The Semantics of Dialogue Acts

Harry Bunt
TiCC, Tilburg Center for Cognition and Communication
Tilburg University, The Netherlands
harry.bunt@uvt.nl

Abstract

This paper presents an update semantic for dialogue acts, defined in terms of combinations of very simple '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 dialogue act representations as defined in the Dialogue Act Markup Language (DiAML), part of the recently proposed 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 interpretation of the behaviour of dialogue participants and in the design of spoken dialogue systems. But in spite of their popularity, their status is nearly always reduced to that of informal, intuitive concepts which lack proper definitions (see Poesio and Traum, 1998 for one of the few attempts at formalization). A wide range of alternative dialogue act taxonomies and inventories have been proposed, causing considerable terminological and conceptual confusion, and problems for reusing annotated corpora. This has motivated the International Organisation for Standards ISO to develop a standard for interoperable dialogue act annotation, ISO 24617-2 (see ISO 2010). This proposed standard is partly based on the comprehensive DIT^{++} taxonomy, which has added to the earlier DIT taxonomy (Bunt, 1994) a number of concepts from other proposals and studies. Semantically, the DIT^{++} taxonomy is based on the dynamic approach to utterance meaning taken in Dynamic Interpretation Theory (DIT), which views dialogue acts as corresponding to update operations on the information states of participants in the dialogue; an approach commonly known as the 'information-state update approach' to meaning in dialogue – see e.g. Bunt (2000); Traum & Larsson (2003). A dialogue act, on this approach, has two main components: a semantic content, which describes the objects, properties, relations, or actions that the dialogue act is about, and a communicative function, which specifies how an addressee should update his information state with the semantic content.

Utterances in dialogue 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 functions to functional segments. The DAMSL annotation scheme for example (DAMSL = Dialogue Act Markup using Several Layers) distinguishes nine 'dimensions' as mutually exclusive groups of function tags.

Bunt (2006) introduces a notion of dimension based on the observation that participation in a dialogue involves, beyond activities strictly related to performing the underlying task, sharing information about the processing of utterances, managing the use of time, taking turns, and various other types of communicative activity, and defines dimensions as corresponding to such aspects of communication. Each dimension in this sense constitutes a category of communicative activity, and the dialogue acts involved in these activities are concerned with different types of information: 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 semantic content.

Petukhova & Bunt (2009a; 2009b) formulate criteria for distinguishing dimensions, and apply these in the analysis of the dimensions that occur in 18 existing annotation schemes, showing that the 10 dimensions of DIT^{++} form a well-founded set of dimensions. These are the following:

(1)
1. Task/Activity: dialogue acts for performing the task or activity underlying the dialogue
2. Auto-Feedback: providing information about the speaker's processing of previous utterances.
3. Allo-Feedback: the speaker expresses opinions or elicits information about the addressee's processing of previous utterances;
4. Contact Management: dialogue acts for establishing and maintaining contact;
5. Turn Management: concerned with grabbing, keeping, giving, or accepting the speaker role;
6. Time Management: the speaker indicates to need some extra 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). Similarly for commissive and directive functions. These functions are therefore called *general-purpose* functions, as opposed to *dimension-specific* functions. The DIT^{++} taxonomy therefore consists of two parts: a taxonomy of *general-purpose functions* and one of *dimension-specific functions* - see Appendix A and `http://dit.uvt.nl`.

2 DiAML: Dialogue Act Markup Language

The Dialogue Act Markup Language (DiAML) which is part of the ISO standard under development for dialogue act annotation (see Bunt et al., 2010, and `http://semantic-annotation.uvt.nl`) has been designed in accordance with the ISO Linguistic Annotation Framework (Ide & Romary, 2004), which makes a distinction between *annotation* and *representation*; 'annotation' refers to the linguistic 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. This distinction is implemented in the DiAML definition by a syntax that specifies, besides a class of XML-based *representation structures*, also a class of more abstract *annotation structures*. These two components are called the *concrete* and *abstract syntax*, respectively.

The abstract syntax defines a class of set-theoretical structures, called 'annotation structures'. It 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 combining these elements. The conceptual inventory consists of finite sets of elements called 'functional segments', 'dimensions', 'communicative functions', 'qualifiers', and 'rhetorical relations'.

An annotation structure consists of a set of *entity structures* and a set of *link structures*. Entity structures contain semantic information about a functional segment; link structures describe semantic relations between segments. The most important kind of entity structure is a so-called *'dialogue act structure'*, which is a quadruple $\langle S, A, d, f\rangle$ where S and A are the sender and addressee of a dialogue act; d is a dimension; and f is a communicative function or a pair $\langle f, q\rangle$, where q is a list of qualifiers.

