Dialogue pragmatics and context specification

Harry Bunt

1 Introduction

Pragmatics is commo,nly understood to be concerned with studying the relations between linguistic phenomena and properties of the context of use. The understanding of these relations is important in many areas of theoretical and applied research, from grammatical analysis to sociolinguistic field studies. One area where the importance of these relations has become particularly clear is the design of language understanding systems. Such systems are extremely limited, brittle, and unpractical if they do not have powerful ways to make use of contextual information in computing the meanings of utterances. The question of how this can be achieved in an effective and principled way forms one of the major obstacles in building such systems. Computational pragmatics, the study of how contextual information can be effectively brought to bear in language understanding and production processes, hopes to contribute to removing this obstacle.

One way in which contextual information is needed in language understanding, is in making it clear in what way ambiguous and vague expressions should be interpreted. For interactive language understanding systems, the relation between language and context is of an even more fundamental nature, as the very notion of language understanding involves the construction and maintenance of contexts that change as a result of the interaction. The idea that understanding involves context change is closely related to views of linguistic behaviour in terms of actions, speech act theory being paramount among those. It has been suggested that a definition of illocutionary force, the central concept of speech act theory, in terms of context changes would be the best way to further develop the theory (Gazdar 1981; see also Levinson 1983). We will argue that a context-change approach can indeed solve a number of difficulties that speech act theory has to face when applied to the analysis of realistic dialogues. When we construe context to be the totality of conditions that influence the understanding and generation of linguistic behaviour, and note that a communicative action obviously changes these conditions, we must conclude that a communicative action operates on a given context to produce a new one. A context-change framework thus provides a natural way to ground action-based approaches to communication. Whether this approach is fruitful depends crucially on whether we can put our hands on a notion of context that is both sufficiently powerful to form an adequate basis and sufficiently restricted to be manageable. The work reported in this chapter aims to contribute to the establishment of such a notion.

Understanding the linguistic behaviour of a dialogue partner implies, in an action-based approach, understanding the underlying motivation. This simple observation has important consequences for the relation between *utterance meaning* and *dialogue mechanisms*: on the one hand, the interpretation of utterance meanings as intended context-changing operations tells us why utterances are performed, while on the other hand the mechanisms explaining why a dialogue may develop the way it does, provides insight into the functional aspects of utterance meaning. We therefore believe that the study of utterance meanings and dialogue mechanisms is most fruitfully pursued in combination, within a single theoretical framework. Starting with Bunt (1989), we have been developing such a framework, called *Dynamic Interpretation Theory*. Sections 2–3 review the essentials of this framework, comparing it with other action-based approaches to language.

The consideration of utterance meaning from a context-change point of view brings a focus on those aspects of context that *can* be changed through communication. We call those aspects 'local', in contrast to 'global' aspects, that cannot be changed through communication and tend to remain constant throughout a dialogue. Section 4 is concerned with the kinds of local context information that should be taken into account in information dialogues, by analysing the semantics of dialogue acts of the various categories that are found in these dialogues.

Section 5 analyses the logical properties of the various types of local context information as a step toward their formal and computational modelling, looking in particular at their logical complexity, depth of recursion, and time dependence. In section 6 we outline two formalisms that seem promising for representing the most complex types of local context information, viz. constructive type theory and modular partial models.

Section 7 concludes this chapter, reviewing the main points that have been discussed in relation to context modelling, dialogue analysis, and the contextual interpretation of utterances.

2 Action-based approaches to language and dialogue

2.1 Speech act theory

Speech act theory has been a major source of inspiration for all action-based approaches to language, and has been fruitful both in the development of pragmatics and as an conceptual framework for thinking about human-computer dialogue. Still, there are several important points where speech act theory is conceptually not quite satisfactory, or is not entirely adequate for being applied to real dialogue and to dialogue system design.

1. In speech act theory, a central goal of utterance interpretation is taken to be the assignment of an illocutionary force and a propositional content. But there is considerable unclarity as to exactly which illocutionary forces should be distinguished, and why. We therefore question whether illocutionary forces are a satisfactory end point in the analysis of utterance meaning.

   A great deal of work in speech act theory has been concerned with such questions as: *Which illocutionary forces should be distinguished?* and *How can they be grouped into taxonomies?* Taxonomies have been provided based on a characterization in terms of 'illocutionary point', 'direction of fit', and a number of further aspects (Searle 1975b), which shows, in fact, that notions such as illocutionary point and direction of fit are the more basic semantic concepts, while illocutionary force is a defined notion. In Dynamic Interpretation Theory we will take a different course, introducing the notion of 'communicative function', which is similar to illocutionary force, but providing a semantic definition of this notion in terms of context changes.

2. Standard speech act theory suggests, although this is often left unclear, that every utterance corresponds with one illocutionary act, except in the case of indirect speech acts. It is thus customary to speak of 'the illocutionary force' of an utterance. We believe that communication has many 'dimensions' that a speaker can address simultaneously, and that utterances should often be considered to have several functions at the same time. We think it is therefore also more fruitful, in many cases, to consider an utterance as multifunctional rather than as (functionally) ambiguous.

   The standard treatment of indirect speech acts considers an utterance as having, besides its 'directly' expressed meaning, another illocutionary act as an additional meaning. For instance, Searle (1975a) analyses *Can you pass me the salt?* as being, in addition to a question about the hearer's ability to pass the salt, also a request to pass the salt. We believe that this analysis is unsatisfactory, however, since the indirectly expressed request is not quite the same illocutionary act as a direct request. In a direct request the speaker *presupposes* that the hearer has the ability to pass the salt; in the indirect formulation this condition is *examined.* We therefore prefer to analyse such utterances not as expressing additional illocutionary acts, but as having *additional intentions.* (See below, section 3.3.)

3. Although speech act theory by its very nature considers the interactive use of language to be of primary importance, it has curiously little to say about utterances that are characteristic for spoken dialogue. Pervasive phenomena in spoken dialogue, such as the use of feedback utterances (*OK, Quite so, Yes, Hm, You think so?,..*), hesitations, self-corrections, greetings, contact and attention signals, and apologies, have not been analysed in a speech act theoretical way to any great depth although they are considerably more common than the promises and performative sentences which enjoy popularity in the speech act literature. Of many of these rather neglected utterance types, speech act theory tells us little more than that they can be classified as 'expressives' – which is not very useful.

4. Finally, for application in the design of dialogue systems, we need a formalized theory taking into account precisely those types of communicative acts that are relevant in the situation where the system is to be used. Such a theory should on the one hand be based on general principles, like speech act theory and the communicative activity analysis approach we will discuss next, but should on the other hand also acknowledge that the set of communicative action types to be considered depends on the social environment, the linguistic community, the use of media, the kind of task for which the communication is to be instrumental, the precise (e.g., temporal) relations between the underlying task and the communicative activity, and so on.

2.2 Communicative Activity Analysis

A very general action-based approach to language has been developed over the years by Allwood and co-workers, called *Communicative Activity Anal-*

ysis (CAA: Allwood 1976, 1994, and in this volume; Allwood, Nivre, and Ahlsén 1990). Like speech act theory, CAA takes the view that communication is action, which in turn is seen as constituted by a combination of behavioral form, intention, context, and result. CAA provides a conceptual analysis of action, social activity, cooperation, and ethics in communication, with considerable depth and generality.

According to CAA, any type of human activity has four levels of organization: physical, biological, psychological, and social. CAA focuses on the psychological and social levels. At the psychological level, two sublevels are distinguished: (a) perception, understanding, and emotion; (b) motivation, rationality and agency. Motivation, rationality, and agency are subsequently used to give an analysis of ethics and cooperation in communication. Cooperation is characterized by the following conditions on the agents involved:

1. they attempt to perceive and understand the other's actions;
2. they have a joint purpose;
3. they act ethically: they don't lie, don't impose on each other, etc.;
4. they trust each other with respect to 1-3.

At the social level, four sublevels are distinguished: (a) culture and social institutions; (b) language and linguistic communities; (c) social activities and roles in activities; (d) communication. Communication is thus considered as a *level of organization of social activity.* Social activity in turn is characterized by four parameters: (1) purpose, function or type; (2) roles (rights, obligations); (3) instruments (machines, media); (4) other physical environment. These parameters are assumed to be involved at all four levels of organization in social activity, in particular in the communicative level. At this level, the notion of 'purpose' or 'function' is constituted by the communicative intentions associated with utterances. Communicative acts are defined as follows: *"A communicative act may be defined as a contribution or a feature/part of a contribution which can be connected with a communicative intention (purpose, goal, function) or a communicative result."* (Allwood, this volume). Three types of function of communicative acts are distinguished:

- Own communication management (OCM) – enabling an agent to produce and edit his contributions;

- Interactive communication management (ICM) – enabling an agent to manage the interaction with respect to such aspects as feedback, sequencing, and turn management;

- other communicative functions, such as questioning, asserting, and requesting.

Every utterance (or 'contribution') is assumed to have a functional structure with three components: (1) functions obligated by the preceding discourse; (2) functions obligating for the succeeding discourse; (3) 'optional' functions, which are neither obligated not obligating. Obligations are analysed as deriving either from general rational and ethical requirements on communication, from ICM requirements, or from the interaction between the goals of non-management directed communicative acts and the embedding activity context.

Concerning the latter source of obligations, it is claimed that communicative intentionality has two aspects: an *expressive* aspect, which is to express a certain attitude (belief, desire, intention,..), and an *evocative* aspect, which is to evoke a certain reaction from the addressee. For instance, the expressive intention of a questioning act is to express a desire for information, and the evocative to get the listener to provide the desired information. On the assumption that dialogue participants are ethical, cooperative, motivated rational agents, it is argued that addressees of dialogue acts should continually evaluate their willingness and ability to continue, to perceive, to understand, and to comply with the evocative intention, and should respond in accordance with the result of this evaluation. The various kinds of obligations created by communicative acts and by the embedding activity context, in particular the pairing of obligating and obligated aspects of communicative acts, are assumed to be responsible for the dependencies and regularities that may be observed in dialogues.

We are in full agreement with most of the conceptual analyses provided by CAA, and indeed much of it underlies the concepts of DIT as well. For the analysis of phenomena in human - human conversation, CAA offers a rich conceptual framework, but as a basis for designing computer dialogue systems, we believe CAA not to be sufficiently concrete in some respects. More specifically, some points of criticism on CAA as it stands are the following.

1. Perhaps because of its broad character, the conceptual analysis in CAA is often expressed in rather broad terms, leading to characterizing a concept by listing a number of 'aspects' or things that are 'involved', where it is often unclear to what extent the listings are meant to be exhaustive, or in what way something is 'involved'.

2. The CAA taxonomy of communicative functions into OCM, ICM, and other functions, is not very satisfactory. First, it is rather ugly to have

a rest category to which many of the functions of communicative acts belong. Second, the distinction between OCM and ICM functions at top level seems to us debatable; it may be argued that there are more important distinctions to be made between classes of communicative functions than the OCM/ICM distinction, and notably, that some of the distinctions made within the category of ICM functions (where subcategories of feedback functions, turn management functions, and sequencing functions are distinguished), are better made at a higher level than the ICM/OCM distinction. We will suggest a different and more detailed classification, supported both by conceptual analysis and by the similarity of the type of context information addressed by different types of communicative acts.

3. 'Evocative intentions' play a fundamental role in CAA, being primarily responsible for the reactions that a communicative action evokes. We believe that it is preferable to assume that communicative acts have a *goal*, which the speaker is expressing as part of his expressive intention, such that hearers determine their response, in accordance with the assumptions of rationality and cooperation, based on their understanding of this goal and the current context.

   For instance, consider questioning acts again. In DIT, we assume that an agent S who has the goal to have the information X, when asking a question with the corresponding content to H, relies on H's cooperativity in order that H, recognizing S's goal, should act so as to satisfy this. This does not necessarily mean that H should recognize an evocative intention that H provide X, since any action that H may do to satisfy S's goal would be adequate. For instance, if S asks *What time is it?*, it would be adequate for H to point at the clock just behind S, even if H himself, not wearing his glasses, is unable to see what time it is. The adequacy of this reaction would be hard to explain on the basis of evocative intentions. We simply assume that, having understood the speaker's goal, a cooperative dialogue agent will determine (rationally) what actions best to perform to help the speaker achieve his goal. This gives more options to the hearer than mere recognition of the evocative intention.

4. Although CAA stresses the fact that context is important, the precise relations between communicative behaviour and context remain rather vague. For example, the fact that what counts as an appropriate reaction to a request depends, as we have just seen, on the context and not just on the recognition of evocative intentions, does not play a

clear role in CAA. Context in CAA is in the first place the embedding social activity, and the obligations created by communicative activity. But CAA remains silent about other aspects of context, in particular about the *informational* context, consisting of dialogue participants' beliefs, goals and other attitudes. We believe that any theoretical framework for dialogue analysis should give a central position to the information states of the participants, since this is primarily what their communicative actions address.

3 Dynamic Interpretation Theory

3.1 Dialogue acts

Dynamic Interpretation Theory has emerged from the study of spoken (human - human) information dialogues, and aims at uncovering fundamental principles in dialogue, both for the purpose of understanding natural dialogue phenomena and for designing effective, efficient and pleasant computer dialogue systems. Information dialogues, that serve the purpose of exchanging factual information, are of fundamental importance for dialogue analysis since any dialogue involves the exchange of information. Many other kinds of dialogue have additional purposes, such as improving personal relationships, or convincing somebody of a certain point of view. Other reasons for focusing on the study of information dialogues are that such dialogues can be obtained under controlled experimental conditions more readily than many other kinds of dialogue, and that information dialogues are of obvious practical interest for human-computer interaction.

An information-exchange task naturally gives rise to questions, answers, checks, confirmations, etc. In addition, natural information dialogues also contain other elements such as greetings, apologies, pause requests, attention signals, and acknowledgements. We refer to the first type of elements as *task-oriented* acts and to the latter as *dialogue control* acts. Task-oriented acts are directly motivated by the task or purpose underlying the dialogue and contribute to its accomplishment; dialogue control acts are concerned with the interaction itself, and serve to create and maintain the conditions for smooth and successful communication. Dialogue 1 shows a fragment of an information dialogue (a telephone dialogue with the information service at Schiphol, Amsterdam Airport).[1]

[1] Dutch original, translated into English.

1	I:	Schiphol information, good morning
2	C:	good morning, this is De Bruin speaking
3	C:	can you tell me which planes will leave for Frankfurt between twelve and three o'clock?
4	I:	just a moment, please
5	I:	hello?
6	C:	yes
7	I:	at twelve fifty-five the KLM will leave...
8	C:	yes
9	I:	the KL 243...
10	C:	the KL 243
11	I:	correct
12	I	and at one a.m. the Garuda leaves...
13	C:	yes
14	I:	and it will make its first intermediate stop in Frankfurt
15	C:	yes
16	I:	and... between twelve and three you said?
17	C:	yes
18	I:	yes, and there is one at thirteen thirty of Turkish Airlines.
19	C:	Turkish Airlines?
20	I:	yes
21	C:	oh
22	C:	OK, thanks very much
23	I:	you're welcome
24	C:	goodbye
25	I:	goodbye

Dialogue 1. Human-human telephone information dialogue between client (C) and information service at Schiphol, Amsterdam Airport (I).

Communicative acts in natural language take the form of utterances or parts of utterances. In DIT, utterance meanings are viewed in terms of context changes; to describe these changes, we distinguish (a) the information the speaker is introducing into the context; and (b) the way in which that information has to be inserted in the context in order to play its intended role. We call (a) the *semantic content* and (b) the *communicative function* of the utterance.[2]

Communicative functions are defined more precisely as *the ways in which dialogue participants use information to change the context.* Examples of

[2]The notions of communicative function and semantic content are, of course, closely related to the concepts of illocutionary force and propositional content in speech act theory. We prefer to use a slightly different terminology in order to avoid the suggestion that we are simply using the framework of speech act theory.

communicative functions are INFORM, CHECK, WH-QUESTION, CONFIRM, DISCONFIRM, YN-QUESTION, CORRECT, THANK, APOLOGY, INTERRUPT. The phrase ... *dialogue participants use* ... is important in the definition, in that it means that *every communicative function is required to correspond to observable features of communicative behaviour.* Mathematically, a communicative function can be construed as a function F that, applied to a semantic content p, yields a context update function $F(p)$; given a context Γ, this function computes an updated context Γ'.

The concepts of communicative function and semantic content are analytic devices, convenient in describing the intended context changes that constitute the meaning of an utterance. Similarly for the combination of a communicative function and a semantic content, which we call a *dialogue act*.[3] The status of these concepts is comparable to that of semantic representations in semantic theories like Montague Grammar. Dialogue acts are useful in the description of utterance meaning, but they do not have a fundamental theoretical status. Analogous to the formal representations in a model-theoretic semantic framework, which have their meaning defined in terms of the properties of a model, dialogue acts have their meaning defined in terms of changes in (a representation of) the context.

Using 'utterance' to designate anything contributed by a speaker during one turn in a conversation, an utterance may correspond to more than one dialogue act, and thus be multifunctional, for several reasons.[4] First, an utterance may consist of several sentences or phrases that each express a dialogue act. Moreover, utterances or utterance parts often carry more than one functional meaning, because of (1) indirectness: a question like *Do you know the arrival time?* may function indirectly as a request to tell the arrival time; (2) 'functional subsumption': a promise like *I will come tonight* is, besides a promise, also an informative statement; (3) 'functional multidimensionality': dealing with the underlying task is very often combined in one utterance with dialogue control aspects; for example, an answer to a question also offers feedback information, since it implicitly indicates that the question was understood and accepted.

We emphasized that dialogue acts are viewed in DIT as useful, rather than theoretically essential. For building computer dialogue systems, this usefulness is quite important, just as it is virtually impossible to build a language understanding system that does not construct semantic representations.

[3]The introduction of dialogue acts goes back to Bunt (1979).
[4]On multifunctionality see also Allwood, Nivre, and Ahlsén (1990; 1992).

3.2 Communicative functions

As mentioned above, we divide dialogue acts into *task-oriented* ('TO') and *dialogue control* ('DC') acts, the distinction being that acts of the first kind are directly motivated by the underlying goal or purpose of the dialogue, while those of the second kind serve to maintain the conditions for successful communication.

