A Context-Change Semantics for Dialogue Acts

Harry Bunt

Abstract This chapter presents an update semantics for dialogue acts, defined in terms of combinations of 'elementary update functions'. This approach allows fine-grained distinctions to be made between related types of dialogue acts, and relations like entailment and exclusion between dialogue acts to be established. The approach is applied to the inventory of dialogue act types in the DIT^{++} taxonomy, using dialogue act representations as defined in the Dialogue Act Markup Language (DiAML), which is part of the recently established ISO standard 24617-2 for dialogue act annotation.

1 Introduction

The notion of a dialogue act plays a key role in studies of dialogue, in particular in the analysis of communicative behaviour and in the design of spoken dialogue systems and embodied conversational agents. In empirical studies of human conversation, dialogue acts are often used to characterize different types of communicative behaviour. In studies of utterance meaning, dialogue acts are used to relate utterances to information states and how these are changed by communication. In spoken dialogue systems, dialogue acts are used in dialogue management, i.e. in the processes of deciding how to continue an ongoing dialogue.

Over the years, a variety of dialogue act inventories and taxonomies has emerged, including the TRAINS inventory (Allen et al., 1994); the MRDA annotation scheme (Dhillon et al., 2004); the DIT taxonomy (Bunt, 1994); the HCRC Map Task scheme (Carletta et al., 1996); DAMSL (Allen & Core, 1997), Switchboard-DAMSL (Jurafsky et al., 1997); COCONUT (Di Eugenio et al, 1998), the Verbmobil scheme (Alexandersson et al., 1998), the MALTUS tag set (Popescu-Belis, 2004) and the AMI (2005) annotation scheme (`http://corpus.amiproject.org`). Each of these schemes has been used to build annotated dialogue corpora.

Tilburg Center for Cognition and Communication (TiCC) and Department of Philosophy, Tilburg University, The Netherlands, e-mail: `harry.bunt@uvt.nl`

In order to support the creation of *interoperable* annotated corpora, the International Organisation for Standards ISO has recently developed a standard for dialogue act annotation (ISO 24617-2, 2012; see also Bunt et al., 2010; 2012), which is largely based on the DIT^{++} taxonomy, a comprehensive domain-independent schema which was constructed by adding to the DIT taxonomy a number of concepts from DAMSL and other schemes and dialogue studies (see Bunt, 2009 and `http://dit.uvt.nl`).

DIT^{++} is based on the dynamic approach to utterance meaning of Dynamic Interpretation Theory (DIT), which views dialogue acts semantically as update operations on the information states of the dialogue participants; an approach that is also known as the 'information-state update' or 'context-change approach' to utterance meaning – see e.g. Bunt (2000); Traum & Larsson (2003). On this approach, the two most important components of a dialogue act are its semantic content, which describes the objects, properties, relations, or actions that the dialogue act is about, and its communicative function, which specifies how an addressee should update his information state with the semantic content.

Utterances are often multifunctional, i.e., they have more than one communicative function. Dialogue analysis and annotation frameworks are therefore often 'multidimensional' in the sense of allowing the assignment of multiple dialogue act tags to utterances; this is e.g. the case for DAMSL, COCONUT, and MRDA. The multifunctionality of utterances is due not just to the fact that an utterance may contain parts that have different functions, but also to the phenomenon that they may contain segments that have more than one communicative function (see Bunt, 2011). In order to accurately describe the relation between dialogue acts and stretches of speech, text, or other forms of communicative behaviour, the notion of a *functional segment* has been introduced in the DIT^{++} annotation framework, defined as a minimal stretch of communicative behaviour that has at least one communicative function (Geertzen et al., 2007). Functional segments may be discontinuous, may overlap, may spread over multiple turns, and may contain parts contributed by different speakers. The following dialogue fragment illustrates some of these phenomena.

(1)
1. A: could you tell me what departure times there are for flights to Frankfurt on Saturday morning?
2. B: sure, there's a Lufthansa flight at ... let me see... 7.45,...
3. A: yes,
4. B: and a KLM flight at 08.15,...
5. A: yes,
6. B and then there's a flight by Philiipine airlines,...

The response to A's request includes an enumeration of items, which B communicates one by one in separate turns (of which the example shows the first two items and part of the third). B's first utterance consists of several functional segments, of which the first (*"sure"*) has the functions of taking the turn and accepting A's request; the discontinuous second segment (*"there's a Lufthansa flight at [...] 7.45"*) provides a part of the information that A requested while at the same time indicating that there's more to come; and the third segment that is embedded in the second (*"let*

me see") has a time management function (Stalling). In fact, the dialogue act providing the information that A requested corresponds to a discontinuous, multi-turn functional segment formed by *"there's a Lufthansa flight at [...] 7.45"*, utterance 4, utterance 6, and subsequent utterances containing further parts of B's answer. A's utterance 3 *"yes,"* has (by virtue of its intonation) both the function of indicating positive feedback (viz. that A has understood what B said and accepted the information provided) and of giving the turn back to B, encouraging him to go on.

The multidimensional annotation schemes mentioned above use an implicitly defined notion of dimension as a set of mutually exclusive tags. By contrast, Bunt (2006) based the design of the DIT^{++} scheme on a notion of dimension which reflects the observation that participation in a dialogue involves, beyond activities strictly related to performing a certain task, also other types of communicative activity such as a sharing information about understanding and accepting each other's utterances; monitoring contact and attention; managing the use of time; taking turns; and correcting a speaking error made by oneself or by another speaker. A dimension in dialogue act analysis is defined as corresponding to such a type of communicative activity. Dialogue acts belonging to different dimensions are thus concerned with different types of semantic content: feedback acts with the success of processing previous utterances; turn management acts with the allocation of the speaker role, task-related acts with the dialogue task; and so on. Dimensions thus classify the semantic contents of dialogue acts.

Petukhova & Bunt (2009a; b) formulate criteria for distinguishing dimensions, and apply these in the analysis of the structure of 18 existing annotation schemes. They show that the DIT^{++} taxonomy has a well-founded set of ten dimensions (nine of which have been retained in ISO standard 24617-2); namely:

(2)
1. Task/Activity: dialogue acts for performing the task or activity underlying the dialogue;
2. Auto-Feedback: dialogue acts providing information about the speaker's processing of previous utterances;
3. Allo-Feedback: dialogue acts expressing opinions or eliciting information about the addressee's processing of previous utterances;
4. Contact Management: dialogue acts for establishing and maintaining contact;
5. Turn Management: dialogue acts concerned with grabbing, keeping, giving, or accepting the speaker role;
6. Time Management: dialogue acts indicating that the speaker needs some time to formulate his contribution;
7. Discourse Structuring: dialogue acts for explicitly structuring the conversation;
8. Own Communication Management: dialogue acts for editing the speaker's current utterance;
9. Partner Communication Management: dialogue acts to assists or correct the current speaker;
10. Social Obligations Management: dialogue acts that take care of social conventions such as greetings, apologies, and expressions of gratitude.

Some communicative functions are specific for a particular dimension; for instance *Turn Accept* and *Turn Release* are specific for turn management; *Stalling* and *Pausing* for time management. Other functions can be applied in any dimension; for instance a *Check Question* can be used with task-related semantic content, but also for checking correct understanding (feedback). More generally, all types of question and inform can be used in any dimension, and the same is true of directive acts such as *Suggest*, *Request, Instruct*, and *Accept Offer*, and of commissive acts such as *Offer, Promise,* and *Accept Request*. These functions are called *general-purpose* functions – see Figure 1 in the Appendix, and `http://dit.uvt.nl` for the taxonomy of general-purpose communicative functions of ISO 24617-2 and DIT^{++}.

The *dimension-specific* communicative functions, which can only be used with a particular type of semantic content to form a dialogue act in that particular dimension, also form a set with a hierarchical organization. The DIT^{++} and ISO 24617-2 taxonomies thus consist of two parts: a taxonomy of *general-purpose functions* and one of *dimension-specific functions*. Figure 2 in the appendix shows the taxonomies of dimension-specific communicative functions in ISO 24617-2 and DIT^{++}. There are some differences between the two, since ISO 24617-2 does not have the Contact Management dimension, lacks communicative functions for different levels of processing for feedback, and lacks a few other fine-grained distinctions that are made in DIT^{++}.

This chapter describes a computational semantics for dialogue acts formed with a communicative function of the DIT^{++} taxonomy. This description takes the form of the definition of the semantics of the annotation language DiAML (Dialogue Act Markup Language), which forms part of the ISO 24617-2 standard. Expressions in DiAML describe dialogue act information, associated with a functional segment. This information consists for each dialogue act of its communicative function; the type of semantic content; the speaker and the addressee(s); semantic relations of various kinds to other dialogue acts or functional segments; communicative function qualifiers (if any); and the functional segment by which the dialogue act is expressed (verbally, nonverbally, or with a combination of modalities). Section 2 describes the DiAML language, with the way its semantics is organized, using operations that update the dialogue participants' information states. Section 3 discusses the notion of information state, or 'dialogue context'. Section 4 describes in some detail the semantics of the DIT^{++} communicative functions. Section 5 draws general conclusions and indicates perspectives for future work.