The concrete syntax defines a rendering of annotation structures in XML. It is defined in accordance with the methodology for defining semantic annotation languages described in Bunt (2010), which introduces the notion of an *ideal representation format*, defined as one where every representation represents a uniquely determined annotation structureThe semantics of the language is then 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.[1] The concrete syntax of DiAML is illustrated in (3) and (2). 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, with value 'answer' qualified as 'uncertain'. (#-prefixed elements are assumed to be identified in the metadata of the source material or in another layer of annotation.)

(2)

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*
TA	fs2.2:	*The next train to Utrecht leaves I think at 8:32.*

(3)

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

3 Context Model Structure and Content

As the proposed semantics of dialogue acts is in terms of information-state 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. An information state will be assumed to have a number of components, an assumption which is shared between all proposals for information states (e.g. Poesio & Traum, 1998; Bunt, 2000; Ahn, 2001; Cooper, 2004); moreover, certain types of information can be argued to be required in information states. The details of an information-state update semantics also 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 *mutual* beliefs, as e.g. argued in Bunt (2000). We consider here only the updates of a single addressee's information state, disregarding mutual beliefs; this is anyway the basis for more complex approaches involving multiple information states and mutual beliefs. In DIT, it is customary to speak of 'contexts' or context models', rather than 'information states', and we will use this terminology in the rest of this paper.

A fundamental requirement for an adequate context model is that, for a given range of dialogue act types, the model contains the kinds of information that are updated by a dialogue act. Bunt (forthc.) argues that an agent's context model does not necessarily have a separate component for each DIT dimension, but that it is convenient to distinguish the following five components:

(4)
1. Linguistic Context, which contains a record of the dialogue history, information about discourse plans (if any), and wishes concerning the occupation of the speaker role;
2. Semantic 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;

[1] See Bunt (2010) for formal definitions and proofs relating to alternative representation formats sharing the same abstract syntax, and Ide & Bunt (2010) for applying this to the GrAF framework for linguistic annotation.

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) and for experimentation by Petukhova et al. (2010).

An update semantics has to take into account that update operations should not undermine the consistency of the context model. A dialogue participant may change his mind during the dialogue, as an effect of receiving some unexpected information, which can have the effect that the participant brings in new information which contradicts something that was already grounded, and hence cannot simply be added without making the context model inconsistent. Rather then building consistency checks into the semantics of each dialogue act, we exploit the DIT distinction of five levels of utterance processing: (1) attention, (2) perception, (3) understanding, (4) evaluation, and (5) execution. 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, it performs the updates. 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.[2] Updating the pending context is a matter of simply *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. We will use the notation (5a) to specify the update consisting of adding the information z to component A'_i i of A's pending context. If f is the update (5a) and g the update $A'_j =\!\!+u$, then (5b) designates the combination of the two updates.[3]

(5) a. $A'_i =\!\!+z$
b. $f \sqcup g$

An analysis of the definitions of the DIT^{++} communicative functions shows that a formal description of the update effects of dialogue acts with a general-purpose function requires the basic concepts listed in Table 1. For convenience, we also 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 the argument $\top$ representing the 'empty' condition, hence **WilDo**(S,α) abbreviates **WilDo**$(S,\alpha,\top)$, and so on.

description	notation	meaning
believes that	**Bel**(S,p,σ)	*S believes that p*; σ indicates whether this is a firm belief or an uncertain belief (σ can have the values 'firm' and 'weak')
knows value of	**Know-val**(S,z)	*S possesses the information z*
has goal	**Wantl**(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* fulfilled; C_α may be the universally true statement $\top$
is committed to do	**CommitDo**(S,α,C_α)	*S is committed to perform the action* α *if the condition* C_α *is* fulfilled; the condition C_α may be 'empty' ($\top$)
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 fulfilled* C_α may be 'empty' ($\top$)
is considering to be done	**ConsidDo** (X,α,Y,C_α)	*X is considering the action* α*, to be performed by Y, if the condition* C_α *is fulfilled* C_α may be 'empty' ($\top$)
is in the interest of	**Interest**(Y,α)	action α is of interest to agent Y.

Table 1: Basic semantic concepts for general-purpose communicative function interpretation

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

[3] The combined update $(f \sqcup g)$ is undefined if the order of performing the two updates would make a difference.

Dimension	*Primitives*
Auto- and Allo-feedback	**Attended, Perceived, Understood, Accepted, Executed, Attention-Problem, Perception-Problem, Interpretation-Problem, Evaluation-Problem, Execution-Problem**
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 Man.	**Available, Thankful, Regretful, Knows-id, Final**

Table 2: Dimension-specific semantic primitives

Dimension-specific communicative functions are always concerned with a specific category of semantic content, which requires certain specific semantic primitives for its representation. Table 2 lists the basic concepts for describing their update semantics.