Dialogue control acts are subdivided into three subsystems, concerned with *feedback, interaction management,* and *social obligations management* (see fig. 1). Feedback acts further divide into those providing information about the speaker's processing of inputs, either reporting or resolving problems (negative feedback), or reporting success (positive feedback), and those providing or eliciting information about the partner's processing of a previous contribution from the speaker. Feedback acts of the first kind are called *auto-feedback* acts, those of the second *allo-feedback* acts (**?**). Interaction management acts handle various aspects of the interactive situation, such as taking turns, pausing and resuming, structuring the discourse, and monitoring attention and contact. Social obligations management acts deal with socially indicated obligations such as welcome greeting, introducing oneself, thanking, apologizing, and farewell greeting.

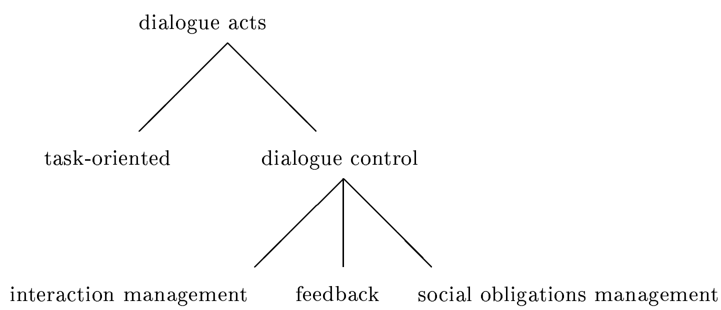


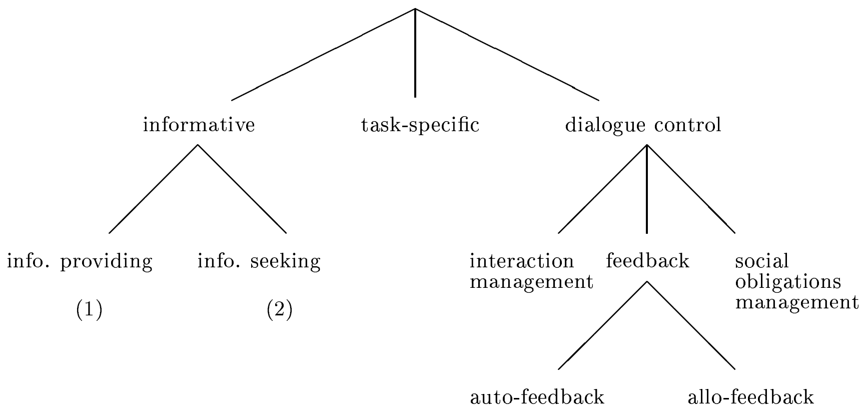
Figure 1: *Types of dialogue acts.*

An important point to note is that the TO/DC-distinction applies to dialogue acts, not to communicative functions. Whether a dialogue act is a task-oriented or a dialogue control one depends not only on its communicative function, but possibly also on its semantic content. For example, there are communicative functions specific for dialogue control purposes, corresponding to specific utterance forms, but a dialogue control act can also be formed by combining a general-purpose communicative function for information transfer, such as INFORM, with a semantic content relating to the interaction rather than to the task domain, as in *I did not hear you what you said*, or *I am very grateful to you for providing this information.* One might say that *Thanks* and *I am very grateful to you for providing this information* realize the same dialogue act in different ways, with a different distribution of its substance over communicative function and semantic content.

It may be noted that questions, informs, answers, verifications, confirmations, etc. occur not just in information dialogues, but in virtually *any* kind of dialogue. Different types of dialogue may have dialogue acts with different sets of communicative functions, relating to the kind of underlying task. For example, in a negotiation dialogue one finds proposals, rejections, and acceptations (Hulstijn 2000; Alexandersson et al. 1998; Jekat et al. 1995; Maier 1994). In such a dialogue the utterance *How about Friday the 13th?* can be analysed as having the communicative function of a proposal (due to the *how about X* form). The speaker may realise the same effect by using a nonspecific utterance form corresponding to an INFORM function: *Friday the 13th would be OK for me.*

Figure 2 provides a schematic overview of the subsystems of communicative functions we have identified, where task-specific communicative functions can be used only to build TO acts; dialogue control functions can be used only to build DC acts, and informative functions can be used to build either kind of dialogue act, depending on the semantic content. Note that the 'task-specific' functions in an information dialogue are just the information-seeking and -providing functions; therefore, information dialogues constitute the one and only kind of dialogue for which there are in fact no task-specific communicative functions. This illustrates once again that information dialogues form a basic kind of dialogue.

For identifying a communicative function we have two criteria, that follow immediately from its definition: (1) the function defines a specific way of changing the context; (2) the function corresponds to specific features of communicative behaviour. Applying these criteria to task-oriented dialogue acts in information dialogues, a hierarchical system of informative functions has been developed in Bunt (1989). The hierarchical structure reflects that some communicative functions are more specific than others; functions lower in the hierarchy inherit the goal and the enabling conditions of dominating functions. At the highest level of this hierarchy we find two subclasses of communicative functions, those concerned with *information seeking* and those with *information providing* (indicated by (1) and (2) in fig. 2).



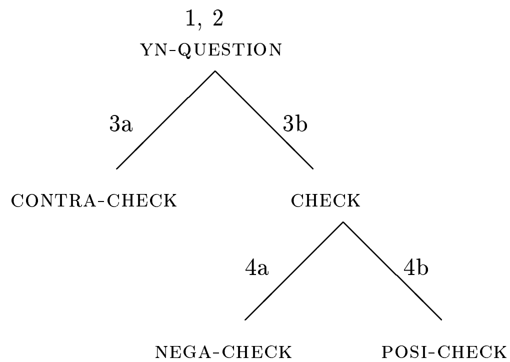
(1), (2): hierarchies of communicative functions defined in (Bunt 1989).

Figure 2: *Subsystems of communicative functions.*

The hierarchical organization of communicative functions in DIT is important both in the interpretation and generation of utterances in a dialogue system, as we will see below. A small part of the hierarchy described in Bunt (1989) is illustrated in fig. 3. The hierarchical relation between functions is defined by the inheritance of preconditions: all functions inherit conditions 1 and 2 from the YN-QUESTION; the POSI-CHECK and NEGA-CHECK inherit 3b from the CHECK.

For a discussion of the system of dialogue control acts, the reader is referred to Bunt (1994). The subsystems of feedback and social obligations management are discussed in **?**), while interaction management acts are the focus of Bunt (1996). We will review these subsystems below (section 4.2) when we analyse the relations between the various types of dialogue acts and different aspects of context. It may be noted that many of the distinctions made here correspond to distinctions proposed for dialogue annotation in the Discourse Research Initiative (Allen and Core 1997).

We have seen above that every communicative function corresponds to certain observable features of communicative behaviour. For every communicative function F_i there is a characteristic set χ_i of utterance features such that an utterance having those features will be assigned the function F_i. We write $\chi_i \rightsquigarrow F_i$ to indicate this. This does not mean that the assignment



Conditions designated by the numbers (semantic content p):
1: Speaker wants to know whether p
2: Speaker believes Hearer knows whether p
3a: Speaker weakly believes that not p
3b: Speaker weakly believes that p
4a: Speaker believes that Hearer weakly believes that p
4b: Speaker believes that Hearer weakly believes that not p

Figure 3: *Subfunctions of yes/no questions.*

of communicative functions to utterances is a straightforward matter, for the set of features ϕ_u of any given utterance in general does not coincide with the characteristic set χ_i of any communicative function. An utterance may be multifunctional, a subset of ϕ_u corresponding with one function and another subset with another one. An utterance may be *functionally ambiguous*: ϕ_u is part of the characteristic set $\chi(F)$ of more than one function F. For example, suppose we encounter an utterance u with a set of features ϕ_u, such that $\phi_u \subset \chi(F_1)$ as well as $\phi_u \subset \chi(F_2)$. The DIT algorithm for communicative function assignment first checks whether F_1 and F_2 are members of the same (sub-)hierarchy; if they are, u is assigned the *most specific function F_3 in the hierarchy which is less specific than F_1 and F_2* (the 'least upper bound' of F_1 and F_2). If F_1 and F_2 are not members of the same (sub-)hierarchy, so they have no 'least upper bound', then u is considered to be truly ambiguous. Note that the assignment of a 'least upper bound' function, if there is one, amounts to interpreting utterances functionally *as specific as unambiguously allowed by the utterance features.*

To see the consequences of this approach, we consider the assignment of a communicative function to an utterance that might be taken to have a specific function, given the discourse context. Consider the utterance: *At 10.45*, occurring in the following dialogue fragment:

A: The KLM 238 from Jakarta is still expected at 4.45ΓΓ
B: At 4.45.

The utterance by *A* can be functionally classified as a CHECK, which means essentially that the speaker has a weak belief that the contents are true, and has the goal to obtain a strong belief about this. The utterance features supporting this interpretation are the combination of the declarative sentence type with a question mark (or, in spoken form, a question-indicating intonation).[5] The utterance *At 4.45.* may be viewed as having the communicative function of a CONFIRM, since it is a response to a verification. We claim that this view is doubtful, however. First, note that the utterance features (prepositional phrase, declarative) are compatible with several interpretations, e.g. WH-ANSWER, CONFIRM, DISCONFIRM. The utterance features alone do not allow us to choose between these. Since CONFIRM and DISCONFIRM are more specific than WH-ANSWER, which is their 'least upper bound', the strategy described above says that we should interpret the utterance as a WH-ANSWER. A more specific interpretation would make no sense, as the more specific context-changing effect it would in general accomplish does not occur here: the knowledge that *A* weakly believed that the KL238 is expected at 4.45 has already been conveyed by the preceding CHECK. By the same token, an utterance that has the form of an INFORM, when used in reply to a question, is adequately interpreted as such, rather than as an ANSWER. More generally, the use of context information for recognizing the communicative functions of an utterance is not meaningful, when we regard dialogue acts as context-changing operations. (This may be different when dialogue acts have some other significance, for instance in computing what a dialogue system should do next.)

The use of contextual information for interpreting utterances *qua* function is also the basis of dealing with indirect speech acts. We will discuss this in the next section.

[5]Beun (1989) has shown that speakers in a substantial amount of cases do not use a rising intonation toward the end of the sentence to indicate a questioning rather than an informing force. They may rely on the hearer's recognition of the fact that the speaker considers the hearer as an expert w.r.t. the semantic content and therefore does not have the intention to tell the hearer something, but rather to check something.

3.3 Understanding and utterance meaning

When a receiver R understands a communicative act, performed by a sender S, the action has the effect that R forms certain beliefs about S's goals, S's information, and other aspects of S's mental state (hopes, preferences, expectations,..). About the fundamental characteristics of senders and receivers of dialogue acts we make the following idealizing assumptions.

Rationality People communicate to achieve something, which we tall the underlying 'task', and they do this in a rational fashion. They form communicative goals in accordance with underlying goals and desires, choose appropriate actions to further their communicative goals, and organize the interaction so as to optimize the conditions for successful communication.

Sociality Communication is a social activity, and is thus subject to cultural norms and conventions. An important aspect of this is **Cooperativity**, i.e. taking the partner's goals, limitations, and other aspects into account in the choice of the function and form of one's communicative actions.

With these assumptions, we can distinguish the following sources of motivation to perform a dialogue act, and relate these to the different types of dialogue act in fig. 1.

1. Communicative goals motivated by goals of the task, for the performance of which the dialogue is instrumental. The assumption of rationality predicts task goals to lead to communicative goals in a rational way. This is the drive behind the task-oriented dialogue acts for which the speaker's communicative goals derive from his own task goals.

2. Recognized partner goals. According to the cooperativity assumption, the recognition of such goals is sufficient reason for a dialogue agent to form the intention to act (this in contrast with approaches where a cooperative agent is assumed to adopt partner goals, which then give rise to own communicative goals. A dialogue agent can act cooperatively either directly on the basis of recognized communicative goals, or on the basis of inferred task goals. This gives rise to TO-acts motivated by task goals.

3. Uncertainties and actual or anticipated problems that may arise in the conditions for successful communication. By the rationality assumption, this leads an agent to perform communicative actions to

improve the conditions, in the interest of successful communication. This is the motivation behind interaction management and feedback acts.

4. Social obligations, such as being friendly, thankful and respectful, by the sociality hypothesis put a pressure on dialogue participants to greet, make and accept apologies, express gratefulness, and perform other social obligations management acts.

While understanding a speaker's motivation for acting is important, there is more to utterance understanding than just that. Compare, for example, the following utterances: (1) *Is there a later flight?*, (2) *There's also a later flight, isn't it?*, and (3) *There's no later flight?* All three utterances can be taken to express the speaker's goal to know whether the proposition p: *there is a later flight*, is true, but they differ in the speaker's assumptions. Utterance (2) may be taken to express a CHECK, which has the 'enabling condition' that the speaker weakly believes that p; utterance (3), moreover, may be taken to express the speaker's additional assumption that the partner believes that not p (a 'NEGA-CHECK', see fig. 3). In order to fully understand the speaker's communicative act, and to respond appropriately, a receiver has to recognize such additional beliefs. Perceiving and understanding an utterance thus means that at least two aspects of the context change: the understander's beliefs about the speaker's goals and beliefs, and, trivially, the discourse context, which is extended with the new utterance.

As mentioned above, in Dynamic Interpretation Theory we construe utterance meaning in a 'dynamic' way, in terms of context changes. Communicative action, as opposed to physical action, cannot change anything in the physical world, but only something in the 'mental worlds' of the communicating agents, so we should look for changes of the kind just considered. Not all the changes that the hearer's state of belief may undergo, as the result of processing an utterance, can be considered as part of the meaning of the utterance. For instance, suppose a speaker asks a question which leads the hearer to suspend the interaction and compute the requested information; the computation of the answer is an effect of the question, but not a part of its meaning. (A question can only be answered *after* it has been understood.) In DIT, we restrict the context-changing effects that form part of the meaning of an utterance to those effects that constitute the understanding of the utterance. Now some listeners may dig deeper for an understanding than others; this means that they assign more complex meanings to utterances. What we call *the* meaning of an utterance is the one that corresponds to the speaker's *intended* meaning. We thus define utterance meaning as: *intended change of context corresponding to understanding the utterance.* When agent A understands an utterance by agent B, A's context changes accordingly. Successful communication takes place, according to A and B, when they mutually believe this to be the case (cf. Bunt 1989). In other words, *communication is (mutually believed) change of context through understanding.*

The fact that a hearer may dig deeper or less deep for understanding a speaker relates to the phenomenon of indirect speech acts. We now turn to the analysis of this phenomenon in terms of the assignment of communicative functions and their context-changing meaning.

In the case of a direct speech act, we have an utterance u with utterance features ϕ_u, such that $\phi_u \rightsquigarrow F_1$. Taking the semantic content X_1 and the initial context Γ_0 into account, $F_1(X_1)$ causes a context change $\Gamma_0 \rightsquigarrow \Gamma_1$ where $\Gamma_1 = \Gamma_0 \cup \{\gamma_{11}, .., \gamma_{1m}\}$, the γ_{1i} corresponding to the goal- and enabling conditions c_{1i}.[6]

Now suppose u is intended as an (additional) indirect speech act $F_2(X_2)$ (with $F_2 \neq F_1$ or $X_2 \neq X_1$, or both), which has the characteristic context conditions $c_{21}, .., c_{2n}$ and which would thus cause the context change $\Gamma_1 \rightsquigarrow \Gamma_1 \cup \{\gamma_{21}, .., \gamma_{2n}\}$. There are two ways in which this can happen:

1. Utterances with the features ϕ_u are conventionally used to express the additional 'indirect' intentions and enabling conditions $c_{21}, .., c_{2k}$, as is the case for questions of the form *Can you X?*, or *Do you know X?*. The utterance u can then be said to additionally express any dialogue act $F_2(X_2)$ of which the characteristic conditions $c_{21}, .., c_{2n}$ consist of these $c_{21}, .., c_{2k}$ plus a set of conditions $c_{2l}, .., c_{2n}$ which are already satisfied in the context Γ_1, i.e., $\Gamma_1 \vdash \{\gamma_{2l}, .., \gamma_{2n}\}$, because any such dialogue act would, in combination with $F_1(X_1)$, create the context $\Gamma_0 \cup \{\gamma_{11}, .., \gamma_{1m}\} \cup \{\gamma_{21}, .., \gamma_{2k}\}$, which is the context conventionally created by u. Since this is a purely conventional relation between utterance features and their interpretation, there is in fact nothing 'indirect' at stake here.

2. The hearer can infer from the context Γ_1, created by $F_1(X_1)$, that the speaker's state of beliefs and intentions satisfies the conditions $\{c_{21}, .., c_{2n}\}$, which would be characteristic for the act $F_2(X_2)$, i.e. $\Gamma_0 \cup \{\gamma_{11}, .., \gamma_{1m}\} \vdash \{\gamma_{21}, .., \gamma_{2n}\}$. The utterance u could then be

[6] This formulation assumes that context changes are monotonically increasing. The argument that follows does not depend on this assumption, however.

said to express any dialogue act $F_2(X_2)$ which has the characteristic conditions $c_{21}, .., c_{2n}$, since the performance of that act would create the context that has been created. This is for instance the case when the utterance *I have to work*, in reply to the invitation: *Can I take you to the movies tonight?* is interpreted as a rejection. It is relevant for the addressee to infer that his invitation is not accepted, but this should not be regarded as part of computing the utterance meaning. Since inferences do not create new information, on the context-change view of utterance meaning it would be be incorrect to say that any additional communicative act is taking place, unless we use a notion of context that distinguishes between information that is available in explicitly stored form and information that is implicitly available, in that it can be inferred. (In section 6.1 we will consider context representations where such distinctions are in fact made.)