2 DiAML: Dialogue Act Markup Language

The Dialogue Act Markup Language (DiAML) has been designed in accordance with the ISO Linguistic Annotation Framework[1], which makes a distinction between *annotation* and *representation*. The term 'annotation' refers to the linguistic

[1] ISO 24612:2012; see also Ide & Romary (2004).

information that is added to segments of language data, independent of format; 'representation' refers to the format in which an annotation is rendered, independent of content. Annotation standards are required to be defined not at the level of a representation formats, but at the more abstract level of annotations.

This distinction has been implemented in the DiAML definition by applying a multilevel design methodology, called CASCADES (Bunt 2010; 2013a; 2013b), which defines an annotation language by means of a syntactic component that specifies, besides a class of XML-based *representation structures*, also a class of set-theoretical structures called *annotation structures*. These two parts of the definition are called the *concrete* and the *abstract syntax* of the language, respectively.

2.1 Abstract syntax

An abstract syntax consists of: (a) a specification of the elements from which annotation structures are built up, called a 'conceptual inventory', and (b) a specification of the possible ways of constructing annotation structures using these elements.

a. Conceptual inventory
The conceptual inventory of DiAML consists of six finite sets:

1. a set of dimensions (ten in the case of DIT^{++}; nine in ISO 24617-2);
2. a set of communicative functions;
3. a set of qualifiers, that can be associated with communicative functions; this set is partitioned into subsets for different aspects of qualification, such as certainty, conditionality, and sentiment;
4. a set of rhetorical relations, that can hold between dialogue acts or their content;
5. a set of dialogue participants;
6. a set of functional segments of primary data.

The set of functional segments is specific for a particular annotation task; since annotation means associating linguistic information with segments of primary data, an annotation language must have elements for identifying relevant segments, which in the case of dialogue act annotation correspond to functional segments. The set of dialogue participants is also specific for a particular annotation task, and is assumed to be specified in the metadata of the dialogue under consideration. The four other sets of concepts in the conceptual inventory are independent of any particular annotation task.

b. Annotation structures
An annotation structure is a set of two kinds of elements, called *entity structures* and *link structures*. An entity structure contains semantic information about a functional segment; a link structure describes a semantic relation between segments. Formally, an annotation structure is a set $\{\epsilon_1, \ldots, \epsilon_k, L_1, \ldots, L_m\}$ of one or more entity structures ϵ_i and zero or more link structures L_j.

An entity structure in DiAML is a nested pair

(3) $\epsilon = \langle s, \langle \alpha, \Delta \rangle \rangle$

consisting of a functional segment s, a 'dialogue act structure' α, which characterises a single dialogue act without the relations that it might have to other dialogue units, and a 'dependence structure' Δ, which describes the semantic dependence relations between the dialogue act α and other dialogue units.

A 'dialogue act structure' is a sixtuple

(4) $\alpha = \langle S, A, H, d, f, q \rangle$

where S is the sender of the dialogue act; A is a non-empty set of addressees; H is a (possibly empty) set of other dialogue participants (such as overhearers or side-participants; see Clark, 1996); d is a dimension; f is a communicative function; and q is a (possibly empty) set of qualifiers. In order to avoid details which are irrelevant to the purpose of this chapter, we will only consider cases where the set H of participants who are neither speakers nor addressees is empty, and where there is only a single addressee - we will use A to indicate this addressee, rather than the set consisting of this lone addressee.

A 'dependence structure' is a pair consisting of a (possibly empty) set of entity structures E, whose members α has a dependence relation with, and the element δ which specifies the nature of a dependence relation (functional or feedback - see below):

(5) $\Delta = \langle E, \delta \rangle$

The other kind of component of an annotation structure besides entity structures, a link structure, is a triple consisting of an entity structure ϵ, a non-empty set E of entity structures, and a rhetorical relation ρ, which relates the dialogue act α in ϵ to the entity structures in E.

(6) $L = \langle \epsilon, E, \rho \rangle$

The 'dependence structures' that an entity structure may contain[2] and that make entity structures potentially recursive, are semantic relations between a dialogue act and one or more other units in dialogue that must be taken into account in order to determine its semantic content. Two such relations are distinguished in DiAML, called 'functional dependence' and 'feedback dependence'.

A functional dependence relation occurs when a dialogue act is semantically dependent on one or more dialogue acts that occurred earlier in the dialogue, due to having a communicative function which is responsive in nature. This is for example the case for answers, whose meaning is partly determined by the question which is being answered, as is immediately obvious for an answer like *"No"*, whose meaning

[2] If the set E in a dependence structure $\Delta = \langle E, \delta \rangle$ is empty, then this amounts to there being no dependences. We will designate a dependence structure $\Delta = \langle \emptyset, \delta \rangle$ by $\emptyset$.

depends almost entirely on the question that is answered. Similarly for the acceptance or rejection of offers, suggestions, requests (where *"Yes"* may illustrate the point), and acceptance of apologies and thankings.

Feedback-providing and eliciting acts provide or elicit information about the processing of something that was said earlier in the dialogue, such as its perception or its interpretation, and their meaning often depends on that (or those) earlier contribution(s) to the dialogue. Positive feedback utterances like *"OK"* and *"Yes"*, and negative ones like *"What?"* and *"Excuse me?"* illustrate this phenomenon.

Responsive dialogue acts and feedback acts are semantically incomplete without the specification of functional and feedback dependence relations, which are therefore part of the entity structures that are used to annotate such acts.

A dialogue act may, finally, also be related to other dialogue acts through rhetorical relations, as in (7).

(7)
1. A: it ties you on in terms of the technology and the complexity that you want
2. A: like for example voice recognition
3. A: because you might need to power a microphone and other things

In this example, from the AMI corpus,[3] we see three functional segments, where the second segment is related to the first through an *Exemplification* relation, and the third through an *Explanation* relation.

Different from functional and feedback dependence relations, rhetorical relations are not part of the meaning of a dialogue act, but add semantic information to the way a self-contained dialogue act is related to other dialogue acts (or how their semantic contents are related – see Petukhova et al., 2011). They therefore turn up in a different way in annotation structures, namely in link structures.

2.2 Concrete Syntax

The concrete syntax defines a rendering of annotation structures in a particular format, such as XML. It is defined in accordance with the methodology for defining semantic annotation languages described in Bunt (2010; 2013a), which introduces the notion of an *ideal representation format*, defined as one where (1) every annotation structure defined by the abstract syntax can be represented, and (2) every representation represents one and only one annotation structure. The semantics of the language is defined for the structures defined by the *abstract* syntax. This has the effect that any two 'ideal' representation formats are semantically equivalent; every representation in one such format can be converted by a meaning-preserving mapping into any other such format.[4] The representation format defined by the con-

[3] `http://corpus.amiproject.org`

[4] See Bunt (2010) for formal definitions and proofs relating to alternative representation formats sharing the same abstract syntax, and Bunt (2013a) for a procedure to derive a concrete syntax from an abstract syntax.

crete syntax of DiAML is illustrated in (8). P2's utterance is segmented into two overlapping functional segments: one (fs2.1) in the Auto-Feedback dimension and one (fs2.2) in the Task dimension (TA), with value 'answer' qualified as 'uncertain'. (Values with a "#" prefix are defined outside the XML element in which they occur; either in the metadata or in another layer of annotation.)

(8) a. Segmented dialogue fragment:

1. P1:	*What time does the next train to Utrecht leave?*
TA:	fs1: What time does the next train to Utrecht leave?
2. P2:	*The next train to Utrecht leaves I think at 8:32.*
AuFB	fs2.1: The next train to Utrecht leaves
TA	fs2.2: The next train to Utrecht leaves I think at 8:32.

b. DiAML annotation structure:
$AS = \langle\{\epsilon_1, \epsilon_2, \epsilon_3\}, \emptyset\rangle$, where
- $\epsilon_1 = \langle$fs1, $\langle\alpha_1, \emptyset\rangle\rangle$;
- $\epsilon_2 = \langle$fs2.1, $\langle\alpha_2, \langle\{$fs1$\}$,feedback$\rangle\rangle$;
- $\epsilon_3 = \langle$fs2.2, $\langle\alpha_3, \langle\{\alpha_1\}$,functional$\rangle\rangle$

c. DiAML representation:

```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1"
  sender="#p1" addressee="#p2" dimension="task"
  communicativeFunction="setQuestion"/>
<dialogueAct xml:id="da2" target="#fs2.1"
  sender="#p2" addressee="#p1"
  communicativeFunction="inform"
  dimension="autoFeedback"
  feedbackDependence="#fs1"/>
<dialogueAct xml:id="da3" target="#fs2.2"
  sender="#p2" addressee="#p1" dimension="task"
  communicativeFunction="answer"
  certainty="uncertain"
  functionalDependence="#da1"/>
</diaml>
```

2.3 DiAML Semantics

A dialogue act structure captures the *functional* part of a dialogue act; it does not include the full semantic content but only a dimension which classifies the content. The semantics of a dialogue act structure is therefore defined as a function that can be applied to a given semantic content to form the interpretation of a full-blown dialogue act. For a dialogue act without functional or feedback dependences this is expressed by (9), which defines the interpretation $I_a(\langle s, \alpha, \emptyset\rangle)$ of the entity structure that associates the dialogue act structure α with the functional segment s. This interpretation is a function applied to the semantic content $\kappa_1(s)$ of that segment.