For expressing the semantics of a feedback act which is underspecified for the level of processing, we introduce in (6) the predicates **Succes-Processing**, defined as successful at least at the level of understanding, and **Unsuccessful-Processing**, defined as unsuccessful at the level of understanding or lower.

(6) a. **Succes-Processing = Understood** $\vee$ **Accepted** $\vee$ **Executed**
b. **Unsuccessful-Processing = Interpretation-Problem** $\vee$ **Perception-Problem** $\vee$ **Attention-Problem**

4 Dialogue Act Semantics

In this section we outline a semantics of dialogue acts in the form of an update semantics for the 'dialogue act structures' defined by the DiAML abstract syntax. A dialogue act structure does not correspond to a full-blown dialogue act representation, since it does not include the full semantic content, but only the dimension which classifies the semantic content. The semantics of a dialogue act structure should therefore be something which can be combined with a semantic content in order to form the interpretation of a full-blown dialogue act. This is precisely the case, for the recursive interpretation of a dialogue act structure $\langle S, A, d, f\rangle$ is defined through the recursive valuation function V as specified in (7). Of the four arguments of V in the left-hand side of (7), *S, A,* and d are elements of the categories of the DiAML conceptual inventory, so there is no recursion in their interpretation; for such elements, the valuation function is defined by a value assignment function F, playing the same role as that of a 'model assignment' function in model-theoretic semantics; F for example assigns to a sender and an addressee certain individuals, identified in the metadata of an annotated dialogue (cf. $\#p1$ and $\#p2$ in (3)). To the dimension argument d, F assigns that component of an information state that should be updated.

(7) $V(<\text{S, A, d, f}>) = (V(\text{f}))(F(S), F(A), F(d))$

4.1 The Update Semantics of Communicative Functions

A communicative function will be interpreted as a function which, applied to a given speaker, addressee, and dimension, results in a function which can be applied to a semantic content in order to obtain a context-update specification. Since related communicative functions often share parts of their defining preconditions, we will construct such interpretations as *combinations of elementary update functions*, each of which takes care of the update corresponding to a single dialogue act precondition; see Table 3 and Table 4 for illustration: Table 3 lists the definitions of the update semantics of the communicative functions of the information-providing class, while Table 4 lists the elementary elementary update functions used in these definitions.

4.1.1 General-Purpose Communicative Functions

The class of general-purpose communicative functions in the DIT^{++} taxonomy falls apart into the *information-transfer functions* and *action-discussion functions*, further subdivided into information-providing and information-seeking functions, and commissives and directives, respectively.

a. Information-Providing and Information-Seeking Functions The class of information-providing functions has a hierarchical structure, with the communicative function Inform as the mother of all information-providing functions; all other functions are specializations of this function. These functions all have in common that (1) the speaker wants the addressee to possess certain information which (2) the speaker assumes to be correct.

Using the epistemic operators introduced in Section 5, these preconditions are formalized as follows:

(8) 1. **Want**$(S, U,$ **Bel**$(A, p, \sigma))$
2. **Bel**(A, p, σ)

The semantics of the Inform function, specified in Table 3, binds the variable σ, representing the belief strength for both the elementary update functions involved. (See further below, section 4.2.)

The update semantics in terms of combinations of elementary update functions often brings out immediately that some communicative functions are specializations of others (as visualized in Appendix A), for instance, the update semantics of the Answer function shares with the Inform function the updates defined by the elementary update functions U_1 and U_2, and adds to that the effects of U_7 and U_9; the semantic of the Confirm function adds to that the update defined by U_8. Hence Confirm is a specialization of Answer, which is a specialization of Inform, or in other words Confirm entails Answer entails Inform.

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)	=	$\lambda s.\lambda X.\lambda Y.\lambda D_i.\lambda p.U_1(X,Y,D_i,\neg p,s) \sqcup U_2(X,Y,D_i,\neg p,s) \sqcup U_8(X,Y,D_i,\neg p,s)$ $\sqcup U_9(X,Y,D_i,p) \sqcup U_7(X,Y,D_i,p)$
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)	=	$\lambda X.\lambda Y.\lambda D_i.\lambda z.U_{10}(X,Y,D_i,p) \sqcup U_{11}(X,Y,D_i,p) \sqcup U_4(X,Y,D_i,p)$
F(SetQuestion)	=	$\lambda X.\lambda Y.\lambda D_i.\lambda z.U_{10}(X,Y,D_i,P) \sqcup U_{11}(X,Y,D_i,P) \sqcup U_{13}(X,Y,D_i,P)$
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

As an illustration of the update semantics of information-providing functions, consider the case of the answer in (9.2).

