Feedback in Human-Computer Interaction: Resolving Ontological Discrepancies

Robbert-Jan Beun

Rogier M. van Eijk

(This report will be as published as Beun, R.J. & van Eijk, R.M. (2005) Feedback in Human Computer Interaction: Resolving Ontological Discrepancies. In: v. Oostendorp, H., Breure, & L., Dillon, A. (eds.) Creation, Use and Deployment of Digital Information. London: Erlbaum.)

institute of information and computing sciences, utrecht university
technical report UU-CS-2003-055
www.cs.uu.nl

Feedback in Human-Computer Interaction: Resolving Ontological Discrepancies

Robbert-Jan Beun & Rogier M. van Eijk
Institute of Information and Computing Sciences
Utrecht University, PO Box 80089,
NL-3508 TB Utrecht, the Netherlands
{rj,rogier}@cs.uu.nl

Introduction

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. But when we interact with computers, we also want them to react to our actions as a cooperative partner and to provide adequate feedback in case of communication flaws. In fact, we want them to be endowed with characteristics that closely mimic human conversation. Whereas the visual interactive qualities of interfaces have improved a lot, computers are still unable to generate the basic communication structures in a similarly powerful and 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 or other errors.

The goal of this paper is to discuss some of the theoretical principles that drive a conversation and to present a computational framework that enables us to generate elementary feedback sequences at the knowledge level of dialogue participants. One of the prerequisites for successful communication – both human/human and human/computer – is that the participants have a shared conceptualization of the meaning of the communication symbols. Here, we will focus on the feedback process that regulates the repair of communication flaws caused by conceptual disparities between a computer system and its user when using particular terms in a communication language. For reasons to be explained below, we will call these disparities *ontological discrepancies*.

In our approach, two agents – the user-agent and the computer-agent – play a dialogue game in which feedback is produced to transfer relevant information with respect to a particular agreement about the meaning of the words in the vocabulary. In line with Beun (2001), agents and their behaviour are modelled, respectively, by mental states in terms of goals and various types of beliefs[1] and by the rules that generate feedback sequences

[1] In this paper, we will not be concerned with the differences between belief and knowledge and the terms will be used interchangeably.

and that determine the change of the mental states as a result of the subsequent dialogue contributions. 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 Levinson (1983) and Good (1989)).

Describing the properties and the dynamics of the mental states 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:

1. What type of mental states should be included to model the dialogue's basic structural properties? (See Taylor, et al., 1996)
2. How do the various feedback contributions change the existing mental states? (See e.g. Gazdar, 1981 and Bunt, 1989)
3. How do these changes influence the generation of new contributions?

In what follows, we will show some of the necessary ingredients for a corrective feedback process when a user and a computer system have different conceptualizations of a particular domain of discourse. We will carefully describe which aspects of the situation have to be modeled to generate adequate conversational sequences and show how the information that the system holds about a domain and about the user influences the course of the dialogue between the system and its user. Based on a precise description of this type of information, the system is able to play the conversational game with the user. As a result, an interface designer may obtain rules with sufficient detail for adequate feedback, given the system's ability to model the various types of information about the domain *and* about the user.

Cooperative feedback in communication

The term 'feedback' originates from the area of cybernetics and refers to the information that a system receives from its environment about the consequences of its behavior (Wiener, 1948). Feedback information is often used to regulate the behavior and guides, in case of purposeful behavior, the actions towards a particular goal. The concept of feedback in cybernetics has had many successful applications in engineering and physics, because the performance of electronic and other physical systems can be modeled quite accurately. We know, within certain limits, how the output of a system - for instance, an electronic circuit - depends on the value of the input variables and we have accurate mathematical tools that predict the system's respons to a change of the input variables as a result of feedback information.

In both human-human and human-system communication, feedback is used for a broad range of communicative responses at various levels and has an enormous diversity, varying from a simple nod or a beep that indicates the receipt of a message to a written comment that evaluates the quality of a scientific paper. However, for various reasons, we have no accurate mathematical theory for natural communicative behavior and the application of cybernetic models to human communicative activities has only a limited scope of relevance (Spink & Saracevic, 1998).

