
Language resource management — Semantic annotation framework — Part 2: Dialogue acts

Gestion des ressources linguistiques — Cadre d'annotation sémantique (SemAF) — Partie 2: Actes de dialogue

Document type: International Standard
Document subtype: Not applicable
Document stage: (40) Draft International Standard
Document language: E



Reference Number
ISO/DIS 24617-2(E)

©ISO 2010

This ISO document is a DRAFT INTERNATIONAL STANDARD.

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

*ISO copyright office
Case postale 56 CH-1211 Geneva 20
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org*

Published in Switzerland

Contents

Page

Foreword	1
1 Scope	2
2 Normative references	2
3 Terms and definitions	2
4 Purpose and justification	5
5 Basic concepts and metamodel	5
6 Approaches to the definition of communicative functions	9
7 Annotation schemas	10
7.1 Schema structure	10
7.2 Multidimensionality and multifunctionality	11
7.3 Multidimensionality, clustering, and dimensions	12
7.4 Dimension-specific and general-purpose functions	12
8 Dialogue segmentation	13
9 Core dimensions and dialogue acts	15
10 Core dialogue acts	17
10.1 General-purpose functions	19
10.2 Dimension-specific functions	20
10.2.1 The Task dimension	21
10.2.2 Feedback	21
10.2.3 Turn Management	22
10.2.4 Time Management	22
10.2.5 Discourse Structuring	22
10.2.6 Own and partner communication management	22
10.2.7 Social obligations management	23
10.3 Function qualifiers	23
11 DiAML: Dialogue Act Markup Language	24
11.1 Abstract syntax	25
11.2 Concrete XML-based syntax	26
11.3 DiAML semantics	28
12 Principles for extending and restricting the standard	30
12.1 Main design principles	30
12.2 Schema extension	31
12.3 Schema restriction	32
Annex A (normative) Annotation guidelines	33
A.1 General issues in DA annotation	33
A.1.1 Preliminaries	33
A.1.2 Dialogue settings and participants	33
A.1.3 Annotation purposes and unusual annotation situations	34
A.1.4 Explicit and implicit, implied and indirect functions	34

A.1.5	General advice for annotators	35
A.2	Segmentation	36
A.3	Annotation representation in DiAML	37
A.3.1	Encoding general-purpose functions	38
A.3.2	Encoding dimension-specific functions	39
A.3.3	Encoding communicative function qualifiers	41
A.3.4	Encoding functional dependences, feedback dependences, and rhetorical relations	44
Annex B	(normative) Completely annotated examples	46
B.1	Short dialogue fragments	46
B.2	Annotated dialogues	48
B.2.1	Annotated human-computer dialogue	48
B.2.2	Annotated Map Task dialogue	53
Annex C	(normative) DiAML schema	59
C.1	Overview	59
C.2	Example	60
Annex D	(normative) Data categories for core concepts	63
D.1	Overview	63
D.2	Dialogue participants	63
D.3	Functional segments	63
D.4	Dimensions	64
D.5	Communicative functions	65
D.5.1	General-purpose functions	65
D.5.2	Feedback functions	71
D.5.3	Turn management functions	72
D.5.4	Time management functions	73
D.5.5	Own and partner communication management functions	73
D.5.6	Discourse structuring functions	74
D.5.7	Social obligations management functions	75
D.6	Qualifiers	76
D.6.1	Certainty	76
D.6.2	Conditionality	77
D.6.3	Partiality	77
D.6.4	Sentiment: Emotion and Attitude	78
Annex E	(informative) Data categories for non-core communicative functions and dimensions	79
E.1	Non-core dimensions	79
E.2	Non-core communicative functions	79
Annex F	(informative) A survey and analysis of dimensions and communicative functions in existing annotation schemas	80
Annex G	(informative) Editors, contributors and meetings	91
G.1	Editors and contributors	91
G.2	Meetings and Workshops	93
	Bibliography	94

Language resource management — Semantic annotation framework — Part 2: Dialogue acts

Foreword

International Standard 24617-2 was prepared by Technical Committee ISO/TC 37, *Terminology and Other Language Resources*, Subcommittee 4, *Language Resource Management*, Working Group 2, *Representation schemas*, following up on the EU-supported project LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems) in collaboration with TC 37/SC 4 ad-hoc Thematic Domain Group 3, Semantic content.

The main parts of ISO 24617-2 are:

- a) Scope
- b) Terms and definitions
- c) Basic concepts and metamodel
- d) Defining communicative functions
- e) Dialogue segmentation
- f) Core dimensions
- g) Core dialogue acts
- h) Specification of DiAML, a formal annotation and representation language for dialogue acts
- i) Principles for schema extension and restriction

In addition, there are four normative and three informative annexes. Annex A (normative) contains guidelines for using the concepts, defined in this standard, and the DiAML language for annotating multimodal dialogues with dialogue act information. Annex B (normative) provides a number of completely annotated examples. The normative Annex C contains the definition of an XML-based representation format for DiAML-annotations. Annex D (normative) contains the data categories for the core communicative functions of this standard. These are a subset (with minor modifications) of the semantic data categories that were compiled in the EU project LIRICS, and approved by ISO/TC 37/SC 4/TDG 3, Semantic content for inclusion in the ISO Data Category Registry. Annex E (informative) contains a number of data categories for non-core communicative functions. Annex F (informative) summarizes a study, performed as part of the project of establishing this standard, establishing and testing criteria for identifying core dimensions and core communicative functions, based on a survey of 18 existing dialogue act annotation schemas. Annex G (informative) provides editorial and authorship information for the current document, with a list of editors, contributors, and meetings. The document concludes with a bibliography.

1 Scope

Utterances in interactive discourse, such as spoken dialogue, have one or more communicative functions that characterize the type of communicative action which the participants are performing; these functions carry an essential part of the meaning of dialogue utterances. An adequate characterization of this aspect of meaning requires a coherent system of well-defined communicative functions. This standard provides empirically as well as theoretically well-motivated concepts for defining communicative functions, for identifying dimensions of interaction that dialogue acts may address, and for functional dialogue segmentation. The standard specifies data categories for a set of core communicative functions for multidimensional dialogue act annotation, starting from proposals made jointly by the LIRICS project and the TC 37/SC 4 ad-hoc Thematic Domain Group (TDG) 3 on Semantic Content.

2 Normative references

For this international standard there are three main normative references:

- ISO 8879: 1986 (SGML) as extended by TC2 (ISO/IEC JTC 1/SC34 N029: 1998-12-06).
- ISO 19757-2, Document Schema Definition Language, part 2.
- TEI-ISO 24610-1:2006 Language resource management: Feature structures, Part 1: Feature structure representation.

The first reference allows the use of XML as a markup language for semantic annotation; the second the use of Relax NG for designing XML representations of annotation structures; and the third the use of features structures to represent annotation structures.

3 Terms and definitions

For the purposes of ISO 24617-2, the following terms and definitions apply.

3.1 addressee
dialogue (3.6) **participant** (3.14) oriented to by the **speaker** in a manner to suggest that his¹⁾ utterances are particularly intended for him, and that some response is therefore anticipated from him/her, more so than from the other participants

NOTE Source: Goffman (1981).²⁾

3.2 allo-feedback act
feedback act (3.9) where the **sender** (3.19) elicits information about the **addressee's** (3.1) processing of what the sender contributed to the dialogue, or where the **sender** (3.19) provides information about his perceived processing by the **addressee** (3.1) of what was contributed before to the dialogue

NOTE The terms 'allo-feedback' and '**auto-feedback**' (3.3) have their origin in the Greek words 'allos' and 'autos', meaning 'other' and 'self', respectively, referring to whose processing is considered combined with 'feedback'.

3.3 auto-feedback act
feedback act (3.9) where the **sender** (3.19) provides information about the sender's own processing of what was contributed before to the dialogue

1) Throughout this document grammatically unmarked pronouns like "he" and "his" as referring in a gender-neutral way,

2) This definition is a *de facto* standard in the linguistic literature.

3.4**communicative function**

property of a **dialogue act** (3.7) , specifying how the act's **semantic content** (3.17) changes the **addressee's** (3.1) **information state** (3.13) upon successful performance of the **dialogue act** (3.7)

3.5**context**

synonym for **information state**

3.6**dialogue**

exchange of utterances (3.23) between two or more **participants** (3.14)

3.7**dialogue act**

communicative activity of a **participant** (3.14) in **dialogue** (3.6) interpreted as having a certain **communicative function** (3.4) and **semantic content** (3.17), and possibly also having certain **functional dependence relations** (3.11), **rhetorical relations** (3.16) and **feedback dependence relations** (3.10)

3.8**dimension**

type of communicative activity in **dialogue** (3.6) which can be performed by **dialogue acts** (3.7) which, upon successful performance, cause a particular type of **information state** (3.13) change

3.9**feedback act**

dialogue act (3.7) which provides or elicits information about the **sender's** (3.19) or the **addressee's** (3.1) processing of something that was uttered in the dialogue

3.10**feedback dependence relation**

relation between a **feedback act** (3.9) and the stretch of communicative behaviour whose processing the act provides or elicits information about

3.11**functional dependence relation**

relation between a **dialogue act** (3.7) which depends semantically on a previous **dialogue act** (3.7) and the previous act that it depends on

EXAMPLE The relation between an answer and the corresponding question; or between the acceptance of an offer and the corresponding offer.

3.12**functional segment**

minimal stretch of communicative behaviour that has one or more **communicative functions** (3.4)

3.13**information state**

the totality of a **dialogue** (3.6) **participant's** beliefs, assumptions, expectations, goals, preferences, hopes, and other attitudes that may influence the **participant's** (3.14) interpretation and generation of communicative behaviour

NOTE In this document the terms **context** and **context model** are considered as synonyms of **information state**.

3.14**participant**

person or artificial agent involved in **dialogue** (3.6)

3.15**qualifier**

predicate that can be associated with a **communicative function** (3.4)

3.16**rhetorical relation**

relation between two **dialogue acts** (3.7), indicating a pragmatic connection between the two

NOTE Relations such as *elaboration*, *explanation*, *justification*, and *concession* have been studied extensively in the analysis of (monologue) text, where they are often called 'rhetorical relations' or 'discourse relations', and are mostly viewed either as relations between text segments or as relations between events described in text segments. Many of these relations also occur in dialogue as relations between dialogue acts. See e.g. Hovy & Maier, 1993; Mann & Thompson, 1988.

3.17**semantic content (of a dialogue act)**

information, situation, action, event, or objects that the **sender** (3.19) of a **dialogue act** (3.7) wants to bring to the attention of the other dialogue **participants** (3.14)

3.18**semantic content category**

semantic content type

kind of information, situation, action, event, or objects that form the **semantic content of a dialogue act** (3.17)

EXAMPLE Task-specific actions and information; information about the processing of what was said before; the allocation of the speaker role.

3.19**sender**

dialogue (3.7) **participant** (3.14) who produces a **dialogue act** (3.7)

3.20**speaker**

sender (3.19) of a **dialogue act** (3.7) in the form of speech, possibly combined with nonverbal communicative behaviour

NOTE A dialogue participant may say something while another participant occupies the **speaker role** (3.21), therefore the term 'speaker' is not synonymous with 'participant who occupies speaker role'.

3.21**speaker role**

role occupied by a **dialogue** (3.7) **participant** (3.14) who has temporary control of the **dialogue** and speaks for some period of time

NOTE Source: DAMSL Revised Manual.

3.22**turn**

stretch of communicative activity produced by one **participant** (3.14) who occupies the **speaker role** (3.21), bounded by periods where another **participant** (3.14) occupies the **speaker role** (3.21)

3.23**utterance**

anything said, written, keyed, gesticulated, or otherwise expressed by a **dialogue** (3.7) **participant** (3.14)

NOTE An utterance is mostly part of what a sender contributes in a turn. See also Clause 8.

4 Purpose and justification

The notion of a dialogue act plays a key role in the construction of annotated dialogue corpora and in the design of spoken dialogue systems and embodied conversational agents. This document proposes an international standard for annotating dialogues with dialogue act information, as part of a broader effort to support the creation of interoperable language resources annotated with semantic information, and thereby support the development of language-based interactive systems.

Over the years a number of dialogue act annotation schemes have been developed, such as those of the TRAINS project in the US (Allen et al., 1994), the Map Task studies in the UK (Carletta et al., 1996), and the Verbmobil project in Germany (Alexandersson et al., 1998). These schemes were all designed for a specific purpose and a specific application domain; they contain overlapping sets of communicative functions and make use of often mutually inconsistent terminology. In the 1990s a group of dialogue researchers came together as the Discourse Research Initiative, and drafted the general-purpose schema for multidimensional dialogue act annotation called DAMSL: Dialogue Act Markup using Several Layers (Allen and Core, 1997; Core et al., 1998). With its focus on multidimensionality and domain-independence, this represented an important step forward in dialogue act annotation. Several variations and extensions of the DAMSL schema have been constructed for specific purposes, such as Switchboard-DAMSL (Jurafsky et al., 1997) and COCONUT (Di Eugenio et al., 1998), and have been used to annotate dialogue corpora. The comprehensive DIT⁺⁺ scheme (Bunt, 2006; 2009) combines the multidimensional DIT schema, developed earlier (Bunt, 1994), with concepts from these and other schemes.

Preparatory studies in ISO TC 37/SC 4 have indicated that the area of dialogue act annotation is sufficiently mature for a new and sustained effort to design a comprehensive general framework for dialogue act annotation, and have sparked off the EU-funded LIRICS project. In this project a set of core dialogue acts from the DIT⁺⁺ taxonomy has been defined in the form of data categories following ISO standard 12620 (LIRICS D4.3, 2006). These data categories have been tested for their usability and coverage in the manual annotation of a multilingual test suite (LIRICS D4.4., 2007) and form an important part of the background of the present proposal.

The standard described in this document defines a number of domain-independent core concepts for dialogue act annotation plus a formal language for expressing such annotations, and is 'open' in the sense that it also provides guidelines and general principles for extending the set of core concepts, for example with domain-specific concepts, and for selecting coherent subsets of core concepts.

5 Basic concepts and metamodel

The term 'dialogue act' is often used rather loosely in the sense of 'speech act used in dialogue'. Indeed, the idea of interpreting communicative behaviour in terms of actions, such as questions, promises, and requests goes back to speech act theory (Austin, 1962; Searle, 1969). But where speech act theory is primarily an action-based approach to meaning within the philosophy of language, dialogue act theory is an empirically-based approach to the computational modeling of communication, in particular of linguistic and nonverbal communicative behaviour in dialogue.

Describing communicative behaviour in terms of dialogue acts is a way of characterizing the meaning of the behaviour. Informally speaking, dialogue acts are such actions as providing information, requesting the performance of a certain action, apologizing for a misunderstanding, and providing feedback. More formally, dialogue acts can be viewed as corresponding to update operations on the information states of understanding participants in the dialogue; this approach is commonly known as the 'information-state update' or 'context-change approach' to the analysis of dialogue – see e.g. Bunt (2000); Traum & Larsson (2003). For instance, when an addressee understands the utterance *Do you know what time it is?* as a question about the time, then the addressee's information state is updated to contain (among other things) the information that the speaker does not know what time it is and would like to know that. If, by contrast, an addressee understands that the speaker used the utterance to reproach the addressee for being late, then the addressee's information state is updated to include (among other things) the information that the speaker *does* know what time it is. Distinctions such as that between a question and a reproach concern the *communicative function* of a dialogue act, which is one of its two main components. The other main component is its *semantic content*, which describes the objects, prop-

erties, relations, actions and events that the dialogue act is about. The communicative function of a dialogue act specifies how an addressee should update his/her information state with the information expressed in the semantic content, when (s)he understands the speaker's utterance.

We define a dialogue act here as *a semantic unit of communicative behaviour in dialogue, which has a certain communicative function (possibly more than one) and semantic content*. Dialogue act *annotation* is the activity of marking up stretches of dialogue with information about the dialogue acts which are performed, and is often limited to marking up communicative functions.

A dialogue act being a semantic unit in communicative behaviour, the question arises what stretches of communicative behaviour are considered as corresponding to dialogue acts. Spoken dialogues are traditionally segmented into *turns*, defined as stretches of communicative behaviour produced by one speaker, bounded by periods of inactivity of that speaker. Turns in this sense can be quite long and complex, and are for most purposes too coarse as the stretches of behaviour to assign communicative functions to. Communicative functions can be assigned more accurately to smaller units, which called *functional segments*, and which are defined as the functionally relevant minimal stretches of communicative behaviour. See further Clause 8 of this document for more details about dialogue segmentation.

Inherent to the notion of a dialogue act is that there is an agent whose communicative behaviour is interpreted, usually called the 'speaker', or 'sender', and one or more agents who are addressed and whose information state the speaker wants to influence; these participants are called 'addressees'. Dialogue studies often focus on two-person dialogues, in which case the dialogue acts have only one addressee, but in multi-party dialogues there is more than one addressee. For natural face-to-face dialogue, where the dialogue acts are often only partially expressed verbally, and sometimes entirely by nonverbal behaviour (such as nodding), it is best to use the term 'sender' for the agent who performs a dialogue act. Besides sender and addressee(s), there may be various types of side-participants who are present but do not or only marginally participate. Clark (1996) distinguishes between 'side-participants', 'bystanders', and 'overhearers', depending on the role that they play in the communicative situation.

Of the two most central aspects of a dialogue act, the communicative function and the semantic content, the former corresponds intuitively to the *type of action* that is performed, and as mentioned above, dialogue act annotation often takes the form of assigning communicative functions to stretches of dialogue. A semantically more complete characterization also provides information about the *type of semantic content*. The DAMSL annotation schema organises the information into four 'layers', called Information Level, Communicative Status, and Forward- and Backward-Looking Functions. Information Level has three possible values: Task, Task Management, and Communication, which indicate whether the semantic content of the dialogue act is concerned with performing the task which underlies the dialogue, or with discussing how to perform the task, or with the communication. This is a 3-way distinction of semantic content categories. The present standard makes a more fine-grained distinction of content type, distinguishing within the category of communication-related information a number of subcategories, such as feedback information, turn allocation information, and topic progression information. These types of semantic content are also called 'dimensions', and are discussed in more detail in Clause 7.

Many dialogue acts are semantically dependent on one or more dialogue acts that occurred earlier in the dialogue. This is for example the case for answers, whose meaning is partly determined by the question which is being answered, and for the acceptance or rejection of offers, suggestions, requests, and apologies. The following example illustrates this, where the meaning of (1.1) clearly depends very much on whether it is an answer to the question (1.2) or to the question (1.3).

- (1) 1. A: I'm expecting Jan, Alex, Claudia, and David, and maybe Olga and Andrei.
 2. B: Do you know who's coming tonight?
 3. B: Which of the project members d'you think will be there?

For dialogue acts which have such a semantic dependence on dialogue acts that occur earlier in the dialogue, the marking up of the links to these 'antecedent' dialogue acts allows the annotation not just to express e.g. that an utterance is an answer, but also to express to which question it is an answer. This type of relation between dialogue acts is called a *functional dependence relation*.

Dialogue acts may also be semantically related through other relations as in the examples shown in (2).

- (2) a. 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
 4. A: so that's one constraint there
- b. 1. S: how do I increase the contrast
 2. E: go to the settings menu
 3. S: yes
 4. E: change the contrast
 5. S: yes
 6. E: and confirm the setting
 7. S: okay

In the first example, from the AMI corpus,³⁾ we see a sequence of four functional segments contributed by the same participant, where the second segment is related to the first through an *Exemplification* relation; the third is related to the first through an *Explanation* relation, and the fourth is related to the preceding three segments through a *Summarization* relation. In the second example, from the DIAMOND corpus,⁴⁾ we see three utterances contributed by participant E which are interrelated through the *Sequencing* relation. Such relations are known as *rhetorical relations*. In the metamodel shown in Figure 1, such relations would occupy the same position as functional dependence relations, and they both constitute a semantic relation between dialogue acts. Functional dependence and rhetorical relations can thus be viewed as forming one relational semantic category. In view of the wide diversity of proposed sets of rhetorical relations (see e.g. Mann and Thompson, 1988; Hovy and Maier, 1993; Sanders et al., 1992), this standard does not propose any specific set of such relations, but only provides a semantic category where a particular set of such relations may be specified.

Feedback-providing and eliciting acts relate to what happened earlier in the dialogue in a slightly different way. Feedback acts are concerned with the *processing* of what was said before - such as its perception or its interpretation. The following examples illustrate this.

- (3) 1. A: Is this flight also available on Thursday?
 2a. B: Yes, it's available on Thursday as well.
 2b. B: On Thursday you said?

B's utterance 2a is used to give an answer to the question expressed in 1; its meaning depends on that of the question, and it responds to the *dialogue act* expressed in utterance 1; this is another example of a functional dependence relation. With utterance 2b, by contrast, B checks whether he understood correctly what A said, and as such this feedback act refers to the *utterance*, rather than to its interpretation. This type of dependence relation is called a *feedback dependence relation*.

Note that nonverbal feedback, for instance in the form of nodding, and the use of backchannels like *m-hm*, may have a feedback dependence relation to what is being said at that moment, rather than to what was previously said. This is also the case for speech editing acts like self-corrections (*on Tuesday I mean Thursday*) and completions of what the partner is trying to say.

Example (1) also illustrates another phenomenon that is frequently found in dialogue, namely that speakers may have incomplete or uncertain information. The use of *maybe* in (1.1) expresses that A is uncertain about part of the information that he provides. An information-providing dialogue act may also be *incomplete*, as example (17) in Clause 10 illustrates. In addition, speakers may express an emotion towards the information or event that is being discussed, as in (4.2), or express a reservation in the form of a condition, as in (4.3), where an offer is conditionally accepted:

3) <http://corpus.amiproject.org>

4) <http://ls0143.uvt.nl/diamond>

- (4) 1. A: Would you like to have some coffee?
- 2. B: That would be wonderful, thank you!
- 2. B: Only if you have it ready.

For the annotation of conditions, emotions, (un-) certainty, and (in-) completeness, this standard makes use of so-called *functional qualifiers*, which can be attached to communicative functions. (See Clause 10.3 for more detail.)

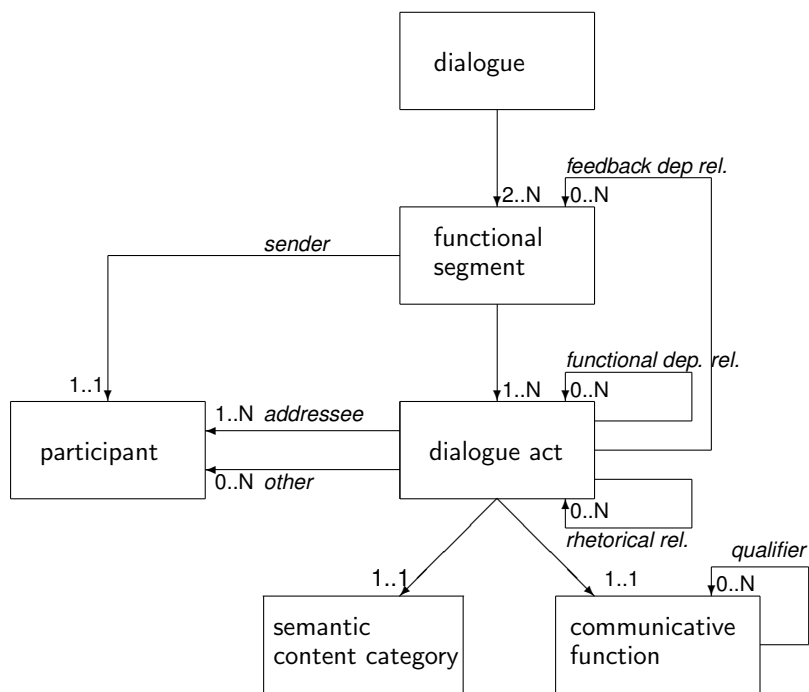


Figure 1 — Metamodel for dialogue act annotation.