(9) $I_a(\epsilon) = I_a(\langle s, \langle \alpha, \emptyset \rangle \rangle) = I_a(\alpha)(\kappa_1(s))$

The interpretation $I_a(\epsilon)$ of a dialogue act structure without qualifiers is defined as the interpretation of its communicative function, applied to the interpretations of the other components of the dialogue act structure, where the function F assigns values to the constants of DiAML:

(10) $I_a(\langle S, A, d, f \rangle) = I_a(f)(F(S), F(A), F(d))$

To the sender and an addressee of a dialogue act (S and A) the function F assigns certain individuals, identified in the metadata of the dialogue; to the dimension argument d, a component is assigned of an Information State (IS) to be updated. The interpretation of a dialogue act with communicative function qualifiers is discussed in Section 4.2; if the communicative function f has no qualifiers, then $I_a(f) = F(f)$; see Section 4.1 for the definition of $F(f)$.

A link structure $L = \langle \epsilon, E, \rho \rangle$ is interpreted semantically as a set of updates that create rhetorical links between the representations of the dialogue acts in ϵ and E in the participants' ISs. This assumes that the dialogue acts that occur in a dialogue are represented as such in an IS, an assumption that is commonly made in proposals for dialogue context modelling (see Section 3). More specifically, the assumption is that an IS has a part (the 'Dialogue History'), where a record is kept of the communicative events in the dialogue, typically in the form of a transcription of what was said, with an interpretation in terms of dialogue acts. The updates corresponding to link structures then come down to the addition of rhetorical links between these representations.

The semantics of an annotation structure $\{e_1, .., e_n, L_1, .., L_k\}$, consisting of the entity structures $\{e_1, .., e_n\}$ and the link structures $\{L_1, .., L_k\}$, is defined as the sequential application of the update functions corresponding to the constituent entity and link structures, following the textual order $<_T$ of their functional segments, where the update operations corresponding to textually coinciding ('$=_T$') entity structures are unified rather than sequenced. This is expressed in (11), where the notation 'α ; /$\sqcup$ β' is used to indicate that the operation α should be followed (';') by the operation β if $\alpha <_T \beta$, and should be unified ($\sqcup$) if $\alpha =_T \beta$.

(11) $I_a(\{e_1, .., e_n, L_1, .., L_k\}) =$
$I_a(e_1)$;/$\sqcup$ ...;/$\sqcup$ $I_a(e_n)$;/$\sqcup$ $I_a(L_1)$;/$\sqcup$ ... ;/$\sqcup$ $I_a(L_k)$

The semantics of an entity structure with dependence relations is defined as follows, where s_ϵ is the functional segment of entity structure ϵ; f_α is the communicative function of α; κ_{2a} computes the semantic content of a dependent dialogue act from its local content $\kappa_1(s_{\epsilon_1})$ and the contents of the dialogue acts that α, depends on (given the communicative function f_α and the nature of the dependence relation δ).

(12) $I_a(\langle s, \alpha, \langle E, \delta \rangle \rangle) = I_a(\alpha)(\kappa_{2a}(\kappa_1(s), \{\kappa_1(s_\epsilon) | \epsilon \in E\}, f_\alpha, \delta))$

3 Context Model Structure and Content

3.1 Types of Context Information

As the proposed semantics of dialogue acts is in terms of IS updates, the question arises as to what exactly is an information state in this context; what information does it contain, and how is it structured. The dialogue act semantics described in this chapter does not make any assumptions about a particular formalism that is used to represent information states; proposals in the literature include Discourse Representation Structures (Poesio & Traum, 1997); Constructive Type Theory (Ahn, 2001); Modular Partial Models (Bunt, 2000); Record Types (Cooper, 2004) and typed feature structures (Keizer et al., 2011; Petukhova et al., 2010). An IS is assumed to have a number of components that contain different kinds of information, such as a dialogue history and a representation of the state of the underlying task or activity.

The details of an IS update semantics depend on whether only the information state of an *addressee* is considered to be updated by dialogue acts, or also that of the *sender*, and on whether these updates involve nested or mutual beliefs (as e.g. argued in Bunt, 1989). In this chapter we consider only the updates of a single addressee's information state; approaches involving multiple ISs and mutual beliefs are readily extrapolated from this. In DIT, it is customary to speak of 'contexts' or 'context models', rather than 'information states', and this terminology will also be used in the rest of this chapter.

A requirement for an adequate notion of context model is that, for a given range of dialogue act types, it contains the kinds of information that can be updated by a dialogue act. For the dialogue acts of the DIT^{++} taxonomy, we require the context models to include the following kinds of information: properties of the dialogue task (and task domain); success/problems in processing previous utterances; allocation of the speaker role; allocation of time; presence and contact; structuring of the discourse; success/problems in utterance production; social obligations and interactive pressures. It can be argued (see Bunt, 2000) that an agent's context model does not need to have a separate component for each dimension of the taxonomy, but that it is convenient to distinguish the following five components:

(13)
1. Linguistic Context, which contains a record of the dialogue history, information about discourse plans (if any), and preferences concerning the occupation of the speaker role;
2. Task Context, which contains the agent's information and goals relating to the dialogue task, as well as his assumptions about the dialogue partner's task-related goals and beliefs;
3. Cognitive Context, which contains information about the agent's cognitive processes concerned with the processing and production of dialogue utterances, including time estimates for these processes;
4. Physical/Perceptual Context, which contains information about physical and perceptual properties of the interactive situation;

5. Social Context, which contains information relevant for interpreting and generating 'social' acts like greetings, apologies, expressions of gratitude.

Versions of such a 5-component context model have been implemented in the PARADIME dialogue manager (Keizer and Bunt, 2006; 2007; Keizer et al., 2011) and for theoretical studies by Petukhova et al. (2010).

A context-update semantics has to take into account that update operations should not undermine the consistency of the context model. A dialogue participant may for example change his mind on something in the course of a dialogue, possibly as an effect of receiving new information which contradicts something that the participant believed. Updates are therefore not simply additions of information. Rather then building consistency checks into the semantics of each dialogue act, we exploit the DIT distinction of several levels of utterance processing: (1) attending, (2) perceiving, (3) understanding, (4) evaluating, and (5) executing. The level of *understanding* determines the meaning of a dialogue segment in terms of dialogue acts. The *evaluation* level checks whether the corresponding updates would keep the current context model consistent. If so, the updates are performed. One way to implement this approach is to add to a context model a part called the *pending context*, which serves as a buffer for items to be inserted in the main context once their consistency with the current content of the main context has been established.[5] Updating the pending context is then simply a matter of adding items to it. For convenience we will assume the pending context $A*$ of an agent A's context model to be structured in the same way as the main context; a piece of information which is found to be consistent with the main context can then simply be moved from its pending context component to the corresponding component of the main context. The notation (14) will be used to designate the operation of adding the information z to component $A*_i$ of A's pending context:

(14) $A*_i =\!\!+ z$

3.2 Semantic Primitives

The definitions of the communicative functions in the DIT^{++} and ISO 24617-2 taxonomies make use of a number of formal concepts needed to describe update effects. This involves such concepts as an agent believing something, an agent wanting to know something, and an agent being committed to do something. Table 1 lists the basic concepts that are required for formulating the update semantics of dialogue acts with a general-purpose function, with the terms used to designate them in the rest of this chapter.

For convenience, we introduce the following abbreviations: **Bel**(S, p) abbreviates **Bel**$(S, p,$ firm); **Wk-Bel**(S, p) abbreviates **Bel**$(S, p,$ weak); **Assumes**$(S,\ p)$ abbreviates **Bel**$(S, p)\vee$**Wk-Bel**(S, p). In all action-related attitude operators we suppress

[5] This approach has been implemented in the multimodal DenK dialogue system; see Kievit et al. (2001).

description	notation	meaning
believes that	**Bel**(S, p, σ)	*S believes that p*; σ indicates the strength of this belief (σ = 'firm' or σ = 'weak')
knows value of	**Know-val**(S, z)	*S possesses the information z*
has goal	**Want**(S, p)	*S has the goal that p*
is able to do	**CanDo**(S, α)	*S is able to perform the action* α
is willing to do	**WilDo**(S, α, C_α)	*S is willing to perform the action* α *if* the condition C_α is satisfied.
is committed to do	**CommitDo**(S, α, C_α)	*S is committed to perform the action* α *if* the condition C_α is satisfied.
is committed to refrain from doing	**RefrainDo**(S, α, C_α)	*S is committed to refrain from performing the action* α *if the condition* C_α *is satisfied.*
is considering	**ConsidDo** (X, α, Y, C_α)	*X is considering the performance of action* α *by agent Y, if condition* C_α *is satisfied.*
is in the interest of	**Interest**(Y, α)	action α is in the interest of agent Y.

Table 1 Semantic primitives for the interpretation of general-purpose communicative functions. (C_α may be the universally true condition $\top$.)

the argument $\top$ representing the 'empty' condition, hence **WilDo**(S, α) abbreviates **WilDo**$(S, \alpha, \top)$, and so on. These semantic primitives are similar to those proposed by Poesio and Traum in their axiomatization of dialogue acts (Poesio and Traum, 1997), which is however limited to a small set of general-purpose functions and positive auto-feedback functions, and does not consider the other dimensions, nor communicative function qualifiers.

Since dimension-specific communicative functions are concerned with a specific kind of semantic content, certain specific semantic primitives are required for representing their semantics; these are listed in Table 2.