When we look at feedback phenomena in conversations between humans, sequences in terms of speech acts appear to be rather chaotic and seem hardly subjected to any rules. Questions can be followed by answers, denials of the relevance of the question, rejections of the presuppositions of the question, statements of ignorance, and so on (see e.g. Levinson, 1983). An example of general rules for cooperative contributions, and conversational feedback in particular, are the Gricean maxims for conversation, such as 'tell the truth' (quality), 'say enough, but not too much' (quantity), 'be relevant' (relevance) and 'use the appropriate form' (manner) (Grice, 1975). Clearly, not all people follow the four rules to the letter, but Grice's point is that, contrary to particular appearances in conversation, the principles are still adhered to at some deeper level. An example is given in the following dialogue (from Sacks cited by Levinson, 1983):

A: I have a fourteen-year old son
B: Well, that's all right
A: And I also have a dog
B: No, I am sorry

At first sight, B's feedback looks unacceptable or even bizarre, and his contributions appear to be a blunt violation of the Gricean maxim of relevance. However, once the circumstances are given - A is trying to hire a room from the landlord B - the dialogue sounds perfectly normal. So it seems that we hardly have a criterion for the acceptability of feedback information without contextual knowledge. In other words, sequencing rules in conversation can never be captured in terms of speech act terms only, such as question/answer and assertion/denial, and our determination of the correctness of a sequence of utterances is heavily influenced by the circumstances of the conversation. From this we conclude that modeling and generating acceptable feedback sequences involves at least an explicit representation of the circumstances of the dialogue in terms of shared background knowledge and intentions of the dialogue participants.

In human-system interaction - where a system is represented by some kind of electronic equipment, such as a computer or a video player - a diversity of heuristics for feedback is suggested. Nielsen, for instance, states that a system should continuously inform the user about what it is doing and how it is interpreting the user's input (Nielsen, 1993). More detailed heuristics concern the different degrees of persistence in the interface, respons times and corrective feedback in case of errors. Just as the Gricean maxims form guidelines for the acceptability of human conversational sequences, the heuristics offer an important and practical handle for a systematic evaluation of user interfaces. However, both type of rules are underspecified in case an interface designer wants to realize the actual implementation. In other words, the rules have some explanatory power, but no predictive power and do not provide the designer with sufficient detail about the type, content and form of the feedback that has to be generated in a particular situation.

In this paper, we will concentrate on feedback sequences that are compulsory at the knowledge level; we will focus in particular on the generation of those sequences that support the establishment of a common ontology between two dialogue participants. Before we will explain the details of the generation mechanism, we will first elaborate on the role of ontologies in communication.

Communication about ontological discrepancies

In communication, a distinction is usually made between the conceptualization of a particular domain of discourse and the communication symbols that refer to various aspects of the domain. In some respect, the communication symbols, and more specifically a particular language in terms of a lexicon and grammatical rules, can be considered as an interface between the conceptualizations of the two dialogue partners. Whereas we have no direct access to the conceptualizations of humans, the conceptualization of computer systems can be expressed in a so-called *ontology*, which is often defined in a formal language, such as a programming language or a formal logic. An ontology abstracts the essence of the domain of interest and helps to catalogue and distinguish various types of objects in the domain, their properties and relationships.

In the domain of e-commerce, for instance, important concepts are those that relate to the issue of trust. Interactions between participants take place over the internet without any personal contact, so consumer trust in the internet vendor is an issue of major concern (Chen & Dhillon 2003). But what exactly does it mean for a vendor to be trustworthy? Many aspects play a role; it is the purpose of an ontology to give structure to the domain and enable the cataloguing of internet vendors.

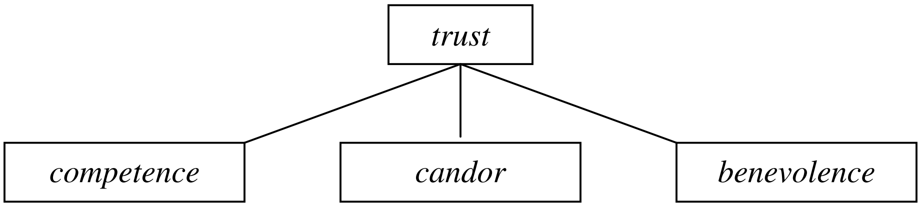

Figure 1: Ontology O_1 of trust

Figure 1 depicts an example of a simple ontology O_1 of trust, which organizes some of the relevant concepts in a tree-like structure. The concept *trust* constitutes the root of this ontology. It has three subordinate nodes: the concepts *competence*, *candor* and *benevolence*. The ontology defines consumer trust as the conjunction of the vendor's competence, candor and benevolence. The above ontology can also be expressed in predicate logic:

$$\forall Y: (trustworthy(Y) \leftrightarrow (competent(Y) \wedge candid(Y) \wedge benevolent(Y))$$

which expresses that trustworthy internet vendors are those vendors that are competent, candid and benevolent. The ontology enables the computer system to reason and to communicate about issues of trust. So, in principle, a user can now ask questions such as 'Is this particular vendor trustworthy?' or 'What does it mean to be trustworthy?' Note that ontologies, in general, can be complex structures. For instance, the concept *candor* may in turn constitute the root of a subtree with concepts like *honesty*, *consistency* and *reliability* as its subordinate nodes; these subordinate nodes may each have other subtrees, and so on. Complex structured ontologies are beyond the scope of the current paper, however. Here, we will only consider ontologies like O_1 that define a non-basic concept in terms of one or more basic concepts.

Suppose we have another agent that holds a deviated ontology O_2 (see Figure 2). In ontology O_2 consumer trust in internet vendors is defined as a conjunction of the vendors' competence and punctuality. Similar to ontology O_1 this ontology defines the concept

competence as a subordinate node of *trust*, but unlike O_1 it contains *punctuality* as a subordinate instead of *candor* and *benevolence*. We refer to such disparities between O_1 and O_2 as *ontological discrepancies.*

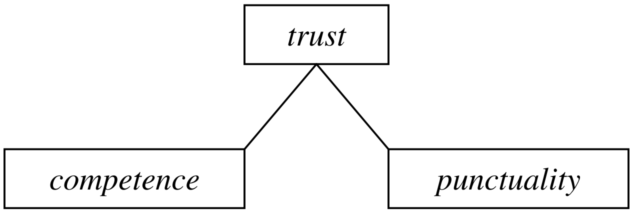

Figure 2: Ontology O_2 of trust

It goes without saying that in successful communication and in collaborative performance of tasks, agreement with respect to the ontology is crucial or, at least, the dialogue partners should be aware of existing discrepancies. In case of discrepancies, various troublesome situations may arise. For instance, one of the participants by mistake may assume that a particular concept is shared, while the other has no knowledge about it. Or worse, both dialogue partners initially have different conceptualisations, while the relevant discrepancy remains unnoticed. Ontology discrepancies may cause serious communication flaws and the generation of adequate utterances in order to repair these flaws is an essential part of modelling a proper feedback process.

In this paper, we will make some important simplifications. First, we assume that every word in the communication language with respect to the domain of discourse has one and only one corresponding concept in the ontology.[2] So, for instance, the word 'trustworthy' of the vocabulary corresponds to the concept *trust* in the underpinning ontology. Second, we will not consider the problem of the detection of ontology discrepancies and only concentrate on the feedback process of avoiding or repairing the discrepancy. Third, we will not consider cases where the two partners use different words for the same conceptualization.

A corresponding concept and its substructure in the ontology of a word in the communication language will also be called the 'interpretation' or the 'meaning' of that word. For instance, the word 'trustworthy' has the interpretation of Figure 1 in O_1 and Figure 2 in O_2. Initially, three basic settings with respect to ontology discrepancies will be distinguished:

[2] This assumption allows the dialogue partners to hold different vocabularies, but does not allow words in the communication language to be ambiguous.

a. the ontology of both agents agrees with respect to a particular word, i.e. there is no discrepancy and the interpretations of both agent are the same
b. both agents have different interpretations with respect to a particular word
c. only one agent has an interpretation of a particular word

Note that c. together with the first assumption implies that also the corresponding word is absent from the other participant's vocabulary. Moreover, the first assumption helps us to avoid rather marginal cases where the participants use words that have no corresponding interpretation in both ontologies.

To make things a little more concrete, let us first consider a situation where a user U intends to purchase a particular item X and where different internet vendors have the item in their catalogue. Assume vendor Y is the cheapest and the user uses a computer-agent C who mediates the transaction. The user may start the dialogue as follows:

U: 'I want to purchase item X from vendor Y. Is Y a trustworthy vendor?'