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

Applying the semantics of the Answer function (see Table 3) to the participants C and D and the semantic content of (9.2), we obtain:

(10) F(Answer)(C, D, Task, EU25=BP20) = U_1(C,D,SemC, EU25=BP20) $\sqcup$
$\sqcup$ U_2(C,D,Task, EU25=BP20) $\sqcup$ U_9(C, D, Task, EU25=BP20) $\sqcup$ U_7(C, D, Task, EU25=BP20) =
$D'_{SemC} \mathrel{=\!\!+}$ **Bel**(D, **Want**(C, **Bel**(D, EU25=BP20))); $D'_{SemC} \mathrel{=\!\!+}$ **Bel**(D, **Bel**(C, EU25=BP20));
$D'_{SemC} \mathrel{=\!\!+}$ **Bel**(D, **Bel**(C, **Want**(D, **Know-val**(D, EU25=BP20)))); $D'_{SemC} \mathrel{=\!\!+}$ **Bel**(D, **Bel**(C, **Assume**(D, **Know-val**(C, EU25=BP20))))

Hence the following beliefs are added to D's pending Semantic Context: (1) C wants D to know that EU25=BP20; (2) C believes that EU25=BP20; (3) C believes that D wants to know whether EU25=BP20; and (4) C believes that D assumes C to know whether EU25=BP20.

$U_1(X, Y, D_i, p, s)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Want}(X,\mathbf{Bel}(Y, p, s)))$
$U_2(X, Y, D_i, p, s)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Bel}(X, p, s))$
$U_3(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Assume}(X, p))$
$U_4(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \textbf{Wk-Bel}(X, p))$
$U_5(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Bel}(X, \mathbf{Assume}(Y, p)))$
$U_6(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Assume}(X, \mathbf{Assume}(Y, p)))$
$U_7(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Bel}(X, \mathbf{Assume}(Y, \textbf{Know-val}(X, P))))$
$U_8(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Assume}(X, \textbf{Wk-Bel}(Y, p))$
$U_9(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Bel}(X, \mathbf{Want}(Y, \textbf{Know-val}(Y, p))))$
$U_{10}(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Want}(X, \textbf{Know-val}(X,)))$
$U_{11}(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Assume}(X, \textbf{Know-val}(Y, p))$
$U_{12}(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Bel}(X, p \vee \neg p))$
$U_{15}(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Assume}(X, p_1\ xor\ p_2))$
$U_{15a}(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Want}(X, \mathbf{Bel}(X, p_1) \vee \mathbf{Bel}(X, p_2))))$
$U_{16}(X, Y, D_i, p)$	$Y_i' =\!\!+$ $\mathbf{Bel}(Y, \mathbf{Assume}(X, \mathrm{Bel}(Y, p_1) \vee \mathbf{Bel}(Y, p_2))))$

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

b. Commissive and Directive Functions For the classes of commissive and directive communicative functions, we provide for reasons of space the semantics of only a small selection of functions; see Bunt (2011a) for more.

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 a selection of commissive and directive functions

As an example of the interpretation of a directive dialogue act, consider the request in (11.2):

(11) 1. A: (...)
2. B: Could you 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:

(12) F(Request)(A, B, Auto-Feedback, $\langle$Repeat(u1), unconditional$\rangle$) = $\lambda C_\alpha.\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.)$
$U_{23}(X, Y, D_i, \alpha, C_\alpha \sqcup U_{26}(X, Y, D_i, \alpha)$(A, B, Auto-Feedback, Repeat(u1), $\top$) =
= U_{23}(A,B, CC, Repeat(u1), $\top$) $\sqcup$ U_{26}(A,B, C, Repeat(u1)) =
$B'_{CC} =\!\!+$ $\mathbf{Bel}(B, \mathbf{Want}(A,$ [$\mathbf{WilDo}(A, Repeat(u1) \rightarrow \mathbf{CommitDo}(B, Repeat(u1))$]));
$B'_{CC} =\!\!+$ $\mathbf{Bel}(B, \mathbf{Bel}(A, \mathbf{CanDo}(B, Repeat(u1))))$

where 'CC' stands for Cognitive Context.

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)

$U_{17}(X,Y,D_i,\alpha,C_\alpha)$	$Y_i' =\!+$ **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**$(Y,\alpha,C_\alpha)]))$
$U_{24}(X,Y,D_i,\alpha)$	$Y_i' =\!+$ **Bel**$(Y,$ **Want**$(X,$ $\mathrm{CommitDo}(Y,\alpha)))$
$U_{25}(X,Y,D_i,\alpha,C_\alpha)$	$Y_i' =\!+$ **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**$(Y,$ $\mathrm{CanDo}(X,\alpha)))$
$U_{27}(X,Y,D_i,\alpha,C_\alpha)$	$Y_i' =\!+$ **Bel**$(Y,$ **CommitRefrain**$(X,\alpha,C_\alpha))$

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

and those concerned with the addressee's processing, as perceived by the speaker (Allo-Feedback). The elementary update functions for both dimensions are nearly identical, only differing in whose processing is concerned. Tables 7 and 8 show the update semantics of a small, representative subset of the (25) DIT^{++} communicative functions for providing and eliciting feedback.