Dimension	*Primitives*
Auto- and Allo-Feedback	**Attended, Perceived, Understood, Accepted, Executed, Success-Processing**
Turn Management	**Current-Speaker, Next-Speaker**
Time Mangement	**Time-Need, small, substantial**
Contact Management	**Present**
Discourse Structuring	**Ready, Available, Start-Dialogue, Close-Dialogue**
Own and Partner Communication Man.	**Delete, Replace, Append**
Social Obligations Management	**Available, Thankful, Regretful, Knows-id, Final**

Table 2 Dimension-specific semantic primitives

For expressing the semantics of a feedback act, we must distinguish between feedback functions which indicate a certain *level of processing*, and those that do not. The taxonomy of dimension-specific communicative functions for feedback in DIT^{++} is based on the distinction of five levels of processing that a feedback act may address: attending, perceiving, understanding, evaluating and executing. At

each of these levels, positive auto-feedback reports that the sender believes his processing of one or more previous utterances to be sufficiently successful to go on, not requiring a repetition or clarification; negative feedback reports that the sender does not believe that. Similarly, positive allo-feedback reports that the sender believes that the addressee did process one or more utterances successfully, and negative allo-feedback that this is not the case. The semantics of feedback acts which are specific about the level of processing that they refer to, requires the semantic primitives mentioned in Table 2.

Since feedback acts are often not specific for a particular level of processing, DIT^{++} also has level-unspecific feedback functions: one for level-unspecific positive auto-feedback, one for negative auto-feedback, one for positive allo-feedback, one for negative allo-feedback, and one for feedback elicitation. The ISO 24617-2 standard has only these level-unspecific functions. A study reported in Bunt (2012) shows that the dialogue participants interpret level-unspecific feedback acts in different ways depending on the interactive setting, and therefore introduces a semantic primitive **Success-Processing** whose interpretation is context-dependent, one common interpretation being "Well understood and possibly also accepted and executed successfully" - see Bunt (2012) for details. This primitive has therefore been added to the level-specific primitives in Table 2.

4 Dialogue Act Interpretation

The definition of the semantics of the communicative functions in the DIT^{++} and ISO 24617-2 taxonomies is organized in a way that exploits the hierarchical structure of these taxonomies, which reflects the phenomenon that some communicative functions are specializations of others. For example, a confirmation is a special kind of answer, and an answer is a special kind of inform (namely an inform in response to a question); this is reflected in the taxonomy by the communicative function Inform dominating the Answer function, which in turn dominates the Confirm function.

An update semantics of dialogue acts with an Inform, an Answer, or a Confirm function should bring this out by having in common that in all three cases (1) the speaker wants to make certain information available to the addressee, and (2) the speaker assumes that this information is correct. These are (minimally) the updates of an Inform act. An Answer act has additional update effects, reflecting that (3) the speaker believes that the addressee wanted to obtain this information; and (4) the addressee assumed that the speaker possessed the requested information. A Confirm act has a further additional update effect, reflecting that (5) the speaker believes that the addressee had an uncertain belief that this information was correct. The DiAML semantics described below therefore makes use of so-called *elementary update functions*, which update an information state with a single information item, such as (1) or (2) in this example.

The update semantics of the Inform function, as specified in Table 3,is defined as the combination of the elementary update functions U_1 and U_2 (defined in Table 4), which perform the updates illustrated by (1) and (2). The update semantics of the Answer function shares the use of U_1 and U_2 with that of the Inform function, and adds to that the effects of the elementary update functions U_7 and U_9 (defined in Table4); the semantics of the Confirm function further adds to that the update defined by U_8.

4.1 The Semantics of Communicative Functions

4.1.1 General-Purpose Communicative Functions

The class of general-purpose communicative functions in ISO 24617-2 and DIT^{++} falls apart into *information-transfer functions* and *action-discussion functions*, further subdivided into information-providing and information-seeking functions, and commissives and directives, respectively (see appendix). We first consider the class of information-transfer functions.

a. Information-Providing and Information-Seeking Functions
The hierarchy of information-providing functions has the function *Inform* as the mother of all information-providing functions; all other functions are specializations of this function, and therefore have in common that the speaker wants the addressee to possess certain information which the speaker assumes to be correct.

Using the epistemic operators introduced in Section 3.2, these conditions can be formalized as sown in (15), where (15.b) says that the speaker S believes that the content p is true, with certainty σ and (15a) says that S wants the addressee A to also have that belief.

(15) a. **Want**$(S,$ **Bel**$(A, p, \sigma))$
b. **Bel**(A, p, σ)

When addressee A understands an utterance by S as an *Inform* with the content denoted by p, then the update effects on the pending context part of A's IS will be that A believes that the two conditions in (15) hold.

If a speaker is uncertain about the content of an *Answer* or an *Inform* (σ = *weak*), then his goal cannot be that the addressee believes for sure that the content is true; if, on the other hand, the speaker is certain, then it would be strange if he would want the addressee to be uncertain. The argument σ should therefore have the same value in both conditions in (15). The semantics of the Inform function, specified in Table 3, has this effect. (See also below, section 4.2, on certainty qualifiers.)

As an illustration of the update semantics of information-providing functions, consider the case of the answer in (16.2).

(16) 1. D: twenty-five euros, how much is that in pounds?
2. C: twenty-five euros is something like 20 pounds

F(Inform)	$= \lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,p,s) \sqcup U_2(X,Y,D_i,p,s)$
F(Agreement)	$= \lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,p,s) \sqcup U_2(X,Y,D_i,p,s) \sqcup U_5(X,Y,D_i,p)$
F(Disagreement)	$= \lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,\neg p,s) \sqcup U_2(X,Y,D_i,\neg p,s) \sqcup U_5(X,Y,D_i,p)$
F(Correction)	$= \lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,p_1,s) \sqcup U_2(X,Y,D_i,\neg p_1,s) \sqcup U_6(X,Y,D_i,p_2)$
F(Answer)	$= \lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,p,s) \sqcup U_2(X,Y,D_i,p,s) \sqcup U_9(X,Y,D_i,p) \sqcup U_7(X,Y,D_i,p)$
F(Confirm)	$= \lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,p,s) \sqcup U_2(X,Y,D_i,p,s) \sqcup U_8(X,Y,D_i,p) \sqcup U_9(X,Y,D_i,p,s) \sqcup U_7(X,Y,D_i,p)$
F(Disconfirm)	$= \lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,\neg p,s) \sqcup U_2(X,Y,D_i,\neg p,s) \sqcup U_8(X,Y,D_i,p,s) \sqcup U_9(X,Y,D_i,p) \sqcup U_7(X,Y,D_i,p)$
F(Question)	$= \lambda X.\lambda Y.\lambda D_i.\lambda z.U_{10}(X,Y,D_i,z) \sqcup U_{11}(X,Y,D_i,z)$
F(Prop.Question)	$= \lambda X.\lambda Y.\lambda D_i.\lambda p.U_{10}(X,Y,D_i,p) \sqcup U_{11}(X,Y,D_i,p) \sqcup U_{12}(X,Y,D_i,p)$
F(CheckQuestion)	$= \lambda X.\lambda Y.\lambda D_i.\lambda p.U_{10}(X,Y,D_i,p) \sqcup U_{11}(X,Y,D_i,p) \sqcup U_4(X,Y,D_i,p)$
F(SetQuestion)	$= \lambda X.\lambda Y.\lambda D_i.\lambda z.U_{10}(X,Y,D_i,z) \sqcup U_{11}(X,Y,D_i,z) \sqcup U_{13}(X,Y,D_i,z)$
F(ChoiceQuestion)	$= \lambda X.\lambda Y.\lambda D_i.\lambda p.U_{15a}(X,Y,D_i,p) \sqcup U_{15}(X,Y,D_i,p) \sqcup U_{16}(X,Y,D_i,p)$

Table 3 Update semantics for information-providing and information-seeking communicative functions

Applying the semantics of the Answer function (see Table 3) to the participants C and D and the semantic content of (16.2), we obtain:

F(Answer)(C, D, Task, EU 25 = BP 20) =
U_1(C,D,TaskC, EU 25 = BP 20) $\sqcup$
U_2(C,D,Task, EU 25 = BP 20) $\sqcup$
U_9(C, D, Task, EU 25 = BP 20) $\sqcup$
(17) U_7(C, D, Task, EU 25 = BP 20) =
$D{*_{TaskC}} =+$ **Bel**(D, **Want**(C, **Bel**(D, EU25=BP20)));
$D{*_{TaskC}} =+$ **Bel**(D, **Bel**(C, EU25=BP20));
$D{*_{TaskC}} =+$ **Bel**(D, **Bel**(C, **Want**(D, **Know-val**(D, EU25=BP20))));
$D{*_{TaskC}} =+$ **Bel**(D, **Bel**(C, **Assume**(D, **Know-val**(C, EU25=BP20))))

Hence the following beliefs are added to D's pending Task Context:

(18) (1) C wants D to know that EU 25 = BP 20;
(2) C believes that EU 25 = BP 20;
(3) C believes that D wants to know whether EU 25= BP 20;
(4) C believes that D assumes C to know whether EU 25 = BP 20.

b. Commissive and Directive Functions

Table 5 specifies the semantics of a representative selection of the commissive and directive communicative functions; Table 6 defines the elementary update functions used in the semantics of these functions.