Now suppose that U and C have agreement about the domain words 'purchase', 'item', 'vendor' and the references of X and Y, and that the computer-agent understands the grammatical rules to interpret U's utterance. Depending on the situation, C has various possibilities to continue the dialogue, for instance:

a. C may simply respond to the question (e.g. 'Yes, it is', 'No, it is not', 'I don't know')
b. C may answer the question and add extra information about the interpretation of the word 'trustworthy' (e.g. 'Yes, Y is competent, benevolent and candid', 'No, because Y is not competent')
c. C may give a tentative answer (e.g. 'In case you mean by trustworthy that Y is …, yes, Y is trustworthy')
d. C may ask a counter-question for the interpretation of the word 'trustworthy' (e.g. 'Do you mean that Y is …?', 'What do you mean by trustworthy?')

This may result in the following sample dialogues. In dialogue 1, for instance, C explicitly states his[3] interpretation of the word 'trustworthy' and, since the user does not deny the interpretation, it can be concluded that she agrees.

Dialogue 1
U: Is Y a trustworthy vendor?
C: Yes, Y is competent, candid and benevolent.
U: Ok, thanks.

[3] Throughout this paper, the computer-agent will receive masculine pronominalisation and the user will be feminine.

In dialogue 2 neither of the two participants accepts the other as an expert on the interpretation and neither contributes extra information that forces a decision. The dialogue ends in an infinite loop.

Dialogue 2
U: U: Is Y a trustworthy vendor?
C: Yes, Y is competent, candid and benevolent.
U: Well, to my opinion a trustworthy vendor is competent and punctual.
C: To my opinion, a trustworthy vendor does not need to be punctual.
U: Well, it does.
C: No, it doesn't.
…

Clearly, infinite dialogues have to be avoided and in the generation rules in section 4 we will give preference to a strategy where C tentatively accepts the interpretation by U (see also Dialogue 3). This implies that C has to make at least a distinction between his own private beliefs and his beliefs about U's beliefs.

Dialogue 3
U: Is Y a trustworthy vendor?
C: Yes, Y is competent, candid and benevolent.
U: Well, to my opinion a trustworthy vendor is competent and punctual.
C: In that case, no, Y is not trustworthy, because he is not punctual.
U: Ok, thanks.

Note that we assume that the word 'punctual' is also part of the C's vocabulary and, therefore, underpinned in C's ontology. From the dialogue fragment we cannot conclude, however, that C also adopts the user's interpretation of the word 'trustworthy' as his own interpretation.

C also has various strategies in the second turn. If, for instance, C has a private interpretation, but a shared one is doubted, he may simply ask for the meaning without manifesting his own interpretation.

Dialogue 4:
U: Is Y a trustworthy vendor?
C: What do you mean by trustworthy?
U: A trustworthy vendor is a vendor that is competent and punctual.
C: In that case, he is trustworthy.
U: Ok, thanks.

A problem in Dialogue 4 is that the dialogue suggests that C has no private interpretation. So, depending on C's tenacity, U may incorrectly come to believe that U's interpretation of the word is a shared belief. Not manifesting C's interpretation may in these cases be considered as a violation of the Gricean quantity maxim. A possible solution is given in

Dialogue 5 where C immediately provides the extra information after U has manifested her interpretation of the word:

Dialogue 5:
U: Is Y a trustworthy vendor?
C: What do *you* mean by trustworthy?
U: To my opinion, a trustworthy vendor is a vendor that is competent and punctual.
C: Well, I think that a trustworthy vendor is competent, benevolent and candid.
U: I prefer my own interpretation.
C: In that case, he is trustworthy.

In Dialogue 5, C gives another cue by stressing the pronominal reference 'you', which already implicates that there may be other interpretations as well, in particular C's interpretation. In the framework below, we will avoid the extra turns and in case of absence of a shared interpretation, C will not ask for one, but explicitly state his interpretation of the word (if he has a private one).

In conclusion, in order to be able to distinguish between the above situations the content of the computer-agent's respons not only depends on his beliefs about the domain of discourse, but also on his beliefs about the user's beliefs and their shared beliefs. In case the user indicates an interpretation that differs from the computer's concept *trust*, it should be decided whose interpretation should prevail. In principle, this is a matter of expertise: the interpretation of an expert usually takes preference over that of a non-expert. If neither of the participants is more an expert than the other, they could start an argumentation about the interpretation. In this paper, however, we will assume that the computer treats the user as an expert on the interpretation, and thus gives priority to her interpretation while answering the initial question. We will adopt the strategy that the computer-agent accepts tentatively the user's interpretation and drops it after the initial question has been answered.

