Repairing Conceptual Mismatches in Dialogue: a Computational Approach

Robbert-Jan Beun
Rogier M. van Eijk

Department of Information and Computing Sciences
Utrecht University
Technical Report UU-CS-2006-033
www.cs.uu.nl
ISSN: 0924-3275

Repairing Conceptual Mismatches in Dialogue: a Computational Approach

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

Abstract. In this paper, a computational framework is presented for the generation of elementary speech acts to establish conceptual alignment between a computer system and its user. We clearly distinguish between two phases of the alignment process: message interpretation and message generation. In the interpretation phase, presuppositions are extracted from the user's message and compared with the system's semantic model of the application domain, which is represented in Type Theory. Subsequently, in the generation phase, an adequate feedback message is produced in order to resolve detected discrepancies. A conversational strategy is provided that is based upon Gricean maxims and a differentiation of various types of information states of the system, such as private and common beliefs about the domain of discourse, and the goal of the user.

1. Introduction

Contemporary technological developments of interactive systems and the expansion of bandwidth enable designers to incorporate a variety of media and modalities in the computer interface. But merely adding amazing technological feats or increasing bandwidth does not necessarily improve the communication process. When we interact with computers, we also 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.

One of the prerequisites for natural human communication is that participants are able to reason about and to discuss various aspects of the domain of discourse. In order to cope with the complexity of the world around us, people consider the existence of objects, discuss possible behaviour, draw conclusions from the various dialogue contributions, discuss the meaning of the communication symbols and many more. Behind these manifestations of various beliefs and opinions is the participant's need to conceptualise the problem domain and to build a coherent and consistent mental model to achieve some sort of common understanding of our complex world.

The situation in human-computer interaction hardly differs from the communicative situation in the real world. In a computer domain, a user should know that there are objects like files and folders, that files are removable, readable, editable, storable, etc. Although the need to discuss these various aspects of the virtual domain is probably even more compelling than in the real world, these conversational skills are usually absent from the computer interface.

The goal of this paper is to present a computational method that enables us to generate feedback utterances that regulate the repair of conceptual mismatches between a computer system and its user. We clearly distinguish between two phases of the alignment process: message interpretation and message generation. In the interpretation phase, assumptions are extracted from the user's message on the basis of the system's belief about the application domain and the rules of well-formedness of these beliefs. In other words, based on both its own semantic model of the application domain and the user's utterance, the system *supposes* that the user makes particular assumptions about the domain. If conceptual mismatches between the user's assumptions and the system's belief are detected, a feedback message is produced in order to resolve the mismatch. We provide a conversational strategy that is based upon Gricean maxims and the system's dynamic mental state that contains information about the application domain and the conversational partner. The semantic model of the system's beliefs of the application domain will be represented in Type Theory. In order to regulate the communicative behaviour of the system, we also introduce a model that contains pragmatic knowledge with respect to the user, such as the beliefs and goals of the user. In fact, the pragmatic model marks the semantic information as part of particular information states.

The remainder of this paper is organised as follows. First, in section 2, we consider the characteristics of the various communication models that should play a basic role in human-computer interaction. We will introduce the formal semantic model that represents a variety of aspects of the domain of discourse and the formal pragmatic model. In section 3, we discuss the role of the Gricean cooperation principle in the feedback generation process. Section 4 and 5 describe the formalization of the semantic model in terms of Type Theory and the detection of conceptual mismatches. In section 6 the actual conversational strategy is presented. We wrap up in section 7 with some conclusions and directions for future research.

2. Mental, Conceptual and Formal Models

Humans carry a model of the external reality and are able to reason about various aspects of the world, to think about the past and the future, and they can decide which action is appropriate given a certain goal within the circumstances of the activities. In the same manner, users of a particular computer application build their own conceptualisation of the characteristics and the behaviour of the application. The mental representations that reflect the user's understanding of a system is often termed a *mental model*. Mental models are used to predict the system behaviour and to guide the user's actions.

Mental models are based on the user's previous knowledge and experiences, and evolve naturally with the interaction of the system under consideration. They are sometimes being derived from idiosyncratic interpretations of the system and must operate within the constraints of the human processing system (Norman, 1983). Since only a fraction of the computer application and its internal state is observable, mental models are not only dynamic, they are also often inaccurate, incomplete, inconsistent and incoherent.

In this paper we will not be concerned with a detailed analysis of the characteristics of a mental model. In other words, whether a mental model is a picture in the head, a set of propositions, a schema, structural or functional, or any other representation is irrelevant here (cf. van der Veer & Puerta Melguizo, 2003). What is important, however, is that we assume a close relation between the intentional behaviour of users, in particular their communicative actions, and their mental model. For instance, if a user utters the question 'Restart the system', she believes, among many other things, that the system exists and that it can be restarted; if a user clicks on the 'save'-button after editing a file, we assume that the user wants to apply the action 'save' to the file-object unless we have evidence for the contrary. Note that the verbs 'believe' and 'want' are natural language terms that explicitly refer to the internal state of the user. Unintentional non-verbal behaviour may as well reveal particular aspects of a user's mental state – think of the application of lie detectors or particular computer games – but valid conclusions are probably much harder to draw from this type of information. Although we expect that most of our results can be generalized into a theory of action sequences in direct manipulation, we will here concentrate on the intentional communicative behaviour of users in terms of simple sequences of words as part of a linguistic dialogue.