The above characterization of the notion of a dialogue act makes use of the following key concepts, which form the backbone of the metamodel for dialogue act annotation:

- sender
- addressee
- participants in other roles (side-participants)
- functional segment
- dialogue act
- communicative function
- communicative function qualifier

- semantic content category
- functional dependence relation
- rhetorical relation
- feedback dependence relation
- function qualifier

The metamodel in Figure 1 shows how these concepts are related. A dialogue consists of two or more functional segments, as indicated by '2..N' at the head of the arrow relating them. Each functional segment is related to one or more dialogue acts, reflecting the possible multifunctionality of functional segments. Each dialogue act has exactly one sender, one or more addressees, and possibly other participants ('0..N'). It has a semantic content of a certain type, and one communicative function, which may have any number of function qualifiers; and is possibly related to other dialogue acts through functional dependence and rhetorical relations, and to functional segments through feedback dependence relations.

6 Approaches to the definition of communicative functions

Existing dialogue act annotation schemas use either one of the following two approaches to defining communicative functions, or a combination of the two: (1) in terms of the effects on addressees intended by the sender; (2) in terms of properties of the signals that are used. For example, questions, invitations, confirmations, and promises are nearly always defined in terms of sender intentions, while repetitions, hesitations, and dialogue opening and closing are typically defined by their form.

Defining a communicative function by its linguistic form has the advantage that its recognition can be straightforward, but has to face the problem that the same linguistic form can often be used to express different communicative functions. For example, the utterance *Why don't you start?* has the form of a question, and can be intended as such, but it can also be used to invite somebody to start. Similarly for so-called 'declarative questions' (questions in the form of a declarative sentence), like *You're going home tomorrow*, which are intended as questions although they look like statements.

Form-based definitions of communicative function also run the risk of being purely descriptive, rather than semantic. For example, when a speaker repeats something that was said before, this behaviour may be characterized as a 'repetition act'; however, that would only say something about the *form* of the behaviour, nothing about its communicative function. A repetition is for instance often used to check correct understanding, but it can also have different functions, as in the following example where it is used as a confirmation in response to a check question:

- (5) 1. S: there are evening flights at 7.15 and 8.30
 2. C: and that's on Sunday too
 3. S: and that's on Sunday too

We take a strictly semantic approach to the definition of communicative functions. But while we do not take linguistic form to be part of the definition of a communicative function, we do insist that for every communicative function there are ways in which a sender can indicate that his behaviour should be understood as having that particular function, shaping his (linguistic and/or nonverbal) behaviour so as to have certain observable features which are indicative for that function in the context in which the behaviour occurs. This requirement puts all communicative functions on an empirical basis.

A particular case where form and function are not related in a straightforward way is that of indirect speech acts, where a speaker uses a linguistic form that is standardly used to express one type of dialogue act, but in context means something else. This phenomenon has been studied extensively in linguistic pragmatics (e.g., Searle, 1975; Bach & Harnish, 1979). Questions of the form *Do you know [X]* provide an illustration: while an utterance of this form standardly would seem to ask whether the addressees possess the knowledge [X], it is often used

to request the addressees to provide the information [X], if possible. This makes such a question a conditional request.

The DIT⁺⁺ taxonomy of communicative functions (Bunt 2004; Bunt & Girard, 2005; <http://dit.uvt.nl>) views indirect requests as having a communicative function which is slightly different from that of the corresponding direct form, because their performance is thought to have slightly different effects on information states. For example, the difference between *Where is Lee's office?* (SetQuestion) and *Do you know where Lee's office is?* (IndirectSetQuestion) would be that in the indirect version the speaker does not express an expectation that the addressee knows the answer to his question, whereas in the direct version he does. The full complexity of the phenomenon of indirect speech acts is beyond the scope of this ISO standard, but an important class of indirect speech acts can be covered by qualifying them as conditional - see Section 10.3 and Petukhova & Bunt (2010).

Successful communication depends on addressees understanding the communicative functions of the sender's utterances. These functions are inferred from the properties of the sender's communicative behaviour together with a model of the dialogue context. Such a model includes assumptions about each other's beliefs and goals, as well as knowledge of the dialogue history and of the activity which motivates the dialogue. In spoken dialogue it is in general not possible to recognize communicative functions on the basis of linguistic form only, since virtually every linguistic form can be used with different functions.

A general dialogue act annotation schema has more value when it supports both manual and automatic annotation. It should therefore contain concepts with a depth and granularity that matches human understanding of the functions of dialogue utterances, and it should also contain concepts that are suitable for a more surface-oriented form of annotation that relies less on deep understanding. This standard therefore makes use of hierarchies of communicative functions, where functions deeper down in a hierarchy are more fine-grained than those higher up, and function qualifiers, which add further detail to a base communicative function.

7 Annotation schemas

7.1 Schema structure

Existing dialogue act annotation schemes differ in their sets of tags, but more importantly with respect to (1) the underlying approach to dialogue modeling; (2) the definitions of the basic concepts; (3) the coverage of aspects of interaction; and (4) the level of granularity of the tag set. They can be divided into onedimensional and multidimensional ones. Onedimensional schemes have a set of mutually exclusive tags, and are used for coding stretches of dialogue with a single tag. Their tag sets are often quite small, as in the LINLIN schema (Ahrenberg et al., 1995) and in the HCRC schema (Carletta et al., 1996). The simplicity of these tag sets is often considered to make them more reliable and to take less effort to apply consistently by annotators; on the other hand, one-dimensional annotation schemes are also known to have serious limitations (see e.g. Klein et al., 1998; Larsson, 1998; Popescu-Belis, 2005).

Multidimensional schemes are intended for encoding stretches of dialogue with multiple tags, doing justice to the multifunctionality of natural dialogue behaviour. Such schemes typically have a relatively large tag set. There are several advantages to the structuring of such a tag set into clusters of communicative functions:

- Clustering semantically related tags improves the transparency of the tag set, as each cluster is concerned with a certain kind of information. This also makes the coverage of the tag set clearer, since each cluster typically corresponds to a certain class of dialogue phenomena.
- A structured tag set can be searched more systematically and more 'semantically' (i.e. on the basis of semantic differences and similarities) than an unstructured one.
- Many of the tags within a cluster are usually mutually exclusive (such as 'signal understanding' and 'signal non-understanding'); this has the advantage that an annotator (human or machine), once having chosen a tag within a cluster, does not need to consider the rest of that cluster any further. If a cluster is hierarchically organised, as is the case in the present standard, with finer-grained functions being dominated by less fine-grained ones (such as 'confirmation' being more fine-grained than 'answer'), then the most sensible use of these tags is to choose the most specific tag for which there is sufficient evidence.

7.2 Multidimensionality and multifunctionality

Participation in a dialogue involves several activities beyond those strictly related to performing the task or activity for which the dialogue is instrumental. In natural conversation, the participants among other things constantly “evaluate whether and how they can (and/or wish to) continue, perceive, understand and react to each other’s intentions” (Allwood, 1997). They share information about the processing of each other’s messages, elicit and provide feedback, and manage the use of time, turn allocation, contact and attention, and various other aspects. Communication is thus a complex, multi-faceted activity, and for this reason dialogue utterances are often multifunctional.

Multifunctionality comes in a variety of forms. Allwood (1992) made a distinction between sequential and simultaneous multifunctionality, illustrated by the following example:

- (6) A: Yes! Come tomorrow. Go to the church. Bill will be there. OK?
B: The church, OK.

Sequential multifunctionality occurs when a turn has several parts which each have a different communicative function. (If they have the same function, then it may be wise to apply the ‘maximum functional-segment principle’ (Larsson, 1998) and avoid splitting up the markable into smaller parts.) In the example we see A’s utterance containing five functional segments with respectively the communicative functions *feedback giving*, *request*, *request*, *statement*, and *response elicitation*. The occurrence of sequential multifunctionality depends on the way in which a dialogue is segmented (see also Clause 8), and disappears when sufficiently small segments of dialogue behaviour are considered as markables.

Simultaneous multifunctionality, by contrast, persists even when minimal segments are used as markables. The following example illustrates this:

- (7) 1. A: Do you know what date it is?
2. B: Today is the fifteenth.
3. A: Thank you.

In (7.3), A’s utterance has the function of thanking, and will mostly be taken to imply that A has understood and accepted the information in (7.2) - i.e., as having a positive feedback function. But “Thank you” does not *always* express positive feedback; a speaker who finds himself in a rather unsuccessful dialogue may just want to terminate the interaction in a polite way. The feedback function of the thanking behaviour in example (7) can be inferred along the following lines: By saying *Thank you*, A thanks B, so there must be something that A is thankful for. This can only be what B just said, and that can only constitute a reason for thankfulness if A considers B’s utterance as relevant and useful, which means that A accepted B’s utterance as an answer to his question, which in turn implies that A believes that B understood that question. The feedback function in such a case can be viewed as a conversational implicature (Grice, 1979), i.e. as a contextually plausible consequence.

The relation between thanking and giving positive feedback is different from that between a propositional answer (‘yes’ or ‘no’) and a confirmation – in this case the relation is one of *entailment*, since every confirmation by its very nature is also an answer. Entailment relations occur in an annotation schema when the definition of one communicative function is a special case of that of another. It is not obvious that such cases should be considered as instances of multifunctionality; a speaker who wants to issue a confirmation can hardly have the intention to *additionally* also give an answer, since the recognition of that intention is already part of the recognition of a confirmation.

There are also cases of multifunctionality where the different functions do not have any logical relation. This is for example the case for turn-initial hesitations, as in the following example:

- (8) 1. A: Is that your opinion too, Bert?
2. B: Ehm,.. well,... I guess so.

In the first turn of (8), speaker A asks a question to B and assigns the turn to B (by the combined use of B's name, the intonation, and by looking at B). In (8.2) B performs a stalling act in order to buy some time for deciding what to say; the fact that he starts speaking without waiting until he has made up his mind about his answer indicates that he accepts the turn, which was given to him. So the segment *Ehm,... well,...* has both a stalling function and a turn-accepting function. Note, incidentally, that A's utterance is also multifunctional: it asks a question about B's opinion and it assigns the turn to B.

A dialogue act annotation schema can reflect the multifunctionality of dialogue utterances in two (non-exclusive) ways: (1) in the structuring of the tag set into clusters or 'dimensions' of communicative functions; (2) in accompanying instructions to annotators for how to apply multiple tags taken from an unstructured tag set. If the tag set is fairly extended and does not have any structure, it is next to impossible to formulate good instructions for how to use the tags in multiple tagging, since there is no easy way to refer to groups of tags. Therefore, the recognition that utterances in dialogue tend to be multifunctional naturally leads to the introduction of dimensions in a dialogue annotation schema.

7.3 Multidimensionality, clustering, and dimensions

The clusters of communicative functions that can be found in existing annotation schemes are typically chosen on the basis of conceptual similarity of the constituent functions. An early version of the DIT schema, for example, has a cluster of 'information-seeking functions' for a range of question types, and a cluster of 'information-providing' functions for various kinds of informs and answers (Bunt, 1989).

The DAMSL schema (Core & Allen, 1997) is organized into 'layers' and 'dimensions'. Four layers are distinguished: Communicative Status, Information Level, and Forward Looking and Backward Looking Communicative Functions (FLF and BLF); the latter two are indeed clusters of communicative functions (the tags in the other layers are concerned with other kinds of information). The FLF cluster is subdivided into five clusters, including the classes of commissive and directive functions, well known from speech act theory. The BLF cluster has four subclasses: Agreement, Understanding, Answer, and Information Relation. Core & Allen (1997) refer to these nine subclasses as 'dimensions'. While the DAMSL documentation does not discuss or motivate the choice of layers and dimensions, these are clearly useful for introducing structure in the tag set in a way that can help annotators to make their choices, supported by useful annotation guidelines. The dimensions within the FLF and BLF clusters make DAMSL a nine-dimensional schema.

Popescu-Belis (2005) mentions six aspects of utterance function as relevant for choosing dimensions: (1) the traditional clustering of illocutionary forces in speech act theory into Representatives, Commissives, Directives, Expressives and Declarations; (2) turn management; (3) adjacency pairs; (4) topical organization in conversation; (5) politeness functions; and (6) rhetorical roles.

Bunt (2005; 2006) proposes to structure a multidimensional tag set by basing the notion of dimension on the observation that participation in a dialogue involves a range of communicative activities beyond those strictly related to performing the task that underlies the dialogue. Dialogue participants share information not only about the task that is pursued with the help of the dialogue, but also about the processing of each other's messages, about the allocation of turns, about contact and attention, about the use of time, and about various other aspects of the interaction. They thus perform communicative activities of various types, such as giving and eliciting feedback, taking turns, stalling for time, establishing contact, and showing attention. Each of these types of activity is concerned with a different category of information. In this standard, we use the term 'dimension' to refer to these various semantic content categories or to the communicative activities concerned with these information categories. This leads to dimensions such as feedback, turn management, time management, and contact management, besides the dimension formed by the task that motivates the dialogue. In Clause 9, the set of core dimensions is described which forms part of this standard.

7.4 Dimension-specific and general-purpose functions

Using the notion of 'dimension' as defined in this document, not every cluster of related communicative functions qualifies as a dimension. For example, the cluster that can be formed from the various kinds of information-seeking acts (such as Yes/No-questions, WH-questions, check questions, and menu questions) does not constitute a dimension, since questions can be concerned with any aspect of communication, be it the underlying task, feedback, task progression, change of topic, or contact. The same is true of the cluster of information-

giving acts (statements, warnings, answers, confirmations, and so on), and of the commissive and directive acts (request, suggest, instruct, offer, promise, and so on). These clusters of functions fail to qualify as dimensions. The functions in these clusters can be combined with semantic content relating to any dimension, and are therefore called *general-purpose communicative functions*. When combined with a semantic content of a certain kind, they form a dialogue act addressing the dimension corresponding to that kind of content. In that sense, the general-purpose functions could be said to belong to *every* dimension. These functions are discussed further in Clause 10.1; Table 1 provides examples of general-purpose functions used in some of the dimensions distinguished in the LIRICS annotation scheme.

<i>Communicative function</i>	<i>Dimension</i>	<i>Example expression</i>
Propositional Question	Task/Activity	<i>is there an earlier possibility?</i>
Set Question	Task/Activity	<i>what time does the meeting start?</i>
Check Question	Auto-Feedback	<i>so you want to leave at eight o'clock in the morning?</i>
Disconfirm	Allo-Feedback	<i>eight o'clock in the evening.</i>
Inform	Social Obligations Man.	<i>I'm very grateful for your help.</i>
Confirm	Allo-Feedback	<i>slightly yeah, very slightly</i>
Offer	Discourse Structuring	<i>would you like me to repeat the connection one more time?</i>
Decline Offer	Discourse Structuring	<i>no thank you</i>
Request	Turn Management	<i>Peter?</i>
Accept Request	Turn Management	<i>yes, I would like to say something at this point</i>
Request	Time Management	<i>Please give me a minute to look that up</i>
Instruct	Task/Activity	<i>we're going to turn east</i>

Table 1: Examples of general-purpose communicative functions and their expression for some of the dimensions distinguished in the LIRICS annotation scheme.

In contrast with the general-purpose communicative functions, other functions can only be used to address a specific dimension, such as Turn Keep and Turn Release which are specific for the dimension of Turn Management; and Stalling and Pause for the dimension of Time Management. Table 2 shows examples of dimension-specific communicative functions in some of the dimensions of the LIRICS annotation scheme; this class of functions is discussed in more detail in Clause 10.2.

<i>Dimension</i>	<i>Communicative function</i>	<i>Example expression</i>
Auto-Feedback	Perception Negative	<i>Huh?</i>
	Evaluation Positive	<i>True.</i>
	Overall Positive	<i>OK.</i>
Turn Management	Turn Keeping	final intonational rise
	Turn Grabbing	hold gesture with hand
	Turn Giving	<i>Yes.</i>
Time Management	Stalling	slowing down speech; fillers
	Pausing	<i>Just a minute</i>
Contact Management	Contact Checking	<i>Hello?</i>
Discourse Structuring	Interaction Structuring	<i>I repeat:</i>
Social Obligations Management	Apology	<i>I'm sorry.</i>
	Greeting	<i>Hello.</i>
	Thanking	<i>Good morning</i> <i>Thanks.</i>

Table 2: Examples of dimension-specific communicative functions and their expression for some of the dimensions distinguished in the LIRICS annotation scheme.

8 Dialogue segmentation

Many studies in the annotation of dialogue with communicative functions have assumed the dialogue to be segmented at the level of turns or utterances. Turns are for many purposes too coarse-grained as the units to which to assign communicative functions, since they commonly contain smaller parts that have separate communicative functions; example (6) illustrates this. These smaller parts are often called 'utterances'.

Utterances are linguistically defined contiguous stretches of (linguistic) behaviour, like phrases or clauses. Being more fine-grained than turns, the use of utterances as units in dialogue allows for more precise annotation; however, the notion of an utterance does not have a clear definition. Syntactic and prosodic features are often used as indicators of utterance endings, but the detection of utterance boundaries on syntactic or prosodic grounds cannot be done reliably (see e.g. Shriberg et al., 1998; Stolcke et al., 2000; Nöth et al., 2002). In the case of nonverbal or multimodal communication, the notion of an utterance as a *linguistically* defined unit is even more problematic, so it is better not to rely on linguistic properties in the definition of functional units in dialogue.

The stretches of behaviour that are relevant for dialogue act annotation may be discontinuous, may overlap, and may contain parts from more than one turn. They therefore do not always correspond to linguistically defined utterances. For these reasons the notion of a *functional segment* has been introduced as a *minimal stretch of communicative behaviour that has a communicative function* (and possibly more than one).

Example (9) shows that a functional segment may be discontinuous:

- (9) A: Do you know what time the next train leaves?
B: The next train is ... *let me see...* at 7.48.

The segment *The next train is at 7.48*, which answers the preceding question, is interrupted by ... *let me see...* which expresses that the speaker cannot answer immediately, but needs a little time (a ‘Stalling’ act). As a result, the stretch of communicative behaviour that expresses the answer is discontinuous.

The following example illustrates that dialogue acts may also be expressed by *overlapping* stretches of communicative behaviour:

- (10) U: What time is *the first train to the airport on Sunday morning, please?*
S: *The first train to the airport on Sunday morning* is at 06:25.

In this example, S’s response as a whole is an answer to U’s question, and the repeated question part *The first train to the airport on Sunday* can be viewed as expressing positive feedback, displaying S’s understanding of U’s question. So the answer act and the feedback act are expressed by overlapping functional segments.

Example (11) shows that a dialogue act may spread over multiple turns. A asks a question, the answer to which consists of a list of items which B communicates one by one.

- (11) A: Could you tell me what departure times there are for flights to Frankfurt on Saturday?
B: Certainly. There’s a Lufthansa flight in the morning leaving at 08:15,
A: yes,
B: and a KLM flight at 08:50,
A: yes,
B: and a Garuda flight at 10:30,
A: yes,
B: ...

The phenomenon that dialogue acts may be discontinuous and overlap closely relates to the multifunctionality of dialogue behaviour discussed in 7.2. In order to account for the multifunctionality of dialogue behaviour, communicative functions can be assigned to all those segments of behaviour that correspond to a dialogue act, allowing these segments to overlap and to be discontinuous and to spread over multiple turns. For example, consider the 3-way segmentation of S’s utterance in (10).

- (12) 1. U: What time is the first train to the airport on Sunday morning please?
2. S: The first train to the airport on Sunday morning is *let me see...* at 5:45.
TA **The first train to the airport on Sunday morning is** *let me see...* **at 5:45**
AutoFB **The first train to the airport on Sunday morning ; is** *let me see...* at 5:45
TiM The first train to the airport on Sunday morning is **let me see...** at 5:45

In the Task/Activity (TA) dimension, the turn is segmented into the discontinuous functional segment *The first train to the airport on Sunday morning is at 5:45*, which has the function of an answer in this dimension, and the intervening stretch *... let me see...*, which does not have a communicative function in this dimension.

In the Time Management (TiM) dimension the same segmentation applies, but now it's only the segment *... let me see...* which has a communicative function (Stalling).

In the Auto-Feedback (AutoFB) dimension the turn is segmented into the functional segment *The first train to the airport on Sunday morning*, which provides positive feedback on understanding the preceding question, and the contiguous stretch *is ... let me see... at 5:45*, which is not a functional segment.

Segments of verbal behaviour have a natural delineation in terms of the constituent words in their transcription; this is different for nonverbal communicative behaviour, where the notion of a constituent is far from obvious; still, each of the various forms of nonverbal behaviour (hand gestures, head gestures, facial expressions, etc.) do have their own morphology which can be used to characterize their instances (see e.g. Kendon, 2004; McNeill, 2005), and also have a beginning and an end. The definition of a functional segment as "minimal stretch of communicative behaviour that has a communicative function" therefore applies not only to verbal behaviour but also to nonverbal communicative behaviour.

In multimodal dialogue, participants combine different modalities to form multimodal segments of behaviour which have a communicative meaning. In such situations a functional segment has several modality-specific components, such as a stretch of speech, a facial expression, and accompanying head gestures.

9 Core dimensions and dialogue acts

'Core dimensions' are those dimensions whose relevance does not depend on the domain of application. In order to identify such dimensions, Petukhova & Bunt (2009b,c) formulate and test a number of criteria that a core dimension should satisfy.

First, only dimensions should be considered which can be distinguished according to empirically observed behaviour in dialogue. This places the notion of a dimension on an empirical basis.

Second, each dimension should be theoretically justified, corresponding to well-studied and investigated communicative activities that dialogue participants perform, such as turn taking and feedback.

Third, each dimension should be recognizable with acceptable precision by human analysts, in particular human annotators, as well as by automatic annotation dialogue understanding and dialogue annotation systems. Recognizability by human and machine annotators is important in order to make the schema useful for the purpose of annotation, as discussed in more detail in the next two sections. Recognizability by dialogue understanding systems is important because dimensions correspond to the kind of semantic information which is addressed by a dialogue act, which is relevant for determining what aspects of the system's information state should be updated.

A fourth criterion, which applies not so much to the choice of individual dimensions, but rather to the choice of a useful *set of dimensions*, is that of the *independence* (or 'orthogonality') of the set. This criterion requires that each dimension in a multidimensional system can be addressed by dialogue acts independent from addressing other dimensions. More precisely, for every dimension D_i there should be forms of communicative behaviour which express a dialogue act that is concerned with information of the kind that is characteristic for D_i , without also expressing a dialogue act addressing one of the other dimensions. In other words, each dimension is *separately addressable* by dialogue acts.

Finally, a fifth consideration specifically applies to the design of a multidimensional *standard* annotation schema, requiring that only dimensions should be included which are commonly present in existing dialogue act annotation schemes. This is a more practical consideration, making explicit that an annotation standard should capitalize on what is already present in existing good practices.

In sum, the following criteria and considerations can help to make a well-motivated choice of the dimensions in

a multidimensional dialogue act annotation schema:

- (13) Each dimension in a dialogue act annotation schema should be:
- a) theoretically justified, in the sense of forming a well-established and well-studied aspect of communication;
 - b) empirically observed in the functions of dialogue utterances;
 - c) addressable independently of the other dimensions.
 - d) recognizable with acceptable precision by human annotators and by automatic annotation systems;
 - e) present in a significant number of existing dialogue act annotation schemes.

In their study, Petukhova and Bunt (2009a) survey the literature and analyse the contents of 18 existing annotation schemes in order to verify the requirements (13a) and (13e) for a range of proposed dimensions. In order to examine the other three requirements, they present the results of annotation experiments and of a range of statistical and machine-learning tests, applied to dialogue corpora of various kind. These tests include empirical data on co-occurrence relations among dialogue acts and dimensions, tests of independent addressability, measures of semantic relatedness, and data on human and machine recognition of dimensions. The main findings are summarized in Annex F.