The Formal Framework

In this section, we outline a computational framework for the generation of human-computer dialogues in which the computer provides feedback in case of ontological discrepancies. For reasons of legibility, we will not give a full formalization, but describe the various aspects of the framework in a semiformal way. In particular, instead of specifying the behavior rules in their most general form, we will define them in terms of typical examples. The generalizations of the rules should be evident from these examples and will not be further worked out in this paper.

Ontologies are used to define the meaning of non-basic predicates (like *trustworthy*) in terms of basic predicates (like *competent, candid* and *benevolent*). It will be assumed that both agents know the meaning of the basic predicates and that these predicates are part of the agents' shared beliefs, have the same interpretation for both agents and, consequently, the meaning of basic predicates is never part of the discussion. Moreover, we assume that for basic predicates p the computer-agent is able to decide whether $p(X)$ holds for each individual X. So, this implies that the computer-agent is always able to give an answer to the initial question whenever a meaning of the non-basic predicate *p* is available. Below, a basic predicate applied to some individual will be called a 'basic proposition'.

In order to play the dialogue game, the user will be simulated by a so-called user-agent that has an explicit representation of the user's domain conceptualisation in terms of an ontology. An agent's mental state (both user- and computer-agent) consists of the following constructs:

a) information about the domain of discourse (private beliefs)
b) information about the beliefs of the other agent (beliefs about beliefs of the other)
c) information about the shared beliefs of both agents (beliefs about shared beliefs)
d) a pending stack containing dialogue moves that have to be processed (the goals of the agents).

Note that an agent's mental state not only contains propositions, but also the underpinning ontologies.

We assume that both agents can reason about their beliefs by Modus Ponens and that the following dependencies exist between the mental states of an agent:

<u>Rule R1</u>
IF a proposition or a meaning is part of the shared belief of an agent
THEN it is also part of the agent's private belief
AND of its beliefs about the other agent.

It is important to note that the opposite of R1 does not hold.

<u>Rule R2</u>
IF the user-agent believes that the computer-agent believes some basic proposition to hold
THEN the user-agent also itself believes this basic proposition to hold.

In fact, the rule R2 establishes particular roles in the dialogue, since the computer-agent is considered as an expert on basic 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.

The Dialogue Rules

Dialogue contributions or so-called 'moves'[4] are fully determined by the mental state of the agent who performs the move and the rules that are applicable to this state. We will assume that the initial question by the user-agent has been asked in the first turn of the dialogue. 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.

[4] In line with a more Searlian tradition (Searle, 1969), we will also call these moves 'speech acts'.

The second turn

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

- It believes that an interpretation is shared by his partner and so gives the answer (Generation rule 1).
- It believes an interpretation, but does not believe that his partner shares it (Generation rule 2).
- It does not know an interpretation (Generation rule 3).

Generation rule 1

IF the agent is processing *ask(trustworthy(X))*

AND it believes that the meaning *trust* = *competence* $\wedge$ *candor* $\wedge$ *benevolence* is shared

THEN it responds with:

a) *tell(trustworthy(X))*
   if it believes *trustworthy(X)* to hold

b) *deny(trustworthy(X))*
   if it believes *trustworthy(X)* not to hold

Note that we do not consider the case where the computer-agent does not know the answer, since we assumed that he is always able to find an answer to the question as long as the proposition is built up from basic predicates and the interpretation is available.

Generation rule 2

IF the agent is processing *ask(trustworthy(X))*

AND it does not know the other agent's meaning of *trust*

AND its own meaning is *trust* = *competence* $\wedge$ *candor* $\wedge$ *benevolence*

THEN it responds with:

a) *tell(trustworthy(X)* |*trust* = *competence* $\wedge$ *candor* $\wedge$ *benevolence)*
   if it believes *trustworthy(X)* to hold

b) *deny(trustworthy(X)* |*trust* = *competence* $\wedge$ *candor* $\wedge$ *benevolence)*
   if it believes *trustworthy(X)* not to hold

IF the agent is processing *ask(trustworthy(X))*

AND it believes the other agent's meaning of *trust* = *competence* $\wedge$ *punctuality*

AND its own meaning is *trust* = *competence* $\wedge$ *candor* $\wedge$ *benevolence*

THEN it responds with:

c) *tell(trustworthy(X)* |*trust* = *competence* $\wedge$ *punctuality)*
   if it believes both *competent(X)* and *punctual(X)* to hold

d) *deny(trustworthy(X)* |*trust* = *competence* $\wedge$ *punctuality)*
   if it believes either *competent(X)* or *punctual(X)* (or both) not to hold

In Generation rule 2, the computer-agent knows a meaning of the predicate but does not believe that his partner shares this meaning. As a result, a conditional answer (expressed by the symbol '|' in the move) is generated and extra information about the interpretation is added. Note that in rule 2 c) and d) the user-agent is considered to be the expert on the meaning of the non-basic predicate and that this meaning takes priority over the computer-agent's beliefs (see also the fourth turn).