The interesting conclusion from this is that, according to DIT, a dialogue act can only be said to take place if the speaker uses an utterance that can be recognized on the basis of its utterance features as conventionally realizing that act. Indirect speech acts that cannot be recognized but can only be inferred formally have no place as separate communicative acts.[7]

3.4 Dimensions of context

For developing adequate models of dialogue context, we must first of all determine what information to consider as constituting dialogue context. We have initially characterized 'context' as *the totality of conditions that may influence the understanding and generation of communicative behaviour.* Now a speaker's communicative behaviour may be influenced by such factors as how well he has slept last night and whether the sun is shining; a hearer's interpretation may depend on whether he likes the speaker, on whether he is tired, on whether he is under severe time pressure, and so on. It thus seems hard to determine the boundaries of this notion of context. A consideration that can help us to eliminate many things, however, is that we are more specifically interested in aspects of context that can be changed through communication, what we called 'local' aspects. Assuming that e.g.

[7]Interpreting an utterance as an indirect speech act may even be dangerous. In the Verbmobil-1 spoken dialogue translation system, an utterance like *Der Zwölfte ist mein Mutters Geburtstag*, in response to the suggestion to have a meeting in *Geht es bei Ihnen den Zwölften?*, is interpreted as in indirect declination and 'translated' as *The twelfth is impossible for me*, which is of course not a correct translation. (Alexandersson, 1997, personal communication.)

the weather and the hearer's physical condition are not affected by dialogue, such aspects should not be considered as local context information.

In the literature the term 'context' is used in many different ways, referring for example to the preceding discourse, to the physical environment, to the domain of discourse, or to the social situation. We believe that all these notions of context should be taken into account, and have suggested in Bunt (1994) that local contextual factors can be grouped into five categories of conceptually different information: *linguistic, cognitive, physical, semantic*, and *social*. Each of these 'dimensions' has local and global aspects. The local information in these categories factors may be characterized briefly as follows.

- **Linguistic context**: Surrounding linguistic material, 'raw' as well as analysed. This is closely related to what is sometimes called 'Dialogue History' (see e.g. Bilange 1991, Prince and Pernel 1995).
- **Semantic context**: state of the underlying task; facts in the task domain.
- **Cognitive context**: participants' states of processing and models of each other's states.
- **Physical and perceptual context**: availability of communicative and perceptual channels; partners' presence and attention.
- **Social context**: communicative rights, obligations and constraints of each participant.

In DIT we are especially interested in the relations between local context and functional, 'pragmatic' aspects of utterance meaning. These relations are brought out by utterance features such as sentence type, intonation pattern, and use of paralinguistic elements (*Um, Ah, Mm*; silences). For some utterance features it is not always clear whether they should be considered as contributing to functional utterance meaning or to semantic content, or both; this is for instance the case with certain modal adverbs (like *indeed, perhaps, not*). Once a set of utterance features has been chosen to be interpreted pragmatically, we in fact have started a snowball rolling, since this calls for an articulate specification of local contexts and how communicative acts may change them. To the extent that such specifications are not available they can be developed by analysing the requirements arising from the interpretation of the utterance features to be treated. This leads to an iterative approach to context specification and dialogue analysis:

1. A set of utterance features is determined, to be interpreted pragmatically;

2. A set of communicative functions is chosen, often inspired by speech act theory;

3. A formal characterization of communicative functions in terms of context changes is developed, which often leads to reconsidering the system of communicative functions and corresponding utterance features;

4. To account for extensions and refinements in the functional treatment of utterance features, new or refined aspects of local context are introduced.

We are thus dealing with three sets of entities: (a) utterance features; (b) communicative functions; (c) local context aspects. Each of these gets revised in the light of the formal analysis of communicative functions in terms of context changes and the empirical investigation of utterance features, and in this way the analysis of dialogue phenomena and the specification of local aspects of context take place in an iterative fashion.

4 Local context information

For task-oriented dialogue acts, we construed local context models in Bunt (1989) as pairs $< K_A, K_B >$, where K_A is the information state of partner A, and K_B that of B, and where K_A and K_B consist of the domain knowledge of the respective agents, their goals, their assumed shared beliefs, and their recursive beliefs about all these elements.

This approach to local context seems basically adequate for dealing with task-oriented dialogue acts, but not for dialogue control acts, because these are concerned with different sorts of information. In Bunt (1991) we therefore proposed a richer notion of local context as pairs $< C_A, C_B >$, C_A being the local context according to A, with the five dimensions mentioned above, and C_B that according to B. It may be noted that, whatever internal structure C_A and C_B have, it is appropriate that local dialogue context consists of two components corresponding to each participant's view of the current situation. There is no room here for an 'objective' notion of context, since the participants' communicative behaviour depends solely on how they view the situation, not on what the situation 'really' is. Dialogue contexts exist only in the minds of the participants. For the same reason, we will use the terms 'knowledge' and 'belief' indiscriminately, since a dialogue agent cannot discriminate between his beliefs (that may be false) and his knowledge (true beliefs).

We will now consider how the various classes of dialogue acts relate to local context information, and identify the kinds of information that the various components of context should contain. Note that 'component' only has a heuristic significance here; we will see later that representations of local context are best structured in a way that does not correspond exactly to these five components.

4.1 Task-oriented dialogue acts

Task-oriented dialogue acts are directly motivated by a communicative goal derived from an underlying task goal. The understanding of such an act creates in the hearer the belief that the speaker's state of intention and information has the properties expressed by the semantic content and the communicative function. A TO-act thus addresses the hearer's local cognitive context, as any dialogue act does, and aims at changing the semantic context.

Speakers in information dialogues often explicitly talk about beliefs and intentions regarding certain beliefs (*Do you know ...?; I suppose you don't know whether ...?*); therefore the beliefs a hearer may build up about the beliefs and intentions of the speaker may contain several levels of nesting of belief- and intention attitudes. It has been argued by e.g. Clark and Marshall (1981) and Bunt (1989) that successful communication between two agents not only creates nested beliefs of one agent about the beliefs and intentions of the other agent, but also leads to *shared* (or *'mutual'*) beliefs, i.e. the two agents both believing (1) that some propositions hold, and (2) both believing that the other agent believes (1), but also (3) that both agents believe (2), and (4) that both agents believe (3), and so on.

For the generation of a task-oriented dialogue act, an agent's most important information consists of (Bunt 1989):

1. his beliefs about (and intentions regarding) the state of the underlying task;
2. his beliefs about (and intentions regarding) the partner's beliefs about (and intentions regarding) the state of the task;
3. his beliefs about (and intentions regarding) the mutual beliefs about the state of the task and about each other's information state. These

beliefs, as created through communication, are often *weak* and subject to acts of verification.

Since understanding a dialogue act basically means understanding why the speaker generated the dialogue act, the same information is needed on the interpretation side.

4.2 Dialogue control acts

Dialogue control acts with DC-functions typically have no or only marginal semantic content. Their meaning is concentrated in their function, so to speak. Since no articulate semantic content is involved, that would require reasoning with beliefs and intentions, we conjecture that the generation of such acts is not governed by intentions, but is rather triggered by relatively simple context conditions. By the same token, the context-changing effect of these acts does not consist in the creation or modification of complex belief structures, but rather in affecting simple, parameter-like representations.[8]

4.2.1 Feedback acts

Auto-feedback acts are triggered by difficulties that the speaker encounters in processing an incoming utterance (negative auto-feedback), or by successful completion of such processing (positive auto-feedback) (see also Allwood, Nivre, and Ahlsén 1992; Nivre 1995). Allo-feedback acts are triggered by difficulties or errors that the speaker notes in the hearer's processing (negative allo-feedback) or by noted successful completion of such processing (positive allo-feedback). We are thus lead to assume that speakers have a representation of how well their own processing goes, as well as that of the partner. This information forms part of the speaker's local cognitive context: the knowledge of the speaker's own processing we will refer to as his *own processing state*; that of the partner's processing belongs to the speaker's model of the partner.

As noted above, feedback acts may be performed by means of a communicative function specific for this purpose, or by means of an informative communicative function combined with feedback information as semantic content. In the latter case the semantic content can be articulate (*Did you say Thursday?)*. Feedback acts of the latter kind have a semantic content that comes from the speaker's local *linguistic* context.

[8] Below, in section 5.6, we will deal with dialogue control acts with articulate semantic contents.

4.2.2 Interaction management acts

Interaction management (IM) acts are concerned with monitoring various aspects of the interactive situation, and form a rather heterogeneous class of dialogue acts. We consider each of the five subclasses of IM acts distinguished in fig. 4.

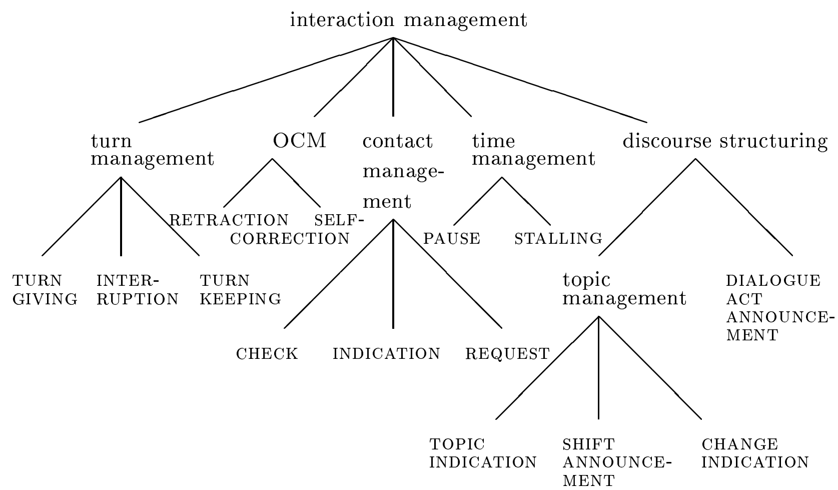


Figure 4: *Interaction management functions.*

Own communication acts: dialogue acts concerned with 'own communication management' (OCM), a term introduced by Allwood et al. (1990), signal difficulties a speaker encounters in the utterance production process. Important in spoken information dialogues are: RETRACTION acts, where the speaker retracts something he just said by mistake, and SELF-CORRECTION acts, where the speaker replaces some erroneously produced material by something else. OCM acts deal with the same kind of information as negative feedback acts, the difference being that they relate to utterance production rather than to input processing. Conceptually, this information is therefore best considered as part of the speaker's own processing state.

Time management acts: two important cases of time management acts in spoken dialogue are that of suspending the interaction (PAUSE) and that of buying time (STALLING). The first occurs when the speaker has to perform some other activity, such as finding information or dealing with some very urgent business, before continuing the dialogue, and estimates that this requires too much time for an unexplained silence. When he thinks that only little time is needed, he may instead STALL, e.g. speaking more slowly and using 'fillers', all the time maintaining contact and keeping the turn. A STALLING act often gives some indication of the progress of the activity that requires some time. A STALLING act may also serve to buy time for dealing with problems in utterance formulation (finding the right words) or in input processing problems; STALLING acts thus often go hand in hand with OCM acts or negative feedback acts.

The occurrence of time management acts indicates that a speaker has estimates of the time needed to perform or to complete his processing. Being process information, this naturally belongs to the processing state part of the local cognitive context.

Turn management acts: In a corpus of 111 naturally occurring spoken information dialogues (see Beun 1989) the most important cases of *turn management* are the following:

1. The information service encourages the client to go on (TURN-GIVING).
2. The speaker needs a little time for producing a response, but wants to keep the turn (TURN-KEEPING).
3. The speaker is interrupted because a communicative error is detected (INTERRUPTION).

Dialogue participants apparently have a view on the allocation of turns. For past turns, this information is represented in the local linguistic context, which is a record of the linguistic events and thus constitutes a 'dialogue history' (cf. Prince and Pernel 1995). For future turns, which a speaker is planning, the local context should contain the same kind of information as the dialogue history, though with less detail. We therefore assume that the linguistic context has both a history part and a future-directed part.

Contact management acts are mostly nonverbal in face-to-face communication. In telephone dialogues they often occur in an explicit form when a speaker is uncertain whether the person at the other end of the line is actually there and is paying attention, especially after a pause. *Hello?* is commonly used to check presence and attention, *Yes* to confirm it.

The occurrence of contact management acts means that speakers make assumptions about the physical and mental 'presence' of their dialogue partner. This forms part of the assumptions about the current physical and perceptual conditions of the interaction, and is thus part of the local physical/perceptual context.

Discourse structuring acts are performed by a speaker to structure the discourse, indicating e.g. that he is closing the discussion of a certain topic, wants to address a new topic, or wants to ask a question, as in *I would like to ask you something, about flights from Munich,...*

Investigations of the articulation of discourse structuring acts suggest that, as far as the planning of topics is concerned, speakers in information dialogues do not go beyond (1) deciding on a set of one or two topics to be addressed; (2) selecting a topic from this set. More generally, discourse structuring acts are based on the speaker's view of the present linguistic context and his plan for continuing the dialogue. When discussing turn management acts we noted that it seems best to consider such information as forming a 'future-directed' part of the local linguistic context.

4.2.3 Social obligations management acts

In natural communication there are certain things one is supposed to do and certain things one is not supposed to do, following general norms and conventions of social behaviour in the culture to which one belongs. For instance, when contacting someone with the purpose of engaging in a dialogue, one may be supposed to exchange greetings and to introduce oneself. We use the term 'social obligations' to describe such phenomena.

For dealing with social obligations, languages have closed classes of utterances with the property that the use of such an utterance puts a pressure on the addressee to react using a particular type of utterance from another closed class. For instance, *Thank you*, creates a pressure to reply with *You're welcome*, or one of its equivalents. In Bunt (1994) we have introduced the notion of *reactive pressures* to capture this phenomenon.[9] In the informa-

[9] We think this term is more appropriate than other terms found in the literature, such as communicative 'obligations' (Allwood 1994); 'adjacency pairs' (Schegloff and Sacks 1973), and 'preferred organization' (Levinson 1983). 'Obligation' is slightly too

tion dialogues we examined, we have found five types of situation where social obligations arise: those where greetings are in order (begin and end of the interaction); those where agents introduce themselves; those where an agent apologizes for a mistake or for the inability to supply requested information, and those where gratitude is expressed.

SOM acts come in initiative-response pairs, in the sense that every type of SOM act has an 'initiative' version where the speaker deals with a particular type of social obligation, and thereby puts pressure on the dialogue partner to also deal with that obligation and/or to play it down in his reaction.[10] In order to account for the occurrence of initiative utterances dealing with social obligations, we have introduced the concept of *interactive pressures* (Bunt 1991). An interactive pressure (IP), like a reactive pressure (RP), is a pressure on an agent to perform a certain communicative action; the difference with a *reactive* pressure is that it is created not by a particular utterance, but by properties of the local context. IPs lead to initiative dialogue acts for social obligation management, if we assume that 'social' communicative agents act to resolve such pressures, thereby evoking the corresponding reactive acts through their RPs (assuming, again, that agents strive to resolve such pressures). IPs and RPs make certain dialogue acts, that an agent is pressured to perform, 'active' in the local social context of that agent.

4.3 Summing up: what's in local context?

By investigating the kinds of information the various types of dialogue acts introduce or modify in the local context, we have identified the following aspects of the conceptual content of local context.

Local cognitive context:

- The agent's processing state, including, for the major aspects of input processing, output generation, and related cognitive and task-specific processing needed for participating in the dialogue:

---

strong, as the 'obligating' utterance does not really *oblige* the addressee to respond in the 'obligated' way. 'Adjacency pair' is also too strong, since the two elements of the pair do not really have to be adjacent, and in fact the second element does not necessarily have to appear at all. 'Preference organization' would seem to have the right kind of strength, but this term belongs to a structural framework of dialogue analysis, where the term 'preference' is not meant to have a cognitive interpretation (Levinson 1983, p. 332-333). Our approach, by contrast, does have a strong cognitive orientation and considers reactive pressures to be an aspect of the local cognitive context.

[10] Some approaches to dialogue give central importance to initiative-response pairs for all kinds of communicative acts; see e.g. Moeschler (1989); Roulet (1985), Bilange (1991).

  - progress, in particular whether the process is ready;
  - any difficulties encountered;
  - results obtained;
  - estimated time needed for completion.
- The agent's beliefs (weak and strong) about the dialogue partner, in particular:
  - about the partner's processing state;
  - about the partner's current information and mutual beliefs relating to the underlying task and to each other's information state.

Local linguistic context: The dialogue so far, including the agent's interpretation of the communicative events. Also the agent's discourse plan, if any ('future linguistic context').

Local semantic context: The agent's current beliefs concerning the underlying task and his current task goals.

Local physical/perceptual context: The agent's assumptions regarding the dialogue partner's physical, perceptual and mental 'presence'.

Local social context: The interactive and reactive pressures on the agent, corresponding to social obligations, feedback, and turn management.

A model of the other agent's model of the local context, within an agent's local cognitive context, is needed since any time a dialogue act is performed concerning some aspect of local context (as in *Please repeat*, concerned with the local linguistic context), an assumption is made about the corresponding component of the other agent's context model (in this example, the assumption that the previous contribution is available in the partner's linguistic context). This causes recursion in the model structure. Using C_{AB} for the local context as A believes B views it, we thus have the conceptual structure for A's view of the local context shown in fig. 5.

5 Representation of local context

Having identified the most important types of local contextual information as required by the various types of dialogue acts, we now turn to an analysis of the logical properties of the various information types as a next step

$$
\begin{array}{llll}
C_A & = & < & A\text{'s local semantic context,} \\
 & & & A\text{'s local cognitive context:} \quad < \text{Own Processing State} \\
 & & & \qquad\qquad\qquad\qquad\qquad\quad \text{Partner Model: } C_{AB} > \\
 & & & A\text{'s local physical and perceptual context,} \\
 & & & A\text{'s local social context,} \\
 & & & A\text{'s local linguistic context} \quad >
\end{array}
$$

Figure 5: *Conceptual structure of an agent's local context.*

toward their formal and computational modelling. Three relevant issues to consider in this connection are the following:

1. the degree of articulation of the information: some information types are more articulate than others;

2. whether communicating agents use the information in full-blown inferential processing or in simpler, special-purpose processing such as parameter value testing;

3. the existence of dependencies between different information types.