In contrast to the concept of mental model is Norman's idea of a *conceptual model* (CM). The conceptual model characterises the relevant objects, their features, relations and behaviour and the interaction with the user. In fact, the conceptual model is supposed to be sound with respect to the application domain and complete with respect to a particular task. It is devised as a tool for the understanding or teaching of the system to the user and often informally communicated in natural language, graphical symbols and pictures. It is important to note, however, that in this paper we will assume that the conceptual model is usually 'in the head' of the designer and, therefore, not directly accessible for other subjects, although it looks as if the communication symbols *are* the conceptual model. In other words, in our view the conceptual model is not a description, it is not on paper, not in the interface or in any other symbolic form; the symbols and their interrelations are only a means to convey the model to the user.

The aspects of soundness and completeness suggest that the conceptual model is an ideal theoretical concept that does not exist in reality. Designers also are subject to inconsistencies and inaccurateness with respect to their own design, especially in case of complex applications where a team of designers is involved. The role of the conceptual model is, therefore, pragmatically defined; we accept a particular model as the conceptual model and hope that, with respect to a particular task, this model closely resembles the reality in terms of behaviour and characteristics of the application.

In our approach, we advocate the idea that we need an explicit representation of the conceptual model inside the machine (for similar approaches, see e.g. Ahn et al., 1995; Rich & Sidner, 1998), i.e. a formal counterpart of the conceptual model that is accepted as the expert model; we will refer to this model as the *formal semantic model* (FSM). The FSM contains semantic knowledge with respect to the domain of discourse, for instance, ontological information that helps to catalogue and distinguish various types of objects in the domain, their properties and relationships (Sowa, 2000) and assertional knowledge in terms of the concepts defined in the ontology.[1] The FSM, which is modelled below as a type theoretical context, is assumed to be semantically grounded in the agent's ability to recognise the inhabitants of certain types in the external world. We will distinguish five types of semantic information:

- information about the existence of a particular type (e.g. 'animals exist')
- information about subtypes (e.g. 'mammals are animals')
- information about the existence of predicates that are applicable to a particular type (e.g. 'warm-bloodedness is applicable to animals')
- information about rules that state that an object of a particular type has a necessary feature (e.g. 'all mammals are warm-blooded')
- information about instances of the information above (e.g. 'flipper is a dolphin')

In order to regulate the communicative behaviour of the system, we also introduce a so-called *formal pragmatic model* (FPM). The FPM contains pragmatic knowledge with respect to the user, such as the words that can be used during the dialogue and the beliefs and goals of the user. In fact, the FPM marks the semantic information as part of particular mental states; we will distinguish five types of pragmatic information:

- *Common vocabulary*: the words that can be used in the communication

[1] A knowledge designer may choose from different formalisms to represent the FSM. In semantic web applications, for instance, a popular formalism is description logic. In description logic a so-called TBox contains a set of terminological axioms, and an ABox contains the assertional knowledge (Baader, McGuinnes & Patel-Schneider, 2003). In the type theoretical formalism used in this paper, the distinction between TBox and ABox is determined by the type-assignment of the expressions.

- *Private belief*: the system's subjective beliefs about the domain of discourse
- *Common belief*: that part of the system's belief that it assumes to be shared with the user
- *Pending belief*: the system's belief about the user's belief
- *Goal*: the system's belief about the user's goal(s)

In order to communicate, there needs to be a certain degree of compatibility between the system and the user; this is expressed by the common vocabulary and the common beliefs (see e.g. Ahn, 2001). The common vocabulary contains the terminology of common concepts and enables the system and user to communicate simple messages.

In line with Clark and Schaefer (1989), Allwood, *et al.* (1992) and Traum (1994), we will assume that successful communication requires some degree of conceptual alignment or common ground and that the goal of grounding of the semantic information is a vital activity in cooperative communication. Prior to a conversation, participants not only have beliefs of a particular discourse domain, they also assume that there is some agreement about these beliefs and they augment manifested beliefs to the 'agreed beliefs' during the conversation – these agreed beliefs will be called the *common belief*. In practice it is hard (and, since we have no direct access to our dialogue partners, even impossible) to decide whether common beliefs are really common, but our main point is that dialogue participants act *as if* these beliefs are common. Among other things, the distinction between private and common beliefs enables us to give concrete form to the Gricean maxim of quantity. If relevant, private beliefs can always be manifested, unless they are part of the common beliefs. Common beliefs give us a criterion to leave out particular information in the dialogue move (otherwise we would manifest information that the user already believes).

It is important to note that the information from the FSM cannot be distributed in a random manner over the belief states of the system. For instance, if the system has a common belief that a predicate is applicable to a particular type, the existence of the type should first be introduced in the private beliefs. We will assume that the system's information state is organised in a well-formed manner, i.e. according to the rules of the type system (see also Borghuis, 1994). Also, the words in the common vocabulary correspond with variables (e.g., ideas or concepts) declared in the common beliefs of the system.

System beliefs about the user's beliefs will be represented by a temporary belief state, called *pending belief*, that exists during the process of interpreting the user's message. Pending belief can be considered as a temporary state to separate the user's beliefs from the system's belief. If the information is accepted, it will be transferred to the system's common belief after the system responds to the user's utterance. If the user's assumptions contradict the FSM, they will be kept separate from the system's belief state and rejected if no agreement can be found.

The system's belief about the user's *goal* gives relevance to the communication process and may be considered as the driving force behind the detection process of the potential mismatches.

Before we explain the details of the semantic model in Type Theory, the detection mechanism of mismatches and the conversational strategy in terms of the Gricean maxims, we will first briefly elaborate on the feedback process in dialogue.

3. Feedback Generation in Dialogue

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

When we look at general 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). Also, 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, response times and corrective feedback in case of errors.

