Conceptual Mismatches and Repair in Human-Computer Interaction

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

Abstract. We present a computational framework 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 ontology. Subsequently, in the generation phase, an adequate feedback message is produced in order to resolve detected discrepancies. We provide a conversational strategy that is based upon Gricean implicatures and a distinction between three types of beliefs: private beliefs about the domain of discourse, beliefs about the beliefs of the other and beliefs about the shared beliefs.

Keywords: conceptual alignment, ontologies, mental models, dialogue game, feedback

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, for instance, a user should know that there are things 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 analyse some of the basic system requirements that pertain directly to the above natural conversational skills. For that, we will present the main building blocks and a computational method that enables us to generate feedback utterances that regulate the repair of conceptual discrepancies between a computer system and its user. The remainder of this paper is organised as follows. First, in section 2, we consider the characteristics of the conceptual model of the user and that of the system. Sections 3 and 4 describe the relevant building blocks of the conceptual alignment process. In section 5, we present a conversational strategy that enables us to generate adequate feedback utterances. We wrap up in section 6 with some conclusions and directions for future research.

2. Mental and conceptual model

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 system or application build their own conceptualisation of the characteristics and the behaviour of that system. 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 system 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 'Is this file running?', she believes, among many other things, that files exist and that they can be running; 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. Also, unintentional non-verbal behaviour may 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.

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; below we will refer to this model representation as the *ontology* of the application. The ontology explicitly states the conceptualisation of a particular domain in a formal language. It 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 (Sowa, 2000). An ontology also enables the computer system to reason about various aspects of the domain.

3. Conceptual alignment

From a point of view of human-computer interaction developers, the challenge is that systems are designed in such way that they support the acquisition by the user of an appropriate mental model and to avoid errors while performing them (van der Veer & Puerta Melquizo, 2003). A system and, probably even more important, its interface should be designed in such a way that the user is able to apply human mechanisms of reasoning and of reuse of available knowledge. Since the user's mental model of the application is usually partial, inconsistent and subject to change, however, conceptual discrepancies with the system's ontology are more the rule than the exception.

In order to avoid miscommunication and task errors, the detection and repair of these flaws is of crucial importance. But how does a system become aware that the user's mental model deviates from the system's ontology and how can these flaws be repaired? In other words, how do the user and the system achieve an alignment of the system's and the user's model?[1]

Here we accept conceptual mismatches between computer and user, and advocate the view that conceptual alignment is established by means of adequate speech act sequences generated by the computer system. We will adopt a dialogue game approach, where the system has a dynamic mental state that contains information about the domain of interest and about its conversational partner. The system's belief state is divided into three parts: its private beliefs about the domain of discourse (i.e. the conceptual model of the application domain), its beliefs about the beliefs of the user and its beliefs about the shared beliefs of system and user; it is assumed that the system's shared belief is a subset of its private belief.

In the alignment process two phases are distinguished: a. *detection* and b. *resolution* of the mismatches. Agents may detect mismatches by, for instance, lexical gaps or inconsistencies that emerge during the conversational process. Subsequently, depending on the type of discrepancy, agents may use different conversational strategies for the resolution process. In (Beun, van Eijk & Prüst, 2004), we have focussed on the detection phase. In this paper we concentrate on the resolution part, in particular the generation of adequate feedback and the conversational strategy based on Gricean implicatures. We will assume that the system is able to interpret a simple language fragment and detects mismatches on the basis of particular inferences drawn from the message. These inferences will be called 'presuppositions', a kind of background assumptions that should be fulfilled in order to understand the meaning of the message. Presuppositions will be compared with the system's ontology and feedback will be generated on the basis of an incorrect match between the two types of information. In fact, presuppositions will constitute a so-called 'pending stack', i.e. the system's belief about the user's belief and will, if accepted, be transferred to the system's shared beliefs after the system responds to the user's utterance. Before we will describe the details of the alignment process, we first elaborate on the relevant aspects of the feedback process.

4. Feedback

In both human-system and human-human 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

[1] It is interesting to note that problems of model alignment also appear in high level system-system communication. Systems developed by multiple parties may have different ontologies of the same domain that dramatically hamper the communication process. There is a range of approaches to achieve ontological alignment. On the one side, developers can agree in advance upon a standard domain ontology and embed it in all future ontology design. On the other side, the existence of ontological variations may be accepted and a dialogue mechanism is designed that solves discrepancies during the communication process. The latter agrees with our approach.

behaviour and the application of cybernetic models to human communicative activities has only a limited scope of relevance (Spink & Saracevic, 1998).