Full-blown inferential processing naturally goes hand in hand with articulate expression of information, while simpler processing is appropriate for information with little internal structure. The articulation of information types and the complexity of the associated processing can be investigated empirically by examining the semantic contents of the dialogue acts addressing the various information types, and by investigating the complexity of subdialogues where the various kinds of context information are the topic of conversation. The existence of dependencies between different information types follows from the inherent connections between perception, input analysis, the maintenance of a model of the discourse situation, and the generation of utterances.

We now consider in turn the five conceptual parts of local context (see fig. 5).

5.1 Local social context

We noted above that social obligations management acts (SOM acts) come in two varieties, initiative and reactive ones. Those of the reactive kind are triggered by reactive pressures (RPs) of initiative SOM acts, those of the initiative kind by interactive pressures (IPs) which arise when the local context satisfies certain conditions.

The rules describing how local context properties give rise to interactive pressures are pairs, consisting of (1) a set of local context conditions; (2) a partial specification of a dialogue act, most typically consisting of the specification of a communicative function and a set of constraints on semantic content and utterance form. The conditional part (CP) describes the properties of the local context that give rise to the pressure; the action part specifies the kind of dialogue act that the agent, to whom the IP rule applies, is pressured to perform. The action part of an IP rule typically does not specify a dialogue act completely, since IP rules leave some freedom for the way to act.

Based on our exploration of corpora of Dutch telephone information dialogues with an information service, the IP rules for opening greetings, self-introductions, and apologies are as follows (slightly simplified). In all instances, the action part indicates an action to be performed by the agent who has the turn, unless otherwise specified.

Opening Greeting

$<$ CP: no dialogue acts have yet been performed
other than SELF-INTRODUCTIONs
IP: OPENING-GREETING-INIT $>$

Self-Introduction

$<$ CP: no dialogue act has yet been performed
IP: SELF-INTRODUCTION-INIT $>$

Apology

$<$ CP: agent X knows that he has made a mistake (e.g., has, misunderstood Y), or is unable to act in a way that furthers any of the known goals of partner Y, or is unable to process a contribution from Y; X has not yet apologized for his inability or error
IP: APOLOGY-INIT by agent X with the error or inability as semantic content $>$

Note that, in the dialogue situation considered here, the IP rule for introducing oneself applies only to the information service, since this partner always opens the dialogue (see Dialogue 1 above). The service is thus always pressured to start by introducing itself. These IP rules are quite specific, and apply only to a particular kind of interactive setting. Articulate IP rules can only be given for specific settings, since social obligations and the way they are handled depend in their details on the global social, physical and perceptual context.

When the conditions in an IP rule are satisfied, this leads to a (partly specified) dialogue act to become 'active' in the local social context. Several IP rules may have their conditions satisfied simultaneously, for instance due to the fact that an agent under pressure does not have the turn; the local social context may therefore contain several 'active' dialogue acts. This may result in a multifunctional utterance containing several SOM acts, such as *Airport Information Service, good morning.* Since each of these acts carries a reactive pressure, the respective RPs accumulate as well. When more than one SOM act is active in the local social context, the most recent one often has priority to be performed first, or only, as in: *Airport Information Service, good morning./Good morning, this is Jansen* and *Thanks a lot, goodbye./Goodbye.* This suggests that a *stack* may be an appropriate organization of the local social context, with decreasing strength of the interactive/reactive pressure for elements lower on the stack. For the representation of the stack elements, dialogue acts, see below (section 5.5, linguistic context).

5.2 Local physical/perceptual context

Of the physical and perceptual context ('P/P context') , which characterizes the ways the dialogue participants can interact with their environment, including each other, the only aspects that can be changed by the dialogue are the effective availability of communication channels and whether the participants are paying attention to each other.

In the case of a telephone dialogue we thus need to represent in the speaker's P/P context his assumptions about the current availability of the telephone line. Whether the participant who is not speaking is paying attention, is a different matter but is for the speaker in the telephone situation indistinguishable form the availability of the communication channel. The contact management acts we have found in telephone information dialogues confirm this; physical and cognitive 'availability' are not addressed separately. A single, binary-valued feature is thus sufficient to represent the partner's assumed physical and mental 'presence'. When the participants can see each other, more features are needed, or a feature with more complex values.

5.3 Local semantic context

The semantic content of task-oriented dialogue acts can be quite articulate, reflecting the complexity of the task domain, and is the subject of the most elaborate discussions in a dialogue; this is also what the agents reason about and use extensively to guide their communicative activity in a rational way.

Semantic context information, moreover, is often embedded within recursive belief attitudes as part of an agent's information about his dialogue partner. See further below, concerning the 'partner model' information in an agent's local cognitive context.

5.4 Local cognitive context

Processing State
We have seen in the previous section that an agent's processing state contains the following elements per process:

1. state of progress;
2. any difficulties encountered;
3. results obtained;
4. estimated time needed for completion.

We have also seen that in natural information dialogues the estimated time needed to complete a process is never considered in precise terms. Although a computer dialogue system could conceivably calculate precise estimates of the time needed for certain processes, it is doubtful that very detailed messages would be useful for the user; we will therefore consider only the representation of the estimated time needed to complete a process in the crude way people naturally do this. This can be achieved by means of a simple attribute-value pair.

Something similar can be said about reporting the state of progress of a certain process. Again, this information can be represented by means of a simple attribute-value pair.

In negative feedback acts, when processing difficulties are reported, the speaker most of the time signals the failure of a process, as in *What did you say?*, or asks for clarification of a particular item, as in *Do you mean this Tuesday?*. This may be represented with two attributes: one representing success or failure, and one that may have a problematic input item as its value. In the case of negative feedback about evaluation, an agent reports conflicts between new information and previously available information. To detect such conflicts clearly may require full-blown inferential processing.

It is not surprising, therefore, that such feedback acts are often not expressed with dedicated feedback functions, but with a general informative communicative function, using a full-blown sentence with articulate semantic content. In information dialogues, such conflicts arise only in connection with *task-domain information.* This means that the representation of difficulties encountered by a process may require links to the local semantic context and to items of semantic analysis in the local linguistic context.

Similarly for positive feedback concerning input interpretation, which typically takes the form of a repetition of part of the previous utterance. In such acts, the agent shows the result of his interpretation process, which should also be linked to the local semantic and linguistic context.

Altogether, we are thus lead to a representation of an agent's processing state information like the attribute-value matrix form shown in fig. 6, where $\boxed{1}$ and $\boxed{2}$ indicate links to the local linguistic or semantic context.

Process P			
PROGRESS	ongoing/almost-ready/ready		
TIME-NEEDED	negligible/small/substantial		
DIFFICULTY		PROC-SUCCESS	success/fail/difficulty/..
		PROBLEM-ITEM	$\boxed{1}$ input-item$_j$/...
RESULT	$\boxed{2}$		

Figure 6: *Processing state information represented in the local cognitive context.*

Partner Model

Agents involved in a dialogue build up all kinds of information about each other. Assuming that all agents operate on the basis of the same types of information, according to the same basic principles of rationality and sociality, and with the same cognitive architecture, an agent A must assume that his dialogue partner B entertains beliefs about the same kinds of things as A himself. Therefore, if we assume that the local context of a dialogue agent A contains, conceptually, the five information types distinguished above, it follows that $A's$ beliefs about $B's$ beliefs follow, conceptually, the same fivedimensional structure, and this recursively, as indicated in figure 5 above.

The most complex kind of local cognitive context information (and the most complex kind of all local context information, in fact) is formed by the participants' recursive beliefs and intentions concerning to the underlying task, since this information combines the inherent complexity of the semantic context information with that of nested propositional attitudes. An adequate representation of this information therefore calls for a logically sophisticated formalism with associated inference machinery.

Candidates for such a formalism are the logics that have been developed in Artificial Intelligence for reasoning about belief and action (see e.g. Halpern 1986, Halpern and Moses 1990, Moore 1985, Levesque 1984, Cohen and Levesque 1990). These logics all have rather limited applicability, however, in not treating quite the epistemic and intentional attitudes that we need for dialogue contexts, or not dealing with interactive agents with such attitudes. The development of adequate knowledge representation formalisms and inference machines is obviously beyond the scope of a dialogue theory. In section 6, we will outline two nonstandard approaches to the modelling of belief contexts that we consider particularly promising.

Later in this section we will consider the recursive beliefs of dialogue agents relating to other than task-related information (see section 5.7).

5.5 Linguistic context and dialogue memory

From the occurrence of feedback acts and OCM acts we can infer that an agent's local linguistic context should contain representations of input utterances as well as of the agent's own contributions to the dialogue, and not just of the 'raw' input- and output signals, but also of aspects of their analysis and evaluation. Linguistic feedback refers to a preceding utterance, requiring a representation of the events earlier in the dialogue; being such a representation, the local linguistic context is a kind of *memory* of what has happened in the dialogue.

For modelling human memory of dialogue the information in linguistic context should be selective: utterances several turns back in a dialogue are not remembered verbatim, but only some words and phrases and the utterance meaning are remembered (where 'semantic memory' is not perfect either). The selectiveness of human memory is often considered to be a design feature, supporting economic use of resources. Computer memory being available in virtually unlimited supply, we may represent the linguistic context available to a computer dialogue partner exhaustively, rather than selectively.

Treating linguistic context as a memory of what has happened in the dialogue has the advantage of providing an elegant way to *avoid all other components of local context to have a memory.* Consider, for example, the modelling of a participant's beliefs and intentions relating to the underlying task, as contained in the local semantic context. It would not be sufficient to only model the agent's *current* beliefs and intentions, for an agent may sometimes discover that something went wrong, and a subgoal that was believed to be achieved and discarded, turns out not to be achieved after all. An agent should then be able to return to a previous state of beliefs and intentions. By systematically associating with each utterance in the linguistic context the changes that the utterance has brought about in the semantic context, we obviate the need to 'remember' these changes in the semantic context; any previous state of the semantic context can be reconstructed from its current state plus the changes represented in the linguistic context. In general, this approach allows all local context components except the linguistic one to represent just the *current* state. Linguistic context, by contrast, by its very definition contains information relating to the previous discourse.[11]

This view on the linguistic context has been implemented in the linguistic context model of the PLUS system, where it was termed the 'Discourse Model' (Bunt and Allwood 1993). The PLUS Discourse Model was defined as a data structure in an object-oriented programming language (see Meyer's chapter in this volume). Figure 7 shows a part of the PLUS Discourse Model, with some simplifications, and at some points using a more intuitive terminology. Formally, this structure can be viewed as a recursive type-theoretical construct.

Note that in this model an utterance is split up into parts, called 'grammatical units', that express one or more dialogue acts. The syntactic-semantic analysis of a grammatical unit is represented as a 'sign' in the sense of HPSG, a recursive attribute-value matrix. The semantically relevant elements are extracted from a sign and represented separately as 'quasi-logical form' (in the language 'ULF'; Geurts and Rentier 1993; similarly, the information that may be relevant for the assignment of communicative functions is collected in a separate representation ('pragmatic features'). A dialogue act in this representation has a communicative function name assigned to it, which is strictly speaking redundant, since the significance of a communicative function is the attitudinal information it conveys, which is represented in the 'goal-attitude' and the 'enabling attitudes'. For example, a CHECK with semantic content p has the goal attitude that S (the speaker) wants to know whether p, and the enabling attitudes that S weakly believes that p,

[11] We have seen that, besides this backward-looking aspect, local linguistic context also has a forward-looking aspect containing a speaker's discourse plans, which is the basis for discourse-structuring acts. It may also be practical to use a *context buffer* for storing the most recent dialogue history, especially in view of the fact that complete, definitive processing of inputs often does not occur immediately. Such a buffer has for instance been implemented in the TENDUM dialogue system (Bunt et al. 1984) and more recently in the DenK system, where it is called the 'pending context' (Piwek 1995).

```
Discourse Model: <list of Utterance >
   Utterance:
   verbatim_form:    <string >
   speaker:          <agent >
   gramm_units:      <list of Grammatical_unit >
      Grammatical_unit:
      synsem_structure:      <sign >
      quasi_logical_form:    <ULF >
      pragmatic_features:    <list of AV pairs >
      dialogue_acts:         <set of Dialogue_act >
         Dialogue_act:
         commun_function:       <CF name >
         semantic_content:      <semantic representation >
         discourse_referents:   <list of discourse referents >
         topic_info:            <topic element >
         goal_attitude:         <goal representation >
         enabling_attitudes:    <list of belief representations >
         reactive_pressure:     <Utterance >
         resolved_pressure:     <Utterance >
         cancel-attitudes:      <list of goal/belief represent's >
         indirect_intentions:   <set of Attitude_Set >
            Attitude_Set:
            ind_goal_attitude:          <goal representation >
            ind_enabling_attitudes:     <list of belief represent's >
            ind_reactive_pressure:      <Utterance >
            ind_resolved_pressure:      <Utterance >
```

Figure 7: *Part of PLUS Discourse model.*

and that S (at least) weakly believes that H (the hearer) knows whether p.

5.6 Articulate dialogue control information

The discussion of local social context, perceptual and physical context, and processing state, has been based on the analysis of dialogue control acts with marginal semantic content and dedicated DC function – which is their most common form. As we have seen, however, a dialogue control act may also be performed by means of a general-purpose informative function and an articulate semantic content. For instance, a STALLING act may be performed by *Um,.., um,..*, but also by saying *Let's see, I'm not sure how to say this in English.* We suggest to analyse such a situation as a temporary shift from the domain of the underlying task to the domain of how to say something in

English, i.e., as a temporary shift of the domain of discourse. The dialogue participants thus use general-purpose informative functions and articulate contents in order to convey beliefs and intentions concerning this temporary domain of discourse. This in contrast with the use of special-purpose DC functions and marginal content, where the domain of discourse remains that of the underlying task.[12] When a dialogue shifts to communication-related information, the state of the task-related information is frozen until the dialogue returns there. The consequences of this approach for context representation and for the design of a dialogue system are discussed in (Bunt 1999).

5.7 Overall organization of context representations

From the above analysis of properties of the information in local dialogue context we can draw conclusions regarding the kind of representational formalisms needed for the various information types and regarding the overall organization of context models.

From a representational point of view, the information types we have seen fall into three categories:

1. The beliefs and intentions about the underlying task that constitute an agent's local semantic context; the beliefs about the partner's beliefs and intentions about the task in the partner model component of the agent's cognitive context, and this recursively. Representation of this information calls for a powerful logical formalism. Similarly for the dialogue control information exchanged by DC acts with a general-purpose informative function and articulate semantic content.

2. The past and planned communicative events in the dialogue, in terms of dialogue acts and their analyses, making up the linguistic context. The linguistic analysis of utterances can be represented in recursive attribute-value matrices (AVMs); for information about dialogue acts we have seen the object-oriented representation used in the PLUS system, which can also be recast in AVM form. The local social context, created by interactive and reactive pressures, also consists of (aspects of) dialogue acts.

3. Process information, in the processing state part of the cognitive context, as well as contact information in the P/P context, is structurally simple and can be represented in simple AVMs.

[12] This view is corroborated by the study of topic shifts in spoken information dialogues reported by Rats (1996).

We consider three aspects of these information types that are relevant for the design of local context representations: *dependencies* between information types; their *temporal properties*, and their *depth of recursion*.

Dependencies
All information in the local context is, by definition, the result of the events in the dialogue; indeed the local context at a certain point in a dialogue can for the most part be reconstructed from the local linguistic context, by following the dialogue history from the beginning to that point. In one respect changes in local context do not relate directly to the dialogue history: an agent's processing state, especially for processing that does not relate directly to inputs (such as task-specific processing, and utterance production), cannot feasibly be computed given only the linguistic context. For the rest, it would seem that the linguistic context is all we need. In practice, this is obviously not the case, since it would mean that to generate a dialogue act an agent would every time have to run through the entire dialogue history. Instead, what the agent needs is a representation of *current* goals and beliefs, turn allocation, interactive and reactive pressures. These all follow through successive updates from the processing of the previous dialogue utterances. Once these context elements have been computed, they can be used independent from how they were created. For instance, the representation of the current beliefs of a dialogue participant about his partner does not have to include the history of how that belief has come about. This has the advantage of simplifying the belief representation system. If necessary, the history of the belief can always be reconstructed from the linguistic context.

The information in an agent's processing state is closely related to the linguistic context, since the processes whose status is represented produce the results represented in the linguistic context, or they take linguistic context information as input and compute information in other context components. The representation of processing state therefore contains links to the other context components, in particular to the linguistic context (see further Bunt 1996).

Temporal properties
We have seen that the linguistic context acts partly as a dialogue memory and allows all other context components to only represent the current state. The linguistic context has a 'temporal size' of as many turns as there are in the dialogue history, plus possibly a few planned future turns.

Depth of recursion
We have seen that beliefs and intentions regarding the underlying task as well as those relating to dialogue control information can be recursive with arbitrary depth of recursion, possibly even infinite (in the case of mutual beliefs).

The linguistic context as modelled in the PLUS Discourse Model does not address to what extent a dialogue system should represent not only its own analysis of the events in the dialogue, but also the assumed user's analysis. A representation of assumptions about the user's analysis would seem useful only if the system has reason to assume differences between it's own analysis and that of the user. More precisely, it would seem sensible to make the following assumptions:

- Unless negative feedback suggests otherwise, the user processes the system's utterances correctly;
- Unless negative (allo-)feedback from the user suggests otherwise, the system believes it analyses the user's utterances correctly.

On these assumptions, which in fact say that, unless there is evidence to the contrary, both participants understand each other correctly, there is no need to maintain separate representations of the user's analysis of the dialogue utterances, except when a communication problem is detected. This can be made operational by adding in a representation like the PLUS Discourse Model an additional attribute user's analysis for each Utterance, the value of which has a verbatim_form and an analysis in the form of a list-valued feature gramm_units: < list of Grammatical_unit >, just like the system's analysis; by default then, these attributes would share their values with the corresponding attributes in the system's analysis of the utterance.

Note that, on this approach, the system's local linguistic context is *not recursive*: it does not contain a representation of how the system assumes the user assumes the system has processed an utterance, and so on.