Focussing on the linguistic aspects of feedback in human-system interaction, several authors have modified and extended the Gricean maxims and Nielsen’s heuristics at various levels of the communication process. Ainsworth and Pratt (1992), for instance, discuss two feedback strategies at the perceptual level for error correction in speech recognition systems in noisy environments. Based on a classification derived from Allwood, Nivre and Ahlsen (1992) and Allwood (1995), Larsson (2003) presents a human-computer feedback system where the levels ‘contact’, ‘perception’, ‘understanding’ and ‘reaction’ are specified. Bernsen, Dybkjær and Dybkjær (1996) present an extension of the Gricean maxims based on a corpus of transcribed dialogues obtained from Wizard of Oz (WOZ) experiments. They supplement the maxims with several rules, such as ‘partner asymmetry’ (e.g., only one partner has a hearing deficiency), ‘background knowledge’ and ‘repair and clarification’.

Just as the Gricean maxims and their extensions form guidelines for the acceptability of human conversational sequences, Nielsen’s heuristics offer a practical handle for a systematic evaluation of user interfaces. However, both type of rules are underspecified in case an interface

U: Restart the internet, please.

S1: ‘Sorry, I don’t understand you’
S2: ‘What do you mean by ‘restart’?
S3: ‘‘Restart’ is only applicable to processes’
S4: ‘The internet cannot be restarted’
S5: ‘The internet is not a process’
S6: ‘Networks are not processes’
S7: ‘No, because I cannot restart the internet’
S8: ‘Do you believe that ‘restarting’ is applicable to the internet?’
S9: …

Figure 1: The presuppositions of the user’s questions are in conflict with the system’s conceptualisation. Depending on its beliefs, the system has abundant possibilities to respond.

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 (see also Frederking (1996) on an discussion about the vagueness of the Gricean maxims). Suppose, for instance, that user U and computer system S have two disparate conceptualisations and that U makes the following request: 'Restart the internet, please'.[2] The system's FSM contains, among other things, a representation for the words 'internet' and 'restart' and knows that internet is a subclass of networks. S also knows that restarting is only applicable to processes and that networks are not processes. Assuming that our computer system should react in a relevant and truthful way, then what should the response of S be? Clearly, we have abundant possibilities for feedback (see Figure1).

Which utterance is the most adequate one and which rules should be applied in order to generate these feedback sequences depends, among other things, on the FPM (e.g., on what the user and system know about the application domain and, more specifically, on the system's knowledge about the user's conceptualisation). For instance, in S6 the response is inadequate if the system does not believe that the user's mental model does not contain the information that the internet is a network. Another parameter is the role played by the system in the interaction; usually, the system acts as an expert who is unwilling to adjust its own FSM, in S8, however, the system reacts as an equal who seems to be willing to reconsider its domain ontology.

Feedback about conceptual mismatches has been studied by several authors. Joshi (1983), for instance, reviews different types of cooperative responses in question-answer systems and distinguishes between responses with regards to so-called extensional and intensional disparities between the views of the user (U) and the system (S). Extensional disparities are related to the content of the database (e.g. false assumptions by the user), whereas intensional disparities are related to the structure of the database (e.g. assumed non-existing relations by the user). In the end of his paper, Joshi generalizes his findings into the following modification of the Gricean quality maxim (MB is an abbreviation for the common beliefs of system and user):

> *'… it should not be possible for U, from what S has said, say Q, to infer something which S believes to be false. If there is such a possibility then after saying Q, S should provide further information to 'square away' the MB's.'* (Joshi, 1983, pp. 238)

In other words, the system should avoid inferences by U that are inconsistent with the system's beliefs by choosing a more appropriate response or by adding extra information. So, for instance, the system's rejection S7 in Figure 1 may imply that the internet can be restarted in general, but that this particular system is unable to do so. Therefore, in order to avoid the unwanted inference, a more appropriate response would be S4 ('The internet cannot be restarted'). Although Joshi's paper does not contain any details with respect to the generation process of specific responses, the modification of the Gricean rule will play a central role in our paper.

More details of the generation process have been worked out in, for instance, McCoy (1988) and McCoy (1989), where two types of misconceptions were considered: misclassification of objects (e.g., U: 'I thought whales were fish.') and misattributes (e.g., U: 'I thought whales have gills.'). In both papers, a number of response strategies were abstracted from a transcript study and associated with a structural configuration of the user model. McCoy's study of the transcripts

---

[2] The example was taken from www.computerflaters.nl. A cliënt was asking the helpdesk: 'Het Internet is erg langzaam. Kunnen jullie het niet opnieuw opstarten?' ('The internet is very slow. Can't you restart it?').

revealed that a response to a misconception can be viewed as consisting of three parts: 1. a denial of the incorrect information, 2. a statement of the correct information and 3. a justification for the denial and the correct response given. To formulate the appropriate response, the proposed generation process heavily relied on the notions of 'perspective' – i.e. context sensitive information, such as the goal of the discourse – and 'object similarity' – i.e. a similarity metric based on the common and disjoint features of objects involved. Contextual information seems indeed indispensable to an adequate correction process, but the measurement of perspectives and object similarity is rather ad hoc and can hardly be verified in different situations because of a lack of details. Another shortcoming is that McCoy's approach does not include an explicit representation of the user's mental model in terms of various types of beliefs.