$U_1(X,Y,D_i,p,s)$	$Y*_i =+$ **Bel**(Y, **Want**(X,**Bel**(Y,p,s)))
$U_2(X,Y,D_i,p,s)$	$Y*_i =+$ **Bel**(Y, **Bel**(X,p,s))
$U_3(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Assume**(X,p))
$U_4(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Wk-Bel**(X,p))
$U_5(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Bel**(X, **Assume**(Y,p)))
$U_6(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Assume**(X, **Assume**(Y,p)))
$U_7(X,Y,D_i,P)$	$Y*_i =+$ **Bel**(Y, **Bel**(X, **Assume**(Y, **Know-val**(X,P))))
$U_8(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Assume**(X, **Wk-Bel**(Y,p))
$U_9(X,Y,D_i,P)$	$Y*_i =+$ **Bel**(Y, **Bel**(X, **Want**(Y, **Know-val**(Y,P))))
$U_{10}(X,Y,D_i,P)$	$Y*_i =+$ **Bel**(Y, **Want**(X, **Know-val**(X,P)))
$U_{11}(X,Y,D_i,P)$	$Y*_i =+$ **Bel**(Y, **Assume**(X, **Know-val**(Y,P))
$U_{12}(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Bel**(X, $p \vee \neg p$))
$U_{13}(X,Y,D_i,P)$	$Y*_i =+$ **Bel**(Y, **Assume**(X, $\exists x.P(x)$))
$U_{14}(X,Y,D_i,P)$	$Y*_i =+$ **Bel**(Y, **Want**(X, **Know-val**(X,P))
$U_{15}(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Assume**(X, p_1 *xor* p_2))
$U_{15a}(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Want**(X, **Bel**(X,p_1) $\vee$ **Bel**(X,p_2))))
$U_{16}(X,Y,D_i,p)$	$Y*_i =+$ **Bel**(Y, **Assume**(X, **Bel**(Y,p_1) $\vee$ **Bel**(Y,p_2))))

Table 4 Elementary update functions used in the semantics of information-transfer functions

F(Offer)	$= \lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{25a}(X,Y,D_i,\alpha) \sqcup U_{20}(X,Y,D_i,\alpha,C_\alpha)$
F(AddressRequest)	$=\lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{17a}(X,Y,D_i,\alpha,C_\alpha) \sqcup U_{18}(X,Y,D_i,\alpha) \sqcup U_{26b}(X,Y,D_i,\alpha)$
F(AcceptRequest)	$=\lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{17}(X,Y,D_i,\alpha,C_\alpha) \sqcup U_{18}(X,Y,D_i,\alpha) \sqcup U_{26b}(X,Y,D_i,\alpha)$
F(DeclineRequest)	$=\lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{27}(X,Y,D_i,\alpha,C_\alpha) \sqcup U_{18}(X,Y,D_i,\alpha) \sqcup U_{26b}(X,Y,D_i,\alpha)$
F(Request)	$= \lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{23}(X,Y,D_i,\alpha,C_\alpha) \sqcup U_{26}(X,Y,D_i,\alpha)$
F(Instruct)	$= \lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{24}(X,Y,D_i,\alpha,C_\alpha) \sqcup U_{26}(X,Y,D_i,\alpha) \sqcup U_{25}(X,Y,D_i,\alpha)$
F(AddressOffer)	$= \lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{17b}(X,Y,D_i,\alpha,C_\alpha) \sqcup U_{25}(X,Y,D_i,\alpha) \sqcup U_{25b}(X,Y,D_i,\alpha)$
F(AcceptOffer)	$= \lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.U_{24}(X,Y,D_i,\alpha) \sqcup U_{25}(X,Y,D_i,\alpha) \sqcup U_{25b}(X,Y,D_i,\alpha)$

Table 5 Update semantics for commissive and directive functions (selection)

As an example of the interpretation of a directive dialogue act, consider the request in (19.2):

(19) 1. B: (...)
2. A: Please repeat that

Applied to the participants A and B and the semantic content Repeat(u1), which situates the Request act in the Auto-Feedback dimension, the definition of the Request semantics in Table 5 leads to the update (20) (where 'CC' stands for Cognitive Context):

$U_{17}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **CommitDo**$(X,\alpha,C_\alpha))$
$U_{17a}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **ConsidDo**$(X,\alpha,X,C_\alpha))$
$U_{17b}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **ConsidDo**$(X,\alpha,Y,C_\alpha))$
$U_{18}(X,Y,D_i,\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Bel**$(X,$**Want**$(Y,$**CommitDo**$(X,\alpha,C_\alpha)))$
$U_{20}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **WilDo**$(X,\alpha,C_\alpha))$
$U_{21}(X,Y,D_i,\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Bel**$(X,$ **Interest**$(\alpha,Y)))$
$U_{23}(X,Y,D_i,\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Want**$(X,$ [**WilDo**$(Y,\alpha,C_\alpha) \rightarrow$ **CommitDo**(Y,α,C_α)]$))$
$U_{24}(X,Y,D_i,\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Want**$(X,\mathbf{CommitDo}(Y,\alpha)))$
$U_{25}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Bel**$(X,$**WilDo**$(Y,\alpha,C_\alpha)))$
$U_{25a}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Want**$(X,$ **Bel**$(Y,$ **WilDo**$(X,\alpha,C_\alpha))))$
$U_{25b}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Bel**$(X,$ **Want**$(Y,$ **Bel**$(X,$ **WilDo**$(Y,\alpha,C_\alpha)))))$
$U_{26}(X,Y,D_i,\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Assume**$(X,$ **CanDo**$(Y,\alpha)))$
$U_{26b}(X,Y,D_i,\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **Bel**$(X,$ **Assume**$(Y,\mathbf{CanDo}(X,\alpha)))$
$U_{27}(X,Y,D_i,\alpha,C_\alpha)$	$Y*_i =+$ **Bel**$(Y,$ **CommitRefrain**$(X,\alpha,C_\alpha))$

Table 6 Elementary update functions used in the semantics of action-discussion functions.

$$
\begin{aligned}
&F\text{(Request)(A, B, Auto-Feedback, }\langle\text{Repeat(u1), unconditional}\rangle\text{) =}\\
&\quad \lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.\ U_{23}(X,Y,D_i,\alpha,C_\alpha) \sqcup\\
&\qquad U_{26}(X,Y,D_i,\alpha)\text{(A, B, Auto-Feedback, Repeat(u1), } \top\text{) =}\\
&\quad U_{23}\text{(A,B, CC, Repeat(u1), } \top\text{)} \sqcup U_{26}\text{(A,B, CC, Repeat(u1)) =}\\
&\quad B{*_{CC}} =+ \textbf{Bel}(B, \textbf{Want}(A, [\textbf{WilDo}(A, Repeat(u1) \rightarrow\\
&\qquad\qquad \textbf{CommitDo}(B, Repeat(u1))]));\\
&\quad B{*_{CC}} =+ \textbf{Bel}(B, \textbf{Assume}(A, \textbf{CanDo}(B, Repeat(u1))))
\end{aligned}
\tag{20}
$$

In words, B's pending cognitive context is extended with two beliefs: (1) that A wants B to commit himself to repeating the previous utterance, if he is willing to do so; (2) that A assumes B is able to repeat that utterance.

4.1.2 Dimension-Specific Communicative Functions

4.1.2.1 Feedback Functions

The communicative functions for providing and eliciting feedback in DIT^{++} fall apart in those concerned with the speaker's own processing of previous utterances (Auto-Feedback) and those concerned with the addressee's processing, as perceived by the speaker (Allo-Feedback). The elementary update functions for these two dimensions are nearly identical, differing only in whose processing is concerned. Tables 7 and 8 show the update semantics of a small, representative subset of the (altogether twenty-five) DIT^{++} communicative functions for providing and eliciting feedback.

4.1.2.2 Turn Management Functions

The communicative functions for turn management serve to decide who has or will have the speaker role. The functions for taking, accepting, grabbing, keeping, releasing, or assigning the turn are therefore all defined in terms of who currently occupies the speaker role and who wants or should have it next. Table 9 defines the

$U_{31}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Want**(X, **Bel**(Y, **Succes-Processing**(X,z)))
$U_{33}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Want**(X, **Bel**(Y, **Perceived**(X,z)))
$U_{35}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Want**(X, **Bel**(Y, **Accepted**(X,z)))
$U_{79}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Want**(X, **Bel**(Y, **Perception-Problem**(Y,z)))
$U_{76}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Want**(X, **Bel**(Y, **Execution-Problem**(Y,z)))
$U_{61}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Bel**(X, **Success-Processing**(X,z)))
$U_{62}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Bel**(X, **Perceived**(X,z)))
$U_{64}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Bel**(X, **Accepted**(X,z)))
$U_{67}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Bel**(X, **Perception-Problem**(X,z)))
$U_{85}(X,Y,z)\ Y *_{CC} =+$ **Bel**(Y, **Bel**(X, **Execution-Problem**(Y,z)))

Table 7 Elementary update functions for the semantics of auto- and allo-feedback functions (selection).