Generation rule 3
IF the agent is processing *ask(trustworthy(X))*
AND it does not have an own meaning of *trust*
THEN it responds with *ask(trust)*

So, if the computer-agents has no meaning, it manifests its ignorance by asking the user-agent for the meaning. In natural language this can be expressed by a WH-question, (e.g., 'What is the meaning of trust?').

The third turn
Just before the third turn, the respons of the computer-agent will be on top of the stack of the user-agent (see the update rules below). Depending on this respons and the mental state of the user-agent, the user-agent has three possible reactions:

- the computer-agent's respons may be accepted (Generation rule 4),
- the respons may be rejected and a meaning may be provided (Generation rule 5),
- the user-agent may indicate that she has a meaning available (Generation rule 6).

Generation rule 4
a) IF the agent is processing *tell(trustworthy(X))*
THEN it responds with *stop*
b) IF the agent is processing *deny(trustworthy(X))*
THEN it responds with *stop*
c) IF the agent is processing *tell(trustworthy(X) |trust = competence ∧ punctuality)*
AND it does not have a different meaning of *trust*
THEN it responds with *stop*

In Generation rules 4, the user-agent accepts the statement by the computer-agent and subsequently stops the dialogue.

Generation rule 5
IF the agent is processing *tell(trustworthy(X) | trust = competence ∧ candor ∧ benevolence)*
AND its own meaning is *trust = competence ∧ punctuality*
THEN it responds with *tell(trust = competence ∧ punctuality)*

In this rule, the meaning *competence ∧ candor ∧ benevolence* of *trust* is rejected because the user-agent has found a meaning that does not correspond to her own. In a rejection, the agent tells the grounds for her rejection, so that the computer-agent has knowledge about the reason of the discrepancy.

Generation rule 6
IF the agent is processing *ask(trust)*

AND its own meaning is *trust = competence* $\wedge$ *punctuality*
THEN it responds with *tell(trust = competence* $\wedge$ *punctuality)*

Generation rule 6 expresses that if a question has been asked by the computer-agent about the meaning of the predicate, the user-agent will manifest her meaning if she has one. Note that we already excluded the situation where the user-agent has no interpretation.

The fourth turn
Depending on its mental state, the computer-agent may apply one of the previous rules, or a rule that stops the dialogue. If, for instance, the user-agent has manifested a meaning and the computer-agent has no meaning available, the meaning will be used by the computer-agent to provide an answer to the initial question. This is expressed in generation rules 2c and 2d.

The Update of Mental States

The update rules yield a new mental state depending on the old state and the move just performed. We will not be concerned with the full details of the update mechanism and assume that the mental states will be updated in accordance with the principles expressed in the rules R1 and R2. In the postconditions we will always represent the weakest conditions. If, for instance, the shared beliefs are represented in the postcondition, the private beliefs and beliefs about the other are automatically updated in accordance with rule R1.

Update rule 1

a) The postcondition of the question *ask(trustworthy(X))* of the user-agent is that the question is pushed on the stack of the computer-agent
b) The postcondition of the question *ask(trust)* of the user-agent is:
the question is pushed on the stack of the computer-agent

These rules express that a question is pushed on top of the stack of the recipient and that the speech act has no further consequences for the mental state of the dialogue partners. Note that the main difference between a. and b. is that a. concerns a question about propositions (yes/no-questions) and b. about the meaning of a term (WH-questions).

Update rule 2
The postcondition of the respons *tell(trustworthy(X))* of the computer-agent is that the respons is pushed on the stack of the user-agent
AND the stack of the computer-agent is popped
AND the user-agent believes that *trustworthy(X)* is part of the shared beliefs
AND the computer-agent believes that *trustworthy(X)* is part of the shared beliefs

Update rule 2 expresses that a proposition is simply added to the shared beliefs of the dialogue participants and pushed on the stack of the partner. The rule for *deny(trustworthy(X))* is similar and omitted here.