4. The Formal Semantic Model in Type Theory

In this paper, the formal semantic model will be expressed in Type Theory (TT). TT, which is actually based on typed λ-calculus, is a powerful logical formalism in the field of theorem proving and programming languages, and there is a growing interest in using TT as a framework for natural language semantics (see e.g., Ranta, 1994; Ginzburg, 2005). In the field of agent communication, TT was used in the DenK-project as a knowledge representation to model various types of beliefs (Ahn, et al., 1995; Bunt, et al., 1998). In the project, an 'intelligent' agent was modelled that supported a human user in its use of a particular domain. Although the system was applied to the domain of an electron microscope, it was intended to be generic in that its architecture and various techniques for knowledge representation and construction were independent of the field of application. The agent's belief states, such as private and common beliefs, were modelled as type theoretical contexts. In the DenK-project the formalism was used to model the ontological assumptions and beliefs about the task domain (the electron microscope) and the cognitive dynamics of the agent's belief state, in particular, the change of beliefs as a result of domain observations and dialogue contributions by the user.

Apart from its intrinsic dynamic properties, TT has important advantages over formalisms such as predicate logic or discourse representation theory. First, and at the heart of type theory, is the formal distinction between objects and types. Types often represent a particular concept, whereas objects can be considered as instances of these concepts. In type theory it can be expressed, for instance, that we have the concept 'network' and that 'the_internet' is an instance of the concept network. Second, TT embodies the notion of contextuality (or sequentiality) that tells us that new assumptions can only be added as long as the current belief state satisfies particular constraints.

To some extent, the notion of sequentiality corresponds to the notion of *presupposition* used in linguistics (Strawson, 1950), where, for instance, the sentence 'Type Theory will be the most prominent logical framework in the near future to model the semantics of natural language utterances' presupposes 'the existence of logical frameworks', 'type theory as a particular instance of these logical frameworks', 'the existence of natural language utterances and their semantics' and so on. Presuppositions can be considered as a kind of background assumptions that should be fulfilled in order to understand the meaning of the message (see also Piwek and Krahmer, 2000). An important characteristic of presuppositions is that they can be inferred from both the sentence and its interrogative form. Since we assume a close relation between the communicative actions of the users and their mental model, presuppositions give us valuable information about the discrepancies between the FSM and the user's model of the application. Hence, we will assume that the system is able to interpret a simple language fragment with words taken from the common vocabulary and that it detects mismatches on the basis of presuppositions derived from the message.

Here we will only give a brief introduction and show how particular semantic information can be expressed in a very limited fragment of TT (see e.g. Ahn, 2001; Luo, 1994 for comprehensive introductions). Private beliefs of an agent can be represented in TT as so called contexts; contexts consist of sequences of expressions and list everything that has been assumed so far by the system with respect to the application domain and everything that has explicitly been inferred from these assumptions. The building blocks in TT are expressions of the form

(1) *G: H*

which indicate that an object *G* has type *H*. These expressions are called statements. We can express, for instance, that a particular object (namely *flipper*) is an inhabitant of type *dolphin*

(2) *flipper: dolphin*

Concepts themselves, such as *dolphin*, are inhabitants of a special type named *sort* (which is denoted by **s*). For instance, the statement

(3) *dolphin: *s*

expresses that *dolphin* is a concept.

The notion of sequentiality plays a role in the order in which statements can be added. For instance, statement (2) cannot be added unless the context already contains statement (3).

Predicates are considered as functions that take a particular concept as an argument and that yield a truth value. For that reason, we also have to introduce propositions; we will do this by introducing a new kind of sort: **p*. For instance, the statement

(4) *warm_blooded: animal → *p*

expresses that the predicate *warm_blooded* takes inhabitants of the type *animal* as an argument and yields propositions as its result. For this we explicitly have to introduce the concept *animal* first:

(5) *animal: *s*

Predicates can be applied to instances of its argument type. The result is of the type 'proposition'. So, for instance, if *tommy* is an instance of *animal*, the statement

(6) *warm_blooded (tommy): *p*

expresses that the application of the predicate *warm_blooded* to *tommy* is of the type 'proposition'. It does *not* yet state that it is a *true* proposition. In type theory, propositions are considered as types themselves and a proposition is true if there exists an object (i.e. the proof) with the proposition as its type. So, the statement

(7) *x: warm_blooded (tommy)*

expresses that the proposition *warm_blooded (tommy)* is true, where *x* denotes the proof object.

Necessary features of objects are modelled by means of the product '*Π*' operator. An inhabitant of the product type

(8) $\Pi x{:}A.P(x)$

is a function that for each inhabitant of type A yields an inhabitant (i.e. a proof object) of the proposition $P(x)$.[3] So, for instance, the statement

(9) $f{:}\ \Pi x{:}dolphin.warm\_blooded(x)$

expresses that f is a function that for each instance x of the type dolphin yields a proof object for the proposition *warm_blooded(x).* In other words, it expresses that all dolphins are warm-blooded. So, for instance,

(10) *f(flipper): warm_blooded (flipper)*

expresses that the proposition *warm_blooded (flipper)* is true, where *f(flipper)* is the proof object.

Inheritance is introduced by the subsumption operator '<', which indicates that inhabitants of a more specific type can be applied in every case where inhabitants of the more general type may be applied (types on the left are lower in the hierarchy):

(11) $mammal < animal{:}\ {*}s$
(12) $dolphin < mammal{:}{*}s$

So, any predicate that take as an argument inhabitants of the type *animal* can take inhabitants of the type *mammal* as well. Similarly, any predicate that take as an argument inhabitants of *mammal* can take inhabitants of *dolphin* as well.