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.

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, these rules are still vague and not all people follow them 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.

Just as the Gricean maxims form guidelines for the acceptability of human conversational sequences, Nielsen’s 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. Suppose, for instance, that user U and computer system S have two disparate conceptualisations and that U asks the question: ‘Is this file running?’ S’s ontology contains, among other things, a representation for the words ‘file’ and ‘running’ (and knows which file is referred to) and knows that files are a subclass of items. S also knows that running is only applicable to processes and that processes are not items. Assuming that our computer system should be relevant and truthful, then what should the response of S be? Clearly, we have abundant possibilities for feedback (see Figure1).

U: Is this file running?

S1: ‘Sorry, I don’t understand you’
S2: ‘What do you mean by ‘running’?
S3: ‘Running is only applicable to processes’
S4: ‘Running is not applicable to files’
S5: ‘Files are not processes’
S6: ‘Items are not processes’
S7: ‘Do you think that running is applicable to files?’
S8: …

Fig. 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.

Which utterance is the most adequate one and which rules we should apply in order to generate these feedback sequences depends, for instance, 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 files are items. 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 ontology, in S7, however, the system reacts as an equal who seems to be willing to reconsider its domain ontology .

The distinction between the various types of belief 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 shared beliefs.

Shared beliefs give us a criterion to leave out particular information in the dialogue move (otherwise we would include information the user already believes). And, finally, the pending stack gives a criterion to manifest particular information, in particular if the presuppositions in some way contradict the system's belief.

5. A conversational strategy

In order to avoid an enormous diversity of user input types, we restrict ourselves in this paper to two types of questions: 1. questions of which the presuppositions in some way contradict the system's ontology and 2. questions of which the direct answer may cause disalignment between the system and the user's belief. An example of type-1 questions was given in Figure 1. We will now concentrate on type-2 questions. The classification of type-2 questions is motivated by the Gricean maxim of quantity. To explain this, let us first present an example.

Imagine a situation where the 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 discrepancy, 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 and relevance)[2], then the speaker should do so. 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:

- 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

Since we have assumed explicitly that the system believes that toxins are always poisonous, the first maxim holds. The same counts for the second 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, all the maxims are satisfied and, therefore, in order to avoid the conceptual discrepancy, the system should add the extra information.

Depending on the domain ontology and the beliefs of the participants, even more informative responses may be generated. Suppose that we have a domain of animals consisting of mammals and reptiles, and the user asks:

U: Is this dolphin warm-blooded?

[2] Here, we abstract from the maxim of manner and concentrate on the informational content.

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 therefore be part of the system's shared beliefs. Hence, we opt for the following strategy:

> *Conversational strategy*
> *Give the most informative answer, provided that the answer is part of the private beliefs (i.e. the quality maxim) and is not part of the shared beliefs (i.e. the second quantity maxim: do not say too much) and that all background information of the response is part of the shared beliefs (i.e. the first maxim of quantity: say enough).*

Apart from the distribution of the information over the various belief sources, also the type of information may be crucial in determining the correct feedback. For that we distinguish five types of information[3]:

- 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. 'x308 is a dolphin')

Note that these information types cannot be distributed in a random manner over the belief states of the system. For instance, if the system has a shared belief that a predicate is applicable to a particular type, the existence of the type should first be introduced in the shared beliefs. We will assume that the system's belief state is organised in a well-formed manner, i.e. according to the rules of the type system (Beun, van Eijk & Prust, 2004).

Now, suppose the system's beliefs are distributed in the following way over the shared and private beliefs (SB and PB, respectively)[4]:

SB:

- Animals, mammals, dolphins and reptiles exist
- Mammals and reptiles are a subtype of animals, dolphins are a subtype of mammals
- Warm-bloodedness and cold-bloodedness are applicable to animals
- Animals are either warm-blooded or cold-blooded
- x308 is a dolphin (where x308 is a pointed object)

PB:

- All mammals are warm-blooded (and, therefore, dolphins are warm-blooded) and reptiles are cold-blooded

Suppose the user asks the question:

U1: Is this dolphin warm-blooded?

where 'this dolphin' refers to the pointed object x308. Below we have presented the sequence of statements that yields a proof that 'this dolphin is warm-blooded' (behind brackets we have added the background statements):

[3] These types can be expressed in type theoretical formulas. For reasons of legibility, however, we will refrain from giving a formal account here (see Beun, van Eijk & Prüst, 2004 and Ahn, 2001 for further details).