Their study confirms that the following nine dimensions fulfil all the requirements (13) and qualify as core dimensions in a dialogue act annotation schema.

- 1. Task** Dialogues are usually motivated by goals, tasks, or activities which are non-communicative in nature, such as obtaining certain information, solving a problem, improving relationships, acting in a game as a team mates, and so on. Of the core dimensions, the one that corresponds to communication about the performance of the task/activity motivating the dialogue, or the task/activity domain is called the Task dimension.
- 2. Auto-Feedback** The term ‘feedback’ (or more precisely, ‘communicative feedback’) is most often used to refer to the activity of dialogue participants signalling their attention, understanding, and evaluation of what the speaker says. Feedback is an essential aspect of successful communication. Allwood (2000) argues that feedback morphemes and mechanisms, whether they occur as a single utterance or as a part of a large utterance, are probably the most important cohesion device in spoken language. Feedback mechanisms, their linguistic properties, non-verbal expression, durational, temporal and prosodic properties and related phenomena have been studied extensively, e.g. Duncan & Fiske (1977); Allwood et al. (1993); Clark & Krych (2004). Bales (1951) observed that dialogue participants address several levels of processing of the partner’s previous utterances, taking each other into cognitive consideration and showing readiness to communicate, giving attention and receptiveness, recognition, interest and responsiveness to the partner’s contributions. Thus, feedback may be reported on various levels. Allwood et al. (1993), Clark (1996) and Bunt (2000) distinguish several feedback levels: attention (in Allwood (1993) called ‘contact’); perception (in Clark, 1996) called ‘identification’), understanding (in Bunt, 2000 called ‘interpretation’); evaluation (in Clark, 1996, called ‘consideration’ and in Allwood et al., 1993 called ‘attitudinal reaction’), and execution (Bunt, 2000). The term ‘auto-feedback’ is used here in order to make a distinction with ‘allo-feedback’; see next item.⁵⁾
- 3. Allo-Feedback** Dialogue participants do not only discuss and report on their own processing of dialogue utterances (‘auto-feedback’), but they also monitor the attention, perception, understanding and evaluation of the addressees, and pose themselves such questions as: *Is the addressee paying attention? Does the addressee seem to hear what I’m saying? Does the addressee seem to understand what I mean? Does the addressee accept/appreciate what I’m saying?* When appropriate, speakers confirm or correct an addressee’s processing, or elicit information about it (feedback elicitation). This communicative activity, where the speaker elicits or volunteers information about the addressee’s processing of what the speaker has said, is called *allo-feedback*; examples are: *Is this clear enough?, That’s what I meant. I’m afraid you got that wrong.*

5) The terms ‘allo-feedback’ and ‘**auto-feedback**’ (3.3) have their origin in the Greek words ‘allos’ and ‘autos’, meaning ‘other’ and ‘self’, respectively, referring to whose processing is considered combined with ‘feedback’.

4. Turn Management Turn Management acts are concerned with the allocation of the main speaker role, also called the 'floor'. Allwood (1997) defines turn management as the distribution of the right to occupy the sender role in dialogue. He argues that this is rather a normative notion than a behavioural unit. Accordingly, the decision to take the next turn or to offer the next turn to the partner(-s) depends on the speaker's needs, motivations, and beliefs, and on the rights and obligations in a conversational situation.

In dialogues with two or three participants, normally only one participant is speaking at any given moment, while the other participants express their involvement through backchannels (like *m-hm*), nonverbal sounds, and other nonverbal activity. The greater the number of participants in a dialogue, the more one may find different simultaneous speakers (Campbell, 2008) depending very much on the type of interaction. Spontaneous multi-party conversations may be fairly chaotic in this respect; more organized interactive situations, like meetings, tend to have a single participant as the main speaker at any moment.

5. Time Management Fluent speech is relatively rare in spontaneous conversation. Disfluent speech production commonly gives rise to issues of timing: at all the major levels of planning and processing involved in speech production (Levelt, 1989), from retrieving a word to deciding what to talk about next, speakers may experience difficulties which give rise to delays (Clark & Fox Tree, 2002). These delays can be minor, giving rise to (*stalling*) acts, or prolonged (*pausing* acts), where the speaker suspends the dialogue for a while.

6. Discourse Structure Management A dialogue participant may perform a dialogue act in order to give an indication of the intention to close the discussion of a certain topic, or to focus an addressee's attention on a new topic. Such dialogue acts, called *discourse structuring acts* are based on the speaker's view of the state of the underlying task, on the development of a plan that he may have for organising the dialogue, and on assumptions that arise concerning the need to structure the discourse in order for the dialogue to proceed successfully.

7. Social Obligations Management Participating in a dialogue is a social activity, where one is supposed to do certain things and not to do others, and to act in accordance with norms and conventions for social behaviour. Dialogue participants have ethical tasks and obligations, and perform dialogue acts to fulfil these. The golden rule of ethics '*Do unto others what you would have them do unto you*' means in communication '*make it possible for others to be rational, motivated agents*' (Allwood, 2004). Bales (1951) pointed out the importance of social obligation acts such as *acts for giving help and reward*.

Bunt (1996) noticed that social obligation acts are often not just 'social', they are also used for improving the transparency of the dialogue. For example, people greet each other not just in order to be friendly, but also to establish and acknowledge their presence, and they wish each other a good day not only for being nice but also to end a conversation.

8. Own Communication Management A communicative activity which has been studied extensively in human dialogue behaviour as well as in the context of designing spoken dialogue systems, concerns a speaker's monitoring of his speech production. Allwood et al. (2005), introduced the term 'Own Communication Management (OCM)' for describing the communicative activity of a speaker relating to the management, planning, and execution of his speech production. This activity is indispensable in the description of spoken dialogue, and is illustrated by the performance of dialogue acts which are usually called '(self-)repairs', 'restarts', and other speech-editing acts.

9. Partner Communication Management Partner Communication Management (PCM) is concerned with monitoring the partner's speech by the speaker, either providing assistance by completing an utterance that the partner is struggling to complete (*completion*), or correcting (part of) a partner's utterance, believing that partner made a speaking error (*correct-misspeaking*).

This aspect of communication satisfies all criteria for being a core dimension, although it is not recognized in many existing annotation schemes, which may be due to its relatively low frequency in certain types of dialogue.

10 Core dialogue acts

The various annotation schemas for dialogue acts that have been proposed share a number of communicative functions which are of obvious importance in virtually any type of dialogue. These 'core dialogue acts' include

various types of questions, answers, informs, requests, and acknowledgements. Traum and Hinkelman (1992) have used the term ‘core dialogue acts’ to refer to the types of acts that are most familiar from traditional speech act theory. These are often related to the use of performative verbs (such as *promise*, *invite*, and *confirm*) and include the commissive and directive act types (*offer*, *request*, *propose*,...), the ‘reportative’ speech acts used for stating facts (*assert*, *conclude*), and the ‘expressive’ ones for expressing psychological states (*apologize*, *thank*, *congratulate*). In this standard the terms ‘core dialogue act’ and ‘core communicative function’ are used to refer to the types of dialogue acts and their communicative functions that are most commonly found in dialogue and that are not specifically related to particular task domains; the data categories specifying names and definitions of these core communicative functions are part of this standard. These include the most common commissive, directive, and reportative acts known from speech act theory and some of the expressive ones, plus a set of other ones which have not been considered much in speech act theory, such as acts for turn taking and time management.

The choice of communicative functions to be included in a dialogue act annotation schema can be based on similar criteria as the choice of core dimensions. First of all, the criterion of *empirical validity* requires that for every communicative function there are linguistic or nonverbal means which are commonly used by speakers to indicate that their behaviour should be understood as having that function. Second, the criterion of *theoretical validity* requires that every communicative function has a precise definition, which clearly distinguishes it from other functions. In particular, the semantic approach taken in this standard requires precise definitions in terms of intended information state updates.

Another empirical requirement for including a communicative function is that of coverage. For example, the phenomenon that conversational analysts have called ‘adjacency sequences’ means for an annotation schema that if it includes one element of such a pair, then it should preferably also contain the other. For example, a thanking act is often responded to by a ‘downplayer’, and an annotation schema which contains a function tag for encoding thankings should preferably also contain a tag for encoding the responding downplayers.

In order to be appropriate as elements in an annotation standard, two additional requirements, again comparable to requirements for dimensions, are (1) that each communicative function should be recognizable with acceptable precision by humans and preferably also by machines, and (2) that they commonly occur in existing annotation schemas.

Finally, it is advantageous if the set of communicative functions has the property of *semantic connectedness*, which says that any two communicative functions that can be used for addressing a given dimension are either mutually exclusive (i.e., if one of them applies then the other one does not); or one is a specialization of the other. This property has the advantage that an annotator, who has decided that a functional segment has a communicative function in a given dimension D_i , can choose from the set of functions available for D_i the most specific one for which there is sufficient evidence. For example, in (14) B’s utterance provides information to A in response to A’s question, and should therefore be encoded as an information-providing act. This means (see Figure 2) that the choice is between the functions *Inform*, *Agreement*, *Disagreement*, *Correction*, *Answer*, *Confirm*, and *Disconfirm*. The functions *Disagreement*, *Correction* and *Disconfirm* do not apply since there is nothing adversary in what B says. Of the remaining possibilities, *Inform* and *Agreement* are not optimally specific, since they miss the fact that B is responding to a question. Of the two remaining functions, *Confirm* is more specific than *Answer*, and since the expression “*That’s right*” is a sign of confirmation, expressing not only a positive reply but also agreement with A’s expectation (as opposed to “*Yes*”), the appropriate function tag is *Confirm*.

- (14) A: And that’s the first flight tomorrow, right?
B: That’s right.

A multidimensional annotation scheme with orthogonal dimensions and semantically connected sets of communicative functions allows annotators to mark up a functional segment with at most as many functions as there are dimensions.

All in all, the communicative functions included in the present standard satisfy the following six requirements and desiderata:

- (15) Every communicative function is:
1. empirically observed in features of communicative behaviour in dialogue;
 2. theoretically validated as an update operation on information states (i.e. clear semantics);
 3. relevant for obtaining a good coverage of the phenomena in the dimensions considered;
 4. recognizable by humans and machines;
 5. a member of a semantic connected set of functions;
 6. present in a significant number of annotation schemes.

The definition of communicative functions in this standard should be seen in connection with the inclusion of data categories for these concepts in the ISOcat Data Category Registry (DCR) (<http://www.isocat.org>). The definitions of the core dialogue act functions will all be entered in the 'Semantics' profile of the registry, which contains certified data categories for semantic annotation. Additional, optional data categories for communicative functions, and extensions for specific domains or purposes, may in due time also be entered in the ISOcat registry, following ISO certification procedures. This standard includes only small numbers of domain-independent core communicative functions for the various dimensions:

- general-purpose functions:
 - * 5 information-seeking functions;
 - * 7 information-providing functions;
 - * 8 commissive functions;
 - * 6 directive functions.
- dimension-specific functions:
 - * 2 auto-feedback functions;
 - * 3 allo-feedback functions;
 - * 2 time management functions;
 - * 6 turn management functions;
 - * 3 discourse structuring functions;
 - * 3 own communication management functions;
 - * 2 partner communication management functions;
 - * 10 social obligation management functions.

10.1 General-purpose functions

The core general-purpose functions are those domain-independent functions which concern the transfer of information and the discussion of (communicative or other) actions. The information-transfer functions are divided into *information-seeking* functions, where the speaker aims to obtain certain information from the addressee(s), and *information-providing* functions, where the speaker wants to make the addressee(s) aware of certain information. The action-discussion functions fall apart into those where the speaker commits himself to perform certain actions (*commissive* functions), and those where the speaker aims to make the addressee(s) perform certain actions (*directive* functions).

The choice of core communicative functions within each of these four classes is based on an analysis of existing annotation schemas. Tables 10-15 in Annex F provide an overview of the occurrence of general-purpose functions in 12 annotation schemas.

The functions in the information-seeking class are questions of various kinds. Many schemas distinguish several types of question, depending on the type of information that the speaker is looking for and on the speaker's expectations regarding the answer that he will get. These distinctions are supported in natural languages in the distinction of different sentence types. In this standard a distinction is made between *propositional questions*, where the speaker wants to know the truth of a given proposition (also known as 'yes/no questions'); *check*

questions, which are propositional questions where the speaker expects the answer to be positive; *set questions*, where the speaker wants to know which elements of a given set of entities have a certain property (also known as ‘WH-questions’); and *choice questions* (also known as ‘multiple-choice questions’, ‘menu questions’, or ‘alternatives-questions’), where the speaker wants to know which one of a set of listed alternatives applies.

The most obvious case of an *information-providing* function is the *inform*, which also goes by the names *statement* and *assertion*, and which is the function of a dialogue act where the speaker has the aim to bring certain information to the addressee’s attention. More specific cases are the functions *agreement* and *disagreement*, where the speaker believes that the addressee agrees or disagrees, respectively, with the information that is brought to his attention, and the *answer* function, where the speaker provides solicited information. In response to a check question, the speaker may either *confirm* or *disconfirm* the addressee’s expectation.

Important *commissive* functions are *promise* and *offer*, which have in common that the speaker is prepared to commit himself to performing a certain action; the difference is that in the case of a promise this commitment is unconditional, whereas in the case of an offer the commitment will only occur if the addressee accepts the offer.

The prototypical case of a *directive* function is the *instruct*, also known as *command*, where the speaker unconditionally tells the addressee to do something. As in the case of commissives, there is also a conditional directive, namely the *request*, which puts pressure on an addressee to perform the requested action, but does so on the condition that the addressee agrees to so. Note that accepting a request or a suggestion is itself a commissive act, and accepting an offer is a directive act.

While accepting a request implies a commitment to perform the requested action, declining a request can be viewed as a commitment to *not* perform the requested action, and is therefore also a commissive act. Accepting and declining a request are two extremes on a scale of possible responses to a request. In between these two extremes are partially or conditionally accepting a request – see clause 10.3. The communicative function Address Request covers all forms of dealing with a request, with Accept Request and Decline Request as special cases. Similarly for Address Offer and Address Suggestion.

Further subdivisions and more specific types of each of the functions mentioned here can be made; for instance, the DIT⁺⁺ taxonomy distinguishes check questions with a positive and a negative expectation (*posi-check* and *nega-check*). Some taxonomies also distinguish different answer types, such as WH-answer and YN-answer for answers to set questions and propositional questions, respectively. However, these and other more specific functions may be regarded as optional refinements of the taxonomy of core general-purpose functions distinguished here, which is depicted in Figure 2. The mother-daughter relation in this taxonomy reflects increasing specialisation going from mother to daughter; functions which are sisters in the taxonomy are mutually exclusive alternatives. The fact that the set of general-purpose functions forms a tree structure shows their semantic connectedness, and can be exploited in annotation processes by using the tree structure as a decision tree – see Annex A.

The general-purpose functions have as defining characteristic that they can be used to build a dialogue act in any of the dimensions by combining the function with a semantic content of the type of the dimension), The precise definitions of the core general-purpose functions are provided in Annex D.

10.2 Dimension-specific functions

The dimension-specific functions, which can be used only to address one of the dimensions, often do not have a semantic content. For instance, a *Turn Keep* function signals that the current speaker wants to keep the speaker role; this dialogue act does not require any semantic content. The same is true of all other turn management acts, and also of time management acts. Many social obligation management acts, like greetings and goodbyes, likewise do not require a semantic content; others, like expressions of thanks or apologies, may have a semantic content, if the speaker wants to indicate what he is thankful for, or what he apologizes for.

The following subclauses describe the core communicative functions identified for each of the nine core dimensions. Their precise definitions are in Annex D.

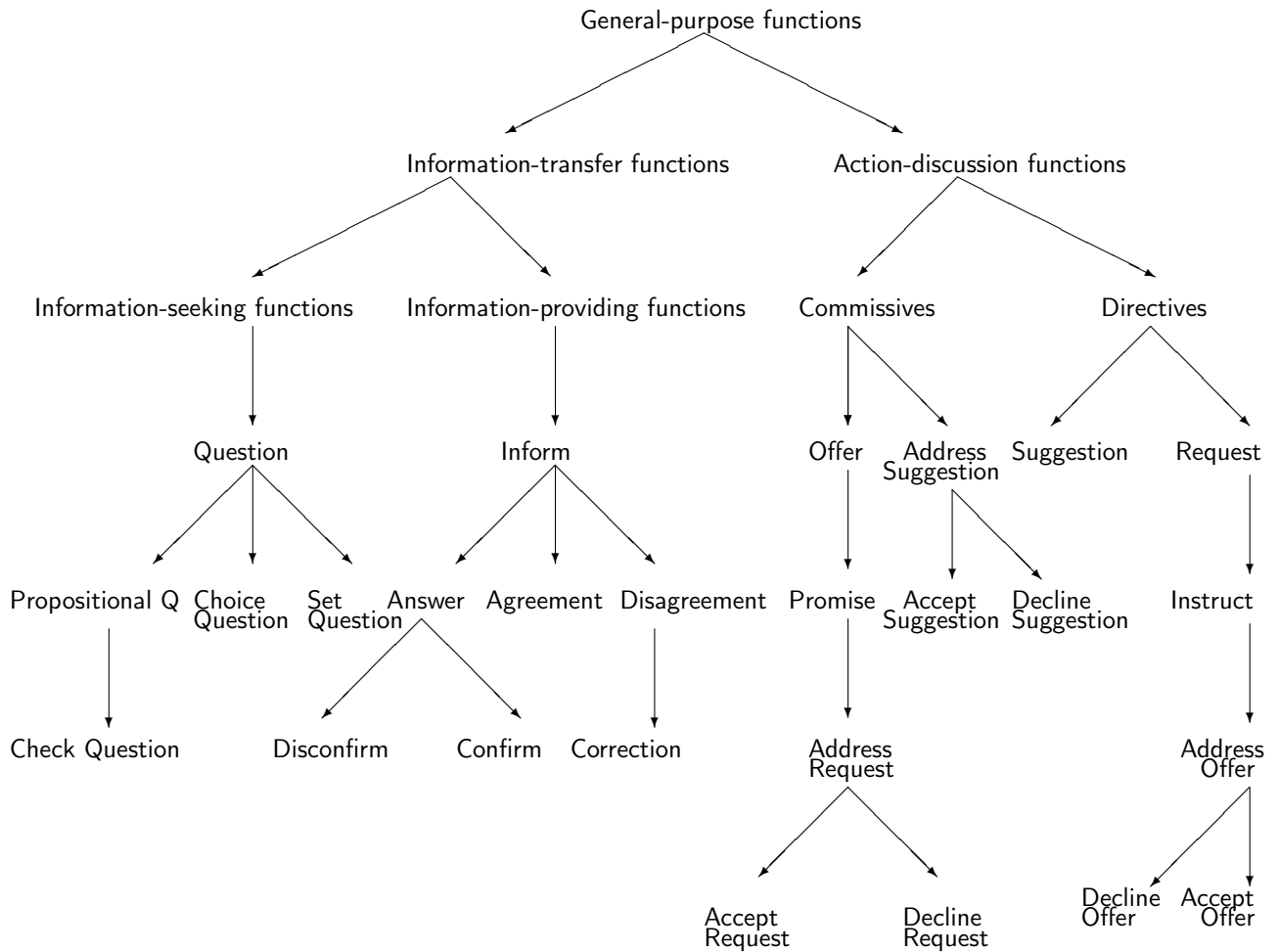


Figure 2 — General-purpose functions

10.2.1 The Task dimension

Dimension-specific communicative functions for the Task dimension are, by definition, functions that are specific for communication about a particular task domain. For example, highly specialized communicative functions such as "accept_date" and "suggest_exclude_location" have been proposed for a task domain concerned with appointment scheduling. In view of the domain-independence of the present standard, no such functions belong to the core communicative functions.

10.2.2 Feedback

Auto- and allo-feedback acts are often performed nonverbally, for instance by nodding, by looking at the speaker (indicating attention), by placing a hand behind an ear ("didn't hear you"), by raising eyebrows, or by frowning.

Feedback-providing acts for auto-feedback as well as for allo-feedback fall apart into positive and negative ones. Positive auto-feedback acts signal that the speaker successfully processed a previous utterance; positive allo-feedback that the speaker believes the addressee processed a previous utterance successfully. Negative auto-feedback acts signal that the sender encountered a processing problem; negative allo-feedback that the sender believes the addressee was unsuccessful in processing a previous utterance. Feedback elicitation acts express that the speaker is uncertain whether the addressee was successful in processing a previous utterance, and wants to know more about the addressee's processing.

Some annotation schemes distinguish various levels of processing to which feedback acts may refer. The SLSA schema (Allwood et al., 1994) distinguishes *contact*, *perception*, *understanding*, *evaluation*; the DIT++ schema distinguishes *attention*, *perception*, *interpretation*, *evaluation*, and *execution*. Feedback signals may be specific about the level of processing they address; for instance, a repetition of what was said before in slightly different terms usually relates to the level of understanding, while a verbatim repetition more likely refers to the level of perception (reporting what was heard). It has also been observed that the duration, speed, size, and number

of repetitions of nodding indicate the level of processing at which head nods signal positive feedback.⁶⁾ Quite often, however, feedback signals are underspecified as to the level of processing they refer to. This is especially true of positive feedback acts. Expressions like *OK* and *Yes* often express positive auto-feedback at the highest level of processing, and thereby by implication at all levels. For annotators it is often impossible to reliably indicate a specific level of processing for feedback messages, therefore the present standard does not include core feedback functions for specific levels of processing, but only the general positive and negative functions *Auto-Positive*, *Auto-Negative*, *Allo-Positive*, and *Allo-Negative*.

10.2.3 Turn Management

The main turn-management activities are taking the turn, assigning the turn, accepting the turn, keeping the turn, grabbing the turn (i.e. interrupting) and releasing the turn are. Turn management functions can be divided into *turn-initial* functions, which can only occur at the beginning of a speaker turn and which are concerned with obtaining the speaker role, and *turn-final* functions, which can occur only at the end of a speaker turn end which are concerned with the end of having the speaker role.

The turn management functions in this standard are defined as the activities that a dialogue participant undertakes *explicitly and specifically* for obtaining, maintaining, or giving up the speaker role. Just starting to speak is not considered as a turn management act, and neither is ceasing to speak considered as giving up the speaker role, or continuing to speak as trying to keep the speaker role. See on this topic also the annotation guidelines in Annex A.

10.2.4 Time Management

Stalling for time is a widespread phenomenon in spoken interaction, and may occur for a variety of reasons. Speakers sometimes need some extra time to decide how to interpret or evaluate an utterance, or on how to respond, and they do not always succeed in immediately finding the right words. This is indicated by slowing down and using fillers like *ehm*, *let me see*, *you know*, *well*, and so on. Simply being silent in such cases would be felt as awkward. Using fillers and slowing down can be used only when the speaker needs just a few seconds, not for, say, several minutes. A speaker who needs more time than just a few seconds, for instance because he needs to find his agenda, or to look something up, or to make a calculation, or because he is interrupted by something urgent, should do something else than stalling. This is where expressions like *just a minute*, *hold on*, *momentito*, *un instant*, *veuillez patienter* are used. They signal that the speaker is briefly suspending his contribution to the dialogue but intends to resume soon. We call this function Pausing.

Interactive computer systems also use Pausing acts, often by means of the same expressions as used in natural conversation, to indicate that the user has to wait a little, that the system is busy and needs some time to complete its processing. Nonverbal means are also used for this purpose, such as an hour glass icon or a bar which gradually fills up.