The types $*p$, $*s$, and types like $animal \rightarrow *p$ and $\Pi x{:}dolphin.warm\_blooded(x)$ themselves are inhabitants of the top-level type '□'. So for instance,

(13) $*s{:}\ \Box$

Contexts are open to new information and can be extended as long as new introductions are adhered to the rules of the type system; these contexts are called *legal contexts.* So, given a particular context, the rules of the type system constrain the way in which statements can be combined into new legal statements. This can be expressed by so called judgements:

(14) $\Gamma \vdash G{:}\ H$

which expresses that object G has type H, given the assumptions in context Γ. A context Γ is defined to be *legal* if:

(15) $\Gamma \vdash *s{:}\ \Box$

The formal rules of the type system are as follows (cf. Ahn, 2001; Kievit, 1998):

(16) $\varepsilon \vdash *{:}\ \Box$ (axiom)

where * can stand for $*s$ or $*p$. This axiom defines the empty context (denoted by ε) to be legal.

[3] In fact, the types $\Pi x{:}A.P(x)$ and $A \rightarrow *p$ are both instances of the more general product type '$\Pi x{:}A.B$'. However, for the purposes of this paper we do not need to consider this extra level of abstraction. For more details see (Ahn, 2001) and (Kievit, 1998).

(17) if $\Gamma \vdash A\colon s$ then $\Gamma, x\colon A \vdash x\colon A$ (start1)

(18) if $\Gamma \vdash x\colon {*s}$ and $\Gamma \vdash y\colon {*s}$ then $\Gamma, x<y\colon {*s} \vdash x < y\colon {*s}$ (start2)

where s can stand for $*s$, $*p$ or $\Box$. These rules allow us to extend a context with a statement.

(19) if $\Gamma \vdash A\colon B$ and $\Gamma \vdash C\colon s$ then $\Gamma, x\colon C \vdash A\colon B$ (weakening1)

(20) if $\Gamma \vdash A\colon B$ and $\Gamma \vdash x\colon {*s}$ and $\Gamma \vdash x\colon {*s}$ then $\Gamma, x<y\colon {*s} \vdash A\colon B$ (weakening2)

These rules guarantee that context extensions are monotonic in the sense that statements remain derivable after an extension of the context.

So for instance, according to the above rules the following context is legal:

(21) $*s\colon \Box,\ animal\colon {*s},\ dolphin\colon {*s},\ dolphin < animal\colon {*s},\ flipper\colon dolphin$

(22) if $\Gamma \vdash A\colon {*s}$ then $\Gamma \vdash A<A\colon {*s}$ (subtype1)

(23) if $\Gamma \vdash A < B\colon {*s}$ and $\Gamma \vdash B < C\colon {*s}$ then $\Gamma \vdash A<C\colon {*s}$ (subtype2)

These rules define the subsumption operator to be reflexive and transitive.

(24) if $\Gamma \vdash A\colon {*s}$ then $\Gamma \vdash A \rightarrow {*p}\colon \Box$ (pred)

(25) if $\Gamma \vdash P\colon A \rightarrow {*p}$ and $\Gamma \vdash a\colon C$ and $\Gamma \vdash C<A\colon {*s}$ then $\Gamma \vdash Pa\colon {*p}$ (apply1)

The rule (pred) defines the formation of predicates. The rule (apply1) states that an application of a predicate to an instance of (a subtype) of the argument type is of type ‘proposition’. So, for example, according to these rules the following is a legal extension of (21):

(26) $animal \rightarrow {*p}\colon \Box,\ warm\_blooded\colon animal \rightarrow {*p},\ warm\_blooded(flipper)\colon {*p}$

(27) if $\Gamma \vdash A\colon {*s}$ and $\Gamma, x{:}A \vdash P(x)\colon {*p}$, then $\Gamma \vdash \Pi x{:}A.P(x)\colon \Box$ (prod)

(28) if $\Gamma \vdash f\colon \Pi x{:}A.P(x)$ and $\Gamma \vdash a\colon C$ and $\Gamma \vdash C<A\colon {*s}$ then $\Gamma \vdash f(a)\colon P(a)$ (apply2)

The rule (prod) defines the formation of product types. The rule (apply2) states that an application of an instance f of a product type $\Pi x{:}A.P(x)$ to an instance a (of a subtype) of its argument type, is an instance (i.e. proof object) of the proposition $P(a)$. So, for instance, according to these rules, the following is a legal extension of (21) and (26):

(29) $\Pi x{:}dolphin.warm\_blooded(x)\colon \Box,\ f\colon \Pi x{:}dolphin.warm\_blooded(x),$
$f(flipper)\colon warmblooded(flipper)$

5. Detecting Conceptual Mismatches

In order to provide feedback about conceptual mismatches we need a computational decision criterion that tells us whether information from the user is incompatible with the FSM of the system (Beun, van Eijk & Prüst, 2004). To achieve this we embrace the notion of *presupposition* to compare the incoming language fragment from the user with the (private belief of the) FSM of the system.

For instance, let assume that the private belief of the FSM contains the information that *warm-blooded* has *animal* as its argument type and that *Africa* is an instance of the type *continent*. Suppose the user asks the following question:

U: Is Africa warm-blooded?

This question presupposes that *warm-blooded(Africa)* is a proposition, and thus that the predicate *warm-blooded* is applicable to the object *Africa*. Since *Africa* is of type *continent* and *warm-blooded* has *animal* as its argument type, this presupposes that the type *continent* is a subtype of *animal*. However, since the latter presupposition is not part of the private belief we conclude that there is a conceptual mismatch between the FSM and the user's mental model.

In general, the mechanism for the detection of conceptual mismatches is the following: each presupposition φ of the semantic information of the user input should be derivable from the private belief Γ of the FSM, formally:

(30) $\Gamma \vdash \varphi$

If property (30) does not hold for one or more of the presuppositions then we say that there is a *conceptual mismatch* between the FSM and the user's mental model.