We have suggested that the information exchanged by means of dialogue control acts with a special-purpose DC function and a marginal semantic content is structurally simple and can be represented in simple attribute-value structures. On the other hand, the 5-component conceptual context model of fig. 5 has embedded in the cognitive context of one agent a context model ascribed to the other agent. Since the embedded context model has the same structure, this implies that all the information types in an agent's context model are infinitely recursive. It thus seems now that this fully recursive structure is inappropriate for some information types. Consider the case of physical and perceptual context information, conveyed by contact management acts.

Contact management acts tend to have negligible semantic content and clearly seem not to be based on communicative planning or full-blown reasoning. The representation of the information they convey therefore does not need to involve mutual belief and intention attitudes. On the other hand, it would not be sufficient if a dialogue system would represent only whether the user is 'present', without any assumption about the user's view on the contact situation. There are at least two reasons why more is needed:

1. To interpret a contact management act by A, agent B must be able to register a contact problem for A.
2. To perform a contact management act, agent A should assume the partner B to be unaware of the problem that A sees, for else it would be unnecessary for A to draw attention to it.

The physical and perceptual context information of an agent A should therefore contain at least the following elements:

1. the assumed status of B's physical and perceptual presence;
2. B's assumed view of the status of A's physical and perceptual presence;
3. the status of B's physical and perceptual presence, as A assumes B assumes A views it.

Each of these elements can be represented with a 3-valued parameter 'PRESENCE', with values 'positive', 'negative', and 'doubtful'. Taking the maximum depth of assumptions, beliefs, and 'views' of agents about each other as 'depth of recursion', this means that the information in local P/P context has recursion depth 3.

The same can be said about the processing state information relating to feedback and own communication management acts with DC functions, where the depth of recursion can also be set at 3.

SOM acts arise through interactive and reactive pressures, not through communicative planning and reasoning with nested beliefs and intentions. IP rules may contain assumptions of one agent about the other, however, (see the example of the Apology rule given above) with a complexity that again corresponds to a recursion depth of 3.

Figure 8 summarizes the temporal and recursion-depth characteristics of the various information types.

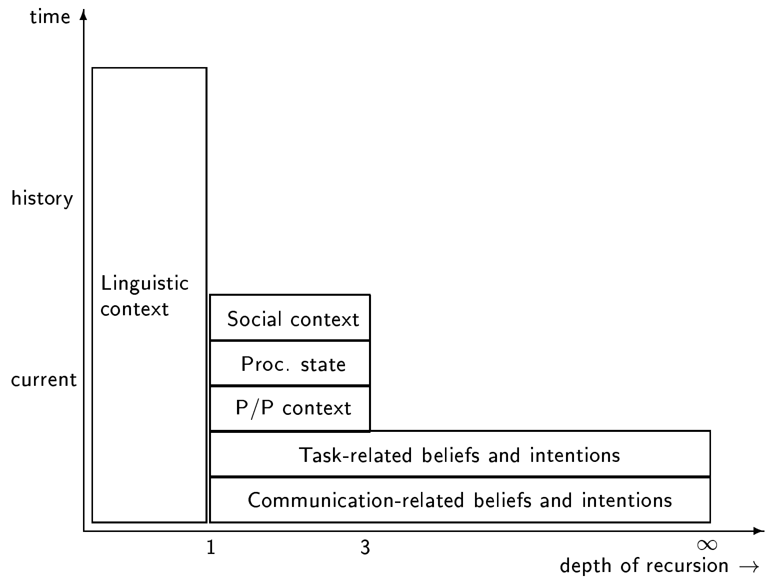


Figure 8: *Temporal extension and recursion depth of local context components.*

Overall organization
Not being fully recursive, the information in physical and perceptual context, processing state, and social context on this approach no longer forms part of the recursively embedded partner model in the local cognitive context (cf. fig. 5). In fact, this puts the entire notion of recursive context embedding into question: the only fully recursive context elements are the beliefs and intentions conveyed by task-oriented dialogue acts and by dialogue control acts with informative communicative functions.

This analysis thus leads us to conclude that the 5-component conceptual model of local context that we started out with, does not correspond to the most sensible design of a formal context representation, but that instead the organization suggests itself which is represented schematically in fig. 9 (see also Bunt 1997). The four components in this organization correspond to the three-way distinction of information types made above (beginning of section 5.7), the social and linguistic components containing information of the same kind but with a different organization: the local social context is a stack of highly underspecified description of communicative events (objects of type Utterance in the PLUS Discourse Model), while the local linguistic context is a list of fully specified objects of the same type in chronological order.

C_A = < A's local epistemic & intentional context:
    task-related information and
    communication-related information,
A's processing state and
    physical and perceptual context,
A's local social context,
A's local linguistic context >

Figure 9: *Organization of an agent's local context representation.*

6 Formalisms for context modelling

In this section we will outline two formalisms that we consider to be particularly promising for computational modelling of local contexts, *Constructive Type Theory* and *Modular Partial Models*. The first is a proof-theoretical formalism that has been implemented in the DenK multimodal dialogue system (Bunt et al. 1998), the second is a model-theoretic formalism that has been developed in particular for the representation of nested beliefs and intentions occurring in a dialogue participant's cognitive and semantic contexts.

6.1 Constructive Type Theory

Constructive type theory (CTT) is a member of the family of powerful and versatile logical formalisms for knowledge representation and reasoning, known as *type theories* or *pure type systems*. The development of these systems was originally motivated by research into the formal properties of logical connectives in intuitionistic logic. De Bruijn (1980) developed a variant of pure type systems called *Automath*, which he used to represent an entire mathematics textbook and to automatically verify all the proofs. Closely related variants are the *calculus of constructions* (Coquand 1985) and Martin-Löfs *intuitionistic type theory* (Martin-Löf 1984).

Sundholm (1986) was the first to apply type theory to natural language semantics, in particular to the analysis of donkey sentences. Ranta (1991) used type theory as the semantic input of a generation algorithm for English sentences; Sundholm (1989) reconstructed Barwise and Cooper's generalized quantifiers in type theory. More recently, type theory has also been applied succesfully to presuppositions (Krahmer and Piwek 1999), to anaphora (Beun and Kievit 1996), and to question-answer relations in dialogue (Piwek 1998). Krause (1995) has shown that type theory provides a good basis for *abductive reasoning*, which has been argued to be especially important in context-based natural language processing (see e.g. Hobbs et al. 1993, and the chapters by Bunt and Black and by Oberlander and Lascarides in this volume).

Type-theoretic semantic representations are *constructive* in nature in that they only allow the use of terms that are well-formed according to a specific background of term introductions, a so-called *context*. In type-theoretic semantics such a background set is interpreted as the knowledge or beliefs of an agent. Semantic interpretation is thus relative to an agent and therefore inherently intensional; it assigns meanings in terms of the knowledge of an agent. Whether or not such interpretations correspond to anything in an external reality is another matter.

An unusual feature of type theory is that it views proofs as abstract objects, on a par with individual concepts; proofs have representations in the language, and are fully integrated within the formalism. (Moore about this below.) This means that in type theory we can represent not only *what* an agent believes, but also *how* he comes to believe it, by having explicit representations of the proofs that justify his beliefs. The constructivist character of the framework shows again here, in that proofs may be constructed only from steps that can be found in a given context.

Type theory is based on typed lambda calculus. The language of explicitly typed lambda calculus consists of expressions of the general form $A : B$, expressing that an object A has type B, which is also glossed as 'A is an inhabitant of type B'. To show that $A : B$ holds in a given context Γ, one has to show that either Γ contains that expression, or that it can be obtained from the expressions in Γ by means of the type deduction rules. These rules spell out how complex terms may be constructed as inhabitants of complex types, given (in the context under consideration) the component terms as inhabitants of other types.

Barendregt (1991) noticed that many existing systems of typed lambda calculus can be uniformly represented using a single format, parametrized for the elementary types or *sorts* that it uses, the 'pure type systems' format (see also Barendregt 1992).

Formally, the expressions of CTT can be defined as follows.

Definition 1 (sorts). The set of *sorts* is $\mathcal{S} = \{\Box, \mathit{type}, t\}$.

The sort $\Box$ is a 'supertype' at the top of the hierarchy of types; the 'mother of all types'. Immediately below the top one may find different high-level types; different choices here lead to different members of the family of pure type systems. In CTT we have chosen the sort *type* as the supertype of all non-mathematical entities, and t as the supertype of all propositional types. We will moreover assume a high-level type e (for 'entity') as a supertype of all types of individuals that we may find in a certain domain of discourse.

Definition 2 (variables). Variables are the elements of a set $\mathcal{V}$ which is disjoint with $\mathcal{S}$.

Definition 3 (types). Types are atomic or complex. The set $\mathcal{T}$ of types is defined as the smallest set such that:

1. sorts and variables are atomic elements of $\mathcal{T}$ (but see below for restrictions on the use of variables as types).
2. if t_i and t_j are types and x is a variable, then the following expressions are complex elements of $\mathcal{T}$:
   (a) $t_i \cdot t_j$, where the dot signifies function application
   (b) $\lambda x : t_i.t_j$ (λ-abstraction)
   (c) $\Pi x : t_i.t_j$ (Π-abstraction).

The Π-types introduced here, so-called *'dependent function types'*, are the types of polymorphic functions where the type of the range t_j may depend on the type of the argument, i.e. t_j may be a complex type expression in which x occurs. For functions of a Π-type where the range does not depends on its arguments we will use the familiar notation '$t_i \rightarrow t_j$'.

Definition 4 (introductions). Introductions are expressions of the form $x : T$ where x is a variable and T is a type.

Note that, by definition 3, variables are types and may thus occur in the right-hand side of an introduction. The legal use of variables is context-dependent, however; every variable must first be introduced in the context under consideration (i.e., must occur as the left-hand side of an introduction). This restriction is captured in the definition of contexts.

Definition 5 (contexts). A context is a sequence of introductions that is well-formed in the sense that every type occurring in the right-hand side of an introduction is either a sort or is the left-hand side of an introduction earlier in the sequence.

Type-theoretical contexts bear some resemblance to the discourse representation structures of DRT (Kamp and Reyle 1993). Roughly, the variable x in an introduction $x : T$ can be seen as corresponding to a discourse referent in DRT, and the type as corresponding to a predicate in a DRT condition. In fact, Ahn and Kolb (1990) have formally shown that CTT-contexts can be regarded as generalizations of DRSs. The following example illustrates the possible use of CTT for semantic representation.

Consider the sentence *A farmer laughs.* The NP gives rise to an introduction of the form $x : farmer$, but $farmer$ has to be introduced first to obtain a well-formed context. So we get, initially:[13]

$[farmer : e,\, x : farmer]$

The VP corresponds to a predicate, applicable to farmers, and must be introduced as such, i.e. as a function from farmers to propositions:[14]

$laugh : farmer \to t$

We can now complete the CTT representation of the sentence by adding the statement that corresponds to the condition $laugh(x)$ in DRT, to obtain the context:

$[farmer : e,\, x : farmer,\, laugh : farmer \to t,\, p : laugh \cdot x]$

This can be read as: farmers are individual objects; x is a farmer; laughing is a property that farmers may have; there is evidence that x laughs. (See below for further explanation of this last bit.) A CTT analysis of a sentence or a discourse thus introduces objects of various kinds and adds them to the context that grows incrementally as the discourse proceeds, similar to

[13] Remember that we assumed, for improved readability, that the type e ('entity') has already been introduced (like a sort, effectively).

[14] Of course, it is advisable to allow the *laugh* predicate to be applicable to a larger domain than just farmers; this can be achieved by allowing polymorphic dependent types or by introducing subtyping (Ahn 1995).

what happens in a DRT treatment.

We have seen that type theory takes an 'object-oriented' view on proofs, as it were, considering proofs as structured objects with properties not unlike individual concepts. This view is based on a fundamental insight, known as the *Curry-Howard-De Bruijn correspondence,* according to which propositions can be interpreted as types in a typed λ-calculus. Under this interpretation, the inhabitants of such a proposition type stand for proofs of the proposition. It turns out that there is a correspondence between provability in standard logic and the existence of inhabitants in typed λ-calculus.

The following example illustrates this approach. A pure type system is a formal system, consisting of a formal language which has contexts as its expressions, plus a set of deduction rules (see Barendregt 1992). The deduction rules make use of contexts, of the introduction statements that we have seen, of the form $x : T$, and also of more general 'statements', i.e. expressions of the form $E : T$ where E may be a complex term, constructed out of several variables by means of the possibilities described in definition 3. For instance, the following rule can be viewed as merging Modus Ponens with function application:

$$\frac{\Gamma \vdash F\colon P \rightarrow Q \quad \Gamma \vdash p\colon P}{\Gamma \vdash (F \cdot p)\colon Q}$$

According to this rule, when we have a context Γ in which F is known to be an inhabitant of type $P \rightarrow Q$ with propositional types P and Q, i.e. F is an already available proof of a proposition of the form $P \rightarrow Q$, and p is a proof of the proposition P, then the term $F \cdot p$ constitutes a proof of the proposition Q.

The occurrence of proofs within the system does not mean that *truth* is a particularly central notion of the framework; on the contrary. Central to type theory is the recording (in type-theoretical contexts) of accumulating information and of what follows from this information. An agent, whose beliefs are represented type-theoretically, is regarded as *explicitly* believing those propositions that are present in the type-theoretical context, and *implicitly* believing those propositions that are not explicitly represented, but for which he can construct a proof. Viewing a type-theoretical context as a representation of the beliefs of an agent, the total set of his beliefs is thus determined by what the agent is able to deduce from the context. For some propositions P, the agent will not be able to construct a proof, nor will he be able to construct a proof of *not* P, hence this framework constitutes a *partial* approach to belief modelling.

A type-theoretical context may be viewed either as an agent's *knowledge* or as his *beliefs*. In epistemic logic the distinction between the two is often expressed by: "knowledge is justified true belief". An agent obviously takes his beliefs to be true, so the only distinction he can possibly make is between justified and unjustified beliefs; in a type-theoretical approach, however, an agent can only have beliefs that are either explicitly or implicitly justified in his context. When an agent's context is viewed as 'beliefs' rather than 'knowledge', one often speaks of the 'evidence' or 'justifications' that the agent has for his beliefs rather than 'proofs'.

In order to give a type-theoretical context the dynamics that is needed to implement a context-change approach to utterance meaning or dialogue act interpretation, it is desirable to distinguish the situation where an agent is able to derive a certain belief from his context, from that where he has actually constructed a proof that motivates the belief (i.e., the agent has constructed a complex term inhabiting the corresponding proposition type). One way of doing that is by extending context with *definitions*.

Definitions in type theory have been introduced by De Bruijn (1980). The idea is that, if E is a complex term such that the context allows the derivation of the statement $E : T$ of a certain type T, i.e. $\Gamma \vdash E : T$, then we can add to the context Γ an abbreviation of E. Such an abbreviation is of the form $z = E : T$, where z is a fresh variable. Using such definitions, we can model an agent reaching a certain conclusion by adding to the context a new term as an abbreviation for the complex proof term inhabiting the conclusion type. For instance, in the above example of type-theoretical Modus Ponens, we could extend the context under consideration with the introduction $q = F \cdot p : T$.

In recent years, research in the DenK project has shown the great potential of type-theoretical contexts as formalizations of context information (Borghuis 1994; Ahn 1995; Ahn 2000). The expressive capabilities and proof methods of standard type theory have to be enriched, however, for adequate modelling of the states of information and intention of agents participating in a dialogue, since standard type theory takes into account only the beliefs of a single agent. An extension for two agents each with their own epistemic modalities has been defined by Borghuis (1994). Further extensions are required for the type-theoretical representation of the time-dependent aspects of the behaviour of objects in the application domain, and also for the representation of and reasoning about the temporal aspects of natural language utterances (see Ahn and Borghuis 1996; Ahn 2000).

In the DenK system, CTT is used both for the semantic representation of the utterances exchanged by the user and the system, and also for implementing the system's knowledge of the task domain (the global semantic context) and the shared beliefs of user and system derived from the dialogue (according to the system, i.e. the system's view of the 'local semantic context'). In fact, the system's representation of the global semantic context constitutes a formal model of the domain, which is the working of a modern electron microscope, in the form of a giant type-theoretical context. This CTT context is divided into two parts, called *'common'* and *'private'*. Common is the part of the context containing the information that the system believes to be shared with the user; private contains the beliefs that the system does not believe to be shared (which is most of the global semantic context).

We have seen earlier in this chapter that some aspects of local dialogue context are conveniently represented by means of simple *features*: attribute-value combinations, rather than by complex logical expressions (see also Bunt 1999. In Bunt and Sloot (1996) we have shown how features can be introduced in familiar logical languages; features can also be introduced in CTT as follows. We take a feature *attribute* conceptually to be a function that pairs objects in its domain with values in its range, and the combination of an attribute with a value as a *predicate*. For instance, to model that a certain process has the property expressed by the feature [PROGRESS =: READY], we introduce PROGRESS as a function from processes to 'progress values'; we introduce READY as an object of type *'progress value'*, and we define the combination to be of the type of a predicate applicable to processes, i.e. a function from processes to propositions.

In a formal language that has lambda abstraction, a feature $[A =: v]$, is semantically equivalent to a lambda abstract $\lambda x : A(x) = v$, if v is not nested (i.e., v is not of the form $[A' =: v']$, and one can simply use lambda abstraction instead of features (and conjoined abstractions instead of feature matrices). For nested feature specifications a more complex equivalent lambda abstract can be given (see Bunt and Sloot 1996). CTT is a language with lambda abstraction, so one way to use the logical machinery of CTT for information in attribute-value format is to translate this into lambda expressions. (In other words, feature specifications on this approach are viewed as abbreviations of certain lambda abstractions.)

Alternatively, feature specifications can be introduced into the language at object level. In CTT, this amounts to adding the corresponding clause to the definition of CTT expressions, plus a deduction rule[15] saying that, if F is a type of the form $t_i \to t_j$ and v is of type t_j, then the attribute-value

[15]The precise formulation, in terms of Π-types, is: $\frac{\Gamma \vdash F:(\Pi x{:}A.B) \;\; \Gamma \vdash b:B}{\Gamma \vdash (F =: b):(\Pi x{:}A.t)}$

combination $(F =: v)$ is of type $t_i \to t$.