10.2.5 Discourse Structuring

Dialogue participants may structure the interaction explicitly by opening and closing the dialogue, by introducing, changing, or closing a topic, by indicating what they intend to do next, or what they would like another participant to do next (*Peter, will you introduce the next item?*). When the discourse structure is addressed explicitly by dialogue acts, most often general-purpose functions are used.

10.2.6 Own and partner communication management

Own communication management, occurring when a speaker edits his own speech while contributing to the dialogue, most commonly takes the form of self-corrections (also called 'repairs') and retractions. The most common forms of Partner Communication Management are the correction of speaking errors and the completion of an utterance which the partner is struggling to complete.

6) Petukhova & Bunt, 2009e.

10.2.7 Social obligations management

Of the numerous dialogue acts that can be performed for social functions, some are found very frequently in all kinds of dialogue. These include greetings and valedictions, at the beginning and end of a dialogue, respectively. Introducing oneself is also common in many interactive situations. Apologies are often used when a dialogue participant has misunderstood another participant, or is unable to fulfill a request or to answer a question. Thanking occurs frequently in those situations where one participant performs a service or provides help, and is also often used to initiate the closing of a dialogue. All these dialogue acts tend to come in initiative-response pairs, such as an initial and a response greeting, an apology and its acceptance, and a thanks and a 'downplayer' (*de nada; pas de quoi*).

Domain-specific functions such as Congratulation, Condolence, and Compliment also belong to this category, but not to the core communicative functions.

10.3 Function qualifiers

A limitation of virtually every dialogue act taxonomy is that it fails to capture subtleties in the performance of communicative actions relating to such phenomena as modality, conditionality, partiality, and accompanying emotions and attitudes. For example, it is customary to distinguish only two possible responses to an offer: acceptance and refusal. An offer may however be responded to in less clear-cut ways, and can for instance be accepted conditionally, as in (16.3), or partially, as in (16.2):

- (16) 1. A: Can I offer you some coffee and chocolates?
 2. B: Only coffee please.
 3. B: Coffee would be nice, but do we have time for that?

Suggestions and requests can be accepted partly, conditionally, and with certain modalities. Information-providing acts may also express the speaker's awareness that he possesses incomplete or uncertain information, as illustrated in (17):

- (17) 1. A: Do you know who'll be coming tonight?
 2. B: I have a hunch that Mary won't come.
 3. B: Peter, Alice, and Bert will probably come.
 4. B: I heard that Tom and Anne might come.

The responses 2, 3 and 4 in (17) all constitute partial answers; response 2 is in addition an uncertain answer.

Many dialogue acts can also be performed with the additional expression of the sender's emotional stance with respect to the semantic content of the act or his attitude toward the addressee, for instance:

- (18) a. A: Can I offer you a cup of coffee?
 B: Lovely!
 b. A: The first flight tomorrow morning is at seven-thirty.
 B: Perfect!
 c. A: What about a fresh cup of coffee?
 B: Ah, you're wonderful!

In (18a), B's acceptance of A's offer carries the additional attitudinal information that B would very happy to have a cup of coffee; in (18b) B's positive feedback carries the information that B is very satisfied with the information he obtained; and in (18c) B's acceptance of A's offer additionally expresses B's positive feelings toward A.

In order to be able to represent such phenomena, communicative functions like ‘Uncertain Answer’, ‘Conditional Accept Request’, ‘Partial Accept Offer’, and ‘Happily Accept Offer’, could be introduced. A more attractive alternative is to introduce a number of *qualifiers* that may be attached to a communicative function.

A corpus-based study of these phenomena (Petukhova & Bunt, 2010) indicates that (un)certainty, partiality, and conditionality can be captured in most cases by means of a binary distinction. For representing a speaker’s emotional stance with respect to the semantic content of the act, or his attitude towards the addressee, a wide variety of descriptors has been proposed in the literature, ranging from Ekman’s basic 6 emotions (Ekman, 1972) to classifications of several hundred possible values. In view of this, the present standard includes three binary attributes, for representing conditional, partial, and uncertain variants of dialogue acts, and an attribute with an open class of values for dealing with emotions and attitudes.

- **Certainty:** the qualifier ‘uncertain’ can be used with information-providing functions, in order to indicate that the speaker is uncertain about the correctness of the information that he provides; the qualifier ‘certain’ that the speaker does not express any doubt about this (which is the default value of this attribute). These two qualifiers can also be used with commissive functions, in order to express whether the speaker is committing himself with certainty to do something, or whether he is uncertain about this (as in *I may call you later today* and *I will definitely call you tomorrow*).
- **Conditionality:** the qualifier ‘conditional’ can be used with action-discussion functions, which have in common that the speaker assumes that the participant whose action is under discussion is able and willing to perform that action (in the case of commissives; this participant is the speaker; in the case of directives it is the addressee). The ‘conditional’ qualifier indicates that one of these assumptions is dropped (as in *Can you/Will you pass me the salt?*). The default value of this attribute is ‘unconditional’.
- **Partiality:** the qualifier ‘partial’ can be used to indicate that the semantic content to which the communicative function is applied is incomplete relative to the semantic content of the dialogue act with which there is a functional dependence relation. The default value of this attribute is ‘full’.
- **Sentiment:** these qualifiers indicate that the speaker has a certain emotional attitude to the semantic content that the communicative function is applied to, or towards the addressee. Possible qualifier values for this might include ‘pleased’, ‘surprised’, ‘annoyed’, ‘disappointed’, and can be chosen as appropriate for a given domain, task, or interactive setting. This attribute does not have a default value.

Table 2 summarizes this, indicating also which categories of communicative functions can be qualified by which qualifiers.

aspect of qualification	qualifier values	communicative function category
certainty	uncertain, certain	information-providing functions commissive functions,
conditionality	conditional, unconditional	action-discussion functions; responses to action-discussion acts
partiality	partial, full	responsive general-purpose functions
sentiment	surprised, pleased,... [open class]	any communicative function

Table 2. Aspects of qualification, qualifiers, and relevant function categories.

11 DiAML: Dialogue Act Markup Language

The Dialogue Act Markup Language DiAML has been designed in accordance with the ISO Linguistic Annotation Framework (LAF, ISO 24612:2009). This framework draws a distinction between the concepts of *annotation* and *representation*. The term ‘annotation’ refers to the linguistic information that is added to segments of language data, independent of the format in which the information is represented. The term ‘representation’ refers to the format in which an annotation is rendered, independent of its content (Ide & Romary, 2004). According to LAF, *annotations* are the proper level of standardization, rather than *representations*. This distinction is implemented in the DiAML definition by including a syntax that specifies besides a class of *representation structures* also a class of more abstract *annotation structures*, defined as certain set-theoretical structures for which a formal

semantics is defined. These two components of the language specification are called its *concrete* and *abstract syntax*, respectively. The concrete syntax defines a particular rendering of the annotation structures following the methodology for defining semantic annotation languages described in Bunt (2010), where the notion of an ‘ideal representation format’ is defined as one where (a) every annotation structure defined by the abstract syntax can be represented, and (b) every representation defined by the concrete syntax represents a uniquely determined annotation structure. Any two representation formats which are ‘ideal’ in this sense are semantically equivalent, and every representation in one format can be converted by a meaning-preserving mapping into the other format.⁷⁾

The abstract syntax of DiAML is specified in subclause 1 of the present clause; subclause 2 presents the specification of a concrete syntax XML-based representation format; and section 3 deals with the DiAML semantics. Annex C of this document contains an XML schema definition for the concrete syntax presented in subclause 2, and also defines an alternative representation format based on feature structures (as represented in XML following ISO standard 24610-1:2006), illustrating the possibility of using alternative representation formats that share the same abstract syntax and semantics.

11.1 Abstract syntax

The abstract syntax of DiAML defines certain set-theoretical structures, called *annotation structures*, which contain all and exactly those elements that constitute the annotation of a functional segment with dialogue act information according to the metamodel shown in Figure 1.

Formally, an annotation structure is a pair $\langle E, L \rangle$ consisting of a set E of *entity structures*, and a set L of *link structures*. Entity structures contain semantic information about a functional segment; link structures describe semantic relations between functional segments. An entity structure is a pair $\langle s, a \rangle$ consisting of a functional segment s and a set of annotations a . A link structure is a triple $\langle e_1, e_2, r \rangle$ consisting of two entity structures and a relation, or a pair $\langle e_1, e_2 \rangle$ consisting of two entity structures and an implicit semantic relation, or a pair $\langle e, s \rangle$ consisting of an entity structure that denotes a feedback act, and a functional segment to which the feedback applies.

The abstract syntax of DiAML consists of two parts: (a) a specification of the elements from which annotation structures are built up, called a ‘conceptual inventory’, and (b) a set of rules which describe the possible ways of combining these elements (‘annotation construction rules’).

a. Conceptual inventory

- a finite set $Parts = \{P_1, P_2, \dots, P_k\}$ of elements called ‘dialogue participants’;
- a finite set $PR = \{PR_1, PR_2, \dots, PR_k\}$ of elements called ‘participant roles’;
- a denumerable set $FS = \{fs_1, fs_2, fs_3, \dots\}$ of elements called ‘functional segments’;
- a finite set $Dim = \{D_1, D_2, \dots, D_N\}$ of elements called ‘dimensions’;
- a finite set $DSF = \{DSF_1, DSF_2, \dots, DSF_N\}$ where each element DSF_i is a finite set $DSF_i = \{F_{i1}, F_{i2}, \dots, F_{in_k}\}$ of elements called ‘dimension-specific communicative functions’;
- a finite set $GPF = \{F_{01}, F_{02}, \dots, F_{0n}\}$ of elements called ‘general-purpose communicative functions’;
- a finite set QV of finite sets Q_1, \dots, Q_k of elements called ‘qualifiers’; the sets Q_i are called ‘qualification aspects’;
- a finite set $Rhet = \{R_{01}, R_{02}, \dots, R_{0m}\}$ of elements called ‘rhetorical relations’.

b. Annotation construction rules

Entity structures

- an entity structure is a pair $\langle s, a \rangle$ where s is functional segment and a is a set of one or more dialogue act structures.
 - * a *dialogue act structure* is a quadruple $\langle S, A, d, f \rangle$ where $S \in Part$ (the sender of the dialogue act); $A \subset Parts$ (the set of addressees of the dialogue act); d is a dimension ($d \in Dim$); and f is a communicative function.

7) See Bunt (2010a) for formal definitions and proofs relating to alternative representation formats sharing the same abstract syntax, and see Ide & Bunt (2010) for applying this to the GrAF framework for linguistic annotation (Ide and Suderman, 2007).

- * a communicative function is an element of the set of communicative functions, i.e. $f \in GPF \cup DSF_1 \cup DSF_2 \cup \dots \cup DSF_N$ or a pair $\langle f, q \rangle$ where f is a communicative function and q is a qualifier structure.
- * a *qualifier structure* q is a list of qualifiers in which no qualification aspect occurs more than once. Formally: $q \in Q_1 \cup (Q_1 \times Q_2) \cup (Q_1 \times Q_2 \times Q_3) \cup \dots \cup (Q_1 \times Q_2 \times Q_3 \times \dots \times Q_k)$.

Link structures

- A link structure is one of the following:
 - * a pair $\langle da_1, da_2 \rangle$ consisting of two dialogue act structures, where the second has a functional dependence relation to the first (e.g. da_2 is an answer and da_1 is the corresponding question);
 - * a pair $\langle da_2, S_1 \rangle$ consisting of a dialogue act structure da_2 and a set of functional segments S_1 to which the dialogue act has a feedback dependence relation (for relating a feedback act to the segment(s) that it provides or elicits information about);
 - * a triple $\langle da_1, da_2, R_j \rangle$ consisting of two dialogue acts and a rhetorical relation.

11.2 Concrete XML-based syntax

The annotation structures defined by the abstract syntax consist of entity structures and link structures, where an entity structure associates one or more dialogue act structures with a functional segment, and a link structure encodes functional, feedback, or rhetorical relations between entity structures. The core ingredient of all these types of structures is the 'dialogue act structure', which is an n-tuple of elements from the conceptual inventory. Such n-tuples have a straightforward XML-representation if we introduce XML attributes and values in a systematic way as corresponding to the categories and their elements from the abstract syntax.

A concrete DiAML syntax consists of two parts: (a) a vocabulary, specifying names of XML tags, attributes, and values for the various ingredients in the conceptual inventory, and (b) a set of 'representation construction rules', specifying XML elements for entity structures and the various types of link structures, and defining an XML representation of the association of dialogue act structures with functional segments. Note that the vocabulary does not include names of dialogue participants or functional segments; this is because the identity of the participants is assumed to be defined by the metadata of the dialogue that is annotated, so that the annotation representation will simply use these identifiers as values of attributes representing participant roles; and the functional segment to which the annotation applies are assumed to be provided by another layer of processing where they have obtained XML identifiers. (See below for an example.)

a. Vocabulary

- XML attributes and values to represent the elements of the conceptual inventory (dimensions, communicative functions, function qualifiers, and rhetorical relations). Senders and addressees are assumed to be identified in the metadata of the annotated data, and are therefore represented by pointers to these identifiers. The functional segments in the annotated data are assumed to be identified either as spans in the original data or as objects at another level of analysis, like the outputs of a tokenizer; they are therefore also represented by pointers to these external structures. (For more details see Annex C and the examples in Annex B.)
 - * **dimension names: (as attribute values)**
`task, autoFeedback, alloFeedback, turnManagement, timeManagement, discourseStructuring, ownCommunicationManagement, partnerCommunicationManagement, socialObligationsManagement`
 - * **communicative function names (as attribute values):**
`inform, agreement, disagreement, correction, propositionalQuestion, setQuestion, checkQuestion, choiceQuestion, answer, confirm, disconfirm, offer, promise, instruct, suggestion, addressSuggestion, acceptSuggestion, declineSuggestion, addressOffer, acceptOffer, declineOffer,`

autoPositive, autoNegative, feedbackElicitation, alloPositive, alloNegative, turnTake, turnGrab, turnAccept, turnRelease, turnAssign, turnKeep, stalling, pausing, topicShift, completion, correctMisspeaking, selfCorrection, signalSpeakingError, opening, interactionStructuring, initialGreeting, returnGreeting, apology, acceptApology, initialSelfIntroduction, returnSelfIntroduction, thanking, acceptThanking, initialGoodbye, returnGoodbye;

* **qualifier attribute and value names:**

- **attribute** certainty; **values** certain, uncertain
- **attribute** conditionality; **values** conditional, unconditional
- **attribute** partiality; **values** partial, exhaustive
- **attribute** sentiment; **values** surprised, pleased, ... (open class)

* **rhetorical relations:**

elaborate, justify, ... (open class)

b. Representation construction rules

Entity structure representations

- an entity structure representation is an XML element called `dialogueAct`, which has an attribute `target` whose value points to a functional segment, and the attributes `sender`, `addressee`, `otherParticipant`, `dimension`, and `communicativeFunction`, and attributes for the qualification aspects `certainty`, `conditionality`, `partiality`, and `sentiment`.

Link structure representations

- **functional dependence link:** an XML element called `functionalDependence` is defined, which has the attributes `dact` and `functAntecedent` for representing the currently annotated dialogue act and its 'functional antecedent';
- **feedback dependence link:** an XML element called `feedbackDependence` is defined, which has the attributes `dact` and `fbSegment` for representing the currently annotated dialogue act and the segment whose processing the current dialogue act provides or elicits information about;
- **rhetorical link:** an XML element called `rhetoricalLink` is defined, which has the attributes `dact`, `rhetoAntecedent`, and `rhetoRel` for representing the currently annotated dialogue act, the one that it has a rhetorical relation to, and the particular rhetorical relation.

The formal specification of the DiAML concrete syntax using XML is provided in Annex C. The following example illustrates the use of this concrete syntax. In (19) we see a dialogue fragment consisting of a question asked by participant P1 followed by an answer by participant P2. Note that P2's utterance is segmented into two overlapping functional segments: one in the Auto-Feedback dimensions, with positive value, and one in the Task dimension, with value 'answer' qualified as 'uncertain'.

1.	P1:	<i>What time does the next train to Utrecht leave?</i>
TA:	fs1:	<i>What time does the next train to Utrecht leave?</i>
(19)		
2.	P2:	<i>The next train to Utrecht leaves I think at 8:32.</i>
AuFB	fs2:	<i>The next train to Utrecht</i>
TA	fs3:	<i>The next train to Utrecht leaves I think at 8:32.</i>

Dialogue act annotations may be attached to primary dialogue data in a variety of ways. They may be attached directly to stretches of speech, defined by temporal begin- and end points, but often they will be attached to

structures at lower levels of analysis and annotation, such as the output of a tokenizer. Here we assume that the relevant functional segments are identified at another level of XML representation according to ISO standard ISO 24610-1 (joint ISO-TEI standard). The `target` attribute establishes links the annotations with the corresponding functional segments. In the example, we assume that P1's utterance is identified as the functional segment 'fs1', and the two functional segments in P2's turn as 'fs2' (in the Auto-Feedback dimension) and 'fs3' (in the Task dimension).

With these assumptions, the DiAML representation of (19) is as shown in (20).

```

<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"
(20)  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact="#da2" fbSegment="#fs1"/>
<dialogueAct xml:id="da3" target="#fs2"
  sender="#p2" addressee="#p1"
  communicativeFunction="answer" certainty="uncertain"
  dimension="task" />
<functionalDependence dact="#da3" functAntecedent="#da1"/>
</diaml>

```

11.3 DiAML semantics

A fundamental requirement on semantic annotation is that semantic markups should have a well-defined semantics (Bunt & Romary, 2002; Bunt, 2007b). The DiAML language has a formal model-theoretic semantics associated with its abstract syntax, in terms of information-state updates. The details of such a semantics depend on the precise definition of information states, and such a definition is beyond the scope of this standard. Instead, in this section we outline a semantics making no further assumption than that an information state is structured into a number of distinct components, an assumption which is shared between all proposals for information states (e.g. Poesio & Traum, 1998; Bunt, 2000; Ahn, 2001; Cooper, 2004). The details of an information-state update semantics also depend on (1) whether a single addressee is considered or multiple addressees; and (2) on whether not only the information states of addressees are considered to be updated by dialogue contributions, but also the information state of the sender, as e.g. argued in Bunt (2000). We will consider only the updates of a single addressee's information states, which is the basis for more complex approaches involving multiple information states.

In formulating this semantics, we will make use of the following functional concept. For a given set of partial functions F , which all have the domain D and values in the range R , the notation F^* is used to designate the set of those functions which can be defined as unions of functions in F :⁸⁾

$$(21) F^* = \{f \in R^D \mid \exists f_1, \dots, f_m \in F, \text{ such that } f = \lambda x. f_1(x) \cup \dots \cup f_m(x)\}.$$

The semantic definition of the DiAML abstract syntax consist of two parts:

1. the definition of a model, including an interpretation function which assigns semantic structures to the categories of the abstract syntax; and
2. the specification of how the interpretation of an annotation structure can be computed from the interpretations of its constituents.

(22) **Definition.** A model for DiAML is a pair $M = \langle D, F \rangle$, where

8) The union of two (or more) PARTIAL functions is defined only if they assign the same values to those arguments for which they are both (or all) defined.

- D is the model structure, which is a triple $\langle IS, PA, P_u \rangle$, where
 - IS is a set of information states; each information state is structured into components $\{IS_1, IS_2, \dots, IS_k\}$ where k is less than or equal to the number of dimensions distinguished in the abstract syntax;
 - PA is a set of participating agents;
 - P_u is a set of elementary update schemes, i.e. functions which, given a sender, an addressee, and a dimension, specify how a participating agent should use a given semantic content to update his information state.
- F is the interpretation function of the model, which assigns
 - to every communicative function and to every qualifier an update schema from P_u^* , i.e., every communicative function is interpreted as a way of updating an information state;
 - to every participant a member of PA (a participating agent);
 - to every dimension D_j the same component of every information state: $F(D_j) = \{S_j | \exists S \in IS \wedge S = \{S_1, \dots, S_i, \dots, S_k\}\}$.

We describe the computation of the interpretation of DiAML annotations from those of their components by presenting a representative example. Note, first, that the ‘elementary update schemes’ of the model are parameterized update functions with parameters for a sender, an addressee, and an information state component, such as the following ones:

- (23) P_{u1} : update component S_i of Y 's information state by extending it with the information that participant X wants participant Y to know that p
 P_{u2} : update component S_i of Y 's information state by extending it with the information that participant X assumes that p

These two elementary update schemes can be used to specify the semantics of the Inform function as the update schema in (24):

$$(24) F(\text{Inform}) = \lambda X. \lambda Y. \lambda S_i. \lambda z. P_{u1}(X, Y, S_i, z) \cup P_{u2}(X, Y, S_i, z).$$

Applied to the participants S and A and the Task dimension, the update function (25) results, where V (for ‘valuation’) denotes the recursive procedure for computing the interpretations of annotations.

$$(25) \quad V(\langle S, A, Task, Inform \rangle) = F(\text{Inform})(F(S), F(A), F(Task)) \\ = \lambda z. P_{u1}(S, A, IS_{A, Task}, z) \cup P_{u2}(S, A, IS_{A, Task}, z)$$

This update function can be applied to a semantic content in order to obtain a description of how A 's information state is updated. When S informs addressee A that KLM flight 476 departs at 19:15, formalized as DEP(KL476, 19:15), then if A understands S , the component $IS_{A, Task}$ of his information state which contains his task-related information is updated by being extended with two beliefs: 1) A now believes that S wants A to know that DEP(KL476, 19:15), and 2) A now believes that S assumes that DEP(KL476, 19:15) - these beliefs having been added to $IS_{A, Task}$. This update is achieved by applying (23), with $p = \text{DEP}(\text{KL476}, 19:15)$.

Communicative function qualifiers are semantically defined as making the information state updates of the communicative functions that they qualify more specific. The following example of an uncertain Inform shows how this can be accomplished.

- (26) a. S : The KL 476 departs I think at 19:15.
 b. $V(\text{Inform}, \text{uncertain}) = \lambda A. \lambda B. \lambda C_i. \lambda x. F(\text{Inform})(A, B, C_i, x) \cup F(\text{uncertain})(A, B, C_i, x). \\ = \lambda A. \lambda B. \lambda C_i. \lambda x. P_{u1}(A, B, C_i, x) \cup P_{u2}(A, B, C_i, x) \cup P_{u3}(A, B, C_i, x)$

where the update schema P_{u3} is defined as follows:

- (27) P_{u3} : update component S_i of Y 's information state by extending it with the information that participant X has a weak belief that p

The effect of the *uncertain* qualifier is thus that the addressee's task-related information is extended with the information that the speaker has a weak belief (as opposed to a firm belief) that the answer he provides is correct. The P_{u2} update contains the element that S 'assumes' that certain information is correct, where 'assumes' is intended to be underspecified as to the strength of that belief; adding to this that the speaker has a weak belief is equivalent to specifying the 'assumption' to be a weak belief. Similarly for other qualifiers.

The semantics of functional and feedback dependence relations and rhetorical relations adds to this that in the addressee's information state a link should be created between the dialogue act of which the communicative function is interpreted and the dialogue act to which it responds (in the case of a functional dependence relation) or to which it has a particular rhetorical relation, or to the functional segment to which it provides or elicits feedback (in the case of a feedback dependence relation).⁹⁾ The addition of such links in an information state requires such a state to include a dialogue history, which represents dialogue acts that occurred earlier in the dialogue. The details of this depend on the precise representation of the dialogue history in information states.

12 Principles for extending and restricting the standard

The annotation schema defined in this standard, with its core and optional dimensions and communicative functions, cannot be expected to be ideal for every kind of dialogue analysis, for every task domain, for every kind of dialogue, and for every annotation purpose. The general principles underlying the design of the schema and the DiAML annotation language should however also be useful for accommodating extensions, modifications, or restrictions of the schema and the annotation language, as the need arises for particular applications. In this section we summarize the main design principles and formulate guidelines for schema extension and restriction.