F(AutoPositive)	$= \lambda X.\lambda Y.\lambda z.U_{31}(X,Y,z) \sqcup U_{61}(X,Y,z)$
F(AlloPerceptionNegative)	$= \lambda X.\lambda Y.\lambda z.U_{33}(X,Y,z) \sqcup U_{62}(X,Y,z)$
F(AutoEvaluationPositive)	$= \lambda X.\lambda Y.\lambda z.U_{35}(X,Y,z) \sqcup U_{64}(X,Y,z)$
F(AlloExecutionNegative)	$= \lambda X.\lambda Y.\lambda z.U_{76}(X,Y,z) \sqcup U_{85}(X,Y,z)$

Table 8 Semantics of feedback functions (selection)

semantics of these functions, using the elementary update functions defined in Table 10.

For example, assigning the turn to a dialogue partner (using a Turn Assign function) means that the participant who currently occupies the speaker role wants the indicated other participant to occupy the speaker role next. This is expressed in the form of a combination of elementary update functions as shown in (21):

$$
\begin{aligned}
&F(\text{TurnAssign})(A,B) = \lambda X.\lambda Y.[U_{101}(X,Y) \sqcup \\
&\qquad\qquad\qquad\qquad U_{102}(X,Y)](A,B) = \\
&U_{101}(A,B,TurnM) \sqcup U_{102}(A,B) = \\
&B *_{LiC} =+ \textbf{Bel}(B, \textbf{Bel}(A, \textbf{Current-Speaker}(A))) \\
&B *_{LiC} =+ \textbf{Bel}(B, \textbf{Want}(A, \textbf{Next-Speaker}(B)))
\end{aligned}
\tag{21}
$$

In other words, the Linguistic Context component of B's pending context is updated to contain the beliefs that A is the current speaker and wants B to be the next speaker.

$U_{101}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Bel**(X, **Current-Speaker**(X)))
$U_{102}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Want**(X, **Next-Speaker**(Y)))
$U_{103}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Bel**(X, **Current-Speaker**(Y)))
$U_{104}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Wants**(X, **Current-Speaker**(X)))
$U_{105}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Wants**(X, **Next-Speaker**(X)))
$U_{105}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Want**($X, \neg$ **Next-Speaker**(X)))
$U_{107}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Bel**($X, \neg$ **Next-Speaker**(X) $\wedge \neg$ **Next-Speaker**(Y)))
$U_{108}(X,Y)\ Y *_{LiC} =+$ **Bel**(Y, **Bel**(X, **Want**(Y, **Next-Speaker**(X))))

Table 9 Elementary update functions for the semantics of turn management functions.

F(TurnAccept)	$= \lambda X.\lambda Y.U_{103}(X,Y) \sqcup U_{105}(X,Y) \sqcup U_{107}(X,Y)$
F(TurnAssign)	$= \lambda X.\lambda Y.U_{101}(X,Y) \sqcup U_{102}(X,Y)$
F(TurnGrab)	$= \lambda X.\lambda Y.U_{103}(X, Y \sqcup U_{104}(X,Y)$
F(TurnKeep)	$= \lambda X.\lambda Y.U_{101}(X,Y) \sqcup U_{105}(X,Y)$
F(TurnRelease)	$= \lambda X.\lambda Y.U_{101}(X,Y) \sqcup U_{106}(X,Y)$
F(TurnTake)	$= \lambda X.\lambda Y.U_{105}(X,Y) \sqcup U_{107}(X,Y)$

Table 10 Update semantics of turn management functions

4.1.2.3 Time Management Functions

Time management acts are used by a speaker to indicate that he needs some time to compose his utterance, as signalled for instance by protracting (decreasing the speech tempo) or by filled pauses; or that he needs so much time that he suspends the dialogue as in *"Just a moment"*. The semantics of such acts requires a context model that contains information about the amount of time needed by certain cognitive processes; the DIT context model therefore assumes the representation of estimates of amount of time to be represented in the Cognitive Context component, which also contains other information about the speaker's utterance processing and generation. In natural human communication such estimates are rough; an expression like *"Just a minute"* does not mean that the speaker thinks he needs one minute to process the utterance in question. The semantic primitives needed for the semantics of dimension-specific dialogue act (see Table 2) therefore include only the time estimates 'small' and 'substantial', which is adequate for interpreting the two time management functions in ISO 24617-2 and DIT^{++}: 'small' for *Stalling* and 'substantial' for *Pausing*.

Consider for example the update semantics of a *Stalling* act, which uses the elementary update scheme U_{111}, defined in Table 11:

$$
\begin{aligned}
& I_a(\langle \textit{Sys, Usr,} \text{ TimeM, Stalling}\rangle) = F(\text{Stalling})(\textit{Sys}, \textit{Usr}, \text{CC}) \\
(22) \qquad & = U_{111}(\textit{Sys}, \textit{Usr}, \textit{CC}, \textbf{Bel}(\textit{Usr}, \textbf{Time-Need}(\textit{Sys}, \text{small}))) \\
& = \textit{Usr}'_{CC} =+ \textbf{Bel}(\textit{Usr}, \textbf{TimeNeed}(Sys, \text{small}))
\end{aligned}
$$

This update operation adds to the pending cognitive context of *Usr* the information that *Sys* needs a small amount of time.

$U_{111}(X,Y)$	$Y *_{CC} =+$ **Bel**(Y, **TimeNeed**(X, small))
$U_{112}(X,Y)$	$Y *_{CC} =+$ **Bel**(Y, **TimeNeed**(X, substantial))

Table 11 Elementary update functions for the semantics of time management functions.

4.1.2.4 Other Communicative Functions

The semantics of the dimension-specific communicative functions for Contact Management, Discourse Structuring, Own Communication Management, Partner Com-

munication Management, and Social Obligations Management is similar to that of the dimension-specific communicative functions considered above, and can be derived from their defintions as specified in ISO 24617-2 and at `http://dit.uvt.nl`; the most important difference is the use of other, dimension-specific semantic primitives. indexcommunicative function

4.2 Communicative Function Qualifiers

Communicative function qualifiers (Petukhova and Bunt, 2010) make the IS updates of the communicative functions that they qualify more elaborate. Qualifiers come in two varieties, restrictive and additive ones. Restrictive qualifiers make the preconditions of a communicative function more specific, for instance specifying for an answer that there is some uncertainty about the correctness of its content. Additive qualifiers, by contrast, enrich a communicative function with additional information, for instance adding that an offer is accepted *happily*. ISO 24617-2 and DIT^{++} have two classes of restrictive qualifiers, for expressing uncertainty and conditionality, and one class of additive qualifiers, for expressing sentiment. Certainty qualifiers can apply only to information-providing functions; conditionality only to action-discussion functions. Sentiment qualifiers can apply to every communicative function.

The following clauses in the definition of the interpretation function I_a specify the semantic interpretation of a communicative function qualified by a restrictive qualifier, by an additive one, and by both a restrictive and an additive one, respectively:

(23) a. $I_a(\langle f, q_r\rangle) = I_a(f)(F(q_r))$
b. $I_a(\langle f, q_a\rangle) = \lambda S.\lambda z.[F(f)(S,z) \sqcup F(q_a)(S,z)]$
c. $I_a(\langle f_i, q_r, q_a\rangle) = \lambda S.\lambda z.[((I_a(f_i))(F(q_r)))(S,z) \sqcup (F(q_a))(S,z)]$

The semantics of each of the individual qualifiers is defined in Table 12, with the elementary update function U_{500} defined as in (24), where SP_k stands for a predicate that represents a particular sentiment (see Table 12, bottom line).

(24) $U_{500}(X, Y, SP_k, a) : Y *_{CC} = +\mathbf{Bel}(\mathrm{Y}, SP_k(X, a))$

F(certain)	= 'firm'
F(uncertain)	= 'weak'
F(conditional)	= 'cond'
F(unconditional)	= $\top$ (the 'empty' condition)
F(sentiment$_k$)	= $\lambda X.\lambda Y.\lambda a.U_{500}(X, Y, SP_k, a)$

Table 12 Specification of the semantics of communicative function qualifiers.

We consider two examples. The first concerns a restrictive qualifier, illustrating the semantics of an answer qualified as uncertain, as in (25) (where $tdp5$ abbreviates the proposition that the train to Tilburg leaves from platform 5):

(25)
1. A: Does the train to Tilburg leave from platform 5?
2. B: I think so, probably yes.

$I_a(\langle$Answer, uncertain$\rangle)(B, A$,Task$, tdp5)$ =
$A{*}_{TaskC} =+$ **Bel**$(A,$ **Want**$(B,$**Bel**$(A, tdp5,$ weak$)))$;
$A{*}_{TaskC} =+$ **Bel**$(A,$ **Bel**$(B, tdp5,$ weak$))$;
$A{*}_{TaskC} =+$ **Bel**$(A,$ **Bel**$(B,$ **Want**$(A,$ **Know-val**$(A, tdp5))))$;
$A{*}_{TaskC} =+$ **Bel**$(A,$ **Bel**$(B,$ **Assume**$(A,$ **Know-val**$(B, tdp5))))$
This means that A's pending task context is extended with the following pieces of information:

(26)
1. **Bel**$(B, tdp5,$weak$)$, or equivalently: **Wk-Bel**$(B, tdp5)$; i.e., B holds the uncertain belief that $tdp5$;
2. **Want**$(B,$ **Wk-Bel**$(A, tdp5))$, i.e. B has the goal that A also holds this uncertain belief;
3. **Bel**$(B,$ **Want**$(A,$ **Know-val**$(A, tdp5)))$, i.e. B believes that A wants to know whether $tdp5$.
4. **Bel**$(B,$**Assume**$(A,$ **Know-val**$(B, tdp5)))$: B believes that A assumes that B knows whether $tdp5$.

