9. D.A. Good. The viability of conversational grammars. In Taylor et al. [18], pages 135–144.
10. H.P. Grice. Logic and conversation. In P. Cole and J.L. Morgan, editors, *Speech Acts. Syntax and Semantics, Vol. 11*, pages 41–58. Academic Press, New York, 1975.
11. B.J. Grosz and C.L. Sidner. Attention, intentions and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.
12. C.L. Hamblin. Mathematical models of dialogue. *Theoria*, 37:130–155, 1971.
13. E. Hutchins. Metaphors for interface design. In Taylor et al. [18], pages 11–28.
14. S.C. Levinson. *Pragmatics*. Cambridge University Press, Cambridge, 1983.
15. C. Rich, N. Lesh, J. Rickel, and G. Garland. A plug-in architecture for generating collaborative agent responses. In C. Castelfranchi and W. Lewis Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 782–789, New York, New York, 2002. ACM Press.
16. J.R. Searle. *Speech acts: An essay in the philosophy of language*. Cambridge University Press, Cambridge, 1969.
17. J.A. Taylor, J. Carletta, and C. Mellisch. Requirements for belief models in co-operative dialogue. *User Modelling and User-Adapted Interaction*, 6:23–68, 1996.
18. M.M. Taylor, F. Néel, and D.G. Bouwhuis, editors. *The structure of multimodal dialogue*, Amsterdam, 1989. Elsevier Science Publishers.