The above example of the information that process P_1 is ready, can then be represented in CTT as follows:

$$[\quad progval : e,\ process : e,\\ ready : progval,\ progress : process \to progval,\\ p : (progress =: ready) \cdot P_1]$$

The structuring of a type-theoretical context into a part that contains the 'private' beliefs of the system and a part containing the beliefs the system assumes to be mutual, can be viewed as a particular instance of the approach of using so-called *belief spaces* for dealing with nested propositional attitudes. On this approach, the belief state of an agent is treated as a set of propositions which are either explicitly stored or derivable from those that are stored. This set is structured, having e.g. a subset formed by those propositions the agent believes to be shared with another agent. Different sets of propositions, or 'spaces', can be defined in this way to represent information in the scope of a particular, possibly nested, propositional attitude.

The belief-space approach has been pioneered in AI in terms of multiple data bases, and has been formalized by Konolige (1986) (see also Cohen 1978; Allen 1978; Moore 1980). The approach is *deductive* in the sense that the information available to an agent within a particular propositional attitude is defined as those propositions the agent can deduce from a particular set of propositions. Deductive approaches may be contrasted with *model-theoretic* approaches to knowledge representation, where the information available to an agent is defined as those propositions that come out `true` upon recursive evaluation against a model M.

The most familiar form of this is the possible-worlds approach. This approach is computationally very costly, since the facts whose truth an agent has a belief about, have to be represented as true or as false, respectively, in every one of his belief-accessible worlds; and even worse, every elementary fact p that an agent S has *no* belief about, is modelled by adding to each S-belief-accessible world one alternative where p is true and one where p is false. For a realistic domain of discourse, with a large number of potential facts, the representation of an agent's incomplete knowledge involves an astronomic number of worlds (sets of facts); moreover, each of these sets is astronomical in size, since possible worlds are *complete*: every atomic proposition must have a truth value in every world. Note also that, the less an agent knows, the more worlds have to be represented; growth of knowledge is viewed as the elimination of possibilities. The possible-worlds approach is thus good for representing the knowledge of an agent who knows almost everything. Participants in a dialogue typically have highly incomplete local context information, however, particularly about each other's knowledge. In such a situation, the possible-worlds approach would be computationally prohibitively expensive. Ideally, one would prefer to model an agent's knowledge in an incremental rather than an eliminative fashion, representing only the facts he knows, and representing these only once. This leads to *partial models*, where truth values are assigned to only those propositions whose truth is known.

When an agent's information is changed by a communicative act, only certain specific aspects of the agent's beliefs are changed, while most of his beliefs remain the same. It would therefore also be advantageous to design models in a such a way that they can be updated without having to consider the entire belief structure. We have developed such an approach, where information is represented in *'modular partial models'* using structured sets of valuation functions, somewhat akin to Fagin, Halpern and Vardi's 'knowledge structures' (Fagin, Halpern, and Vardi 1984; Fagin and Vardi 1985). We outline this approach in the next section.

6.2 Modular Partial Models

We describe modular partial models for representing the beliefs of two communicating agents.[16] We will describe agents' beliefs in the language of first-order logic extended with a belief operator and call this language DFOL: *Doxastic First-Order Language.* We will later add an intention attitude. We will be specifically concerned with modelling the beliefs of a single agent, including his beliefs about (the beliefs of) another agent, and we will use 'DFOL_s' to indicate the sublanguage of DFOL where all expressions are of the form *S believes that* ϕ, with ϕ a DFOL expression (or of the form *S has the goal that* ϕ, when we have added an intention attitude). We will interpret DFOL_s expressions by means of *modular partial models.*

A modular partial model, or MPM, is in essence a structured set of partial valuation functions. One of the valuations in this set plays a particular role,

[16]The communicative behaviour of an agent in a given context depends on what he believes, not on whether these beliefs are actually true. (Or rather, perhaps, on what he believes he knows; see Thijsse's chapter in this volume.) When discussing modular partial models, intended to model an agent's state of information, we will as before indiscriminately use the terms 'knowledge' and 'belief' for the contents of an information state.

as its extension represents the elementary facts known to the agent whose information state we want to model. The set of valuations is structured by two kinds of relations, corresponding to: (1) the agent-dependence of nested beliefs of one agent about another; (2) logical relations that may give rise to incompleteness in beliefs: disjunction, negation and quantification. The following forms of incomplete information are taken into account:

1. The propositional-logical forms of partiality: disjunctive knowledge and absence of knowledge. Disjunctive knowledge is for instance knowing that John's birthday is the 23rd or the 25th, but not knowing which of the two. Absence of knowledge is e.g. that (S knows that) U does not know that p.

2. For predicate-logical knowledge we have, in addition, the generalization from disjunctive to existentially quantified knowledge, and the possibility of partial and negative knowledge of the extension of a predicate. For instance, you know that there were three Marx brothers, that Groucho and Harpo were two of them, and you don't know the third but you do know it's not Karl.

To deal with these cases, MPMs have the following structural provisions:

1. (a) 'Alternative extensions'. To represent that a valuation F_s captures an agent S's disjunctive knowledge that p or q, F_s has associated with it two partial functions, one assigning `true` to p, the other assigning `true` to q. These functions are called 'alternative extensions of F_s'; they can be thought of as representing alternative ways in which F_s can be extended when more information becomes available.

   (b) 'Absent belief' relations. If an agent S believes that it is not the case that agent U believes that p, then associated with the valuation F_s is a function $F_{s\sim u}$ that assigns `true` to p.

2. (a) 'Anonymous referents'. A kind of 'pseudo-objects' for representing information about individuals whose existence is considered within the scope of a certain propositional attitude. To denote these objects we will use symbols that have some similarity to variables in that distinct symbols do not necessarily correspond to different objects.

   (b) To represent 'negative knowledge' about the extension of a predicate, we split every valuation function F into two functions:

> $F = < F^+, F^- >$, where F^+ assigns 'positive' and F^- assigns 'negative' extensions to predicate terms. (For instance, F_u^+(*Marx-brothers* = {Groucho, Harpo}; F_u^- *(Marx-brothers)* = {Karl}.) To allow the representation of quantified beliefs concerning a predicate of which the extension is only partly known, MPMs have a special relation 'nex' ('negative extension'); see below for its use.

Modular partial models with these provisions can be defined as follows.

Definition 6. A **modular partial model for DFOL**$_\alpha$ is a sixtuple

$M = < D, \mathcal{I}_\alpha, \mathcal{N}, \mathcal{F}, F_\alpha, \mathcal{A} >$, where:
- D is a domain of individuals;
- $\mathcal{I}_\alpha$ is a set of indices, defined as the following strings: $\alpha \in \mathcal{I}_s$; if $i \in \mathcal{I}_s$ then is, iu, $i \sim s$, $i \sim u$ and $i$nex $\in \mathcal{I}_s$; if $i \in \mathcal{I}_s$ and k is a natural number, then $i_k \in \mathcal{I}_s$;
- $\mathcal{N}$ is an indexed set of finite sets N_j of 'anonymous referents', with $N_j \cap D = \emptyset$ for every $j \in \mathcal{I}_s$;
- $\mathcal{F}$ is an indexed set of pairs $< F_i^+, F_i^- >$ of partial functions assigning values to DFOL terms, satisfying the contraints mentioned below;
- $F_\alpha \in \mathcal{F}$;
- $\mathcal{A}$ is an indexed set of subsets of $\mathcal{F}$, specifying the (non-empty) alternative extensions present in the model.

The index α of the valuation F_α is called the *root index* of the model; we will often designate an MPM with root index α by M_α. The model M_s is intended to represent the information state of the agent S. Given a model M_s, we may restrict the index set $\mathcal{I}_s$ to a subset $\mathcal{I}_j$ for some $j \in \mathcal{I}_s$ and restrict the M_s-components $\mathcal{N}, \mathcal{F}$, and $\mathcal{A}$ accordingly. These restricted sets plus the domain D and the valuation F_j then form a substructure of M_s which is itself an MPM with root index j.

Note that the set $\mathcal{I}_\alpha$ of indices as defined above is infinite; for most indices i it will be the case that $F_i = N_i = A_i = \emptyset$. We will use $\mathcal{I}_M$ to denote the subset of indices for which at least one of these sets is not empty. When specifying a particular MPM M, the set $\mathcal{I}_M$ follows from the specification of $\mathcal{F}, \mathcal{N}$ and $\mathcal{A}$, so we will usually omit the specification of $\mathcal{I}_M$.

The indices of the various forms have the following intended significance:

1. F_s assigns to DFOL terms the denotations they have according to agent S.

2. F_{su} does the same according to S's beliefs about U's beliefs. Similarly for F_{sus}, etc.

3. Numerical indices designate alternative extensions: for a given index i, F_{i_k} is an alternative extension of F_i.

4. $F_{s\sim u}$ assigns predicate terms (partial) denotations which S believes that U does not believe them to have. Similarly for $F_{su\sim s}$, etc.

5. Indices of the form $i\,\mathsf{nex}$ are used for the representation of negative existential knowledge (see below).

The explicit representation of negative beliefs creates the danger that 'positive' and 'negative' parts of an MPM contain conflicting information, therefore the set of valuation functions of an MPM M is required to meet the following consistency conditions.

CC1 For all indices $i, j \in \mathcal{I}_M$, polarities $\delta, \delta' \in \{+,-\}$, and nonempty string γ such that $i\gamma \in \mathcal{I}_M$:
a. $F_i^+(P) \cap F_i^-(P) = \emptyset$ for every predicate constant P;
b. $F_i^\delta(c) = F_j^{\delta'}(c)$ for every individual constant c
c. $F_{i\gamma}^+(x) = F_i^+(x)$ for every individual variable x.

CC2 For every index $i \in \mathcal{I}$ and string γ such that $i\gamma \in \mathcal{I}_m$:
$F_{i\gamma}^\delta \cap F_{i\mathsf{nex}\gamma}^\delta = \emptyset$ for $\delta \in \{+,-\}$.

The family of sets $\mathcal{N}$ introduces anonymous referents for those indices j where N_j is not empty. An anonymous referent a is thus introduced at some point in an MPM, namely at the index i where $a \in N_i$. This gives an anonymous referent something like a *scope*: when introduced at index i, it may be used at every index of the form $i\gamma$. This is formalized as follows.

Definition 7. An anonymous referent a is *available* at index i if either a is introduced at that index, i.e. $a \in N_i$, or a is available at some index j such that $i = js$, or $i = ju$, or $i = j \sim s$, or $i = j \sim u$, or $i = j\mathsf{nex}$, or $i = j_k$. We use 'Av_i' to denote the set of anonymous referents, available at index i.

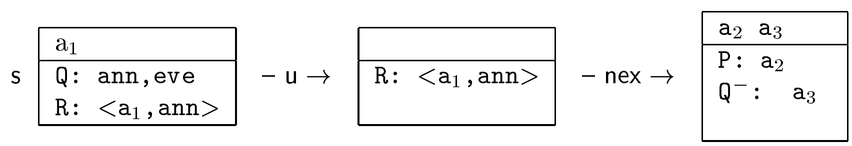
Multiple occurrences of an anonymous referent within its scope represent the same object; an anonymous referent should therefore not be reintroduced within its own scope. Consistency condition **CC3** ensures this.

CC3 For every index $i \in \mathcal{I}$ and nonempty string γ such that $i\gamma \in \mathcal{I}_M$: $N_{i\gamma} \cap \mathsf{Av}_i = \emptyset$.

The valuation functions in $\mathcal{F}$ assign values to the individual and predicate constants of DFOL; for simplicity we will also use these functions to assign values to individual variables (cf. **CC1**). Individual constants have values belonging to the domain D, while variables may be assigned domain objects or anonymous referents as values. More precisely, $F_i(x) \in D \cup \mathsf{Av}_i$, i.e. the value assigned to a variable x at index i is either a domain object or an anonymous referent available at that index.

By way of illustration, consider the model, depicted in a DRT-like form in Fig. 10, which is formally the following set-theoretical structure:

$$
\begin{aligned}
M \quad = \quad & < \{\mathtt{ann}, \mathtt{eve}\},\ \{N_s, N_{s\mathsf{unex}}\},\ \{F_s, F_{su}, F_{s\mathsf{unex}}, \},\ F_s,\ \emptyset > \\
& \text{where} \\
& N_s = \{\mathtt{a}_1\},\ N_{s\mathsf{unex}} = \{\mathtt{a}_2, \mathtt{a}_3\}, \\
& F_s^+ = \{< Q, \{\mathtt{ann}, \mathtt{eve}\} >,\ < R, \{< \mathtt{a}_1, \mathtt{ann} >\} >\}, \\
& F_{su}^+ = \{< R,\ \{< \mathtt{a}_1, \mathtt{ann} >\} >\}, \\
& F_{s\mathsf{unex}}^+ = \{< P,\ \{\mathtt{a}_2\} >\}. \\
& F_{s\mathsf{unex}}^- = \{< Q,\ \{\mathtt{a}_3\} >\}.
\end{aligned}
$$



S believes that $Q(eve)$ and that $Q(ann)$;
S believes that there is an a_1 such that $R(a_1, ann)$ and
U believes that $R(a_1, ann)$;
S believes that U believes that there is no a_2 such that $P(a_2)$.
S believes that U believes that there is no a_3 such that not $Q(a_3)$.

Figure 10: *Example of a simple* MPM.

It may be noted that the modelling of universally quantified beliefs causes a particular problem because an agent may have only partial knowledge of the individuals in the domain. The standard (total as well as partial) model-theoretic semantics of universal quantification is:

$$|| \forall x \in F : \phi(x) \,|| = \bigcap_{d \in ||F||} || \,\phi[d/x]\, ||$$

i.e., every individual d that is an F has the property ϕ (where $||\ \phi\ ||$ designates the value of ϕ according to the model M, with an assignment of values to variables). This doesn't help us very much for constructing an adequate representation of universally quantified beliefs, as the following example shows.

U: Can you tell me whether flight KL403 from Montreal will arrive in timeΓ
S: All flights are diverted to Brussels because of the weather. I'm afraid you have to call Brussels or the KLM for this information.

After the second utterance, U knows that $\forall x \in FLIGHTS$:*Div(x,brussels)*, but this cannot be modelled by U knowing for each flight d that it is divered to Brussels, for U may not know any flight other than the KL403. We must somehow model that U knows that even those flights that he does not know, are diverted or, equivalently, that there are no flights that are not diverted. The latter can be realized in MPM form by means of anonymous referents, allowing the modelling of existential quantification, and a relation between modules that captures the negation of quantification; this relation is indicated by nex. If M_u is an MPM modelling U's knowledge, this amounts to introducing an anonymous referent a_i at index $u\mathsf{nex}$ such that $< \mathsf{a}_i, brussels > \in F^-_{u\mathsf{nex}}(Div)$. (See also the MPM in Fig. 10). Note that this knowledge constitutes a ground for U to know that, in particular, the KL403 is diverted; this illustrates why the second disjunct is needed in clause 1 of the truth definition for MPMs (Definition 8 below).

Normalization in modular partial models
The above definition of MPMs must be supplemented with certain normalization constraints, to avoid unintended and undesirable ways of using alternative extensions, which would have the effect that different MPMs can represent the same information state. Constraints on the proper use of alternative extensions, for instance, rule out a model where S believes that p and q (atomic propositions, i.e. zero-place predicates), not because $F_s^+(p) = F_s^+(q) = \mathtt{true}$, but by $F_s^+(p) = \mathtt{true}$ and F_s having two alternative extensions F_{s1}, F_{s2} such that $F_{s1}^+(p) = \mathtt{false}$ and $F_{s2}^+(q) = \mathtt{true}$. This would amount to modelling the equivalent belief that $p \wedge (\neg p \vee q)$, rather than modelling the belief that $p \wedge q$. The undesirable complexity arises here because an alternative extension assigns a value to a term that already has a value, which goes against the very idea of extension. Constraint **NC1** deals with such cases.

NC1. For every predicate constant P, the valuations at alternative extensions at any index i assigns values that are disjoint with the value assigned at i, i.e., $F^{\delta}_{i_j}(P) \cap F^{\delta'}_{i} = \emptyset$ for any $F_{i_j} \in A_i, \delta \in \{+, -\}$.

As another example, suppose S's belief that $P(a)$ is captured by $P(a)$ being `true` not at index s, but at every index s_k (and being undefined at index s). This would be like modelling that S believes that $p \vee p \vee p \vee ..$ rather than S believes that p. Constraint **NC2** deals with this:

NC2. For every predicate constant P, the valuations at alternative extensions at any index i assign disjoint values, i.e., $F^{\delta}_{i_j}(P) \cap F^{\delta'}_{i_k}(P) = \emptyset$ for any $F_{i_j}, F_{i_k} \in A_i, \delta \in \{+, -\}$.

Such constraints serve to normalize MPMs, making sure that, given certain information to be captured by an MPM, there is a unique MPM that does the job (see further below). An MPM that satisfies the normalization constraints is called a *normal* MPM. Normalization constraints are fairly easily translated into operations for normalizing a give MPM.

Truth in a modular partial model
A modular partial model M_s is intended to represent the beliefs of an agent S, including his beliefs about the beliefs of another agent U. We will write $S \Vdash \phi$ to denote that S believes that ϕ. We thus want DFOL$_s$ expressions $S \Vdash \phi$ to come out true or false when evaluated against a model with root index s. To define the truth conditions of $S \Vdash \phi$ we will use the relations of *verification*, denoted by $\models$, and *falsification*, denoted by $\dashv$. These relations are defined in Definition 8 by simultaneous recursion. In the definition, we use the notation $M_i[a/x]$ to designate the submodel that differs from M_i at most in that $F_j(x) = a$ for all valuations F_j in $\mathcal{I}_{M_i}$.

Definition 8. A formula of the form $S \Vdash \phi$ is true in a modular partial model $M = < D, \mathcal{I}_f, \mathcal{N}, \mathcal{F}, F_s, \mathcal{A} >$ with root index s iff $M \models \phi$.