[4] Since we focus in this paper on the generation of answers to questions we assume that the inference process is given.

a. animals exist
b. mammals are animals (a)
c. dolphins are mammals (b)
d. x308 is a dolphin (c)
e. warm-bloodedness is applicable to animals (a)
f. all mammals are warm-blooded (b, e)
g. x308 is warm-blooded (c, d, f)

We assume that if one of the background statements is part of the private beliefs, then the conclusion is also part of the private beliefs. This implies that, since f. is part of the system's private beliefs, g. is also part of the private beliefs. So, we have:

SB: a, b, c, d, e and PB: f, g.

According to the above strategy, the answer to the user's question *U1* would be:

S1: Yes, because all mammals are warm-blooded.

since f. and g. are part of the private beliefs and not of the shared beliefs and their background statements are part of the shared beliefs.

We make one final amendment to the conversational strategy. In the above situation, the generated response consists of all the private beliefs of the proof (i.e., f. and g.). In other situations, however, only a portion of the private beliefs is to be given. To explain this, let us consider the following distribution of the statements over the shared and private beliefs:

SB: a. animals exist
b. dolphins are animals (a)
c. x308 is a dolphin (b)
d. warm-bloodedness is applicable to animals (a)

PB: e. mammals are animals (a)
f. dolphins are mammals (e)
g. all mammals are warm-blooded (d, e)
h. x308 is warm-blooded (c, f, g)

This situation differs from the previous one in that the existence of mammals is not part of the shared but part of the private beliefs. In this situation, an adequate response to the user's question *U1* would be:

S2: Yes, because all dolphins are warm-blooded.

Note that this is not the most informative answer (i.e., a more informative answer would for instance be "Yes, because dolphins are mammals and all mammals are warm-blooded."). It is however as informative as necessary in that it avoids the hearer drawing any invalid quantity implicature. For instance, the information 'not all mammals are warm-blooded' is *not* an implicature of *S2* for the simple reason that 'mammals are animals' (and therefore also 'dolphins are mammals') is not part of the shared beliefs. Therefore, no information about the warm-bloodedness of mammals is to be included in the response. In order to cover this issue, we opt for adding one extra condition to the conversational strategy:

> *Amendment to the conversational strategy*
> *... and provided that the answer involves types that are part of the shared beliefs.*

This condition implies that in the response we can only refer to instances and subtypes of a particular type if this type is part of the shared beliefs. So for instance, in generating the response S2 from the above proof, we exclude the statements e. and f. and weaken g. to ‘all dolphins are warm-blooded’ because the type mammals is not part of the shared beliefs.

6. Conclusions

In this paper, a computational framework was presented for the generation of feedback utterances in a dialogue between a computer system and its user. For that, we included an explicit representation of the application domain, a so-called ontology, and developed a conversational strategy to establish alignment between the system’s ontology and the user's mental model. This conversational strategy is based upon Gricean implicatures and a distinction between three types of beliefs: private beliefs about the domain of discourse, beliefs about the beliefs of the other and beliefs about the shared beliefs.

In the future, we will extend the basic framework to richer ontologies and we will take into consideration the topic of the conversation (cf. McCoy, 1989). Another important aspect is the acquisition of empirical data to determine what humans actually do in realistic conversational circumstances. Although we have taken a strong theoretical stance, we believe that the various aspects described in this paper form the minimal ingredients that enable a system to generate adequate feedback utterances at the conceptual level in human-computer interaction.

Bibliography

Ahn, R.M.C. (2001). *Agents, objects and events: A conversational approach to knowledge, observation and communication.* Doctoral dissertation. 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.

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.

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

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

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

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.

Veer, G.C. van der & Puerta Melguizo, M.C (2003). Mental models. *In: J.A. Jacko & A. Sears (eds.) The human computer interaction handbook*, p. 52-80. Mahwah: Lawrence Erlbaum.