12.1 Main design principles

The main principles underlying the annotation schema and the DiAML language can be summarized as follows:

- (28) a. Dialogue behaviour is viewed as multifunctional, i.e. each stretch of communicatively meaningful behaviour may have more than one communicative function. The schema is therefore multidimensional, aiming to optimally support the assignment of multiple communicative functions to units in dialogue.
- b. Dimensions are defined in this standard as distinct types of communicative activity, such as giving feedback, managing the allocation of the speaker role, pursuing the underlying task or activity, and taking care of social obligations. Each of these types of activity is concerned with different categories of information (processing of utterances, allocation of participant roles, task/activity, and social obligations, respectively).
- c. Communicative functions are most accurately assigned to *functional segments*, minimal stretches of communicative behaviour that have a communicative function (one or more). Functional segments may be discontinuous, overlapping, and spreading over multiple turns.
Segmenting a dialogue into functional segments is most accurately done in a 'multidimensional' fashion, identifying in each dimension the stretches of communicative behaviour that count as functional segments in the sense of having a communicative function in that dimension.
- d. Communicative functions are defined semantically in terms of how they use a semantic content to change the information state of a dialogue participant who understands the corresponding functional segment (understands: as intended by the speaker).

9) See Lascarides & Asher (2007) for an update semantics of rhetorical relations. They write: "rhetorical relations act semantically like complex update operators, and their interpretation reflects the special semantic influence that they have (...) It also reflects that (...) they change the context."

- e. All dimensions are (1) theoretically justified; (2) empirically observed; (3) recognisable with acceptable precision by human annotators and by automatic annotation systems; (4) addressable independently from other dimensions ('orthogonal').
- f. The set of communicative functions is divided into sets of *dimension-specific* functions, one for each dimension, which can be applied only to information that belongs to that dimension, and a set of *general-purpose functions* which can be applied to any sort of information and form a dialogue act in any of the dimensions.
- g. Communicative functions are required to be: (1) empirically observed; (2) theoretically validated; (3) relevant for obtaining a good coverage of the phenomena in a given dimension; (4) recognizable with acceptable precision by human annotators and by automatic annotation systems.
- h. The set of general-purpose communicative functions is *semantically connected*, in the sense that any two functions are either mutually exclusive alternatives or one is a specialization of the other. This is reflected in Figure 2, where any two functions either have a dominance relation or are alternatives with a common ancestor.
- i. For each dimension, the set of dimension-specific communicative functions for that dimension is semantically connected.
- j. The semantic connectedness of the sets of communicative functions that can be used in any given dimension has the effect that a functional segment never needs to be annotated with more than one function per dimension, when annotators follow the plausible strategy of assigning to a functional segment in each dimension the most specific communicative function (if any) for which there is sufficient evidence. Combined with the orthogonality of the dimensions, this has the consequence that a functional segment is annotated with maximally as many functions as there are dimensions in the annotation schema.

Semantic connectedness of the set of function tags is not a strict requirement; if this requirement is not met for some part of the tag set, then multiple tags from that (sub)set may be assigned to a functional segment.

12.2 Schema extension

The DiAML standard for dialogue act annotation is relatively easily extensible in the following ways:

- the addition of dimensions to the set of core dimensions. The additional dimension Contact Management was for example noted to be a possible candidate for being added, being orthogonal to the core dimensions, theoretically justified, empirically observed, and recognizable with acceptable precision by human annotators and automatic annotation programs;
- the addition of communicative functions to the set of core communicative functions. The DIT⁺⁺ and LIRiCS taxonomies contain examples of communicative functions that satisfy the requirements of empirical and theoretical validity, relevance for adequate coverage, and human and machine recognizability, and that moreover satisfy the desideratum of leading to semantically connected sets of communicative functions;
- the addition of communicative function qualifiers as values of existing qualifier attributes, such as the specification of the values of the `sentiment` attribute as well as the introduction of additional qualifier attributes and values. As in the case of communicative functions, the additional qualifier attributes should leave the set of these attributes "orthogonal", in the sense of dealing with non-overlapping aspects of qualification, and for each attribute the set of values should preferably be "semantically connected" in order to allow the assignment of a unique most specific value to the attribute.
- the specification of rhetorical relations. To avoid ambiguity and redundancy, again, the total set of specified rhetorical relations should be semantically connected.

More concretely, concepts may be added to the core notions of this standard in the following ways.

- a) The set of core DiAML dimensions has the properties summarized in (28e), and moreover meets the requirement of playing a role in many of existing dialogue act annotation schemas. For specific purposes or domains, new dimensions may be added for which these requirements do not hold, because they are in fact an object of investigation. For example, an particular additional dimension does not need to be theoretically justified *a priori*, since the purpose may be to investigate dialogue phenomena which have not been well studied yet. One property that a potential additional dimension should satisfy is that of being orthogonal to the already existing dimensions, in order to avoid redundancy and ambiguity in annotation. (But if a potential new dimension D_m would not be orthogonal to an existing dimension D_k , a possibility would be to leave out D_k and add D_m instead.)
- b) New communicative functions may be added to the set of core functions, provided that they satisfy the requirements summarized above, except possibly the requirements of theoretical validity and presence in a significant number of existing annotation schemes. As in the case of adding dimensions, these requirement may be dropped when adding functions for a specific purpose or a particular domain. In addition, the convenient property of semantic relatedness of the communicative functions available in a given dimension is deserves careful consideration.
- c) Domain-specific communicative functions may freely be added, provided that they meet the requirements for the inclusion of communicative functions in general, as laid out in (15) in Clause 10.
- d) New communicative functions may be freely introduced which are more specific than a function already present in the schema.
- e) Communicative function qualifier values may freely be introduced for the attribute that takes emotional stance and attitude into account.
- f) Additional communicative function qualifier attributes (and their values) may be introduced provided that they capture information which relates in a well-defined way to what is captured by those attributes already present.

12.3 Schema restriction

Subschemas of this annotation standard schema can be defined relatively easily, by leaving out certain ingredients in the following ways.

- A dimension and the corresponding set of dimension-specific communicative functions may be freely left out; by virtue of the orthogonality of the set of core dimensions, whether a particular core dimension is included or not has no influence on the remaining dimensions.
- Communicative functions may be freely left out for which there is a less specific function present in the schema, since in that case the remaining set of communicative functions is still semantically connected.
- It is not recommended to leave out a communicative function for which the schema contains more specific functions while maintaining the more specific functions, since this limits an annotator's possibilities to use a less specific functional tag in the case of little evidence that would warrant a more specific choice.
- Communicative functions may be left out which are considered irrelevant for a particular purpose, if that does not have undesirable limiting effects on the desired coverage of dialogue phenomena.
- Communicative function qualifiers may be partly or completely left out, having the effect that the results of analysis or annotation are equivalent to using the qualifiers with their default values, for those qualifier attributes which have default values, and leaving the qualification aspect underspecified for those attributes which do not have default values.

Annex A

(normative)

Annotation guidelines

Overview

Section A.1 of this annex discusses some general issues in dialogue act annotation. Section A.2 discusses the segmentation of a dialogue into functional segments. Section A.3 provides guidelines for the use of DiAML and the annotation schema defined in this standard. The examples in this annex focus on certain specific issues in dialogue act annotation; examples of fully annotated dialogue fragments can be found in Annex B.

A.1 General issues in DA annotation

A.1.1 Preliminaries

A dialogue has been defined as "a spoken, typed or written interaction in natural language between two or more agents" (DAMSL Revised Manual, p. 1). The term 'agent' in this characterization is intended to cover both human and artificial participants. The present standard is intended to apply to dialogues in a somewhat wider sense, where the participants not only use natural language but also nonverbal means, such as gestures and facial expressions in the case of human participants and embodied conversational agents, and means like highlighting, blinking, or a filling hour glass in the case of computer systems. In the case of human dialogue, the prototypical setting is that of face-to-face communication, where speech is combined with other vocal sounds (laughs, sighs, heavy breathing, etc.), facial expressions, eye gaze movement, and other physical activities including head-, hand-, arm-, and shoulder gestures, forms of touching (stroking, caressing, hugging, shaking hands, patting on the shoulder, etc.), and body posture changes. All these verbal and nonverbal activities may have a communicative meaning which can be made explicit in terms of dialogue acts. This standard has a general emphasis on its use for creating interoperable language resources, but it has been successfully applied also to a range of nonverbal and multimodal behaviours. (See e.g. Petukhova and Bunt, 2009e on the analysis of nodding as feedback signals.)

A.1.2 Dialogue settings and participants

Dialogue act annotation schemes have been developed mostly for situations involving two people in a spoken interaction, with or without visual contact, or involving several people in a setting where they see each other. In both types of situation there is much of the time one participant who occupies the speaker role, i.e. "has temporary control of the dialogue and speaks for some period of time" (DAMSL Revised Manual, Preliminaries, p. 1).

This participant, the 'speaker', speaks either to the single other participant in the case of a two-person dialogue, or to one or more participants in the case of multi-party dialogue - this /these participant(s) is/are the addressee(s) of the dialogue acts performed by the speaker.

There are formalized interactive situations where the role of addressee does not coincide with the person(s) that the speaker is in fact addressing. For example, in debates in the British House of Commons the person who occupies the speaker role is formally addressing the Speaker of the house, but his words are in fact aimed at a particular representative or at the entire group of representatives of a political party. Another type of dialogue setting where the role of addressee is not straightforward is that of a televised interview in front of an audience. In this case the interviewee will typically speak as if addressing the interviewer, while his words are in fact intended primarily for the audience that is present in the studio, or for the viewers at home. Communicative functions are defined in this standard as the way in which the speaker intends to affect the information state(s) of the addressee(s), hence in such situations the annotation of the speaker's utterances should be determined by considering whose information states the speaker is principally trying to influence.

A.1.3 Annotation purposes and unusual annotation situations

This standard is intended for use by human annotators and by automatic annotation systems. It has been tested for being useful for both these purposes.

If the purpose of an annotation effort is to achieve the most accurate annotations, then the annotators involved should use all the sources of information that are available. For a multimodal dialogue, where speech is used in combination with nonverbal behaviour, this means that not only the recorded speech should be available to annotators, but also a video recording of the nonverbal behaviour, or an accurate transcription of that behaviour. Similarly, in the case of a dialogue over the telephone, annotators should not only have the transcribed speech at their disposal but also the original sound recording (or an accurate transcription of the prosody and the relevant nonlinguistic sounds that occur), for being able to interpret the intonation, speech tempo, and nonlinguistic vocal sounds. One important source of information for annotators when deciding on the identification or annotation of a given functional segment may be the recording of how the dialogue continued *after* the segment under consideration. Therefore, if the purpose is to obtain the most accurate possible annotation, annotators should be allowed to use look-ahead.

A.1.4 Explicit and implicit, implied and indirect functions

A functional segment has a communicative function for one of two reasons: 1) by virtue of having linguistic or nonverbal features which, in the context in which the segment occurs, are indicators of that function: or 2) by implication of having another function. In the first case it is common to say that the segment has that communicative function explicitly; in the second case that it has that function implicitly. The following example illustrates this:

- (29) 1. A: Would you like to have some coffee?
2. B: Some coffee would be great, thanks.

A's utterance is an Offer; B's response is an Accept Offer by virtue of its linguistic form and the fact that it occurs immediately after an Offer. Since an offer can only be accepted when it has been understood, B's response by implication also has a positive auto-feedback function.

A functional segment expressing a dialogue act DA1 which has a functional dependence relation to a previous dialogue act DA2, always has an implied auto-feedback function relating to the functional segment where DA2 was expressed. This is one important type of implicit functions that functional segments may have, and it is one of the sources of the multifunctionality of functional segments. More generally, the following types of implicit communicative functions can be distinguished:

- a) A communicative function F_1 may *logically entail* another one F_2 because F_1 is a special case of F_2 . This happens in hierarchies of communicative functions like the general-purpose functions of these standard, where for instance a Confirm is a special case of an Answer, and a Correction is a special case of a Disagreement, which in turn is a special case of an Inform.
- b) A communicative function F_1 may have another function F_2 as a pragmatic *conversational implicature*, i.e. in most situations where a functional segment has the function F_1 it also has the function F_2 , assuming that the dialogue participants behave cooperatively. For example, a thanking act like *Thank you* will normally be understood as also a signal of positive feedback.

Should implicit communicative functions be annotated? Annotating logically entailed functions would be redundant, since by their very nature such functions can be inferred from explicit functions. For conversationally implicated functions the situation is different, since these functions do not necessarily follow from explicit function, so it is in general recommended to do so. An annotator running into the situation where a functional segment has an explicitly expressed communicative function and an implied function, should decide whether the implied function is a logical consequence or a matter of what is plausible in the given context. In the first case the implied function should not be annotated; in the second case it should. For more details about types of implicit functions and strategies for how to deal with them see Bunt (2010b).

Standard speech act theory regards indirect speech acts, such as indirect requests, as just an indirect *form* of the same illocutionary acts. By contrast, this standard incorporates the view that indirect forms signal subtly different packages of beliefs and intentions than direct ones. For example, the direct request *Tell me what time it is please* carries the assumption that the addressee knows what time it is, whereas an indirect request like *Do you know what time it is?* or *Can you tell me what time it is?* does not carry that assumption (it does at least not express that assumption; in fact it questions it), and is best interpreted as *Please tell me what time it is, if you know*.

This example shows that an indirectly formulated request may have a conditional character: the speaker is expressing a request under the condition that the addressee is able to perform the requested action. In this case the annotator may therefore make use of the option to annotate the utterance as having a qualified Request function, with the attribute ‘conditionality’ having the value ‘conditional’. This is represented in DiAML as follows:

```
(30) <dialogueAct xml:id="da1" target="fs1"
      sender="s" addressee="a" dimension="task"
      communicativeFunction="request" conditionality="conditional"/>
```

A.1.5 General advice for annotators

Dialogue act annotation is about indicating the kind of intention that the speaker had; what was he trying to achieve? When participating in a dialogue, this is what an addressee tries to establish. The following general advice for dialogue act annotators derives from this.

a) Do as an addressee would do.

When assigning annotation tags to a dialogue utterance (a ‘functional segment’, more precisely), put yourself in the position of the participant(s) to whom the utterance was addressed, and imagine that you try to understand what the speaker is trying to achieve. Why does he say what he says? What are the purposes of the utterance? What assumptions does the speaker express about the addressee? Answering such questions should guide you in deciding which annotation tags to assign, regardless of how exactly the speaker has expressed himself. Use all the available information that you would have if you were the actual addressee, and like the addressee, try to understand the speaker’s communicative behaviour.

b) Think functionally, not formally.

The linguistic form of an utterance often provides vital clues for choosing an annotation tag, but such clues may also be misleading; in making your choices you should of course use the linguistic clues to your advantage, and don’t let them fool you - the true question is not what the speaker says but what he means.

For example, Set Questions are questions where the speaker wants to know which elements of a certain domain have a certain property. In English, such questions often contain a word beginning with “wh”, such as *which* as in *Which books did you read on your holidays?* or *where* in *Where do your parents live?* But in other languages this is not the case; moreover, even in English not all sentences of this form express a Set Question: *Why don’t you go ahead* is for instance typically a suggestion rather than a question.

Similarly, Propositional Questions are questions where the speaker wants to know whether a certain statement is true or false. Such questions are typically expressed by interrogative sentences, such as *Is The Hague the capital of the Netherlands?* or *Do you like peanut butter?* But not all sentences of this form express a Propositional Question; for example, *Do you know what time it is?* is most often used as an indirect way of requesting to tell the time. Similarly, *Would you like some coffee?* is most likely an offer, rather than a question, and *Shall we go?* a suggestion.

c) Be specific

Among the communicative functions that you can choose from, there are differences in specificity. For instance, a Check Question is more specific than a Propositional Question, in that it additionally carries the expectation that the answer will be positive. Similarly, a Confirm act is more specific than an Answer, in that it carries the additional speaker that the addressee expects the answer to be positive.

In general, try to be as specific as you can. But if you’re in doubt about whether to use a more or a less specific function that both seem reasonable to you, then use the less specific one.

A.2 Segmentation

According to this standard, dialogue acts correspond to *functional segments* as defined in Clause 8. In this definition, a functional segment is characterised as a *minimal* stretch of communicative behaviour that has a communicative function; the requirement of being ‘minimal’ has been added in order to ensure that communicative functions are assigned as accurately as possible to those stretches of behaviour which express these functions. Consider the following example (from a Map Task dialogue):

(31) E: ... and then go direction that moon lander, that thing on those legs

This stretch of behaviour could be marked up as expressing an *Instruct* act and an *Inform* act which explains the term “moon lander”. However, in order to do that accurately it is best to segment this stretch into two functional segments: *fs1* = “and then go direction that moon lander” and *fs2* = “that thing on those legs”, and to be accurate in assigning the *Instruct* function to segment *fs1* only and the *Inform* function to *fs2* only, rather than assigning these functions to the segment *fs3* consisting of the entire utterance. The more fine-grained segmentation also allows us to indicate the fact that the *Inform* in *fs2* is an explanation of something in the *Instruct* in *fs1*, as follows:

```

<dialogueAct xml:id="da1" target="fs1"
  speaker="s" addressee="a" dimension="task"
  communicativeFunction="instruct"/>
(32) <dialogueAct xml:id="da2" target="fs2"
  speaker="s" addressee="a"
  communicativeFunction="inform" dimension="alloFeedback"/>
<rhetoricalLink dact="#da2" rhetoAntecedent="#da1" rhetoRel="explanation"/>

```

There are cases where the identification of the minimal stretch of behaviour that corresponds to a functional segment is not obvious, in particular when a longer stretch could be said to express a particular function, but where it consists of smaller parts which could also be said to express that same function. Example (33) illustrates this (from a Map Task dialogue):

(33) 1. E: and then you go up and around that, a little to the right
 2. A: slightly northeast?
 3. E: yeah, slightly northeast.

E’s utterance 3 as a whole could be said to constitute a *Confirm* act, but each of the two parts ‘yeah’ and ‘slightly northeast’ could also be said to constitute two separate *Confirms*. Larsson (1998) has recommended in such cases to take a *maximal* approach and choose the larger stretch as the unit of annotation. Alternatively, the use of functional segments naturally suggests to always take a minimal approach. Which of these strategies is to be preferred may be determined by the purpose of the annotation, but clearly the minimal approach is more fine-grained.

A functional segment is most often a part of what is contributed by the participant who occupies the speaker role, distinguished by the fact that this part has a separate communicative function. However, when working from a pre-segmented transcription of a spoken dialogue, the segmentation used in the transcript is not necessarily perfect, or not as one would like it to be.

First, there may be cases where one would prefer a given segment to be segmented into smaller segments. In such a case it is best to assign the various tags that one would prefer to assign to the parts of the segment, to the segment as a whole. This could lead to assigning an inconsistent set of tags to a segment; in that case one either has to omit one or more tags, or temporarily accept the assignment of an inconsistent set of tags, and/or add a comment to the annotation to signal this problem. What is the best strategy in such cases depends on the purposes of the annotation and on the options offered by the annotation tool that is used.

Second, it may happen that a turn has been pre-segmented into certain parts where one would prefer to annotate a longer segment, formed by these parts. In such a case it is recommended to annotate all these parts with the same tags.

Third, a given segment may be 'self-interrupted' by a part that has a different communicative function, as in the following example:

(34) Can you tell me what time the train to *ehm*,... Viareggio leaves?

Here we see a Set Question interrupted by a Stalling segment (*ehm*). The preferred segmentation would distinguish in this case one functional segment in the Task dimension, viz. *fs1* = *Can you tell me what time the train to Viareggio leaves?* and one in the time Management dimension, viz. *fs2* = *ehm*,..., leading to the following representation in DiAML:

```
(35) <dialogueAct xml:id="da1" target="fs1"
      speaker="s" addressee="a" dimension="task"
      communicativeFunction="request" conditionality="conditional"/>
      <dialogueAct xml:id="da2" target="fs2"
      speaker="s" addressee="a"
      communicativeFunction="stalling" dimension="timeManagement"
```

If the segmentation has not distinguished the intervening segment as a separate functional segment, then again, it is best to assign the tags for the intervening segments to the entire segment as a whole.

Fourth, it may happen that a dialogue act corresponds to more than one turn, as in the following example, where the utterances in turns 1 and 3 together form an Answer:

(36) 1. A: There are two flights early in the morning, at 7.45 and at 8.15
 2. B: Yes
 3. A: and two more in the evening, at 7.15 and at 8.30

If the pre-segmentation does not distinguish the segment formed by (36.1) and (36.3) as a single functional segment, but treats them as two separate segments, then it is best to give each of these parts the same tag and a function qualifier that expresses partiality (Answer [partial], in this example), and code them all as having a functional dependency relation with the same question. In this way it is clear that they are all partial answers to the same question.

A.3 Annotation representation in DiAML

According to the abstract syntax of DiAML, as specified in Clause 11, a DiAML annotation structure formally consists of a functional segment and a set of annotations, which contain information about sender, addressee(s), communicative functions, function qualifiers, dimensions, and functional and feedback dependence relations, and rhetorical relations. In order to be ISO-compliant, the concrete representation of these structures assumes a three-level architecture, consisting of:

- a) a primary source, which may correspond to a speech recording, textual transcription or any further low-level annotation thereof;
- b) the marking of functional segments from the primary source;
- c) the dialogue act annotation associated with a functional segment.

Functional segments can be identified by means of the `functionalSegment` element, regardless whether is verbal, nonverbal, or multimodal; the `target` attribute is used to point to a functional segment. For more details see Annex C.

According to the metamodel in Figure 1, a dialogue act has one sender, at least one addressee, possibly other participants, one semantic content category, one communicative function (where a communicative function may have qualifiers), and possibly functional and feedback dependence relations. This is reflected in the concrete XML-based DiAML representation of dialogue act annotation in the fact that a `dialogueAct` element has obligatory attributes `sender`, `addressee`, `communicative function`, and `dimension`, and optionally qualifiers, while the optional functional relations, feedback relations, and rhetorical relations correspond to relational XML elements which may but do not have to be added.

For a given functional segment in a dialogue, the sender and addressee roles are usually easy to assign. For assigning communicative functions, see sections A.3.1 and A.3.2. For assigning dimensions, the decision to be made is which category of information or actions is addressed. Is it (1) concerning the underlying task/activity; or (2) concerning the speaker's processing of previous utterances; or (3) concerning the addressee's processing of previous utterances; or (4) concerning the allocation of the speaker role; or (5) concerning the time needed to continue the dialogue; or (6) concerning the editing of what the speaker is saying; or (7) concerning the editing of what the addressee is currently saying; or (8) concerning the structure of the dialogue; or (9) concerning social obligations?

A.3.1 Encoding general-purpose functions

Information transfer functions

All dialogue acts with an information transfer function have the main purpose of making certain information available to the addressee (acts with an Inform function or a function dominated by Inform in the hierarchy shown in Figure 2) or of the speaker obtaining certain information (the Information-seeking functions in Figure 2). The information to be obtained or made available can be of any kind, relating to the underlying task or activity, or relating to the interaction.

In order to decide whether a segment of dialogue has an information transfer function, an annotator should thus decide whether the segment has such a purpose. If so, the annotator can use the subtrees of the Information-providing and Information-seeking in Figure 2 as decision trees, going systematically left-right through the functions at the next level down and checking the defining conditions that distinguish each of these functions from their ancestor and from each other. Since the functions at one level in a subtree are mutually exclusive, at most one of them applies. If one is found that applies, then go down one level to the functions dominated by this function, and repeat the process. Keep doing this until hitting a level where none of the functions apply. In that case choose the function that dominates the functions at that level.