The verification and falsification of a DFOL formula by a normal (sub-) MPM M_i are defined as follows.[17]

[17]The disjunctive clauses in this definition correspond to cases where a formula is verified (falsified) by consequence of a *stronger* formula being verified (falsified). The second clause of 1a, for instance, represents the case where a formula like $P(a)$ is true because $\forall x : P(x)$ is true. See below about 'honest' models, i.e. models which, for a given set of formulas Γ, represent that an agent knows *only* Γ.

1 Let P be a k-ary predicate constant and t a sequence $(t_1,..,t_k)$ of individual constants or variables. Let F_i^* be the same as F_i except possibly for some arguments t_j, for which $F_i^*(t_j) \in \mathsf{Av}_i$ $(j = 1,..,k)$. We write $F(t)$ to abbreviate $< F(t_1),..,F(t_k) >$.

a. $M_i \mathrel{|\!\equiv} P(t) \iff F_i^+(t) \in F_i^+(P)$ or $F_i^+(t) \in F^-_{i\mathsf{nex}}(P)$ or $i \in A_j$ for some $j \in \mathcal{I}_s$ and $M_j \mathrel{|\!\equiv} P(t)$

b. $M_i \mathrel{\equiv\!|} P(t) \iff F_i^-(t) \in F_i^-(P)$ or $F_i^+(t) \in F^+_{i\mathsf{nex}}(P)$ or $M_j \mathrel{\equiv\!|} P(t)$ for some $i \in \mathcal{A}_j, j \in \mathcal{I}_s$.

The remaining clauses apply to any DFOL expressions ϕ, ψ.

2 a. $M_i \mathrel{|\!\equiv} \neg\phi \iff M_i \mathrel{\equiv\!|} \phi$

b. $M_i \mathrel{\equiv\!|} \neg\phi \iff M_i \mathrel{|\!\equiv} \phi$

For $A = S$ and $\alpha = s$, or $A = U$ and $\alpha = u$:

3 a. $M_i \mathrel{|\!\equiv} \phi \vee \psi \iff M_i \mathrel{|\!\equiv} \phi$ or $M_i \mathrel{|\!\equiv} \psi$, or for every index $j \in A_i$, $M_j \mathrel{|\!\equiv} \phi$ or $M_j \mathrel{|\!\equiv} \psi$, or the index i is of the form $\gamma * \alpha$ and $M_\gamma \mathrel{|\!\equiv} (A \Vdash \phi \vee A \Vdash \psi)$

b. $M_i \mathrel{\equiv\!|} \phi \vee \psi \iff M_i \mathrel{\equiv\!|} \phi$ and $M_i \mathrel{\equiv\!|} \psi$

4 a. $M_i \mathrel{|\!\equiv} A \Vdash \phi \iff M_{i\alpha} \mathrel{|\!\equiv} \phi$ or $M_{i\sim\alpha} \mathrel{\equiv\!|} \phi$ or $i \in A_j$ for some $j \in \mathcal{I}_s$ and $M_j \mathrel{|\!\equiv} A \Vdash \phi$.

b. $M_i \mathrel{\equiv\!|} A \Vdash \phi \iff M_{i\alpha} \mathrel{\equiv\!|} \phi$ or $M_{i\sim\alpha} \mathrel{|\!\equiv} \phi$ or $i \in A_j$ for some $j \in \mathcal{I}_s$ and $M_j \mathrel{\equiv\!|} A \Vdash \phi$.

5 a. $M_i \mathrel{|\!\equiv} \exists x\colon \phi \iff$ there is an $a \in D \cup A\mathrm{v}_i$ such that $M_i[a/x] \mathrel{|\!\equiv} \phi$ or for every index $j \in A_i$ there is an $a \in D \cup A\mathrm{v}_j$ such that $M_j[a/x] \mathrel{|\!\equiv} \phi$, or the index i is of the form $\gamma * \alpha$ and $M_\gamma \mathrel{|\!\equiv} A \Vdash \exists x\colon \phi$

b. $M_i \mathrel{\equiv\!|} \exists x\colon \phi \iff$ there is an $a \in N_{i\,\mathsf{nex}}$ such that $M_i[a/x] \mathrel{|\!\equiv} \phi$ and there is no $a \in D \cup \mathsf{Av}_i$ such that $M_i[a/x] \mathrel{|\!\equiv} \phi$.

We introduce conjunction and universal quantification by their usual definitions in terms of disjunction, negation and existential quantification. This has the effect that, for instance, $M_i \mathrel{|\!\equiv} \phi \,\&\, \psi \iff M_i \mathrel{|\!\equiv} \phi$ and $M_i \mathrel{|\!\equiv} \psi$.

6.3 MPMs as representations of information states

The definition of normal modular partial models allows us to prove that every *honest* set D of DFOL_α formulas, that is every set of formulas that characterizes a logically possible state of information, has a unique normal MPM which verifies exactly the formulas of D plus their logical consequences.

The notion of 'honesty' of knowledge has been introduced by Halpern and Moses (1986) in relation to knowledge bases (see also Thijsse 1992.).[18] They noted that not all formulas of an epistemic logical language characterize a state of knowledge. For example, it cannot possibly be the case that a knowledge base only 'knows' that it either knows that p or it knows that q. For if the *only* knowledge in a knowledge base is that it knows p or it knows q, then it does not know p and it does not know q, which is inconsistent with the assumption we started with.

The notion of honesty is also relevant when it comes to modelling states of human knowledge. It is less obvious to what extent logical consistency should be required of such states, but it seems obvious, for instance, that a human agent cannot honestly claim to only know whether p, without knowing that p or that $\neg p$. (This is a special case of the above example, with $\neg p$ for q.)

Figure 11 shows the normal MPM representing that agent S only knows that $p \vee q$.

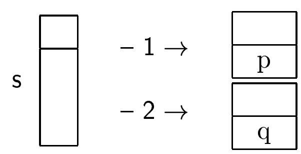


Figure 11. *Normal* MPM *representing that S only knows that $p \vee q$*

The truth definition of DFOL_s has been formulated in such a way that an MPM supports or rejects the truth of a formula only if it has the form $S \Vdash \phi$, i.e. *S believes that ϕ*. Such formulas are always honest if they are not logically false (like $p \wedge \neg p$), since they express a speaker's belief. Notice that the embedded formula ϕ may be a 'dishonest' one, like *'U knows that p or U knows that q'*; the complete expression $S \Vdash (U \Vdash p \vee U \Vdash q)$ is a perfectly honest formula, characterizing a belief state of S. MPMs are thus especially aimed at the representation of the information corresponding to

[18]For an analysis of the notion of honesty in the context of epistemic logic see Van der Hoek, Jaspars, and Thijsse (1996; 2000).

honest formulas. It can be proved that every consistent set D of honest formulas has exactly one normal modular partial model, that supports only the formulas of D plus any formula that logically follows from D. We first prove that an honest formula has a unique normal MPM, up to alphanumeric variation.[19]

Theorem 1. *For any honest DFOL$_s$ formula ψ there exists a unique normal* MPM *M^ψ that supports only ψ and any formula entailed by ψ, while leaving the truth of all other formulas undefined.*

Proof outline. The proof goes by induction on the length of formulas. Every DFOL$_s$ formula being of the form $S \Vdash \phi$, we use induction on the length of ϕ.

1. Length 1. If ϕ is of the form $P(t)$ with argument sequence t, then M^ψ is the MPM where $F_s^+(t) \in F_s^+(P)$ and all other F_i^δ empty for $\delta \in \{+,-\}$. (If ϕ is of the form $\neg P(t)$ then similarly, with $F_s^-(t)$ instead of $F_s^+(t)$.) If ϕ is of the form $\exists x : P(t)$ then M^ψ is the MPM where $F_s^+(t) \in \mathsf{Av}_s \cap F_s^+(P)$ and all other F_i^δ are empty ($\delta \in \{+,-\}$).

2. From length k to length $k+1$. There are three cases to consider: negation, disjunction, and embedding under 'agent believes that'. Let χ be a DFOL formula of length k and M^χ the corresponding normal MPM; $M^\chi = < D^\chi, \mathcal{N}^\chi, \mathcal{F}^\chi, F^{\chi s}, \mathcal{A}^\chi >$.

   (a) Case $\phi = \chi \vee \zeta$ where ζ is of the form $P(t)$ with t a sequence of constants, or of the form $\exists t' : P(t)$ with t' a sequence of variables occurring in t. If χ entails ζ then $M^\psi = M^\chi$. (If χ entailed $\neg\zeta$ then ψ would not be an honest formula.) Else $M^\psi = < D^\psi, \mathcal{N}^\psi, \mathcal{F}^\psi, F^{\psi s}, \mathcal{A}^\psi >$ is constructed from M^χ as follows.

   Let $F_i^{\chi\delta}$ be the valuation with index i and polarity δ in M^χ, and $F_i^{\psi\delta}$ the corresponding valuation in M^χ, and similarly for N_i^ψ etc. Then:

   - $D^\psi = D^\chi$;
   - $N_s^\psi = \emptyset$; $F_s^{\psi\delta} = \emptyset$;
   - $N_{s_1 j}^\psi = N_{s_j}^\chi$ and $F_{s_1 j}^{\psi\delta} = F_{s_j}^{\chi\delta}$ for $j \in \mathcal{I}_{M^\psi}$;

[19] Since the names of anonymous referents are meaningless, an MPM can only be uniquely determined up to alphabetic variation. Similarly for the numbers used to distinguish alternative extensions. Naming and numbering conventions may be introduced to cope with this.

- if $\zeta = P(t)$ then $N^{\psi}_{s_2} = \emptyset$; $Dom(F^{\psi+}_{s_2}) = \{P, \{t_k | t_k \in t\}\}$, $F^{\psi+}_{s_2}(t_k) = F^{\chi+}_{s}(t_k)$ for $t_k \in t$ and $F^{\psi+}_{s_2}(P) = F^{\chi+}_{s}(t)$;
  if $\zeta = \exists t' : P(t)$ then $N^{\psi}_{s_2}$ is a set of as many anonymous referents as there are elements in t'; $F^{\psi+}_{s_2}$ (arbitrarily) assigns a member of $N^{\psi}_{s_2}$ to each variable in t'; otherwise $F^{\psi+}_{s_2}$ is as in the previous case;
- $\mathcal{A}^{\psi} = \{A^{\psi}_{m} | m$ is of the form s_{1j} and $A^{\psi} s_{1j} = A^{\chi}_{s_j}\}$.

(b) Case $\phi = A \Vdash \chi$. The central point in constructing M^{ψ} in this case is that $F^{\psi+}_{s} = F^{\chi-}_{s} = \emptyset$, and for every index $i \neq s$ and polarity δ: $F^{\psi\delta}_{i\alpha} = F^{\chi\delta}_{i}$.

(c) Case $\phi = \neg\chi$. Compared to M^{χ}, the main points of difference are:

- the contents of F^{+}_{s} and F^{-}_{s} are swapped;
- the contents of alternative extensions are treated as negated conjuncts;
- existential beliefs are moved to negative existential beliefs, i.e. $F^{\psi\delta}_{\mathsf{inex}\gamma} = F^{\chi\delta}_{i\gamma}$;
- positive beliefs about U's beliefs are turned into negative beliefs, i.e. $F^{\psi\delta}_{i\sim u\gamma} = F^{\chi\delta}_{iu\gamma}$.

In order to prove the existence of MPMs for sets of honest formulas, we first introduce the notion of the *merge* of two MPMs. The idea of merging two modular partial models is to construct the minimal MPM that contains the beliefs of the two MPMs involved and nothing more.

Definition 9. Given two MPMs $M = < D, \mathcal{N}, \mathcal{F}, F_{\alpha}, \mathcal{A} >$ and $M' = < D', \mathcal{N}', \mathcal{F}', F_{\alpha}, \mathcal{A}' >$, the merge $M \otimes M'$ is the quintuple $< D_{M\otimes M'}, \mathcal{N}_{\mathcal{M}\otimes\mathcal{M}'}$, $\mathcal{F}_{\mathcal{M}\otimes\mathcal{M}'}$, F_{α}, $\mathcal{A}_{\mathcal{M}\otimes\mathcal{M}'} >$, where:

- $D_{M\otimes M'} = D \cup D'$;
- $\mathcal{N}_{\mathcal{M}\otimes\mathcal{M}'} = \{N | N = N_j \cup N'_j, j \in \mathcal{I} \cup \mathcal{I}'\}$;
- $\mathcal{F}_{\mathcal{M}\otimes\mathcal{M}'} = \{< F^{+}_{i} \cup F'^{+}_{i}, F^{-}_{i} \cup F'^{-}_{i} > | i \in \mathcal{I} \cup \mathcal{I}'\}$;
- $F_{(M\otimes M')\alpha} =< F^{+}_{\alpha} \cup F'^{+}_{\alpha}, F^{-}_{\alpha} \cup F'^{-}_{\alpha} >$
- $\mathcal{A}_{\mathcal{M}\otimes\mathcal{M}'} = \mathcal{A} \cup \mathcal{A}'$.

We have seen above that the sets of valuations and anonymous referents have to satisfy certain consistency conditions in order to qualify as a modular partial model, and the merge of two MPMs therefore exists only if the sets $\mathcal{F}_{\mathcal{M}\otimes\mathcal{M}'}$ and $\mathcal{N}_{\mathcal{M}\otimes\mathcal{M}'}$ satisfy these conditions. Intuitively, this means that two MPMs can only be merged if they do not contain conflicting information. If they can be merged, the merge is clearly uniquely determined by the definition. Moreover, the merge can be shown to form an 'honest' representation of the beliefs in the MPMs involved. The following theorem establishes this, generalizing from the case of two MPMs to any countable number of them.

Theorem 2. *For any countable honest set D of DFOL formulas there is a uniquely determined (up to alphanumeric variation) normal modular partial model supporting only the formulas of D and any formula that logically follows from D, while leaving the truth of any other formula undefined.*

Proof outline. The proof goes by induction on the cardinality of D.

1. If D is a singleton set, then Theorem 1 applies.

2. If D contains two or more formulas, then choose an arbitrary member ϕ from D. Let $D = D' \cup \{\phi\}$. D having cardinality k, the normal MPM $M^{D'}$ exists, as does M^{ϕ}, by Theorem 1. The normalized merge M^n_D, i.e. the MPM obtained by normalizing $M_{D'} \otimes M_{\phi}$, has the property that any simpler MPM lacks some information that would be needed to support all the formulas in $D' \cup \{\phi\}$. Moreover, it supports any formula ψ entailed by $D' \cup \{\phi\}$ and does neither support nor falsify any other formula ψ, as can be proved again by induction on the length of ψ.

6.4 Modular partial models and context modelling

The modular partial models considered so far are too simple to be of much use in modelling the task-related beliefs and intentions in an agent's local semantic and cognitive context. In particular, we should take additional attitudes into account, as mentioned above: weak beliefs, mutual beliefs and (epistemic) intentions.

The extension of MPMs with additional doxastic attitudes is technically rather straightforward (at least when we model intentions by means of goals; cf. Bratman 1987 and Hobbs 1990), but does complicate the truth definitions considerably because of the interactions between the various attitudes, such as the following:

- Weak and strong beliefs should be mutually compatible, in the sense that one cannot weakly believe that ϕ and at the same time strongly believe that $\neg\phi$. Also, having a weak belief about ϕ implies an 'awareness' of ϕ, and thus the strong belief that $\phi \vee \neg\phi$.

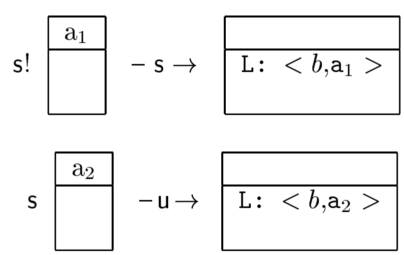
- Epistemic intentions should also be compatible with strong beliefs. One cannot want to know whether ϕ while at the same time knowing that $\neg\phi$, for instance. Also, intentions presuppose awareness, so one cannot honestly want to know whether ϕ without believing that $\phi \vee \neg\phi$.

- Mutual belief entails simple belief of both dialogue partners, as well as nested beliefs with arbitrary level of nesting. For instance, if S weakly believes ϕ to be mutually believed, than S weakly believes that U believes that S believes that ϕ.

Leaving the formalization of these extensions aside, let us indicate for a concrete example of a dialogue act what the MPM might look like when it comes to modelling the preconditions on the local semantic and cognitive context. Consider the example of a WH-QUESTION such as *Where does Bill live?* We consider the two most important preconditions, when S asks this question to U:

C1: S wants to know where Bill lives;
C2: S believes that U knows where Bill lives.

Using the index $s!$ for *S has the goal*, we may represent this with the MPM depicted in Fig. 11.



S wants there to be an a_1 such that S knows that Bill lives in a_1;
S believes that there is an a_2 such that U knows that Bill lives in a_2.

Figure 11: *Example of an* MPM *with an intention attitude.*

This MPM visualizes the following set-theoretical structure:

$$M_s \quad = \quad <\emptyset,\ \{N_s, N_{s!}\},\ \{F_s, F_{su}, F_{s!}, F_{s!s}\},\ F_s,\ \emptyset>$$
where
$N_s = \{\mathtt{a}_2\}$, $N_{s!} = \{\mathtt{a}_1\}$,
$F^+_{su} = \{< L, \{< \mathtt{b}, \mathtt{a}_2 >\} >\}$,
$F_{s!s} = \{< L, \{\mathtt{b}, \mathtt{a}_1\} >\}$.

Apart from the addition of the relevant index types, the definition of MPMs remains the same. The truth definition of DFOL (Definition 8) needs one extra clause for the intention attitude; using $S \lhd \phi$ to designate that S wants that ϕ, this clause goes as follows:

Definition 10. A formula of the form $S \lhd \phi$ is true in a modular partial model M iff $M_{s!} \models\!\!\equiv \phi$. The additional clause in the truth definition is as follows (where A stands for S or U (the agents) and α for the corresponding index s or u, respectively):

6 a. $M_i \models\!\!\equiv A \lhd \phi \iff M_{i\alpha!} \models\!\!\equiv \phi$ or $M_{i\sim\alpha!} \equiv\!\!| \phi$ or $i \in A_j$ for some $j \in \mathcal{I}_s$ and $M_j \models\!\!\equiv A \lhd \phi$.