Update rule 3

The postcondition of the respons *tell(trust = competence* $\wedge$ *punctuality)* of the user-agent is that the stack of the user-agent is popped
AND the computer-agent believes that the user-agent believes *trust = competence* $\wedge$ *punctuality*

Rule 3 states that in case the statement contains an explanation of the meaning, it is added to the belief state of the partner about the other and the stack of the performer of the speech act is popped. Note that rule 2 concerns the manifestation of a domain proposition and rule 3 the manifestation of the interpretation of a non-basic predicate.

Update rule 4

The postcondition of the respons *tell(trustworthy(X)* | *trust = competence* $\wedge$ *candor* $\wedge$ *benevolence)* of the computer-agent is that this respons is pushed on the stack of the user-agent
AND the user-agent believes that the computer-agent believes *competent(X)*, *candid(X)* and *benevolent(X)* to hold.

The rule for *deny(trustworthy(X)* | *trust = competence* $\wedge$ *candor* $\wedge$ *benevolence)* is similar to update rule 4 and omitted here. Note that Rule 4 does not express that the computer-agent's interpretation is accepted.

Example

We conclude with an example of a dialogue between the computer agent C and the user U that is generated by the above rules. The computer-agent employs the ontology O_1, which defines the non-basic concept *trust* as a conjunction of the basic concepts *competence*, *candour* and *benevolence*; the user employs ontology O_2. Initially, C does not believe that the meaning of the ontology is a shared belief and also believes that the firm Asbelix is not very punctual. The user starts with a question:

1. U: 'Is Asbelix a trustworthy internet vendor?'

This question is pushed upon the stack of the computer-agent (update rule 1a). According to his interpretation (based on O_1), the firm Asbelix is indeed trustworthy, but since the computer-agent does not know that the meaning is shared it responds with (generation rule 2a):

2. C: 'Yes, because Asbelix is competent, candid and benevolent'

This respons is pushed upon the stack of the user and the user now comes to believe that Asbelix is competent, candid and benevolent (update rule 4 and rule R2). The user-agent, however, employs the ontology O_2, which defines *trust* to be the conjunction of the basic concepts *competence* and *punctuality*. It manifests this discrepancy by responding with (generation rule 5):

3. U: ‘To my opinion, trust amounts to competence and punctuality’

The stack of the user is popped and the computer-agent now knows the user’s interpretation of trust (update rule 3) and, since it believes that this particular vendor Asbelix is not punctual, it gives the following answer to the initial question (generation rule 2b):

4. C: ‘Well, if trust amounts to competence and punctuality then Asbelix is not a trustworthy internet vendor’

This respons is pushed upon the stack of the user-agent and the user now knows that Asbelix is not punctual (update rule 4 and rule R2). The user-agent has received an acceptable answer to its question and ends the dialogue (generation rule 4c)

5. U: ‘OK, thank you’.

Note that if in the second turn, the computer agent would not have manifested its interpretation of trust, the ontological discrepancy would have remained unnoticed, possibly leading the user-agent to draw incorrect conclusions from the answer.

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 Carlson, 1985) 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.

How does this framework contribute to the interaction between a computer system and its user and what are the limitations of the framework? This question brings us back to the idea stated in the introduction that a computer interface may also be considered as a cooperative agent. This was already suggested in Hutchins (1989) where a model for human-computer interaction was presented that distinguishes between two types of interaction: symbolically and physically. The essential difference between the two types of interaction is that actions of the first type (e.g. speech acts) 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 (see also Ahn, et al 1995). Nowadays, interface designers focus mainly on the direct type of interaction (remember the graphical apparatus for direct manipulation and observation discussed in the introduction) and are hardly concerned with the rules for interaction and feedback that can be observed in natural human communication.

Viewing a computer interface as a cooperative dialogue partner 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' (for instance, the computer's belief about ontology O_2) 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 Generation rule 1, where shared information is not discussed. Beliefs by the agent about private beliefs of the user influence feedback contributions in another way. In Generation rule 2, extra information is added because the computer agent has no knowledge about the beliefs of the user (a and b) or because the agent believes that the user has a distinct belief (c and d). In fact, the distinction between the various beliefs enables a designer to give concrete form to the Gricean maxim of quantity that we mentioned earlier, since shared beliefs give the designer a criterion to *leave out* particular information in the dialogue move and beliefs about the other's beliefs give a criterion to *include* particular information.