Action discussion functions

All action discussion functions have in common that their semantic content describes the an action, possibly with specifications of manner or frequency of performance. The actions under discussion can be of any kind: actions for moving the underlying task forward, or actions for managing the interaction, or actions for dealing with social obligations.

This class of communicative functions falls apart into the classes of Commissive and Directives, familiar from speech act theory. Commissive acts all have as their common property that the sender expresses a commitment to performing an action, while directive acts are characterised by the sender having the goal that the addressee commits himself to performing an action. In order to decide whether a segment of dialogue has a commissive or a directive function, an annotator should decide whether the segment has the purpose of expressing or trying to impose such a commitment. If so, the annotator can use the subtrees of Commissives and Directives (see Figure 2) as decision trees, in the same way as for choosing an information transfer function.

A.3.2 Encoding dimension-specific functions

In contrast with general-purpose communicative functions, dimension-specific functions can often be recognised by their use of fixed forms; all the dimensions have particular fixed forms and formulaic expressions.

Auto- and Allo-Feedback

Feedback acts have the purpose of providing or eliciting information about the processing of utterances in dialogue. Both auto- and allo-feedback providing functions are divided into positive and negative ones. Positive feedback is very often expressed implicitly, and should in such a case most probably not be encoded, as argued in A.1.4. Negative feedback is virtually always explicit, and as such easy to recognise. Some of the frequently used fixed forms for negative auto-feedback are *Huh?*, *What?* and equivalent expressions in other languages, and nonverbal signals such as raising eyebrows, frowning, or cupping a hand behind an ear.

Repetitions and rephrases are common forms of auto-feedback. A distinction can be made between the case where the speaker literally repeats (part of) what was said before ('echos') and the case where he rephrases (part of) what was said. For example:

- (37) 1. A: I would like to travel next Saturday, in the afternoon.
 2. B: Next Saturday in the afternoon I have a flight leaving at 16:10.
 3. B: On Saturday May 8 after 12 p.m. I have a flight leaving at 16:10.

In (37.2) B literally repeats part of A's question, thereby displaying what he perceived what A said. In (37.3), by contrast, B paraphrases parts of A's question, and this can be taken to indicate not only what B heard but also how B interpreted what A said (which in this example may be particularly relevant for the interpretation of 'next Saturday'.)

On the other hand, positive feedback is often expressed in a rather inarticulate fashion by fixed forms like *OK* or *Yes*, *Sure*, etc. which may be taken to express overall successful processing of what was said, and correspond to the communicative function `autoPositive`.

It may be worth noting that there is a systematic relation between auto- and allo-feedback acts. This is for the following reason. A dialogue act in the Allo-Feedback dimension is concerned with the addressee's processing of a previous utterance, e.g. A: *What do you think I said?*; when the addressee responds to that, e.g. B: *I thought you said Tuesday* then the speaker of this response is speaking about his own processing of a previous utterance, hence the response is an act in that participant's Allo-Feedback dimension. This is more generally the case: the response to an Allo-Feedback act is an Auto-Feedback act.

The reverse is also true. When a participant A encounters a processing problem and tries to resolve it, as in A: *Do you mean this Saturday?*, and the addressee responds to that like in B: *That's right*, then in the response the speaker is talking about the addressee's processing, hence this is an act in the Allo-Feedback dimension.

Turn Management

Turn management functions are characterised by the sender having the goal to obtain, to keep, or to hand over the speaker role. For an annotator, the issue to decide on is thus whether the sender's behaviour expresses such a goal. Consider, for example, the case of a question-answer pair:

- (38) 1. A: Do you know what time it is?
 2. B: It's nearly twelve fifteen.

Does B, in answering A's question, express the goal to occupy the speaker role? This is not obvious, but it should be noted that B's primary aim is to answer A's question, and that in order to do so he cannot avoid taking the speaker role; this suggests that B did not have a separate goal to have the speaker role.

Similarly, does A, by asking a question, express that he wants B to occupy the speaker role next? The answer to

this question is clearly No, since A can continue for a while occupying the speaker role after asking the question, as in the following example:

- (39) 1. A: Do you know what time it is? I need to catch the twelve seventeen train. Oh dear it's already too late, I see.
2. B: Yes, it's twelve fifteen now.

Note that in example (39) participant A continued in the speaker role simply by continuing to speak. This raises a rather troubling question: does continuing to speak indicate the speaker's goal to keep the turn? In that case, one should assign a turn-keeping function to nearly everything that a speaker says. A recommendation for how to go about assigning turn-management functions would be to only assign such a function to those stretches of communicative behaviour which have the sole (or the main) purpose to obtain, to keep, or to get rid of the speaker role. Just starting to speak, continuing to speak, or ceasing to speak should not be annotated as expressions of Turn management functions.

A particularity of the Turn Management dimension is that the dimension-specific functions are divided into two subclasses, that could in fact be considered as separate dimensions. Usually only the first segment in a turn has a turn-initial function and only the last one a turn-final one. The non-final utterances in a turn do not have a turn-final function, except when the speaker signals (for example by using a rising intonation or a filled pause) that the utterance is not going to be the last one in the turn, that he wants to continue. In that case the utterance has a Turn Keeping function.

When a speaker accepts a turn that the addressee has assigned to him through a Turn Assign act, the relevant segment should be annotated as having the turn-initial function Turn Accept *only* when the speaker performs a *separate act* for the purpose of accepting the turn (such as nodding, or clearing his throat, or saying something like *Yes* or *OK*). The verbal as well as nonverbal activities that a speaker performs to seize the turn should be marked as Turn Grabbing, but the segment that follows *after* he has seized the turn should not be marked as having a turn-initial Turn Management function.

Time Management

Time management functions are concerned with the sender buying some time. Only two cases seem to occur:

- a) the speaker is unable to say immediately what he intended to say (Stalling);
- b) the speaker suspends the dialogue for a while (Pausing).

In both cases there may be several reasons why the sender wants to buy some time. In the first case this is most probably because he is looking for the right words to express what he wants to convey or that he is gathering (or calculating) the information that he wants to convey. In the second case this may be because he is aware that collecting/computing the information requires more time than is reasonable to take while continuing the dialogue, or is too complicated to allow him to continue to participate in the dialogue, or it may be that something more urgent came up, or that a disturbing sound was prohibiting him from continuing to interact.

Stalling acts take the form of filled pauses (*ehm, let me see, well,..*), often occurring together with slowing down and short silences. Pausing acts explicitly claim or request some time: *Just a minute, Wait a second, I'll be right back, etc..* Fully explicit requests like *Please wait while I check the current status* should not be marked as Pausing acts, but as requests in the Time Management dimension, using the general-purpose function `Request`.

Own and Partner Communication Management

In Own Communication Management (OCM) acts the speaker is editing his own speech. The speaker interrupts himself, being aware that he said something wrong, retracts something that he just said (*Oh sorry no,...; Or no wait,..*), and/or replaces something he just said by something else (*I want to travel on Tuesday Thursday*).

Partner Communication Management (PCM) acts similarly edit the addressee's speech, who at that moment occupies the speaker role. Two important cases are the correction of the addressee/current speaker (Correct

Misspeaking) when there was an obvious slip of the tongue, and the completion of what the addressee/current speaker is struggling to say (Completion). In both cases the sender of the PCM act barges in and grabs the turn, or takes the turn which has become available because the addressee/current speaker is hesitating.

Discourse Structuring

Discourse Structuring acts are concerned with the explicit structuring of the dialogue. Such acts occur frequently at the beginning and near the end of a dialogue. A dialogue needs to be opened in some way, and there are conventional ways of doing so. In multi-party dialogue an expression that is frequently used to open the dialogue is *Okay!* The same utterance is often used (though with a different intonation) to indicate that a dialogue can be closed, signaling positive feedback concerning the entire preceding dialogue. There do not seem to exist dialogue acts that have no other function than closing a dialogue; conventionally, a dialogue is considered closed when the participants have exchanged farewell greetings.

During a dialogue, the topic is often changed implicitly, simply by talking about a new topic. This happens especially if the new topic is closely related to the previous one, for instance by being a subtopic of the previous topic, or by being another subtopic of a more general topic. Implicit topic management should not be encoded; the fact that a new topic is addressed is a property of the semantic content of the Inform, the Question, or whatever dialogue is performed which addresses this new topic. Only explicitly signaled topic (actual or intended) shifting should be annotated as such.

Social Obligations Management (SOM)

The kind of social obligations that should be annotated depend on the kind of dialogue. Welcome and farewell greetings that play a role in starting and ending a dialogue are domain-independent, however, as are apologies and their acceptances, acts for introducing oneself, and thanking acts and their acceptances. All of these types of acts have conventional forms ('formulas') in every language. They also tend to come in pairs: an initial greeting puts pressure on the addressee to send a response greeting; introducing oneself puts pressure on the addressee to also introduce himself; an apology puts pressure on the addressee to accept the apology; a thanking puts pressure on the addressee to downplay what he is thanked for (like in *It was nothing; It was my pleasure*); and a farewell greeting puts pressure on the addressee to produce a response farewell greeting.

SOM acts can also be performed by means of general-purpose functions. For instance, *I'm extremely grateful for your help* and *I hope to see you next year in Hong Kong* are Informs in the SOM dimension.

It is worth noting that utterances which serve a 'social' purpose such as greetings, thanks, and apologies are often used to serve other purposes as well. Greetings like *Hello!*, for example, can be used also for opening a dialogue (a Dialogue Structuring function). Also, an expression of thanks can be used to signal that the speaker intends to terminate the dialogue, and can also be used for positive feedback.

A.3.3 Encoding communicative function qualifiers

Function qualifiers are available in DiAML for encoding various ways in which a speaker can specify certain conditions, qualifications, or feelings accompanying a dialogue act. For the encoding of certainty, of conditionality, and of partiality DiAML has binary-valued attributes one of which is the default value. For the encoding of feelings the `sentiment` attribute is available which has an open class of values and no default value; if no value of the attribute is specified in an annotation this means that no such information is present.

a. Certainty The sender of a dialogue act can express certainty or uncertainty about the correctness of the information provided in an information-providing act, or about his commitment to perform an action in a commissive act. This is illustrated in (40) for information-providing acts, where the expressions "*I have a hunch that*", "*probably*", "*might*", and "*I'm not sure if*" are indicators of the speaker's uncertainty. When these expressions are omitted, as in (41), the resulting sentences no longer contain any suggestion that the speaker is uncertain about the correctness of what he says. This illustrates that the default value, corresponding to the unmarked case, is *certain*.

- (40) 1. A: Do you know who'll be coming tonight?
 2. B: I have a hunch that Mary won't come.
 3. B: Peter, Alice, and Bert will probably come.
 4. B: I heard that Tom and Anne might come.
 5. B: I'm not sure if Bill will come.

- (41) 1. A: Do you know who'll be coming tonight?
 2. B: Mary won't come.
 3. B: Peter, Alice, and Bert will come.
 4. B: I heard that Tom and Anne [will] come.
 5. B: Bill will come.

Speakers may also signal being *very certain*, as exemplified in (42). For such cases, the DiAML encoding with `certainty="certain"` is recommended,

- (42) 1. Mary will definitely not come.
 2. Peter, Alice, and Bert will come for sure.
 3. I certainly agree with that.

In commissive acts, the speaker may signal uncertainty or great certainty concerning the commitment to perform certain actions. The examples in (43) show certain and uncertain promises, and uncertain acceptances of an offer and a request.

- (43) 1. I'll probably come. [*uncertain promise*]
 2. I'll definitely call you. [*certain promise*]
 3. Some coffee? That might be nice. [*uncertain accept offer*]
 4. Could you take us through this? I guess so. [*uncertain accept request*]

Summarizing, for the coding of a sender's certainty associated5. B: I'm not sure if Bill will come. with the performance of an information-providing act or a commissive act, look for expressions of uncertainty and expressions of great certainty. Certainty and the lack thereof are not only indicated by verbal expressions, but also by prosody gaze direction, and several types of gestures. Prominent nonverbal expressions of uncertainty include gaze aversion, head waggles, lip pouting, lowering eyebrows, and self-touching.

Warning: verbal expressions of uncertainty, in particular adverbs, should sometimes be interpreted as part of the semantic content of a dialogue act, rather than as a qualification of the communicative function. The following examples illustrate this:

- (44) 1. I'll probably come around eight o'clock.
 2. I'll definitely come before nine.

In these examples, *probably* and *definitively* apply to the time that is mentioned, not to the sender's certainty about his commitment to come.

For deciding whether to use a certainty qualifier in the annotation of a functional segment, the decision tree shown in Figure 3 can be used.

b. Partiality The qualifier *partial* is available in DiAML to encode that a speaker reacts to only part of the semantic content of a previous dialogue act, as illustrated in (45) by partial acceptance of an offer, partial agreement with the previous speaker, and a partial answer of a question..

- (45) a. 1. A: Can I offer you some coffee and chocolates?
 2. B: Only coffee please.

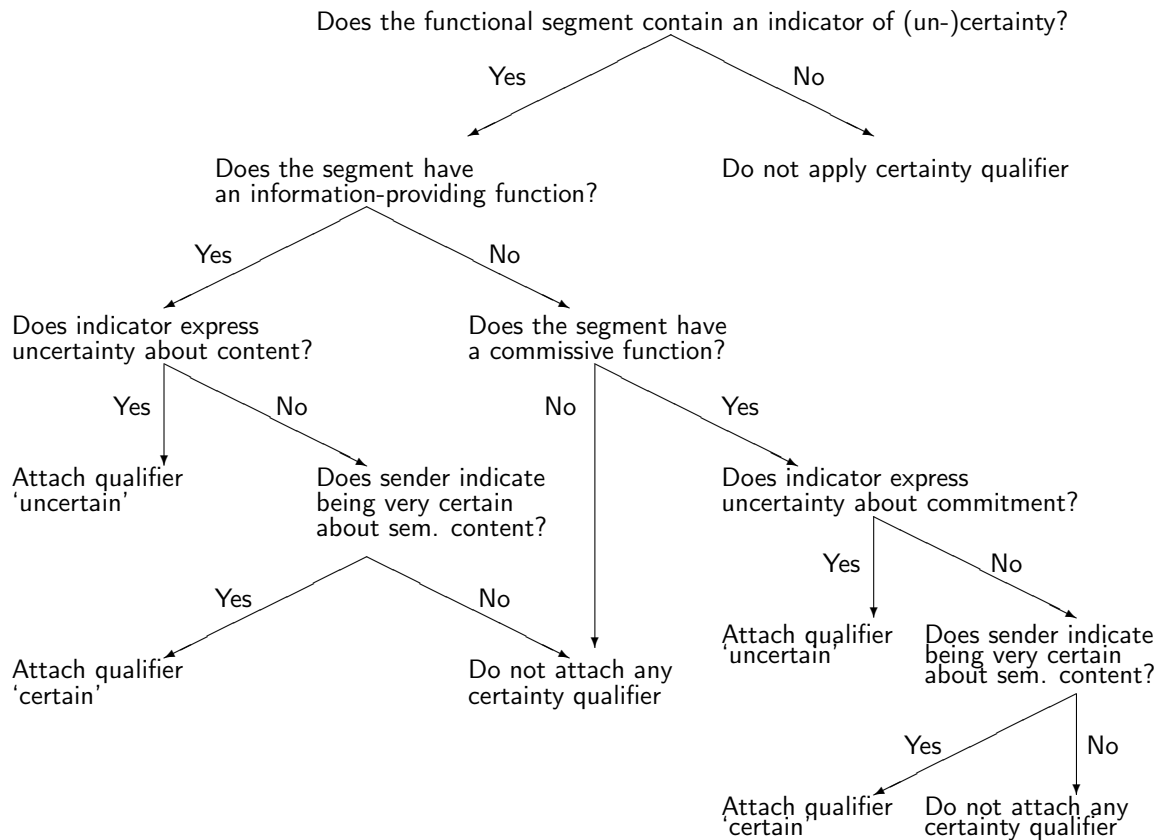


Figure 3 — Decision tree for applying certainty qualifiers

- b. 1. A: The new student is brilliant and imaginative.
 2. B: He's certainly imaginative
- c. A: Do you have any rooms available on Saturday the 26th, and at what price?
 B: Yes, I have rooms available. How many do you need?

An expression such as *only* is a potential clue of partiality, but in general one has to trace the previous dialogue act to which the current one is a response, and see whether the response takes all or only part of the semantic content of the previous act into account.

c. Conditionality Conditionality refers to the possibility (with respect to ability and power), the necessity, or the willingness to perform an action; the qualifiers *conditional* and *unconditional* can therefore be attached to action-discussion functions and to responses to dialogue acts with such a function. The following examples illustrate this phenomenon.

- (46) a. A: Would you like to have some coffee?
 B: Thanks, only if you have it ready.
- b. A: Can you do the presentation, if you're ready?
 B: I can do that if you like.
- c. A: I'll send you an email if you give me your address.

- d. A: Can we just go over that again?
 B: Just very quickly. I have to hurry you on here.
 C: I don't think we have time for that, unless you make it very short.
- e. A: I can make the buttons larger.
 B: No, only if we want basic things to be visible.

In (46a) we see the conditional acceptance of an offer; in (46b) a conditional request, with a conditional acceptance; in (46c) a conditional promise; in (46d) a two conditional acceptances of a request; and in (46e) a conditional rejection of a suggestion. Similar to the case of certainty qualifiers, omission of the expressions indicating a condition leads to expressions that signal unconditional dialogue acts, hence the default value is *unconditional*, and does not need to be marked up. Explicit expressions of 'unconditionality' are hard to find, other than the adverb *unconditional*, which is hardly ever used in natural dialogue.

Conditional dialogue acts can often be recognised by the use of conditional expressions such as *if ...* or *unless*, and *just* (as in (46d), first case) but just like in the case of certainty, these expressions can also be part of the semantic content rather than qualifiers of the communicative function. For deciding whether to add a conditionality qualifier to the annotation of a communicative function, the decision tree can be used which is displayed in Figure 4.

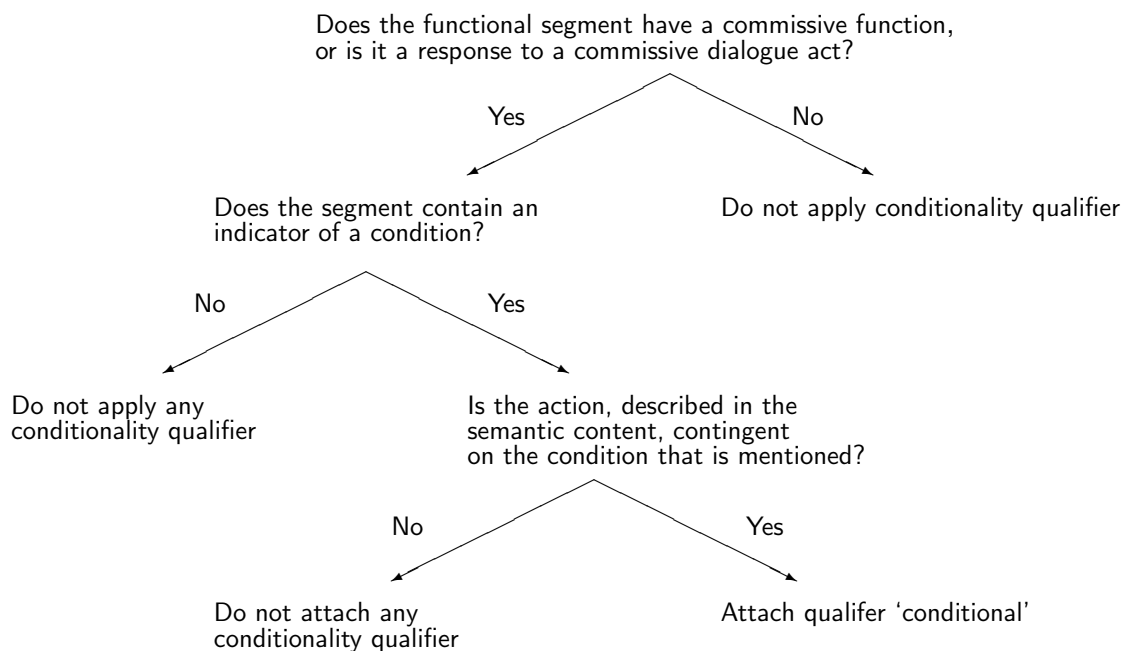


Figure 4 — Decision tree for applying conditionality qualifiers

d. Sentiment A particular sentiment associated with the performance of a dialogue act may be annotated if the sender indicates an emotion or an attitude concerning the semantic content or the addressee, verbally or nonverbally, or both. Example (18) shows some verbal expressions of sentiment. Nonverbal expressions of sentiment exist in abundance and in great variety, including for instance smiling (happiness), eyebrow raising (surprise), pressing lips together (angst), and sighing (sadness). Specific guidelines for sentiment annotation cannot be given here, in view of the class of sentiment qualifiers not being specified in this standard.

A.3.4 Encoding functional dependences, feedback dependences, and rhetorical relations

Functional dependence

A dialogue act A1 is functionally dependent on a previous dialogue act A2 (its 'functional antecedent'), if its

communicative function by its very nature responds to another dialogue act. This is the case for the following core communicative functions defined in this standard:

- (47) - Answer, Confirm, Disconfirm;
 - Agreement, Disagreement, Correction;
 - Address Request, Accept Request, Decline Request;
 - Address Suggestion, Accept Suggestion, Decline Suggestion;
 - Address Offer, Accept Offer, Decline Offer;
 - Turn Accept;
 - Return Greeting, Return Self-introduction, Accept Apology, Accept Thanking, Return Goodbye

Encoding a functional dependence relation means identifying the functional antecedent and linking the two dialogue acts by means of a `functionalDependence` element. The identification of a functional antecedent is not straightforward if (a) the current dialogue act does not respond to a single dialogue act but to a combination of dialogue acts, as in (48), or (b) responds to an implicit dialogue act.

- (48) 1. U: Can you what time there are trains from Harwich to York?
 2. S: What day would you like to travel?
 3. U: Tomorrow morning.
 4. U: On Tuesday morning there are trains at 6:45, 70:30,...(etc.)

In (48), utterance 4 forms a functional segment with function Answer, which responds to the question formed by the dialogue acts expressed by utterances 1 and 3 together. In such a case it is recommended to mark functional dependence relations to both these dialogue acts.

The case (of responding to an implicit dialogue acts is illustrated by (54) in Annex B, where the dialogue system operates on the assumption that the user has a question about train journeys, and queries the user for parameter values until it believes it knows the user's question, which it subsequently answers. This question is not explicit anywhere in the dialogue, and in such a case it is best not to mark up a functional dependence relation.

Feedback dependence

Every auto- or allo-feedback act is about the processing of one or more previous dialogue segments, and therefore has a feedback relation to these segments. This is the case both for feedback acts that have a dimension-specific communicative function (i.e. Auto-Positive, Auto-Negative, Allo-Positive, Allo-Negative, or Feedback Elicitation) and for feedback acts with a general-purpose function.

Encoding a feedback dependence relation means identifying the functional segment(s) that the feedback is about, and linking the dialogue act to these segment(s) by means of a `feedbackDependence` element. For feedback acts with an Auto-Positive or Allo-Positive function the feedback is usually about the previous utterance from another participant, but sometimes the feedback is more global, and can refer to everything that happened so far in the dialogue - in such a case it is best not to annotate a feedback dependence.

Rhetorical relations

Many of the relations which may occur between units in discourse such as Justification, Explanation, Cause-Effect, or Summarization, and which in the linguistic literature are often called 'rhetorical relations' or 'discourse relations', may also occur between dialogue acts. This standard does not specify any particular set of such relations, and therefore does not provide detailed guidelines for their encoding. So-called 'discourse markers'; like *also*, *but*, *because*, *for example*, *so* often signal such relations as Elaboration, Motivation (or Justification, or Cause), Exemplification, and Conclusion (or Summarization), and they are often multifunctional; for example, a protracted turn-initial *An..d,...* may be a functional segment with the functions `Turn Take`, `Stalling`, `Auto-Positive`, and also be the first part of a longer functional segment expressing a dialogue act which has an Elaboration relation to a previous dialogue act. (See Petukhova and Bunt, 2009a on multifunctionality of discourse markers, and Hovy and Maier, 1995; and Mann and Thompson, 1988 more generally on discourse relations and rhetorical relations.)