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

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

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

Table 8: Semantics of feedback functions (selection)

4.1.2.2 Turn Management Functions

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

For example, assigning the turn to somebody (Turn Assign) means that the participant A, who currently occupies the speaker role, wants the indicated other participant, B, to occupy the speaker role next. This is expressed in the form of a combination of elementary update functions as shown in (13):

(13) F(TurnAssign)(A,B) = $[\lambda X.\lambda Y.U_{101}(X,Y,TurnM) \sqcup U_{102}(X,Y,TurnM](A,B)$ =
= $U_{101}(A,B,TurnM) \sqcup U_{102}(X,Y,TurnM)$ =
= $B'_{LiC} =\!+$ **Bel**$(A,$ **Current-Speaker**$(A))$; $B'_{LiC} =\!+$ **Want**$(A,$ **Next-Speaker**$(B))$

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.

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

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

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

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 his speech tempo) or 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 to contain beliefs about the amount of time needed by cetain 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 cognitive processing.

Consider for example consider the update semantics of a Stalling act:

$$\begin{array}{lcl} V(<\mathit{Sys},\mathit{Usr},\ \text{TimeM, Stalling}>) & = & F(\text{Stalling})(\mathit{Sys},\ \mathit{Usr},\ \text{CogC}) \\ & = & U_{111}(\mathit{Sys},\ \mathit{Usr},\ \mathit{CogC},\textbf{Time-Need}(\mathit{Sys},\ \textbf{small})) \\ & = & \mathit{Usr}'_{CC} =\!+\ \textbf{TimeNeed}(\mathit{Sys},\ \textbf{small}) \end{array} \quad (14)$$

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

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

Table 11: Elementary update schemes 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 Communication Management, and Social Obligations Management is quite similar to that of the dimension-specific communicative functions that considered above. the main difference being the use of other, dimension-specific predicates.

4.2 The Interpretation of Communicative Function Qualifiers

Communicative function qualifiers come in two varieties, 'q-specifiers' and 'q-additives'. Q-specifiers make preconditions of the communicative function that they qualify more specific, for instance specifying for an answer that there is some uncertainty about the correctness of its content. Q-additives enrich a communicative function, for instance adding that an offer is accepted *happily*. Currently DIT distinguishes two classes of q-specifiers, the 'certainty' and 'conditionality' qualifiers, and one type of q-additive, for 'sentiment' representation.Qualifiers can apply only to general-purpose communicative functions; certainty qualifiers to information-providing functions, and conditionality qualifiers to action-discussion functions. Sentiment qualifiers can be attached in principle to every communicative function.

For the semantics of qualified communicative functions we thus have three possible cases to consider, where f_i is an unqualified communicative function: (a) $\langle f_i, qs_j\rangle$ where qs_j is a q-specifier; (b) $\langle f_i, qa_k\rangle$ where qa_k is a q-additive; and (c) $\langle f_i, qs_j, as_k\rangle$ where qs_j is a q-specifier and qa_k is a q-additive. The following clauses in the definition of the recursive valuation function V for DiAML specify the semantic interpretation in each of these cases:

(15) a. $V(\langle f_i, qs_j\rangle) = (F(f_i))(F(qs_j))$

b. $V(\langle f_i, qa_k\rangle) = \lambda S.\lambda z.[(F(f_i))(S,z) \sqcup (F(qa_k))(S,z)]$

c. $V(\langle f_i, qs_j, qa_k\rangle) = \lambda S.\lambda z.[((F(f_i))(F(qs_j)))(S,z) \sqcup (F(qa_k))(S,z)]$

The semantics of each of the individual qualifiers is defined as follows:

(16)

F(certain)	=	'firm'
F(uncertain)	=	'weak'
F(conditional)	=	'cond'
F(unconditional)	=	$\top$ (the 'empty' condition)
F(sentiment$_k$)	=	$\lambda X.\lambda u.$ SENTIMENT-PREDICATE$_k(X,u)$

We consider two examples. The first illustrates the semantics of an answer, qualified as uncertain, as in (17) ('p5' abbreviates the proposition that the train to Tilburg leaves from platform 5):