In our framework, we will assume that the presuppositions φ for which property (30) does not hold become part of the *pending belief*. So, formalising the above example, if we have:

(31) Γ = *animal: *s, continent: *s, Africa: continent, warm_blooded: animal* $\rightarrow$ *p

(32) φ = *warm_blooded(Africa): *p*

then $\Gamma \vdash \varphi$ does not hold. The reason for this is that according to the rule (apply1) of the type system, it is required that:

(33) $\Gamma \vdash$ *continent* < *animal: *s*

However, this judgement does not hold. In other words, both the presuppositions *continent* < *animal: *s* and *warm_blooded(Africa): *p* are not part of the private belief but instead of the pending belief.

6. A Conversational Strategy

In the literature various kinds of conceptual mismatches have been identified (see, for instance, Visser *et al.,* 1998; Hameed *et al*., 2002; Agarwal *et al*., 2005). In this paper, we will distinguish two types of mismatches: *direct* and *indirect mismatches*. Direct mismatches result from an incorrect match between the private belief of the system's FSM and the presuppositions derived from the user's message. For example, in the imperative:

U: Restart the internet, please!

the user presupposes that the process 'restart' is applicable to an object called 'the internet', which is in conflict with the system's FSM. If we take the system to be an expert on the semantic domain then in order to resolve the mismatch a cooperative response would involve a denial of the incorrect information, i.e. a denial of the information that is part of the pending belief (cf. Fig. 1). Direct mismatches are a result of the a priori conceptualisation of the domain of discourse by user and system; the assumptions that cause the mismatch are, therefore, part of the semantic content of the linguistic realization.

Indirect mismatches, on the other hand, are not part of the semantic content of the message, but are mismatches that result from the system's answer. They are caused by pragmatic inferences that can be deduced on the basis of the rules of the dialogue game, such as Gricean implicatures. Imagine a situation where a user observes a number of bottles with the description 'toxin'. Suppose that, for whatever reason, the user asks the question:

U: Is this toxin poisonous?

A simple affirmation by the system (e.g. *'Yes'*) triggers an inference by the user that may cause a serious conceptual mismatch, namely that not all toxins are poisonous. The inference can be concluded from the first part of the Gricean maxim of quantity that states that dialogue participants should contribute as much as possible given the goal of the interaction. In fact, the inference is a so-called quantity implicature (Levinson, 1983). If a speaker can contribute a stronger proposition – in this case that toxins are always poisonous – and the stronger proposition also satisfies the other Gricean maxims (i.e. quality, the second part of quantity, relevance and manner), then the speaker should add, in line with Joshi's proposal, extra information. Consequently, if the speaker withholds the stronger information, the hearer may conclude that the information does not hold, especially in cases where the speaker is supposed to be the expert of the discourse domain. So, in order to avoid the discrepancy, the system has to add extra information to the affirmation, for example:

S: Yes, because toxins are always poisonous.

An important question is whether we should always include the extra information. The answer to this question depends on whether the extra information satisfies the remaining maxims:

- Manner: The system believes that the user knows the words 'toxins' and 'poisonous'.
- Quality: The system believes that toxins are always poisonous
- Quantity 2: The system believes that the user does not believe that all toxins are poisonous
- Relevance: The system believes that the information 'toxins are always poisonous' is relevant

The words 'toxin' and 'poisonous' were part of the user's question, so the first maxim holds. Since we have assumed explicitly that the system believes that toxins are always poisonous, the second maxim holds. The same counts for the third maxim: if the user is cooperative, it may be concluded from her question that she believes that toxins can be poisonous or not poisonous. Therefore, she does not believe that toxins are always poisonous (Quantity 2). The relevance maxim cannot be proven and we will, therefore, assume that, if one of the participants wants to know whether an object of a particular type has a particular characteristic, then by default it is always relevant that the participant knows that all these type of objects have that characteristic. It is hard to think of a situation where this is not the case and it certainly holds for the 'toxin'-case. So, in order to avoid the conceptual discrepancy, the system should add the extra information. An important difference between the direct and the indirect mismatch is that in the direct case the user's message is always rejected and that in the indirect case the system's response is always affirmative.

Depending on both the content of the FSM and FPM, even more informative responses may be generated. Let us consider the simple domain of animals again. Suppose that the user asks:

U: Is this dolphin warm-blooded?

Then an adequate response could be:

S: Yes, because all mammals are warm-blooded.

In the response the information that dolphins are a subclass of mammals is included as background information and should be, in order to be cooperative, part of the common belief of the FSM.

Summarising, we opt for the following conversational strategy:

<u>*Conversational strategy*</u>
Give the most informative answer (i.e. the relevance maxim) provided that

- *the answer is expressed in terms of the common vocabulary (i.e. the manner maxim),*
- *the answer is part of the private belief or the denial of the answer is part of the pending belief (i.e. the quality maxim),*
- *the answer is not part of the common belief (i.e. the second quantity maxim: do not say too much), and*
- *all presuppositions of the response is part of the common belief (i.e. the first maxim of quantity: say enough).*

We will illustrate this strategy with some examples. First, we consider the case of indirect mismatches. Let us consider the user question:

U: Is this dolphin warm-blooded?

In Table 1 we have presented the sequence of statements of the FSM that constitute the proof of the proposition *warm-blooded(this_dolphin)*. In the table we have indicated for each of the statements its presuppositions and the part of the FPM it belongs to.