Annex B (normative) Completely annotated examples

This annex contains examples of dialogue fragments annotated with dialogue act information, represented in DiAML. The first subsection contains annotations of some simple examples of very short dialogue fragments. The second subsection contains the annotation of a complete spoken human-computer dialogue (from a Wizard-of-Oz experiment), and an extended fragment of a multimodal human-human dialogue from the HCRC Map Task corpus.

B.1 Short dialogue fragments

Example (49) is a two-turn dialogue fragment, which is also used in Annex C. Each of the two turns constitutes a single functional segment in only one dimension (Task), and does not display any multifunctionality.

For the anchoring of DiAML annotations to the primary text, it is assumed that these two functional segments are defined at another level of analysis (see Annex C) as having the XML identifiers "fs1" and "fs2", respectively. In this case, fs1 is everything said by participant P1 in turn 1, and fs2 is everything said by P2 in turn 2. What P1 says is in this case interpreted as an indirect request to tell where he should check in; this type of indirect request can be annotated according to this standard as a conditional request (*Please tell me where I should check in for Munich, if you have that information*); this is represented in DiAML as a dialogue act with the communicative function `Request` with the qualifier `conditional`. Participant P2 responds by providing the information that P1 wants to obtain, hence this segment should be annotated as an `Answer` which has a functional dependence relation to P1's request. This is illustrated in (49b).

- (49) a. 1. P1: Do you know where I should check in for Munich?
2. P2: For Munich go to counters 31 to 40.

```

<diaml xmlns:"http://www.iso.org/diaml/">
  <dialogueAct xml:id="da1" target="#fs1"
    speaker="#p1" addressee="#p2"
    communicativeFunction="setQuestion" dimension="task"
    conditionality="conditional"/>
  b. <dialogueAct xml:id="da2" target="#fs2"
    speaker="#p2" addressee="#p1"
    communicativeFunction="answer" dimension="task"/>
  <functionalDependence dact="#da2" functAntecedent="#da1"/>
</diaml>

```

Example (50) is again a two-turn dialogue fragment. The second turn, P2's utterance, is segmented into two overlapping functional segments: one in the Auto-Feedback dimension and one in the Task dimension. The functional segment in the Task dimension is an answer to the question in the first turn, and it is qualified as 'uncertain' since the speaker signals his uncertainty about the correctness of the answer he provides.

1. P1: Do you know what time the next train to Utrecht leaves?
TA fs1: Do you know what time the next train to Utrecht leaves?
- (50) a. 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.

```

<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"
b.  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact="#da2" fbSegment="#fs1"/>
<dialogueAct xml:id="da3" target="#fs2"
  sender="#p2" addressee="#p1"
  communicativeFunction="answer" dimension="task"/>
<functionalDependence dact="#da2" functAntecedent="#da1"/>
</diaml>

```

Example (51), a three-turn dialogue fragment of a dialogue from the HCRS Map Task corpus, again has turns coinciding with functional segments. The example illustrates the use of general-purpose functions for addressing another dimension than that of the task. Participant P1 checks that he understood the previous instruction correctly, producing a Check Question in the Auto-Feedback dimension. Participant P2 confirms P1's understanding, thus addressing P1's processing of that same instruction, i.e. performing a Confirm act in the Allo-Feedback dimension.

The example also illustrates the use of feedback and functional dependency relations. P1's contribution has a feedback relation to the functional segment ("fs1") expressing the previous instruction, while P2's contribution has both a functional dependency relation to the Check Question that it reacts to, and a feedback dependency relation to the functional segment of that same previous instruction

- (51) a. P1: Move up
 P2: Slightly northeast?
 P1: Yeah very slightly.
 (From HCRC Map Task, Carletta et al., 1998)

```

<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da2" target="#fs2"
  sender="#p2" addressee="#p1"
b.  communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<feedbackDependence dact="#da2" fbSegment="fs1"/>
<dialogueAct xml:id="da3" target="#fs3"
  sender="#p2" addressee="#p1"
  communicativeFunction="confirm" dimension="alloFeedback"/>
<functionalDependence dact="da3" functAntecedent="#da2"/>
</diaml>

```

Example (52) shows a two-turn dialogue fragment of a dialogue from the TRAINS corpus (Allen et al., 1994) which illustrates the use of a dimension-specific function (Correct Misspeaking) in the dimension of Partner Communication Management (PCM). Notice that PCM act refer to what the main speaker is doing at that moment, as opposed to allo-feedback acts, which refer to what was said in a previous turn. Still, the relation between the Correct Misspeaking act and the functional segment that it refers to is of the same nature as the relation between a feedback act and its trigger, so we use the same 'feedback dependence' relation to indicate this relation.

- (52) a. P1: engine E3 is going to pick up the bananas, back to Avon, dro...
 P2: to pick up the oranges

```

<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
b. <dialogueAct xml:id="da2" target="#fs2"
  sender="#p2" addressee="#p1"
  communicativeFunction="correctMisspeaking"
  dimension="partnerCommunicationManagement"/>
<feedbackDependence dact= "#da2" fbSegment="#fs1"/>
</diaml>

```

Example (53), taken from a dialogue in the AMI multi-party dialogue corpus, shows the use of a general-purpose function in the Turn Management dimension, and the annotation of a functional dependence relation.

- (53) a. P1: Would you like to say something at this point?
 P2: Certainly.

```

<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1"
  sender="#p1" addressee="#p2"
  communicativeFunction="offer" dimension="turnManagement"/>
b. <dialogueAct xml:id="da2" target="#fs2"
  sender="#p2" addressee="#p1"
  communicativeFunction="acceptOffer" dimension="turnManagement"/>
<functionalDependence dact= "#da2" functAntecedent="#fs1"/>
</diaml>

```

B.2 Annotated dialogues

B.2.1 Annotated human-computer dialogue

The dialogue which is analyzed here has been taken from the OVIS corpus of telephone dialogues with an experimental system of the Dutch railways. The dialogues was originally in Dutch and has been translated into English.

In (54) the transcription of the dialogue is shown with its segmentation into functional segments. Segment boundaries are indicated by '|'. Horizontal lines have been added as turn separators for improved readability.

- | | |
|------|--|
| (54) | <hr/> 1. S: good morning, this is the automatic information system of Public
fs1.1 SOM InitialGreeting
Transport Travel Information
fs1.2 SOM SelfIntroduction
this system can only give information about train travel.
fs1.3 Task Inform
from which station to which station do you want to travel?
fs1.4 Task SetQuestion
<hr/> 2. U: from Harwich to York.
fs2 Task Answer
<hr/> 3. S: when do you want to travel from Harwich to York?
fs3.1 = "when do you want to travel from Harwich to York"
fs3.2 = "from Harwich to York"
fs3.1 Task SetQuestion
fs3.2 AutoFb autoPositive
<hr/> |
|------|--|

4.	U:	ehm... Monday evening		
	fs4.1	TimeM		Stalling
	fs4.2	Task		Answer
5.	S:	so you want to leave between eighteen and twenty-four hours?		
	fs5	AutoFb		CheckQuestion
6.	U:	ehm yes, Monday evening		
	fs6.1	AlloFb		Answer [uncertain]
	fs6.2	AlloFb		Inform
7.	S:	you said that you do not want to leave on Thursday April twenty-six between eighteen and twenty-four hours,		
	fs7.1	AutoFb		inform
		can you please correct me?		
	fs7.2	AutoFb		Request
8.	U:	yes I would like to leave on monday... April twenty-six.		
	fs8.1	AlloFb		AcceptRequest
	fs8.2	Task		Inform
	fs8.3	Task		Inform
9.	S:	what time do you want to travel the day after tomorrow?		
	fs9.1	= "what time do you want to travel the day after tomorrow?"		
	fs2.2	= "the day after tomorrow"		
	fs9.1	Task		SetQuestion
	fs9.2	AutoFb		autoPositive
10.	U:	at ehm... around eight o'clock		
	fs10.1:	<i>at [...] around eight o'clock</i>		
	fs10.2	Task		Answer
	fs10.2	TimeM		Stalling
11.	S:	so you want to leave at eight o'clock in the morning?		
	fs11	AutoFb		CheckQuestion
12.	U:	oh yes sorry eight o'clock in the evening.		
	fs12.1	AutoFb		Inform
	fs12.2	SOM		IApology
	fs12.3	AlloFb		Disconfirm
13.	S:	so you want to leave at twenty o'clock?		
	fs13	AutoFb		CheckQuestion
14.	U:	yes		
	fs14	AlloFb		Confirm
15.		I have found the following connection.		
	fs15.1	DiscourseStructuring	InteractionStructuring	
		Departure from Harwich at nineteen fifty-six arrival in London Waterloo at twenty zero four, continuing from there with the train departing at twenty fourteen arrival in Birmingham Central at twenty ten. Continuing from there with the train departing at twenty sixteen arrival in York at twenty forty.		
	fs15.2	Task		Answer
		would you like me to repeat the connection once more?		
	fs15.3	AlloFb		Offer
16.	U:	oh no thank you		
	fs16.1	AutoFb		DeclineOffer
	fs16.2	SOM		Thanking
17.	S:	would you like to know another connection?		
	fs17	Task		Offer

18.	U:	do you have another connection?	fs18	Task	PropositionalQuestion
19.	S:	I will have a look for you	fs19.1	TimeM	Promise
	.	no, I don't think that there is something earlier	fs19.2	Task	Answer [uncertain]
20.	U:	is there an earlier possibility?	fs20	Task	PropositionalQuestion
21.	S:	I'm sorry there's no earlier possibility.	fs21.1	SOM	Apology
			fs21.2	Task	Answer
22.	U:	thank you very much	fs22	SOM	Thanking
23.	U:	Public Transportation Information wishes you a pleasant journey	fs23.1	SOM	InitialGoodbye
		goodbye	fs23.2	SOM	initialGoodbye

```

<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fsl.1"
  sender="#s" addressee="#u"
  communicativeFunction="initGreeting"
  dimension="socialObligationsManagement"
<dialogueAct xml:id="da2" target="#fsl.2"
  sender="#s" addressee="#u"
  communicativeFunction="selfIntroduction"
  dimension="socialObligationsManagement"
<dialogueAct xml:id="da3" target="#fsl.3"
  sender="#s" addressee="#u"
  communicativeFunction="inform" dimension="task"
<dialogueAct xml:id="da4" target="#fsl.4"
  sender="#s" addressee="#u"
  communicativeFunction="setQuestion" dimension="task"
(55) <functionalDependence dact="#da5" functAntecedent="#da4"/>
<dialogueAct xml:id="da6" target="#fs3.1"
  sender="#s" addressee="#u"
  communicativeFunction="setQuestion" dimension="task"
<dialogueAct xml:id="da7" target="#fs3.2"
  sender="#s" addressee="#u"
  communicativeFunction="autoPositive" dimension="autoFeedback"
<feedbackDependence dact="#da7" fbSegment="#fs2"/>
<dialogueAct xml:id="da8" target="#fs4.1"
  sender="#u" addressee="#s"
  communicativeFunction="stalling" dimension="timeManagement"
<dialogueAct xml:id="da9" target="#fs4.2"
  sender="#u" addressee="#s"
  communicativeFunction="answer" dimension="task"
<functionalDependence dact="#da9" functAntecedent="#da7"/>
<dialogueAct xml:id="da10" target="#fs5"
  sender="#s" addressee="#u"
  communicativeFunction="checkQuestion" dimension="autoFeedback"
<feedbackDependence dact="#da10" fbSegment="#fs4.2"/>

```

```

<dialogueAct xml:id="da11" target="#fs6.1"
  sender="#u" addressee="#s"
  communicativeFunction="answer" dimension="alloFeedback"
  qualifier="uncertain"
<functionalDependence dact="#da11" functAntecedent="#da10"/>
<dialogueAct xml:id="da12" target="#fs6.2"
  sender="#u" addressee="#s"
  communicativeFunction="inform" dimension="alloFeedback"


---


<dialogueAct xml:id="da13" target="#fs7.1"
  sender="#s" addressee="#u"
  communicativeFunction="checkQuestion" dimension="autoFeedback"
<feedbackDependence dact="#da13" fbSegment="#fs6.2"/>
<dialogueAct xml:id="da14" target="#fs7.2"
  sender="#s" addressee="#u"
  communicativeFunction="request" dimension="autoFeedback"
<feedbackDependence dact="#da14" fbSegment="#fs6.2"/>


---


<dialogueAct xml:id="da15" target="#fs8.1"
  sender="#u" addressee="#s"
  communicativeFunction="acceptRequest" dimension="alloFeedback"
<functionalDependence dact="#da15" functAntecedent="#da14"/>
<dialogueAct xml:id="da16" target="#fs8.2"
  sender="#u" addressee="#s"
  communicativeFunction="inform" dimension="task"
<dialogueAct xml:id="da17" target="#fs8.3"
  sender="#u" addressee="#s"
  communicativeFunction="inform" dimension="task"


---


<dialogueAct xml:id="da18" target="#fs9.1"
  sender="#s" addressee="#u"
  communicativeFunction="setQuestion" dimension="task"
<dialogueAct xml:id="da19" target="#fs9.2"
  sender="#s" addressee="#u"
  communicativeFunction="autoPositive" dimension="autoFeedback"
<feedbackDependence dact="#da18" fbSegment="#fs8.3"/>


---


<dialogueAct xml:id="da20" target="#fs10.1"
  sender="#u" addressee="#s"
  communicativeFunction="answer" dimension="task"
<functionalDependence dact="da20" functAntecedent="da18"/>
<dialogueAct xml:id="da21" target="#fs10.2"
  sender="#u" addressee="#s"
  communicativeFunction="stalling" dimension="timeManagement"


---


<dialogueAct xml:id="da22" target="#fs11"
  sender="#s" addressee="#u"
  communicativeFunction="checkQuestion" dimension="autoFeedback"
<feedbackDependence dact="#da22" fbSegment="#fs10.2"/>


---


<dialogueAct xml:id="da23" target="#fs12.1"
  sender="#u" addressee="#s"
  communicativeFunction="inform" dimension="autoFeedback"
<dialogueAct xml:id="da24" target="#fs12.2"
  sender="#u" addressee="#s"
  communicativeFunction="apology"
  dimension="socialObligationsManagement"
<dialogueAct xml:id="da25" target="#fs12.3"
  sender="#u" addressee="#s"
  communicativeFunction="disconfirm" dimension="alloFeedback"
<functionalDependence dact="da25" functAntecedent="da22"/>


---



```

```

<dialogueAct xml:id="da26" target="#fs13"
  sender="#s" addressee="#u"
  communicativeFunction="checkQuestion" dimension="autoFeedback"
</dialogueAct>
<dialogueAct xml:id="da27" target="#fs14"
  sender="#u" addressee="#s"
  communicativeFunction="confirm" dimension="alloFeedback"
</dialogueAct>
<functionalDependence dact="da27" functAntecedent="da25"/>
<dialogueAct xml:id="da28" target="#fs15.1"
  sender="#s" addressee="#u"
  communicativeFunction="interactionStructuring"
  dimension="discourseStructuring"
</dialogueAct>
<dialogueAct xml:id="da29" target="#fs15.2"
  sender="#s" addressee="#u"
  communicativeFunction="answer" dimension="task"
</dialogueAct>
<dialogueAct xml:id="da30" target="#fs15.3"
  sender="#s" addressee="#u"
  communicativeFunction="offer" dimension="alloFeedback"
</dialogueAct>
<dialogueAct xml:id="da31" target="#fs16.1"
  sender="#u" addressee="#s"
  communicativeFunction="declineOffer" dimension="autoFeedback"
</dialogueAct>
<functionalDependence dact="da31" functAntecedent="da35"/>
<dialogueAct xml:id="da32" target="#fs16.2"
  sender="#u" addressee="#s"
  communicativeFunction="thanking"
  dimension="socialObligationsManagement"/>
<dialogueAct xml:id="da33" target="#fs17"
  sender="#s" addressee="#u"
  communicativeFunction="offer" dimension="task"/>
<dialogueAct xml:id="da34" target="#fs18"
  sender="#u" addressee="#s"
  communicativeFunction="propositionalQuestion" dimension="task"/>
<dialogueAct xml:id="da35" target="#fs19.1"
  sender="#s" addressee="#u"
  communicativeFunction="promise" dimension="timeManagement"/>
<dialogueAct xml:id="da36" target="#fs19.2"
  sender="#s" addressee="#u"
  communicativeFunction="answer" dimension="task"
  qualifier="uncertain"/>
<functionalDependence dact="da36" functAntecedent="da34"/>
<dialogueAct xml:id="da37" target="#fs20"
  sender="#u" addressee="#s"
  communicativeFunction="propositionalQuestion" dimension="task"/>
<dialogueAct xml:id="da38" target="#fs21.1"
  sender="#s" addressee="#u"
  communicativeFunction="apology"
  dimension="socialObligationsManagement"/>
<dialogueAct xml:id="da39" target="#fs21.2"
  sender="#s" addressee="#u"
  communicativeFunction="answer" dimension="task"/>
<functionalDependence dact="da39" functAntecedent="da37"/>
<dialogueAct xml:id="da39" target="#fs22"
  sender="#u" addressee="#s"
  communicativeFunction="thanking"
  dimension="socialObligationsManagement"/>

```



```

<dialogueAct xml:id="da40" target="#fs23.1"
  sender="#s" addressee="#u"
  communicativeFunction="initGoodbye"
  dimension="socialObligationsManagement"/>
<dialogueAct xml:id="da41" target="#fs23.2"
  sender="#s" addressee="#u"
  communicativeFunction="initGoodbye"
  dimension="socialObligationsManagement"/>
</diaml>

```

B.2.2 Annotated Map Task dialogue

The following excerpt from a dialogue in the HCRC Map Task corpus (Carletta et al., 1996), illustrates the occurrence of nonverbal and multimodal segments. There are several occurrences of heavy breathing (in or out) which may have a communicative meaning; in the transcription these are indicated by VOC_intbreath and VOC_outbreath, respectively, where the prefix VOC_ indicates that this is a sample of vocal (nonverbal) behaviour. In turn 11 there is an occurrence of a lip smacking gesture, indicated in the transcription similarly by LIPGES_lipsmack. In the latter case, the relevant functional segment of the sender's behaviour is multimodal, consisting of (1) a verbal segment, where the sender says *erm* in a very slow fashion, surrounded by periods of silence; (2) the smacking of the lips; and (3) heavily breathing in. This is an illustration of the phenomenon, discussed in section 8, that a functional segment in general has several components consisting of sender behaviour in various communicative channels, together making up a multimodal unit. The encoding of functional segments is not part of the present standard, and may be determined by the particular annotation tool that is used, but might for example look as in (56). In this particular example the vocal (but nonverbal) behaviour and the characterisation of lip gestures are described simply by named values; in other cases, like head gestures, the representation will be more complex and involve the representation of several features such as duration, direction, speed, and number of repetitions.

```

(56) <functionalSegment xml:id="fs1"
      textSegment="#ts1"
      vocalSegment="#vs1"
      lipSegment="#ls1"/>
      <textSegment xml:id="ts1" from="w11.16" to="w11.17"/>
      <vocalSegment xml:id="vs1" value="inbreath"/>
      <lipSegment xml:id="ls1" value="lipsmack"/>

```

- (57)
- | | |
|-------|--|
| 1. | P1: okay, starting off, we are .. above .. a caravan park |
| | fs1.1 TurnM: turnTake |
| | DS: opening |
| | fs1.2 DS: interactionStructuring |
| | fs1.3 Task: inform |
| <hr/> | |
| 2. | P2: mmhmm |
| | fs2 autoFb autoPositive |
| <hr/> | |
| 3. | P1: we are going to go due south NONVOC_noise ... # |
| | fs3.1 Task: inform |
| | fs3.2 TimeM: stalling |
| | TurnM: turnKeep |
| | straight south ... and NONVOC_noise ... then we're going to g--. |
| | fs3.3 OCM: Self-Correction |
| | fs3.4 TimeM: Stalling |
| | TurnM: TurnKeep |
| | fs3.5 Task: Instruct |

	fs3.6	<i>g-</i> OCM: Retraction .. turn straight back round and head north... past an old mill ... on the right ... hand side fs3.7: Task: Instruct
4.	P2:	VOC_outbreath ... B due south and then back up again fs4.1 TurnM: TurnTake fs4.1 TimeM: Stalling fs4.2 AutoFb: CheckQuestion
5.	P1:	yeah south and then straight back up again fs5.1 AlloFb: Confirm fs5.2 Task: Instruct with an old mill on the right fs5.3 Task: Inform [Elaborate] and you're going to pass on the left-hand side of the mill fs5.4 Task: Instruct
6.	P2:	right okay fs6 AutoFb: AutoPositive TurnM: TurnTake
7.	P1:	okay and then we're going to turn ... VOC_inbreath east fs7.1 = "and then" fs7.2 = "we're going to turn east" fs7.3 = "turn ... VOC_inbreath" fs7.1 AutoFb: AutoPositive TurnM: TurnGrab fs7.2 Task: Instruct fs7.3 TimeM: Stalling TurnM: TurnKeep
8.	P2:	mmhmm fs8 AutoFb: AutoPositive
9.	P1:	not ... straight east ... slightly sort of northeast ... fs9.1 Task: Inform VOC_nonvocal ... fs9.2 TimeM: Stalling TurnM: TurnKeep
10.	P2:	s- slightly northeast fs10.1 TurnM: TurnGrab fs10.2 AutoFb: CheckQuestion
11.	P1:	slightly slightly yeah very slightly VOC_inbreath ... and fs11.1 Task: Confirm fs11.2 Task: Inform fs11.3 TimeM: Stalling TurnM: TurnKeep we're going to continue straight along #... GES_lipsmack VOC_inbreath ... erm ... quite a wee distance fs11.4 = "we're going to continue straight along quite a wee distance" fs11.5 = "... GES_lipsmack VOC_inbreath" fs11.4 Task: Instruct fs11.5 TimeM: Stalling TurnM: TurnKeep on that course and then we're going to turn north again fs11.6 Task: Instruct
12.	P2:	right mmhmm fs12.1 AutoFb: AutoPositive fs12.2 AutoFb: AutoPositive

13. P1: NONVOC_noise and ... | immediat- | well |
 fs13.1 TurnM: TurnTake
 TimeM: Stalling
 fs13.2 OCM: Retraction
 fs13.3 TurnM: TunKeep
 a distance below that turning point there's a fenced meadow
 fs13.4 Task: Inform
 | ... VOC_inbreath ... | but you should be avoiding that by quite
 fs13.5 TimeM: Stalling
 TurnM: TurnKeep
 a distance
 fs13.6 Task: Instruct
-
14. P2: okay
 fs14 AutoFb: AutoPositive
-
15. P1: okay | so we've turned | and we're going up north again
 fs15.1 AutoFb: AutoPositive
 fs15.2 Task: Inform
 fs15.3 Task: Instruct

```

(58) <diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#fs1.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnTake" dimension="turnManagement"/>
<dialogueAct xml:id="da2" target="#fs1.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="opening" dimension="discourseStructuring"/>
<dialogueAct xml:id="da3" target="#fs1.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="interactionStructuring"
  dimension="discourseStructuring"
</dialogueAct>
<dialogueAct xml:id="da4" target="#fs1.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da5" target="#fs2"
  sender="#p2" addressee="#p1"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da5" fbSegment="#fs1.3"/>
<dialogueAct xml:id="da6" target="#fs3.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da7" target="#fs3.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da8" target="#fs3.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da9" target="#fs3.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="selfCorrection" dimension="ownCommManagement"/>