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

(18)
$$
\begin{aligned}
&V(\langle B, A,\text{Task}, p5, \langle\text{Answer, uncertain}\rangle\rangle) = V(\langle\text{Answer, uncertain}\rangle)(A, B,\text{Task}, p5)\\
&\quad = B'_i =\!\!+ \textbf{Bel}(B, U_1(A, B,\text{Task}, p5,\text{weak}) \sqcup U_2(A, B,\text{Task}, p5,\text{weak}) \sqcup U_9(A, B, Task, p)\\
&\qquad\qquad\qquad \sqcup U_7(A, B, Tak, p)\\
&\quad = A'_{SemC} =\!\!+ \textbf{Bel}(A, \textbf{Want}(B,\textbf{Bel}(A, p, \text{weak}))); A'_{SemC} =\!\!+ \textbf{Bel}(A, \textbf{Bel}(B, p, \text{weak}));\\
&\qquad A'_{SemC} =\!\!+ \textbf{Bel}(A, \textbf{Bel}(B, \textbf{Want}(A, \textbf{Know-val}(A, p))));\\
&\qquad A'_{SemC} =\!\!+ \textbf{Bel}(A, \textbf{Bel}(B, \textbf{Assume}(A, \textbf{Know-val}(B, p))))
\end{aligned}
$$

This means that A's pending semantic context is extended with the following pieces of information:

(19) 1. **Bel**$(B, p5,$weak), or equivalently: **Wk-Bel**$(B, p5)$; i.e., B holds the uncertain belief that $p5$;
2. **Want**(B, **Wk-Bel**$(A, p5)$), i.e. B has the goal that A also holds this uncertain belief;
3. **Bel**(B, **Want**(A, **Know-val**(A, p))), i.e. B believes that A wants to know whether $p5$.
4. **Bel**(B,**Assume**(A, **Know-val**(B, p))): B believes that A assumes that B knows whether $p5$.

Second, example (20) illustrates the semantics of an unconditional Accept Offer with a happy sentiment (as in A: *How about a cup of coffee?* B: *Oh yes, that would be wonderful!*), using (15c).

(20)
$$
\begin{aligned}
&V(\langle\text{AcceptOffer, unconditional, happy}\rangle) =\\
&\quad = \quad \lambda S.\lambda z.[[F(\text{AcceptOffer})(F(\text{unconditional}))](S,z) \sqcup [F(\text{happy})](S,z)]\\
&\quad = \quad \lambda S.\lambda z.[[[\lambda X.\lambda Y.\lambda D_i.\lambda\alpha.\lambda C_\alpha.\, U_{24}(X,Y,D_i,\alpha) \sqcup U_{25}(X,Y,D_i,\alpha,C_\alpha) \sqcup\\
&\qquad\quad\; U_{25b}(X,Y,D_i,\alpha,C_\alpha)](\top)](S,z) \sqcup \text{HAPPY}(S,z))]\\
&\quad = \quad [[\lambda S.\lambda Y.\lambda D_i.\lambda z.\lambda C_z.\, U_{24}(S,Y,D_i,z) \sqcup U_{25}(S,Y,D_i,z,\top) \sqcup\\
&\qquad\quad\; U_{25b}(S,Y,D_i,z,\top)] \sqcup \text{HAPPY}(S,z))]
\end{aligned}
$$

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

(21)
$$
\begin{aligned}
= \quad & A'_{Task} =\!\!+ \textbf{Bel}(A, \textbf{Want}(B, \textbf{CommitDo}(A, \text{coffee})));\\
& A'_{Task} =\!\!+ \textbf{Bel}(A, \textbf{Bel}(B, \textbf{WilDo}(A, \text{coffee})));\\
& A'_{Task} =\!\!+ \textbf{Bel}(A, \textbf{Bel}(B, \textbf{Want}(A, \textbf{Bel}(B, \textbf{WilDo}(A, \text{coffee})))));\\
& A'_{CC} =\!\!+ \text{HAPPY}(B, \text{coffee}))]
\end{aligned}
$$

In other words, the Task component of 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 s o; and that A wants B to believe that. Moreover, the understanding that B is happy to get some coffee is represented in the cognitive component of A's pending context.

Concerning the certainty regarding the correctness of provided information, as represented through certainty qualifiers, the unmarked case in natural language is *certain*. A speaker who is quite certain about something may indicate this by expressions like *definitely, most certainly*, but this tends to occur only when doubt or disbelief has expressed about something that was claimed. When there is no expression of uncertainty, the speaker's utterance is therefore interpreted as expressing certainty. For conditionality, the unmarked case is *unconditional*; an unconditional commitment or willingness to perform a certain action can be expressed explicitly, but this tends to occur only if some doubt has been expressed about someone's commitment or willingness. When no conditions for performing an action are expressed, we therefore interpret the utterance as unconditional.

5 Conclusion and Future Work

This paper has outlined an update semantics of dialogue acts, associated with annotation structures defined by the abstract syntax of the DIAML language for semantic annotation, which forms part of ISO standard (24617-2) under development for dialogue act annotation.

Future work that's crying to be done includes further implementation, testing and evaluation beyond what has already been done (see Petukhova, Bunt and Malchanau, 2010; Keizer, Bunt and Petukhova, 2010), and supplementing the approach with an interpretation of the relations between dialogue acts and other units in dialogue (see Petukhova, Prévot and Bunt, 2011).