	FSM	**Presuppositions**	**FPM**
0.	proof of *warm_blooded(this_dolphin)*		goal
1.	*animal:*s*		common vocabulary
2.	*mammal:*s*		
3.	*dolphin: *s*		
4.	*warm_blooded: animal* $\rightarrow$ **p*	(1.)	
5.	*this_dolphin: dolphin*	(3.)	
6.	*mammal < animal:*s*	(1.), (2.)	common belief
7.	*dolphin < mammal: *s*	(2.), (3.)	
8.	*f: Πx:mammal.warm_blooded(x)*	(4.), (6.)	private belief
9.	*f(this_dolphin): warm_blooded(this_dolphin)*	(5.), (7.), (8.)	

Table 1: proof table for the generation of the response to the question 'Is this dolphin warm-blooded?'.

According to the above strategy, the answer to the user's question would be:

S: (9.), because (8.)
(S: Yes, because all mammals are warm-blooded.)

because (8.) and (9.) are part of the private beliefs and not of the common beliefs, their presuppositions are part of the common beliefs and ‘mammal’ and ‘warm-blooded’ are part of the common vocabulary.

Note that if the information (7.) *dolphin < mammal: *s* would have been part of the private belief instead of the common belief then the system’s response would have been:

S: (9.), because (8.) and (7.)
(S: Yes, because all mammals are warm-blooded and dolphins are mammals.)

Let us now consider the case that the type ‘mammal’ is not part of the common vocabulary but of the private belief. In particular, let us assume that the distribution of the semantic information over the FSM is as depicted in Table 2.

	FSM	**Presuppositions**	**FPM**
0.	proof of *warm_blooded(this_dolphin)*		goal
1.	*animal:*s*		common vocabulary
2.	*dolphin: *s*		
3.	*warm_blooded: animal* $\rightarrow$ **p*	(1.)	
4.	*this_dolphin: dolphin*	(2.)	
5.	*dolphin < animal: *s*	(1.), (2.)	common belief
6.	*mammal: *s*		private belief
7.	*mammal < animal: *s*	(1.), (6.)	
8.	*dolphin < mammal: *s*	(2.), (6.)	
9.	*f: Πx:mammal.warm_blooded(x)*	(3.), (7.)	
10.	*f(this_dolphin): warm_blooded(this_dolphin)*	(4.), (8.), (9.)	

Table 2: Proof table for the generation of the response to the question ‘Is this dolphin warm-blooded?’, where the word ‘mammal’ is not part of the common vocabulary.

In this situation, the private belief ‘all mammals are warm-blooded’ is not expressible in terms of the common vocabulary. The strongest information that *is* expressible in the common vocabulary is the information ‘all dolphins are warm-blooded’. Hence, according to the conversational strategy the system responds with:

S: (10.), because (9. plus 8.minus 6.)
(S: Yes, because all dolphins are warm-blooded.)

Let us now address the case of direct mismatches. Consider again the user input:

U: Restart the internet, please!

To be able to understand the input we test whether there are no (direct) conceptual mismatches with the FSM, i.e. whether the presupposition *restart(the_internet):*p* is part of the private beliefs of the FSM. In Table 3 we have depicted for some specific FSM and FPM the results of this test.

	FSM	Presuppositions	FPM
0.	restart(the_internet): *p		goal
1.	process: *s		common vocabulary
2.	internet: *s		
3.	network: *s		
4.	restart: process → *p	(1.)	
5.	the_internet: internet	(2.)	
6.	internet < network: *s	(2.), (3.)	common belief
7.	network < process: *s	(1.), (3.)	pending belief
8.	restart(the_internet): *p	(5.), (6.), (7.)	

Table 3: Proof table for the generation of the response to the imperative ‘Restart the internet, please!’.

Since statements (7.) and (8.) are presuppositions that are part of the pending belief (and thus not part of the private belief) there is a direct conceptual discrepancy. According to the conversational strategy these presuppositions should be denied in the response:

S: (not 8.), because (not 7.)
(S: It is impossible to restart the internet, because networks are not processes.)

Note that as a result of different type-assignments (‘*s’ in (7.), ‘*p’ in (8.)), different terms are used in the linguistic realization for the denial of the incorrect presupposition (‘are not’ and ‘is impossible’, respectively).

7. Conclusion

From a point of view of human-computer interaction developers, one of the challenges is that systems are designed in such way that they support the user’s acquisition of an appropriate mental model in order to avoid errors while performing on them. In this paper, we have presented a framework that enables a computer system to generate speech act sequences in case of mismatches between the system’s semantic model of the application domain and the user’s mental model. For that we have modelled the system’s model in Type Theory and we have distinguished various types of information states to regulate the system’s communicative behaviour.

We expected that including natural conversational skills in interfaces may contribute to a more efficient and satisfactory human-computer interaction. Due to its complexity, natural language interaction with computer systems is usually avoided and, if included, highly underdeveloped. In this paper, we have taken a strong theoretical stance, since we believe that a fundamental approach to communication is necessary to solve important problems in human-system interaction. In order to determine what humans actually do in realistic conversational circumstances and to validate the model presented in this paper, the acquisition of empirical data is a necessary step in the research process, however. We will, therefore, collect empirical data from help desk conversations between experts and naive users. We expect that the analysis of the transcripts will lead to a further extension and refinement of the model, such as the inclusion of an ambiguous communication vocabulary and richer semantic descriptions. In order to prevent irrelevant linguistic realizations in the generation process, we believe that including the topic of the conversation should also be taken into consideration (cf. McCoy, 1989).

Although admittedly still incomplete, the presented framework provides a simple and elegant solution to empower a computer system to generate adequate feedback utterances at the conceptual level in human-computer interaction. It remains for the future, however, in how far the introduced model will be sufficiently rich to take care for an adequate feedback process between humans and computer systems at the conceptual level.