The second example concerns the use of both a restrictive and an additive qualifier, illustrated by the semantics (using (23c)) of an unconditional Accept Offer with a happy sentiment, as in (27).

(27)
1. A: How about a cup of coffee?
2. B: Oh yes, that would be wonderful!

(28)
$I_a(\langle$AcceptOffer, unconditional, happy$\rangle)$ =
= $\lambda S.\lambda z.[[I_a($AcceptOffer$)(I_a($unconditional$))](S, z) \sqcup$
$[I_a($happy$)](S, z)] =$
$\lambda S.\lambda z.[[[\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.\lambda C_\alpha.\ U_{24}(X, Y, D_i, \alpha) \sqcup U_{25}(X, Y, D_i, \alpha, C_\alpha) \sqcup$
$U_{25b}(X, Y, D_i, \alpha, C_\alpha)](\top)](S, z) \sqcup U_{500}(X, Y,\textsc{happy}, \alpha)(S, z))]$ =
$\lambda S.\lambda z.\lambda Y.\lambda D_i.\ [U_{24}(S, Y, D_i, z) \sqcup U_{25}(S, Y, D_i, z, \top) \sqcup$
$U_{25b}(S, Y, D_i, z, \top) \sqcup U_{500}(X, Y,\textsc{happy}, z)]$

Applied to the participants A and B and the action of having coffee, we obtain:

(29)
$A{*}_{Task} =+$ **Bel**$(A,$ **Want**$(B, \mathbf{CommitDo}(A,$ have_coffee$)))$;
$A{*}_{Task} =+$ **Bel**$(A,$ **Bel**$(B,$ **WilDo**$(A,$ have_coffee$)))$;
$A{*}_{Task} =+$ **Bel**$(A,$ **Bel**$(B,$ **Want**$(A,$ **Bel**$(B,$ **WilDo**$(A,$ have_coffee$)))))$;
$A{*}_{Task} =+$ **Bel**$(A,$ HAPPY$(B,$ have_coffee$))$

In other words, A's pending context is extended with the beliefs that B wants A to commit himself to arrange coffee; that A is willing to do so; that A wants B to believe that; and that B would be happy to get some coffee.

5 Conclusion

In this chapter we have provided a computational semantics of dialogue acts in the form of updates of an addressee's information state. We have formulated this in the form of a semantics for the annotation structures defined by the abstract syntax of the language DiAML, the Dialogue Act Markup Language for semantic annotation, which forms part of ISO standard 24617-2 for dialogue annotation. The semantics as described in this chapter abstracts away from many of the details concerning the 'information states' or 'context models' of dialogue participants, to which the update operations apply, but by way of example we have adopted some of the assumptions of Dynamic Interpretation Theory regarding the structure and content of context models, and we have shown how such a choice can be useful for implementing update operations for the interpretation of dialogue acts.

This semantics provides an essential part of the foundations of the ISO standard for dialogue annotation, as well as of the DIT^{++} taxonomy of dialogue acts, which slightly extends the ISO standard. Both the ISO 24617-2 and the DIT^{++} annotation schemes go beyond what is commonly done in dialogue act annotation in not just indicating the communicative functions of utterances but also certain ways in which these functions may be qualified for uncertainty, conditionality, or sentiment, and also indicating functional dependence relations, feedback dependence relations, and rhetorical relations between dialogue acts and other dialogue units. The semantics described in this chapter takes these extensions into account.

Future work includes computer implementation, testing and evaluation of context models and their use in dialogue act interpretation and dialogue act generation for the entire ISO 24617-2 and DIT^{++} taxonomies, extending the partial implementations of Petukhova et al. (2010) and Keizer et al. (2010).

Acknowledgements

I thank the members of the Tilburg Dialogue Club, who over the years have contributed to shaping Dynamic Interpretation Theory and the DIT^{++} annotation scheme, as well as PhD students and colleagues in related projects. This includes Volha Petukhova, Jeroen Geertzen, Simon Keizer, Roser Morante, Amanda Schiffrin, Ielka van der Sluis, Hans van Dam, Yann Girard, Rintse van der Weff, Elyon Dekoven, Paul Piwek, Robbert-Jan Beun, René Ahn, and Leen Kievit. Important contributions have also come from collaborative work in ISO project 24617-2 "Semantic Annotation Framework, Part 2: Dialogue Acts", in particular with David Traum, Jan Alexandersson, Andrei Popescu-Belis, Laurent Prévot, Marcin Wlodarzcak, Jens Allwood, Jean Carletta, Jae-Woong Choe, Alex Fang, Kiyong Lee, Laurent Romary, Nancy Ide, Claudia Soria, Dirk Heylen, and David Novick.

References

Ahn, R. (2001). *Agents, Objects and Events*. Ph.D. Thesis, Eindhoven University of Technology.

Allen, J., L. Schubert, G. Ferguson, P. Heeman, C.H. Hwang, T. Kato, M. Light, N. Martin, B. Miller, M. Poesio, D. Traum (1994) *The TRAINS project: A case study in defining a conversational planning agent*. Technical Report 532, Computer Science Department, University of Rochester.

Allen, J. and M. Core (1997). *DAMSL: Dialogue Act Markup in Several Layers (Draft 2.1). Technical Report*. Rochester, NY: University of Rochester.

Alexandersson, J., B. Buschbeck-Wolf, T. Fujinami, M. Kipp, S. Koch, E. Maier, N. Reithinger, B. Schmitz & M. Siegel (1998). *Dialogue acts in VERBMOBIL-2 (second edition)*. Verbmobil Report 226. Saarbrücken: DFKI.

Augmented Multiparty Interaction Consortium (AMI) (2005) Guidelines for Dialogue Act and Addressee Annotation. Unpublished report, University of Edinburgh.

Bunt94 Bunt, H. (1994). Context and Dialogue Control. *Think Quarterly* 3 (1), 19–31.

Bunt, H. (1989). Information dialogues as communicative action in relation to partner modelling and information processing. In M. Taylor, F. Néel, and D. Bouwhuis (Eds.), *The structure of multimodal dialogue*, pp. 47–74. Amsterdam: North-Holland.

Bunt, H. (2000). Dialogue pragmatics and context specification. In Harry Bunt and William Black (Eds.), *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics.*, pp. 81–150. Amsterdam: John Benjamins.

Bunt, H. (2006). Dimensions in dialogue annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genova, Italy*, Paris. ELRA.

Bunt H (2009) A framework for dialogue act specification. In: D Heylen, C Pelachaud, D Traum (eds) Proceedings of EDAMLAAMAS Workshop "Towards a Standard Markup Language for Embodied Dialogue Acts, Budapest, pp 13–24

Bunt, H. (2010). A methodology for designing semantic annotation languages exploiting semantic-syntactic ISO-morphisms. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, Hong Kong.

Bunt, H. (2011). Multifunctionality in dialogue and its interpretation. *Computer, Speech and Language* 25, 222-245.

Bunt, H. (2012). The Semantics of Feedback. In *Proceedings of SeineDial, 2012 Workshop on the Semantics and Pragmatics of Dialogue*, Paris.

Bunt, H. (2013a). A methodology for designing semantic annotations. TiCC Technical Report TR 2013-001, Tilburg University.

Bunt, H. (2013b). Annotations that effectively contribute to semantic interpretation. *This volume*, chapter 4.

Bunt, H., J. Alexandersson, J.-W. Choe, A. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and D. Traum (2010). Towards an ISO standard for dialogue act annotation. In *Proceedings of LREC 2010, Malta*, Paris. ELDA.

Bunt, H., J. Alexandersson, J.-W. Choe, A. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. Traum (2012). A semantically-basedstandard for dialogue annotation. In *Proceedings of LREC 2012, Istanbul*, Paris. ELRA.

Carletta, J., S. Isard, J. Kowtko & G. Doherty-Sneddon (1996) *HCRC dialogue structure coding manual*. Technical Report HCRC/TR-82.

Di Eugenio, B., P. Jordan, and L. Pylkkaenen, (1998). *The COCONUT project: dialogue annotation manual*. ISP Technical Report 98-1 (1998).

Clark, H. (1996). *Using Language*. Cambridge, UK: Cambridge University Press.

Cooper, R. (2004). Information states, attitudes and dependent record types. In N. B. L. Cavedon, P. Blackburn and A. Shimolina (Eds.), *Logic, Language and Computation, Vol. 3*, pp. 85–106. Stanford: CSLI Publications.

Dhillon, R., S. Bhagat, H. Carvey, and E. Schriberg (2004). *Meeting recorder project: dialogue labelling guide. ICSI Technical Report TR-04-002.*

Geertzen, J. (2009). *Dialogue act recognition and prediction. Ph.D. Thesis, Tilburg Universsity.*

Geertzen, J., V. Petukhova, and H. Bunt (2007). A multidimensional approach to utterance segmentation and dialogue act classification. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, pp. 140–149.

Ide, N. and H. Bunt (2010). Anatomy of Annotation Schemes: Mappings to GrAF. In *Proceedings of LAW-IV: Fourth Linguistic Annotation Workshop*, Uppsala.

Ide, N. and L. Romary (2004). International Standard for a Linguistic Annotation Framework. *Natural Language Engineering 10*, 211–225.