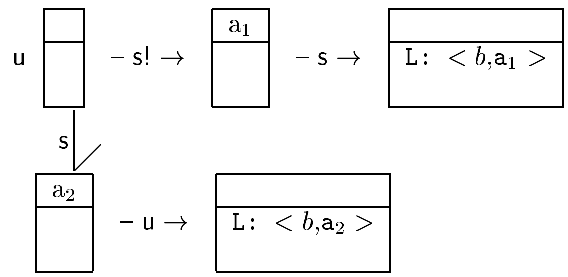
b. $M_i \equiv\!\!| A \lhd \phi \iff M_{i\alpha!} \equiv\!\!| \phi$ or $M_{i\sim\alpha!} \models\!\!\equiv \phi$ or $i \in A_j$ for some $j \in \mathcal{I}_s$ and $M_j \equiv\!\!| A \lhd \phi$.

Continuing the example of S asking where Bill lives: if U understands the utterance correctly, and builds up the belief that conditions C1 and C2 are satisfied, then this can be represented in the form of the MPM shown in Fig. 12, which is to be merged with the MPM that represents U's information state before the dialogue act took place.

This illustrates that an MPM representation of local epistemic and intentional context is simple to update, due to the modular character of MPMs.

7 Conclusions

We have used the analysis of the meanings of dialogue utterances in terms of context changes to obtain insights both into the conceptual content of local dialogue contexts and into the formal and computational modelling of such contexts. We have shown that a strict application of the context-change approach to utterance meaning in some respects refines speech act theoretic analyses, and in other respects has analytic consequences that contradict



U believes that S wants there to be an a_1 such that S knows that Bill lives in a_1;
U believes that S believes that there is an a_2 such that U knows that Bill lives in a_2.

Figure 12: *Effect of updating an* MPM.

standard analyses. In particular, we argued against the standard analysis of indirect speech acts, because it uses contextual information to compute 'indirect' illocutionary forces. The indirect forces would have the effect of adding certain information to the context; however, this is precisely the information that was used to compute these forces, i.e. information that was alread available in the context. From a logical point of view, computing such indirect interpretations would therefore not give rise to additional context changes, and would for a dialogue agent be a waste of effort. The same goes for using contextual information to compute a more specific communicative function (like CONFIRM instead of ANSWER).

A similar argument can be developed for the use of context in computing the propositional aspect of utterance meanings. The syntactic composition of an utterance and the meanings of its constituent words can be taken to determine a 'direct' propositional content of an utterance, which is typically vague and ambiguous, since the determination of word senses, the resolution of anaphora, ellipsis, PP attachment, quantifier scope, etc. are only possible on the basis of contextual information. If an utterance u has a 'direct' propositional content σ_u, which may be underspecified in various respects, and if contextual information can be used to compute a more specific content σ'_u, then the question arises again whether it would be sensible for a dialogue agent to spend the effort to do so. Again, it would seem not, since computing σ'_u would strictly depend on information already in his context, so there would be no additional context change. Another way to put this is that, *in a context that allows the computation of a specific propositional meaning* σ'_u*, the underspecified meaning* σ_u *leads to the same context change as* σ_u*; in other words, in such a context, the underspecified meaning is equivalent to the more specific meaning.*

The suggestion that it may be most efficient for a dialogue agent to build direct illocutionary forces and 'direct', underspecified contents, holds only if dialogue agents can operate directly with such constructs. In particular, dialogue agents would be required to operate with 'underspecified' beliefs and intentions. To explore this in any detail goes beyond the scope of this chapter, but by way of illustrating the possibilities, consider the following example, using modular partial models. Suppose agent S is considering the adoption of the disjunctive belief that p or q. If S already knows that $\neg q$, S could disambiguate the disjunction to p, but the MPM formalism is indifferent about this, since the belief state depicted in Fig. 13a, corresponding to S believing that $((p \vee q) \wedge \neg q)$, is equivalent to the state depicted in Fig. 13b, where S has only the belief that p. In other words, when S interprets the input in an underspecified way as carrying the information that $p \vee q$, then in the context where S already knows $\neg q$, it will be *as if* S *interprets the input in the more specific way where it conveys the information* p.

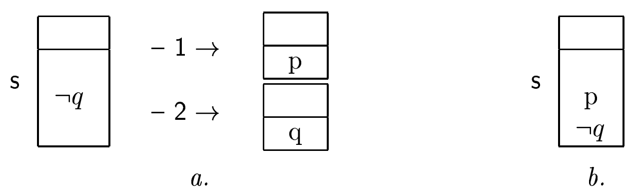


Figure 13. MPM *representing* S *believes that* $((p \vee q) \wedge \neg q)$ *without (a) and after normalization (b).*

The two MPMs are logically equivalent, but the one in Fig. 13a does not satisfy normalization constraint **NC1**. It seems appealing to separate normalization from interpretation, since normalization does not change a context, from the point of view of information content. If agent S interprets an input that would provide the information $p \vee q$, while S already knows $\neg q$, the interpretation process might simply update S's state in the way indicated in Fig. 13a (merging the MPMs for *S believes that* $\neg q$ and *S believes that* $p \vee q$.) An agent may, in addition, be concerned about optimizing the representation of his state, and thus simplify his model to that of Fig. 13b by applying a normalization operation. The attraction of this view is that normalization, which is indeed a form of optimization of representations, is detached from understanding.

Our dialogue-based explorations of conceptual and formal aspects of context modelling have led us to distinguish five conceptual 'dimensions' of dialogue contexts: the semantic, cognitive, physical-perceptual, linguistic, and social ones, where two kinds of information were identified within the cognitive dimension: knowledge of the dialogue partner and knowledge of one's own state of processing. Investigating the logical properties of the information in these dimensions for the *local* aspect of context, i.e. the context as far as it can be changed through communication, has led us to the conclusion that local context representations are best structured as consisting of four components:

1. beliefs and intentions of the agent about the task domain, the communication, and the partner's beliefs and intentions;

2. processing state information and information about the physical and perceptual context;

3. interactive and reactive pressures to perform communicative acts relating to social obligations (local social context);

4. linguistic context: a record of the events that make up the dialogue so far, plus a representation of aspects of future dialogue that may have been planned.

Considering the formal and computational modelling of these contexts, we argued that partial logics seem most appropriate, and outlined two such formalisms: the proof-theoretic formalism of constructive type theory, as applied and further developed in the DenK project, and the model-theoretic formalism of modular partial models. Both formalisms seem promising, but further work is needed to unequivocally establish their adequacy for effective modelling of all the types of information that have been distinguished.

Acknowledgements

I would like to thank the members of the PLUS, DenK, TENDUM and ΔELTA dialogue projects, for stimulating discussions that have helped to shape the ideas described in this chapter, in particular Jens Allwood, Joakim Nivre, Gérard Sabah, Robbert-Jan Beun, René Ahn, Harry Bego, Frens Dols, Wessel Kraaij and Wim van Loo. Special thanks are due to Elias Thijsse for detailed comments on an earlier version of the section on modular partial models.

References

Ahn, R. (1995). Logical model of the electron microscope. DENK Report 95/15, Tilburg: ITK.

Ahn, R. (2000). Agents, objects and events: a computational approach to knowledge, observation and communication. Ph.D. thesis (forthc.)

Ahn, R. and T. Borghuis (1996). Chronicle. DENK Report 96/91, Tilburg: ITK.

Ahn, R. and H. Kolb (1990). Discourse representation meets constructive mathematics. In L. Kálmán and l. Pólos (eds.), *Papers from the Second Symposium on Logic and Language*. Budapest: Akademiai Kiadó.

Alexandersson, J., B. Buschbeck, T. Fujinami, M. Kipp, S. Koch, E. Maier, N. Reithinger, B. Schmitz and M. Siegel (1998). Dialogue acts in VERBMOBIL-2, Second Edition. Verbmobil Report No. 226, Saarbrücken: DFKI.

Allen, J. (1978). *Recognizing Intention in Dialog.* Ph.D. thesis, Dept. of Computer Science, University of Toronto.

Allen, J. and M. Core (1997). Draft of DAMSL: dialogue act markup in several layers. Technical Report, Discourse Research Initiative, Schloss Dagstuhl.

Allwood, J. (1976). *Linguistic communication as action and cooperation.* Gothenburg Monographs in Linguistics 2, Göteborg University.

Allwood, J. (1994). Obligations and options in dialogue. *THINK Quarterly 3*(1), 9–18.

Allwood, J., J. Nivre, and E. Ahlsén (1990). Speech management: on the non-written life of speech. *Nordic Journal of Linguistics 13*, 1–48.

Allwood, J., J. Nivre, and E. Ahlsén (1992). On the semantics and pragmatics of linguistic feedback. *Journal of Semantics 9*, 1–29.

Barendregt, H. (1991). Introduction to generalized type systems. *Journal of Functional Programming 1*(2).

Barendregt, H. (1992). Lambda calculi with types. In S. Abramsky and D. Gabbay and T. Maibaum (eds.), *Handbook of Logic in Computer Science.* Oxford: Oxford University Press.

Beun, R.J. (1989). *The recognition of declarative questions in information dialogues.* Ph.D. thesis, Tilburg University.

Beun, R.J. and L. Kievit (1996). Resolving definite descriptions in dialogue. In *Proc.* 5^{th} *International Conference on Pragmatics*, Mexico City.

Bilange, E. (1991). *Modélisation du dialogue oral personne–machine par une approche structurelle: théorie et réalisation.* Ph.D. thesis, Univ. Rennes 1.

Borghuis, T. (1994). *Coming to terms with modal logic.* Ph. D. thesis, Eindhoven University of Technology, Eindhoven.

Bratman, M. (1987). *Intention, plans, and practical reason.* Cambridge, MA: Harvard University Press.

Bunt, H. (1979). Conversational principles in question-answer dialogues. In D. Krallmann (ed.), *The Theory of Questions*. Essen: Narr Verlag.

Bunt, H. (1989). Information dialogues as communicative action in relation to partner modeling and information processing. In M. Taylor, F. Néel, and D. Bouwhuis (eds.), *The structure of multimodal dialogue*. Amsterdam: North-Holland Elsevier.

Bunt, H. (1994). Context and dialogue control. *THINK Quarterly 3*(1), 19–31.

Bunt, H. (1995). Dynamic interpretation and dialogue theory. In M. Taylor, F. Néel, and D. Bouwhuis (eds.) (2000), *The structure of multimodal dialogue II*, Amsterdam: Benjamins.

Bunt, H. (1996). Interaction management functions and context representation requirements. In *Proc. Twente Workshop on Dialogue Management in Natural Language Systems*. Enschede: Universiteit Twente.

Bunt, H. (1997). Dialogue context modelling. In P. Brezillon, M. Cavalcanti and R. Nossum (eds.) *Formal Aspects in Context*. Dordrecht: Kluwer, 2000.

Bunt, H. (1999). Context Representation for Dialogue Management. In P. Bouquet, L. Serafini, P. Brezillon, M. Beneceretti, and F. Castellani (eds.) *Modeling and Using Context*. Lecture Notes in Artificial Intelligence Vol. 1668. Berlin: Springer.

Bunt, H., R. Ahn, R.J. Beun, T. Borghuis, and C. van Overveld (1998). Cooperative multimodal communication in the DenK project. In H. Bunt, R.J. Beun, and T. Borghuis (eds.), *Multimodal Human-Computer Communication*. Berlin: Springer.

Bunt, H. and J. Allwood (1993). The PLUS discourse model. Esprit project PLUS P5254 deliverable, Tilburg: ITK.

Bunt, H., R.J. Beun, F. Dols, v.d. Linden J., and G. Schwartzenberg (1984). The theoretical foundations of the TENDUM system. In *IPO Annual Progress Report 19*. Eindhoven: IPO.

Bunt, H. and K. v. d. Sloot (1996). Parsing as dynamic interpretation of feature structures. In H. Bunt and M. Tomita (eds.), *Recent Advances in Parsing Technology*. Boston/Dordrecht: Kluwer.

Clark, H. and C. Marshall (1981). Definite reference and mutual knowledge. In A. Joshi, B. Webber, and I. Sag (eds.), *Elements of Discourse Understanding*. Cambridge: Cambridge University Press.

Cohen, P. (1978). *On knowing what to say: planning speech acts*. Ph.D. thesis, University of Toronto. Technical Report 118, Dept. of Computer Science.

Cohen, P. and H. Levesque (1990). Intention in choice with commitment. *Artificial Intelligence 42*, 213–261.

Coquand, T. (1985). Une théorie de constructions. Thèse de troisième cycle, Université de Paris VII.

De Bruijn, N. (1980). A survey of the Project Automath. In J. Seldin and J. Hindley (eds.) *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalisms*. New York: Academic Press.

Fagin, R., J. Halpern, and M. Vardi (1984). A model-theoretic analysis of knowledge: preliminary report. In *Proc. 25th IEEE Symposium Foundations of Computer Science*.

Fagin, R. and M. Vardi (1985). An internal semantics for modal logic. In *Proc. 17th ACM Symposium on Theory of Computing*, Providence, RI.

Gazdar, G. (1981). Speech act assignment. In A. Joshi, I. Sag, and B. Webber (eds.), *Elements of discourse understanding*. Cambridge, UK: Cambridge University Press.

Geurts, B. and G. Rentier (1993). Quasi logical form in PLUS. Esprit project PLUS report, Tilburg: ITK.

Halpern, J. (1986). Reasoning about knowledge: and overview. In J. Halpern (ed.), *Theoretical Aspects of Reasoning about Knowledge: The 1986 Conference*. Los Altos, CA: Morgan Kaufmann.

Halpern, J. and Y. Moses (1986). Towards a theory of knowledge and ignorance: a preliminary report. In *Proc. AAAI Workshop on Non-Monotonic Reasoning*, New Paltz, NY, 125–143.

Halpern, J. and Y. Moses (1990). Knowledge and common knowledge in a distributed environment. *Journal of the ACM 37*, 549–587.

Hobbs, J. (1990). Artificial intelligence and collective intentionality: Comments on Searle and Grosz and Sidner. In P. Cohen, J. Morgan and M. Pollack (eds.) *Intentions in Communication,* 445–459. Cambridge, MA: MIT Press.

Hobbs, J., M. Stickel, D. Appelt, and P. Martin (1993). Interpretation as abduction. *Artificial Intelligence 63*, 69–142.

Hoek, W. v.d., J. Jaspars, and E. Thijsse (1996). Honesty in partial logic. *Studia Logica 56*(3), 326–360.

Hoek, W. v.d., J. Jaspars, and E. Thijsse (2000). Persistence and Minimality in Epistemic Logic. *Annals of Mathematics and Artificial Intelligence*.

Hulstijn, J. (2000). *Dialogue models for inquiry and transaction.* Ph.D dissertation, University of Twente, Enschede.

Jekat, S., A. Klein, E. Maier, I. Maleck, M. Mast, and J. Quantz (1995). Dialogue acts in Verbmobil. *Verbmobil Technical Report No. 65*, Saarbrücken.

Kamp, H. and U. Reyle (1993). *From discourse to logic: introduction to model theoretic semantics of natural language*. Dordrecht: Kluwer.

Konolige, K. (1986). *A deduction model of belief.* Los Altos, CA: Morgan Kaufmann Publishers.

Krahmer, E. and P. Piwek (1999). Presupposition projection as proof construction. In H. Bunt and R. Muskens (eds.) *Computing Meaning, Vol 1,* 281–300. Dordrecht: Kluwer.

Krause, P. (1995). Presupposition and abduction in type theory. In: E. Klein, S. Manandhar, W. Nutt, and J. Siekmann (eds.) *Proc. Edinburgh Conf. on Computational Logic and Natural Language Processing*. Edinburgh: HCRC.

Levesque, H. (1984). A logic of implicit and explicit belief. In *Proc. AAAI'84 Conference*.

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

Maier, E. (1994). Dialogmodellierung in Verbmobil. Technical Report Verbmobil 31, Saarbrücken: DFKI.

Martin-Löf, P. (1984). *Intuitionisitc Type Theory*. Naples: Bibliopolis.

Moeschler, J. (1989). *Modélisation du dialogue, réprésentation de l'inférence argumentative*. Paris: Hermès.

Moore, R. (1980). Reasoning about knowledge and action. Technical Report Artificial Intelligence Center Technical Note 191, Menlo Park, CA.

Moore, R. (1985). A formal theory of knowledge and action. In J. Hobbs and R. Moore (eds.), *Formal Theories of the Commonsense World*. New Jersey: Ablex.

Nivre, J. (1995). Communicative action and feedback. In M. Baker, R.J. Beun, and M. Reiner (eds.), *Dialogue in instruction*. Heidelberg: Springer.

Piwek, P. (1995). Annotated type theory for information representation. DenK Report 95/17. Eindhoven: IPO.

Piwek, P. (1998). *Logic, Information & Conversation*. Ph.D. thesis, Eindhoven University of Technology.

Prince, V. and D. Pernel (1995). Several knowledge models and a blackboard memory for human-machine robust dialogues. *Natural Language Engineering 1*(2), 113–145.

Rats, M. (1996). *Topic management in information dialogues*. Ph. D. thesis, Tilburg University. *ITK Dissertation Series 1996-2.*

Roulet, E. (1985). *L'articulation du discours en français contemporain*. Bern: Peter Lang.

Schegloff, E. and H. Sacks (1973). Opening up closings. *Semiotica 7*(4), 289–327.

Searle, J. (1975a). *Indirect speech acts*. New York: Academic Press.

Searle, J. (1975b). *A taxonomy of illocutionary acts. Minnesota Studies in the Philosophy of Science 7.* Minneapolis: University of Minnesota Press.

Sundholm, G. (1986). Proof theory and meaning. In D. Gabbay and F. Guenthner (eds.) *Handbook of Philosophical Logic, Vol. 3,* 471–506. Dordrecht: Reidel.

Sundholm, G. (1989). Constructive generalised quantifiers. *Synthese 79,* 1–12.

Thijsse, E. (1992). *Partial logic and knowledge representation*. Ph. D. thesis, Tilburg University.