Acknowledgements

I thank the members of the Tilburg Dialogue Club, who over the years have contributed to shaping Dynamic Interpretation Theory, 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 Werff, 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.

References

Ahn, R. (2001). *Agents, Object and Events: A computational approach to knowledge, observation and communication*. PhD Thesis, Eindhoven University of Technology.

Bunt, H. (2000). Dialogue pragmatics and context specification. In H. Bunt and W. 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, pp. 919–924. ELRA.

Bunt, H. (2009). Multifunctionality and muldimensional dialogue semantics. In *Proceedings of Dia-Holmia, 13th Workshop on the Semantics and Pragmatics of Dialogue*, Stockholm, pp. 3–14.

Bunt, H. (2010). A methodology for designing semantic annotation languages. In *Proceedings of the 2nd International Conference on Global Interoperability for Language Resources*, Hong Kong, pp. 29–46.

Bunt, H. (2011a). Formal specification of an update semantics for dialogue acts. TiCC Technical Report TR 2011-001, Tilburg Center for Cognition and Communication.

Bunt, H. (2011b). Multifunctionality in dialogue and its interpretation. *Computer, Speech and Language* (25), 225 – 245.

Bunt, H. (forthc.). Interpretation and generation of dialogue with multidimensional context models. In A. Esposito (Ed.), *Toward Autonomous, adaptive, and context-aware multimedia interfaces*, pp. 81–131. Berlin: Springer.

Bunt, H., J. Alexandersson, J. Carletta, J.-W. Choe, A. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and Traum (2010). Towards an ISO standard for dialogue act formal annotation. In *Proceedings 7th International Conference on Language Resources and Evaluation (LREC 2010)*, Malta. Paris: ELRA.

Cooper, R. (2000). Information states, attitudes and dependent record types. In L. Cavedon, P. Blackburn, N. Braisby, and A. Shimojima (Eds.), *Logic, Language and Computation, Vol 3*, pp. 85–106. Stanford: CSLI Publications.

Core, M. and J. Allen (1997). Coding dialogs with the DAMSL annotation schema. In *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, MA.

Ide, N. and H. Bunt (2010). Anatomy of semantic annotation schemes: Mappings to GrAF. In *Proceedings of the4th Linguistic Annotation Workshop (LAW-IV)*, Uppsala.

Ide, N. and L. Romary (2004). International standard for a linguistic annotation framework. *Natural Language engineering 10*, 211–225.

ISO (2010). *DIS 24617-2: Semantic annotation framework Part 2: Dialogue acts.* ISO, Geneva: Draft International Standard, July 2010.

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*, pp. 197–214. Berlin: Springer.

Morante, R. (2007). *Computing meaning in interaction.* Ph.D. Dissertation, Tilburg University.

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

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

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, Poznan.*

Poesio, M. and D. Traum (1998). Towards an axiomatisation of dialogue acts. In *Proceedings of the twente Workshop on the Semantics and Pragmatics of Dialogue*, Enschede, pp. 207 – 222.