It can be verified that the dialogue rules discussed in this paper yield sixteen different dialogues in terms of speech act sequences, such as *ask/tell/stop*, *ask/deny/stop*, *ask/deny/tell/deny/stop*, *ask/ask/tell/deny/stop,* and that the exact type and order of speech acts depends on the initial state of the beliefs of the dialogue participants. (Note that *ask/tell/tell/tell/stop* is the sequence of speech acts in the example discussed in the previous section; the difference between the three *tell*-moves can be found in the semantic content of the speech act.) In order to avoid infinite sequences of disagreement, we have defined roles that the two dialogue partners play in the game: the computer is considered to be the expert on the propositions with respect of the domain of discourse and the user is considered to be the expert on the meaning of the vocabulary. In our 'simple' version of the dialogue game, these roles are implicitly built into the dialogue rules, but in more refined versions of the framework, roles could be defined explicitly and included as an extra precondition in the generation rules. In even more sophisticated versions, it can be imagined that the roles can be learned by the computer agent or that they differ with respect to the type of predicates that are used to describe the properties of objects (see also below).

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. And, although effective, the distinction between basic and non-basic predicates is too simple. In practice it may also be useful to classify predicates in terms of observable or non-observable. Depending on the communicative situation (e.g. 'Who can observe what?'), this may also have important implications for the role distinction between expert and non-expert and for the dialogue rules.

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. Allen & Perrault, 1980), 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 Taylor et al., 1996).

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 a powerful 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.

Acknowledgements
The authors would like to thank the three anonymous reviewers who criticised an earlier draft of the paper and suggested valuable improvements.

References

Ahn, R.M.C., Beun, R.J., Borghuis, T., Bunt, H.C. & van Overveld, C.W.A.M. (1995) The DenK-architecture: A Fundamental Approach to User-Interfaces. *Artificial Intelligence Review, 8*, 431-445.

Allen, J.F. & Perrault, C.R. (1980) Analysing Intention in Utterances. *Artificial Intelligence, 15*, 143-178.

Beun, R.J. (2001) On the Generation of Coherent Dialogue. *Pragmatics and Cognition. Vol. 9(1),* pp. 37-68.

Bunt, H.C. (1989) "Information dialogues as communicative action in relation to partner modelling and information processing". In: M.M. Taylor, F. Néel & D.G. Bouwhuis (eds.) *The Structure of Multimodal Dialogue*. Amsterdam: North Holland, 47-73.

Carlson, L. (1985) *Dialogue Games. An Approach to Discourse Analysis*. Dordrecht: D. Reidel Publishing Company.

Chen, S.C. & Dhillon, G.S. (2003). Interpreting Dimensions of Consumer Trust in E-commerce. *Information Technology and Management* 4, pp. 303-318.

Gazdar, G. (1981) "Speech act assignment". In: A.K. Joshi, B.L. Webber & I.A. Sag (eds.) *Elements of Discourse Understanding*. Cambridge: Cambridge University Press.

Good, D.A. (1989) "The Viability of Conversational Grammars". In: M.M. Taylor, F. Néel & D.G. Bouwhuis (eds.) *The Structure of Multimodal Dialogue*. Amsterdam: Elsevier Science Publishers. 135-144.

Grice, H.P. (1975) "Logic and Conversation". In: P. Cole & J. Morgan (Eds.): *Speech Acts. Syntax and Semantics, Vol. 11*. New York: Academic Press. pp. 41-58

Hutchins, E. (1989) "Metaphors for Interface Design". In: M.M. Taylor, F. Néel & D.G. Bouwhuis (eds.) *The Structure of Multimodal Dialogue*. Amsterdam: North Holland, 11-28.

Levinson, S. C. (1983) *Pragmatics.* Cambridge: Cambridge University Press.

Nielsen, J. (1993). *Usability Engineering*. San Diego: Morgan Kaufman.

Searle, J.R. (1969) *Speech Acts*. Cambridge: Cambridge University Press.

Spink, A. & Saracevic, T. (1998) Human-Computer Interaction in Information Retrieval: Nature and Manifestation of Feedback. Interacting with Computers, 10, 249-267.

Taylor, J.A., Carletta, J. & Mellish, C. (1996) "Requirements for Belief Models in Co-operative Dialogue". *User Modelling and User-Adapted Interaction, 6*, 23-68.

Wiener, N. (1948). Cybernetics or Control and Communication in the Animal and the Machine. Cambridge (MA): The MIT Press.