8. References

Agarwal, P., Huang, Y. & Dimitrova, V. (2005). Formal approach to reconciliation of individual ontologies for personalisation of geospatial semantic web. In: the *First International Conference on GeoSpatial Semantics (GeoS 2005).* Volume 3799 of Lecture Notes in Computer Science. Heidelberg: Springer-Verlag: 195-210.

Ahn, R.M.C. (2001). *Agents, objects and events: A conversational approach to knowledge, observation and communication.* PhD-thesis. Eindhoven University of Technology.

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

Ainsworth, W.A. & Pratt, S.R. (1992) Feedback strategies for error correction in speech recognition systems. *International Journal of Man Machine Studies, 36*. 833-842.

Allwood, J., Nivre, J., & Ahlsen, E. (1992) On the Semantics and Pragmatics of Linguistic Feedback*. Journal of Semantics, 9*. 1-26

Allwood, J. (1995) An activity based approach to Pragmatics. *Gothenburg Papers in Theoretical Linguistics, Technical Report (GPTL) 75*, University of Göteborg.

Baader, F., McGuinnes, D.L. & Patel-Schneider, P.F. (2003) *The Description Logic Handbook: Theory, Implementation and Applications.* Cambridge: Cambridge University Press.

Bernsen, N.O., Dybkjær, L. & Dybkjær, H. (1996) Cooperativity in Human-Machine and Human-Human Spoken Dialogue. *Discourse Processes, 21*, 213-236.

Beun, R.J., Eijk, R.M. van & Prüst, H. (2004). Ontological feedback in multiagent systems. In *Jennings, N.R., Sierra, C., Sonenberg, L. & Tambe, M. (eds.) Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, p. 110-117. New York: ACM press.

Borghuis, V.A.J. (1994) *Coming to Terms with Modal Logic. On the Interpretation of Modalities in Typed λ-Calculus*. PhD-thesis. Eindhoven University of Technology.

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.

Bunt, H.C. Ahn, R.M.C., Beun, R.J., Borghuis, T. & van Overveld, K. (1998) Multimodal Cooperation with the DenK-system. In H.C. Bunt, R.J. Beun & T. Borghuis (eds.) *Multimodal Human Computer Communication. Volume 1374 of Lecture Notes in Computer Science*. Berlin: Springer-Verlag. 39-67.

Clark, H.H. & Schaefer, E.F. (1989) Contributing to Discourse. *Cognitive Science, 13*. 259-294.

Frederking, R. E. (1996) Grice's Maxims: 'Do the Right Thing'. Presentation at the *Computational Implicature workshop at the AAAI-96* Spring Symposium Series, Stanford.

Grice, H. (1975). Logic and conversation. In *Cole, P. & Morgan, J. (eds.) Speech acts. Syntax and Semantics*, volume 11. New York: Academic Press. 41-58.

Ginzburg, J. (2005) Abstraction and Ontology: Questions as Propositional Abstracts in Type Theory with Records. *Journal of Logic and Computation, 15(2):*113-130

Hameed, A., Sleeman, D., Preece, A. (2002). Detecting mismatches among experts' ontologies acquired through knowledge elicitation. *Knowledge-Based Systems* 15: 265–273

Joshi, A.K. (1983) Varieties of Cooperative Responses in Question-Answer Systems. In: F. Kiefer (ed.) *Questions and Answers*. Dordrecht: Reidel Publishing Company. 229-240.

Larsson, S. (2003) Generating Feedback and Sequencing Moves in a Dialogue System. In: Freedman and Callaway (eds.): *Natural Language Generation in Spoken and Written Dialogue.* Papers from the 2003 AAAI Symposium. AAAI Press, Menlo Park, California. 79-84.

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

Luo, Z. (1994) *Computation and Reasoning: A Type Theory for Computer Science*. Int. Series of Monographs on Computer Science. Oxford: Clarendon Press.

Kievit, L. (1998) Context-*Driven Natural Language Interpretation.* PhD-thesis, Tilburg University, The Netherlands.

McCoy, K.F. (1988) Reasoning on a Highlighted User Model to Respond to Misconceptions. *Computational Linguistics, Vol. 14(3)*. 52-63.

McCoy, K.F. (1989) Generating Context-Sensitive Responses to Object-Related Misconceptions. *Artificial Intelligence 41.* 157-195.

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

Norman, D.A. (1983) Design rules based on analyses of human error. *Communications of the ACM* 26(4): 254-258

Piwek, P. & Krahmer, E. (2000) Presuppositions in Context: Constructing Bridges. In: P. Bonzon. M. Cavalcanti and R. Nossum (eds.) *Formal Aspects of Context, Volume 20 of Applied Logic Series*. Dordrecht: Kluwer Academic Publishers.

Ranta, A. (1994) *Type Theoretical Grammar*. Oxford: Oxford University Press.

Rich, C. & Sidner, C. L. (1998) COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction* 8(3/4): 315-350.

Sowa, J.F. (2000). *Knowledge representation: Logical, philosophical, and computational foundations.* Pacific Grove, CA: Brooks Cole Publishing Co.

Spink, A. & Saracevic, T. (1998). Human-computer interaction in information retrieval: Nature and manifestation of feedback. *Interacting with Computers, 10.* 249-267

Strawson, P. (1950) On Referring. *Mind, 59*. 320-344.

Traum, D. R. (1994) *A Computational Theory of Grounding in Natural Language Conversation*. Ph.D. Thesis. Technical Report 545, University of Rochester.

Visser, P., Jones, D., Bench-Capon, T. and Shave M. (1998) Assessing heterogeneity by classifying ontology mismatches. In: N. Guarino (eds.) *Formal Ontology in Information Systems*, IOS Press: Amsterdam.148–162.