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

Appendix: The DIT++ taxonomy of communicative functions

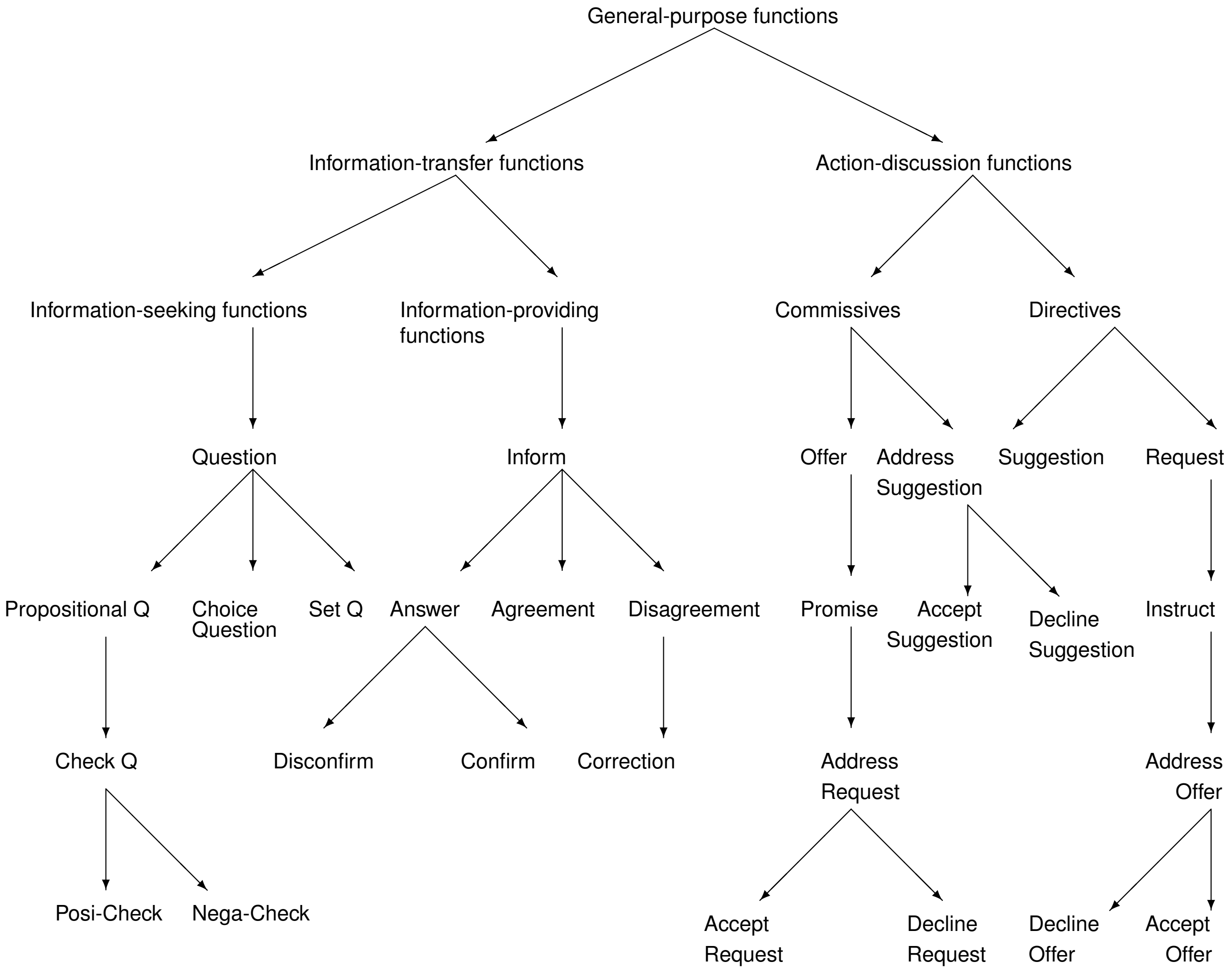

Figure 1: General-purpose functions

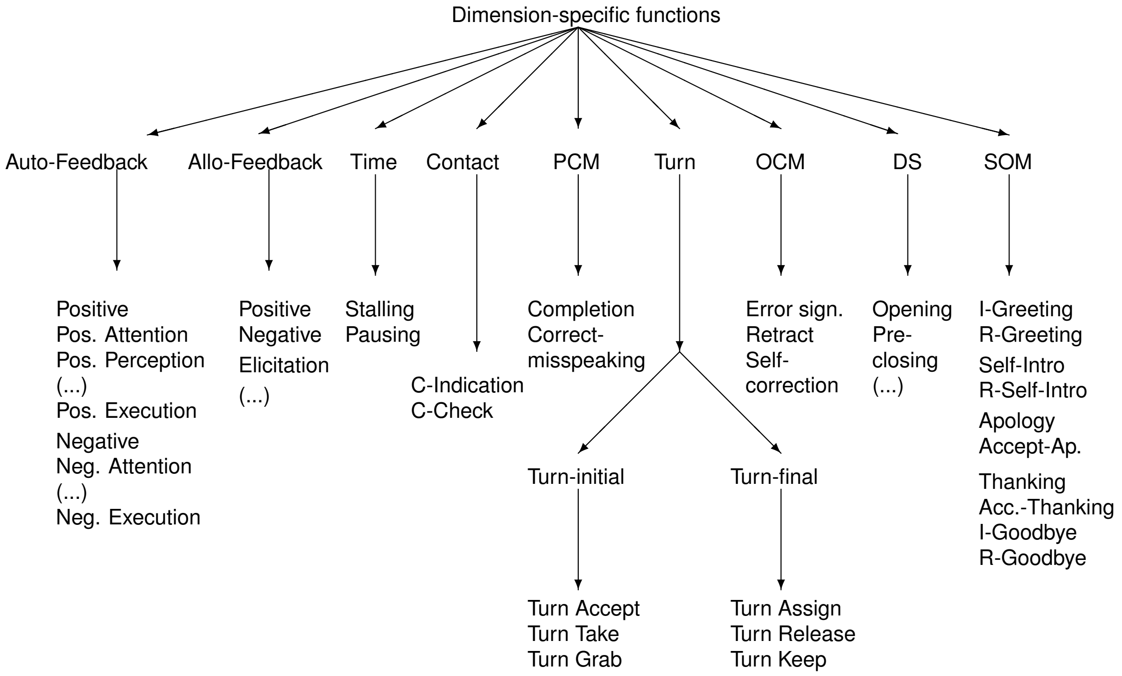

Figure 2: Dimension-specific communicative functions