ISO (2012b, September). *Language Resource Management - Semantic Annotation Framework (SemAF) - Part 2: Dialogue acts*. International Organisation for Standardisation ISO. ISO International Standard 24617-2:2012(E)).

ISO (2012c). *Linguistic Annotation Framework (LAF).* International Organisation for Standardisation ISO. ISO International Standard 24612-2:2012(E)).

Jurafsky, D., E. Schriberg, and D. Biasca (1997) Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual.

Keizer, S., H. Bunt, and V. Petukhova (2011) Multidimensional Dialogue Management. In: A. van den Bosch and G. Bouma (eds.) Interactive Multimodal Question Answering. Springer, Berlin, pp. 118-145.

Keizer, S. and H. Bunt (2006). Multidimensional dialogue management. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, Sydney, Australia, pp. 37–45.

Keizer, S. and H. Bunt (2007, September). Evaluating combinations of dialogue acts for generation. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, pp. 158–165.

Kievit, L., P. Piwek, R.-J. Beun, and H. Bunt (2001). Multimodal cooperative resolution of referential expressions in the DenK system. In H. Bunt and R.-J. Beun

(eds.) *Cooperative Multimodal Communication*, LNAI Series no. 2155, pp. 197–216. Berlin: Springer.

Petukhova, V. and H. Bunt (2009a). The independence of dimensions in multi-dimensional dialogue act annotation. In *Proceedings NAACL HLT Conference, Boulder, Colorado*.

Petukhova, V. and H. Bunt (2009b). Dimensions in communication. TiCC Technical Report TR 2009-003, Tilburg University.

Petukhova, V. and H. Bunt (2010). introducing communicative function qualifiers. In *Proceedings Second International Conference on Global Interoperability for Language resources (ICGL-2), Hong Kong*, pp. 132 – 132.

Petukhova, V., H. Bunt, and A. Malchanau (2010). Empirical and theoretical constraints on dialogue act combinations. In *Proceedings 14th Workshop on the Semantics and Pragmatics of Dialogue (PozDial)*, Poznán, Poland.

Petukhova, V., L. Prévot, and H. Bunt (2011). Discourse relations in dialogue. In *Proceedings 6h Joint ISO-ACL/SIGSEM Workshop on Interoperable Semantic Annotation (ISA-6)*, Oxford, UK.

Popescu-Belis, A. *Dialogue act tagsets for meeting understanding: an abstraction based on the DAMSL, Switchboard and ICSI-MR tagsets*. Technical report, IM2.MDM-09, v1.2 (2004)

Poesio, M. and D. Traum (1997). Conversational actions and discourse situations. *Computational Intelligence 13(3)*, 309 – 347.

Traum, D. and S. Larsson (2003). The information state approach to dialogue management. In J. van Kuppevelt and R. Smith (Eds.), *Current and New Directions in Discourse and Dialogue*, pp. 325–345. Dordrecht: Kluwer.

Appendix: The DIT^{++}/ISO 24617-2 taxonomies of communicative functions

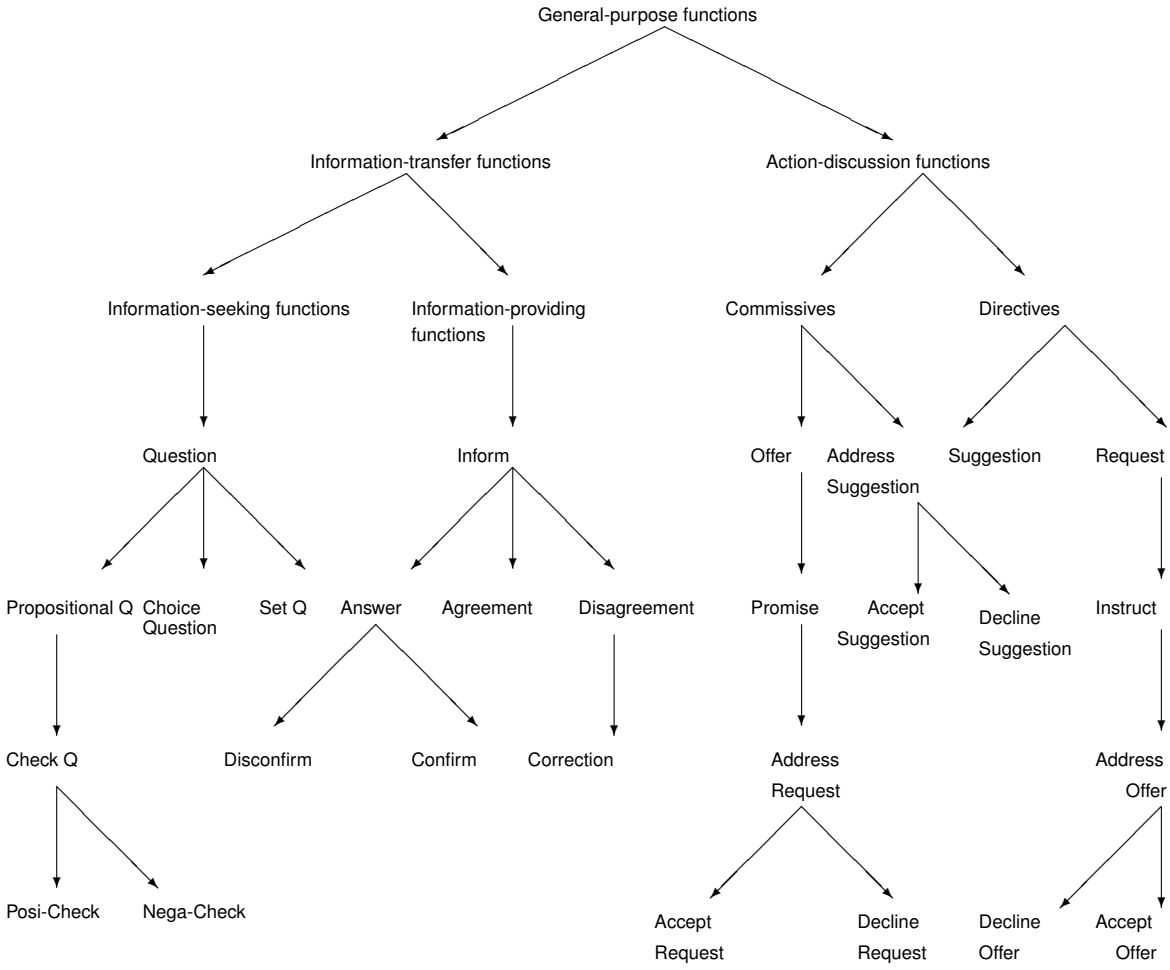

Fig. 1 General-purpose communicative functions in ISO 24617-2 and DIT^{++}

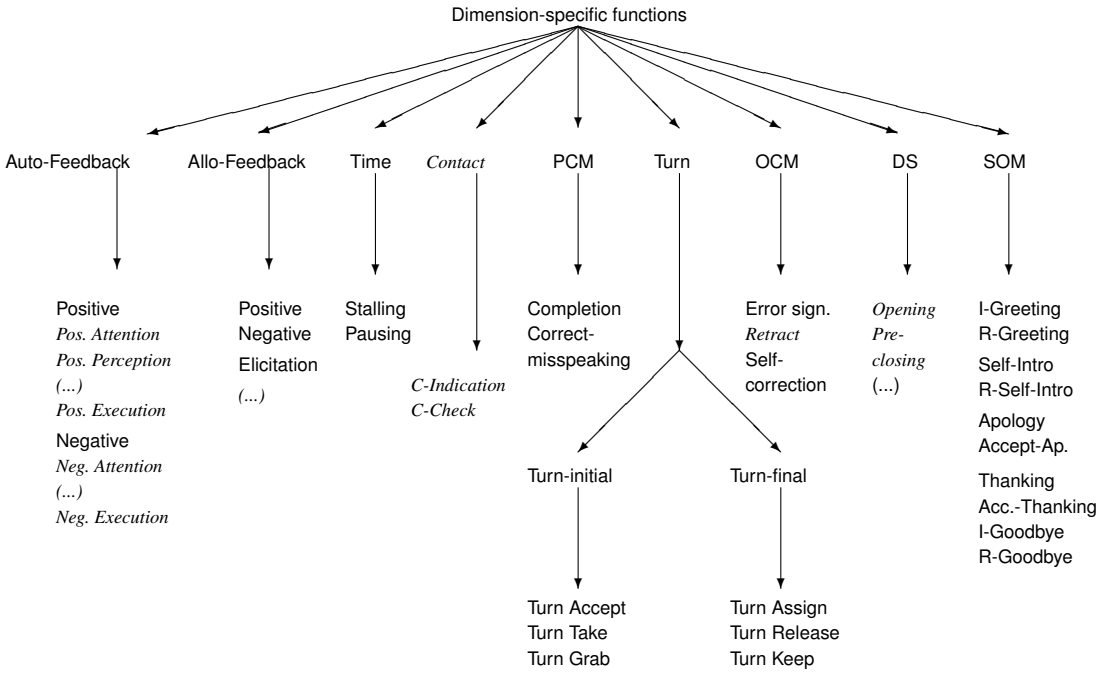

Fig. 2 Dimension-specific communicative functions in ISO 24617-2 and DIT^{++}. Functions and dimensions in italics are defined only in DIT^{++}.