```

```

<dialogueAct xml:id="da10" target="#fs3.4"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da11" target="#fs3.4"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da12" target="#fs3.5"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da13" target="#fs3.6"
  sender="#p1" addressee="#p2"
  communicativeFunction="retraction"
  dimension="ownCommManagement"/>
<dialogueAct xml:id="da14" target="#fs3.7"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>


---


<dialogueAct xml:id="da15" target="#fs4.1"
  sender="#p2" addressee="#p1"
  communicativeFunction="turnTake" dimension="turnManagement"/>
<dialogueAct xml:id="da16" target="#fs4.2"
  sender="#p2" addressee="#p1"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da17" target="#fs4.3"
  sender="#p2" addressee="#p1"
  communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<feedbackDependence dact= "#da17" fbSegment="#fs3.6"/>


---


<dialogueAct xml:id="da18" target="#fs5.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="confirm" dimension="alloFeedback"/>
<functionalDependence dact= "#da18" functAntecedent="#fs4.2"/>
<dialogueAct xml:id="da19" target="#fs5.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da20" target="#fs5.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da20" rhetRelatum="#da19" rhetoRel="elaborate"/>
<dialogueAct xml:id="da21" target="#fs5.4"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>


---


<dialogueAct xml:id="da22" target="#fs6"
  sender="#p2" addressee="#p1"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da23" fbSegment="#fs5.4"/>
<dialogueAct xml:id="da22" target="#fs6"
  sender="#p2" addressee="#p1"
  communicativeFunction="turnTake" dimension="turnManagement"/>


---


<dialogueAct xml:id="da24" target="#fs7.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da24" fbSegment="#fs6"/>
<dialogueAct xml:id="da25" target="#fs7.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnGrab" dimension="turnManagement"/>

```

```

<dialogueAct xml:id="da26" target="#fs7.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da27" target="#fs7.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da28" target="#fs7.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>


---


<dialogueAct xml:id="da29" target="#fs8"
  sender="#p2" addressee="#p1"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da29" fbSegment="#fs7.1"/>


---


<dialogueAct xml:id="da30" target="#fs9.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da31" target="#fs9.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da32" target="#fs9.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>


---


<dialogueAct xml:id="da33" target="#fs10.1"
  sender="#p2" addressee="#p1"
  communicativeFunction="turnGrab" dimension="turnManagement"/>
<dialogueAct xml:id="da34" target="#fs10.2"
  sender="#p2" addressee="#p1"
  communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<feedbackDependence dact34= "#da" fbSegment="#fs9.1"/>


---


<dialogueAct xml:id="da35" target="#fs11.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="confirm" dimension="alloFeedback"/>
<functionalDependence dact= "#da35" functAntecedent="#da34"/>
<dialogueAct xml:id="da36" target="#fs11.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da37" target="#fs11.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da38" target="#fs11.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da39" target="#fs11.4"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da40" target="#fs11.5"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>

```

```

<dialogueAct xml:id="da41" target="#fs11.6"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da42" target="#fs11.6"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>


---


<dialogueAct xml:id="da43" target="#fs12.1"
  sender="#p2" addressee="#p1"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da43" fbSegment="#fs11.5"/>
<dialogueAct xml:id="da44" target="#fs12.2"
  sender="#p2" addressee="#p1"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da44" fbSegment="#fs11.5"/>


---


<dialogueAct xml:id="da45" target="#fs13.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnTake" dimension="turnManagement"/>
<dialogueAct xml:id="da46" target="#fs13.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da47" target="#fs13.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="retraction"
  dimension="ownCommunicationManagement"/>
<dialogueAct xml:id="da48" target="#fs13.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da49" target="#fs13.4"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da50" target="#fs13.5"
  sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da51" target="#fs13.5"
  sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da52" target="#fs13.6"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>


---


<dialogueAct xml:id="da53" target="#fs14"
  sender="#p2" addressee="#p1"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da53" fbSegment="#fs13.6"/>


---


<dialogueAct xml:id="da55" target="#fs15.1"
  sender="#p1" addressee="#p2"
  communicativeFunction="autoPositive" dimension="autoFeedback"/>
<feedbackDependence dact= "#da54" fbSegment="#fs14"/>
<dialogueAct xml:id="da55" target="#fs15.2"
  sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da56" target="#fs15.3"
  sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>
</diaml>

```

Annex C (normative) DiAML schema

This annex introduces the technical scheme for the Dialogue Act markup Language DiAML associated with this standard for the concrete representation of annotations of dialogue data with dialogue act information.

C.1 Overview

This representation relies on a three-level architecture:

- a) a primary source, which may correspond to a speech recording, textual transcription or any further low-level annotation thereof (e.g. tokenization or morphosyntactic annotation according to ISO 24611 - MAF);
- b) the marking of functional segments from the primary source;
- c) the actual dialogue act annotation associated with a functional segment.

This annex provides a specification for this third level (the dialogue act annotation) as well as implementation guidelines for the two others.

The representation of a dialogue act annotated for a functional segment is done by means of the `dialogueAct` element. This element has the following attributes:

- `sender`, `addressee`, `otherParticipant` – see Clause 5;
- `communicativeFunction`, `dimension` - see Clause 11.2 for the possible values of these attributes;
- `certainty`, `partiality`, `conditionality`, `sentiment` – see Clause 11.2 for the possible values of these attributes

Functional relations between dialogue acts, like the one between a question and an answer, are represented by means of the `functionalDependence` element, which has the attributes `dact` and `functionalAntecedent`, pointing to the two related dialogue acts.

The relation between a feedback act and the functional segment(s) that the act provides or elicits feedback about, is likewise represented by means of the `feedbackDependence` element, which has the attributes `dact` and `fbSegment`, pointing to the act and the segment(s), respectively.

Rhetorical relations among dialogue acts are represented by means of the `rhetoLink` attributes, whose values are not fixed by this standard, but would for example include such values as `elaborate`, `justify`, `exemplify`, `clarify`.

Functional segments can be identified by means of the `functionalSegment` element, which groups together the components of multimodal communicative behaviour that may constitute a multimodal functional segment. There may be a verbal component, which in this example is identified in terms of the words in a transcription of the sender's spoken contribution, following joint TEI-ISO standard 24610-1 for referring to the corresponding span in a source document, using the `span` attribute. The `spanGrp` element is available for grouping more than one contiguous span in order to construct a representation of a discontinuous stretch of speech. The `target` attribute, which can denote any TEI pointer reference, is used to point to the (possibly discontinuous) verbal segment.

C.2 Example

The following excerpt exemplifies how the three above mentioned levels may be instantiated in the specific case of a tokenized primary source, encoded in accordance with the TEI guidelines. The source contains two utterances forming a small dialogue fragment, where the second utterance consists of a sentence interrupted by a filled pause (... *erm...*), which is accompanied by a frowning expression and a head gesture, and followed by lip smacking and a sigh, before the verbal contribution continues.¹⁰⁾ Two alternative XML representations are shown of the dialogue act information associated with the primary data, one using the XML encoding of feature structures according to joint TEI-ISO standard ISO 24610-1 and TEI P5, and compliant with W3C XML Schema in general; the other using a direct XML encoding of the DiAML concrete syntax introduced in Clause 11 of this standard.

This example also illustrates the attachment of dialogue act annotations to multimodal data. The transcription of multimodal dialogue is not part of this standard, but the example shows how dialogue act annotations can be related to XML representations of multimodal functional segments. In this example we show, for the sake of illustrating the possibilities, the XML representation of a multimodal segment which consists of a discontinuous verbal segment, a vocal component (heavily breathing out), a head movement (a 'waggle', i.e. left-right motion), a lip gesture (smacking), and an eyebrow gesture (frowning). Other components, like gaze direction or hand gesture, could be added in similar ways.

The `TEI` header contains metadata, including the identity of the dialogue participants.

```
<TEI xmlns:="http://www.tei-c.org/ns/1.0"
      xmlns:xlink="http://www.w3.org/1999/xlink">
  <teiHeader>
    <fileDescr>
      <titleStmt>
        <title>DiAML annotation example </title>
      </titleStmt>
      <publicationStmt>
        <p>...</p>
      </publicationStmt>
      <sourceDesc>
        <p>...</p>
      </sourceDesc>
    </fileDescr>
    <profileDesc>
      <particDescr>
        <person xml:id="p1">
          <p>the first participant</p>
        </person>
        <person xml:id="p2">
          <p>the second participant</p>
        </person>
      </particDescr>
    </profileDesc>
  </teiHeader>
  <text>
    <body>
      <div>
        <head>Simple dialogue fragment</head>
        <u>Do you know where I should check in for Munich</u>
        <u>For Munich go to counters 31 to 40</u>
      </div>
    </body>
  </text>
</TEI>
```

10) The XML-representation of the nonverbal behaviour is not considered here, nor is the time stamping of the verbal components. The representation of this sort of information is typically dependent on the transcription and annotation tools that are used. For example, the ANVIL tool (Kipp, 2001) supports the use of a separate track for each modality for each participant, and in each track an annotator can characterize the observed behaviour using his own imported set of features, and indicate start and ending. This information is represented by the tool in XML, and can be pointed to from DiAML annotations using the `target` attribute.


```

</div>
<div>
  <head>The dialogue turns, segmented into words (TEI-compliant)</head>
  <u>
    <w xml:id="w1">Do</w>
    <w xml:id="w2">you</w>
    <w xml:id="w3">know</w>
    <w xml:id="w4">where</w>
    <w xml:id="w5">I</w>
    <w xml:id="w6">should</w>
    <w xml:id="w7">check</w>
    <w xml:id="w8">in</w>
    <w xml:id="w9">for</w>
    <w xml:id="w10">Munich</w>
  </u>
  <u>
    <w xml:id="w11">Forn</w>
    <w xml:id="w12">Munich</w>
    <w xml:id="w13">go</w>
    <w xml:id="w14">to</w>
    <w xml:id="w15">erm</w>
    <w xml:id="w16">counters</w>
    <w xml:id="w17">32</w>
    <w xml:id="w18">to</w>
    <w xml:id="w19">40</w>
  </u>
</div>
<div>
  <head>Identification of functional segments</head>
  <spanGrp type="functionalVerbalSegment" xml:id="ves1">
    <span type="textStretch" xml:id="ts1" from="w1" to="w10" />
  </spanGrp>
  <spanGrp type="functionalVerbalSegment" xml:id="ves2">
    <span type="textStretch" xml:id="ts2.1" from="w11" to="w14" />
    <span type="textStretch" xml:id="ts2.2" from="w15" to="w19" />
  </spanGrp>
  <fs type="functionalSegment" xml:id="fs1">
    <f name="verbalComponent" fVal="#ves1"/> </f>
  </fs>
  <fs type="functionalSegment" xml:id="fs2">
    <f name="verbalComponent" fVal="#ves2"/>
    <f name="vocalComponent" <symbol value="outbreath"/> </f>
    <f name="headComponent" fVal="#heg1"/>
    <f name="lipComponent" <symbol value="lipsmack"/></f>
    <f name="browComponent" <symbol value="frown"/></f>
  </fs>
  <fs type="headGesture" xml:id="heg1">
    <f name="direction" <symbol value="leftright"/> </f>
    <f name="velocity" <symbol value="slow"/></f>
  </fs>
</div>
<div>
  <head>Representation by means of feature structures
  in TEI/ISO-compliant format</head>
  <fs type="dialogueAct" xml:id="da1" target="#fs1">
    <f name="sender" fVal="#p1"/>

```

```

    <f name="addressee" fVal="#p2"/>
    <f name="communicativeFunction" <symbol value="setQuestion"/></f>
    <f name="dimension" <symbol value="task"/></f>
    <f name="conditionality" <symbol value="conditional"/>
  </fs>
  <fs type="dialogueAct" xml:id="da2" target="#fs2">
    <f name="sender" fVal="#p2"/>
    <f name="addressee" fVal="#p1"/>
    <f name="communicativeFunction" <symbol value="answer"/></f>
    <f name="dimension" <symbol value="task"/></f>
  </fs>
  <fs type="functionalDependence" xml:id="fud1">
    <f name="dact" fVal="#da2"/>
    <f name="functionalAntecedent"="#da1"/></f>
  </fs>
</div>
</body>
</text>
</TEI>

```

An alternative, direct XML encoding of DiAML would look as follows, assuming the same representation of meta-data and functional segments, but replacing the part from <head>Representation by means of feature structures in TEI/ISO-compliant format</head> until </body> by the XML lines below, enclosed within <diaml ...>, </diaml> brackets:

```

<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="answer"
    dimension="task"/>
  <functionalDependence xml:id="fud1">
    dact="#da2"
    functionalAntecedent="#da1"/>
</diaml>

```

Annex D (normative)

Data categories for core concepts

D.1 Overview

The specification of a data category, as defined by ISO standard 12620, has the definition of a concept as its most important part. A definition has a *Source* attribute, which indicates the origin of the definition, and a *Note* attribute that may be used for mentioning alternative and related terms and concepts.

Two optional components of a data category specification are a *Conceptual domain* which lists the special cases of the defined concept, and *Broader concept*, which can be used to indicate that a concept is a special case of a more general concept. For example, the */answer/* data category has the conceptual domain */confirm/*, */disconfirm/*, and the broader concept */inform/*. Together, the values of these two components can be used to define a hierarchical structure in a set of concepts, such as the hierarchy of general-purpose communicative functions shown in Figure 2. Other optional components are an *Explanation*, which may provide useful information that does not strictly belong to the definition of the concept, but helps to understand it and place it in perspective; and *Example*, where the latter has a *Source* attribute for indicating the origin of an example.

D.2 Dialogue participants

Definition	/sender/ Dialogue participant who produces a dialogue act.
– Source	Commonplace
– Note	For a dialogue act in spoken form, possibly in combination with nonverbal communicative behaviour, then the sender is also called ‘speaker’.
– Explanation	The speaker role in spoken dialogue has been defined as that of a <i>participant who “has temporary control of the dialogue and speaks for some time</i> (DAMSL Revised Manual).

Definition	/addressee/ Dialogue participant at whom the sender of a dialogue act is primarily aiming his contribution, intending this participant to respond more than any other participant.
– Source	Goffman (1981)
– Note	Alternative terms: Hearer, Listener, Recipient.

D.3 Functional segments

Definition	/functionalSegment/ Minimal stretch of communicative behaviour that has a communicative function (and possibly more than one).
– Source	Geertzen et al., 2007
– Explanation	A functional segment is <i>minimal</i> in the sense of not being extended in ways that are irrelevant for the segment to have a certain communicative function. This requirement is motivated by the consideration that, whenever a certain segment s_1 of communicative behaviour has a communicative function $F - 1$, a larger segment s_2 which includes s_1 may also be said to have that function. It would be pointless to treat all supersegments of a functional segment as functional segments having the same communicative function.

D.4 Dimensions

Definition	/task/ Category of dialogue acts whose performance contributes to pursuing the task or activity that motivates the dialogue.
– Source	Commonplace
– Note	Related terminology in other schemes: Task (DAMSL) and Task Management (DAMSL); Activity (GBG-IM); Task/Activity (DIT).
– Explanation	The notion of a 'task' is to be taken in a very broad sense here, including any activity which can be said to be aimed at achieving a goal. Such a goal may be very specific, such as knowing the departure time of a particular train, or quite general, such as creating a pleasant atmosphere. Instead of 'task', the term 'activity' is also used.

Definition	/autoFeedback/ Category of dialogue acts where the sender discusses or reports on his processing of previous dialogue contributions.
– Source	DIT
– Note	Related terminology in other schemes: Feedback (e.g. Allwood et al., 1993); Signal Understanding (DAMSL).

Definition	/alloFeedback/ Category of dialogue acts where the sender discusses the addressee's processing of previous dialogue contributions.
– Source	DIT

Definition	/turnManagement/ Category of dialogue acts whose performance is meant to regulate the allocation of the speaker role.
– Source	Allwood et al., 1993
– Note	In the linguistic literature known as the turn-taking system.

Definition	/timeManagement/ Category of dialogue acts which concern the allocation of time to the participant occupying the speaker role.
– Source	DIT

Definition	/discourseStructuring/ Category of dialogue acts which explicitly structure the interaction.
– Source	DIT

Definition	/ownCommunicationManagement/ Category of dialogue acts where the speaker edits his own speech within the current turn.
– Source	Allwood et al., 1993

Definition	/partnerCommunicationManagement/ Category of dialogue acts which are performed by a dialogue participant who does not have the speaker role, and edits the speech of the participant who does occupy that role.
– Source	DIT

Definition	/socialObligationsManagement/ Category of dialogue acts performed for taking care of social obligations such as greeting, thanking, and apologizing.
– Source	DIT

D.5 Communicative functions

D.5.1 General-purpose functions

D4.1.1 Information-seeking functions

Conceptual domain	/question/
Definition	<i>/propositionalQuestion/ /setQuestion/ /choiceQuestion/</i> Communicative function of a dialogue act performed by the sender, S, in order to obtain the information, described by the semantic content, which S assumes that the addressee, A, possesses; S puts pressure on A to provide this information.
– Source	Commonplace
– Note	The notion of 'question' defined here only covers those cases where the sender genuinely wants to obtain the information that he is asking about. It does not include for instance 'exam questions', where the speaker does know the answer to his question but wants to know whether the examinee also knows, nor does it include rhetorical questions, which from a semantic point of view are not questions at all.
Example	"How about you?"
–Source	HCRC MapTask corpus

Broader concept	/propositionalQuestion/
Definition	<i>/question/</i> Communicative function of a dialogue act performed by the sender, S, in order to know whether the proposition, described by the semantic content, is true. S assumes that A knows whether the proposition is true or not, and puts pressure on A to provide this information
– Source	LIRICS
– Note	Related terminology in other schemes: YN-Question (TRAINS), Query-yn (HCRC MapTask)
Explanation	A propositional question corresponds to what is commonly termed a YN-question in the linguistic literature. This standard prefers the term 'propositional question' because the term 'YN-Question' carries the suggestion that this kind of question can only be answered by 'yes' or 'no', which is not the case.
Example	"Does the meeting start at ten?"

Broader concept	/setQuestion/
Definition	/question/ Communicative function of a dialogue act performed by the sender, S, in order to to know which elements of a certain set, described by the semantic content, have a certain property. also described by the semantic content; S puts pressure on the addressee, A, to provide this information. S believes that at least one element of the set has the named property, and S assumes that A knows which are the elements of the set that have the property.
– Source	LIRICS
– Note	Related terminology in other schemes: WH-Question (SWBD-DAMSL, MRDA), Query-w (HCRC MapTask), and WHQ (TRAINS).
Explanation	A set question corresponds to what is commonly termed a WH-question in the linguistic literature. The term 'set question' is preferred because: (a) it clearly separates form from function by removing any oblique reference to syntactic criteria for the identification of such acts; and (b) it is not a language specific term (it may be further noted that even in English, not all questioning words begin with 'wh', e.g. "How?").
Example	"What time does the meeting start?"; How far is it to the station?
	.

Broader concept	/checkQuestion/
Definition	/propositionalQuestion/ Communicative function of a dialogue act performed by the sender, S, in order to know whether a proposition, described by the semantic content, is true, about which S holds the uncertain belief that it is true S. S assumes that A knows whether the proposition is true or not, and puts pressure on A to provide this information
– Source	LIRICS
– Note	Related terminology in other schemes: Check (DIT, HCRC MapTask, TRAINS), Tag Question (SWBD-DAMSL), Request_Comment (Verbmobil)
Example	"The meeting starts at ten, right?"

Broader concept	/choiceQuestion/
Definition	/question/ Communicative function of a dialogue act performed by the sender, S, in order to know which one from a list of alternative propositions, provided by the semantic content, is true; S believes that exactly one element of that list is true; S assumes that the addressee, A, knows which of the alternative propositions is true, and S puts pressure on A to provide this information.
– Source	DAMSL; DIT
– Note	Related terminology in other schemes: Alternatives Question (DIT, LIRICS), QUERY-W (HCRC MapTask), Or-Question/Or-Clause (SWBD-DAMSL, MRDA). Also commonly known as 'menu question' or 'multiple-choice question'.
Explanation	It is not very common in annotation schemes to specifically distinguish the concept of choice questions from that of set questions (although it is common in the literature on interrogatives, see for instance: Tsui 1994). However, whereas it is common for the concept set question to carry the expectation that all members of the set with a given property should be returned by the addressee, for a choice-question the expectation is that there will be exactly one. The different preconditions and effects indicate that these are semantically different concepts, and they have been treated here as such.
Example	"Should the telephone cable go in telephone line or in external line?"
– Source	DIAMOND corpus

D4.1.2 Information-providing functions

Conceptual domain	/inform/
Definition	/agreement/ /disagreement/ /answer/ Communicative function of a dialogue act performed by the sender, S, in order to make the information contained in the semantic content known to the addressee, A; S assumes that the information is correct.
– Source	DIT
– Note	Related terminology in other schemes: Assert (DAMSL, COCONUT), Statement (SWBD-DAMSL, MRDA, Mal-tus).
Explanation	The inform function may also have more specific rhetorical functions such as: explain, elaborate, exemplify and justify; this is treated in this standard by means of rhetorical relations.
Example	"The 6.34 to Breda leaves from platform 2."

Broader concept	/agreement/
Definition	/inform/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S assumes a given proposition to be true, which S believes that A also assumes to be true.
– Source	DIT
– Note	Related terminology in other schemes: Accept (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil, Maltus, SPAAC).
Explanation	DAMSL and SWBD-DAMSL use "Agreement" to refer to various degrees in which some previous proposal, plan, opinion or statement is accepted; "accept" is one of these degrees; "reject" is another.
Example	English: "Exactly"; Dutch "Precies!"; Danish: "Netop!"

Broader concept	/disagreement/
Conceptual domain	/inform/
Definition	/correction/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S assumes a given proposition to be false, which S believes that A believes to be true.
– Source	DIT
– Note	Related terminology in other schemes: Reject (DAMSL, COCONUT, MRDA, Verbmobil) and Denial (TRAINS). DAMSL and SWBD-DAMSL use "Agreement" to refer to various degrees in which a speaker accepts some previous proposal, plan, opinion or statement; "accept" is one of these degrees; "reject" is another.
Example	J: "do you know where to find ink saving?" S: "ehm.. oh I think to the left of the ink cartridge" J: "ehm... no"
– Source	DIAMOND corpus

Broader concept	/correction/
Definition	/disagreement/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that certain information which S has reason to believe that A assumes to be correct, is in fact incorrect and should be replaced by the information that S provides.
– Source	Commonplace
– Note	In this definition /correction/ inherits the elements in the definition of /disagreement/.
Example	"To Montreal, not to Ottawa."

Broader concept	/answer /
Conceptual domain	/inform/ /confirm/ /disconfirm/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make certain information available to the addressee, A, which S believes A wants to know; S assumes that this information is correct.
– Source	Commonplace
Example	S: "what does the display say?" H: "send error document ready"
– Source	DIAMOND corpus

Broader concept	/confirm/
Definition	/answer/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that certain information that A wants to know, and concerning which A holds an uncertain belief, is indeed correct.
– Source	DIT; Verbmobil
– Note	Related terminology in other schemes: Reply-Y (HCRC MapTask); Yes-Answer (SWBD-DAMSL); Affirmative answer (MRDA).
Example	"Indeed"

Broader concept	/disconfirm/
Definition	/answer/ Communicative function of a dialogue act performed by the sender, S, in order to let the addressee, A, know that certain information that A wants to know, and concerning which A holds an uncertain belief, is incorrect.
– Source	DIT
– Note	Related terminology in other schemes: Reply-N (HCRC MapTask); No-Answer (SWBD-DAMSL); Dispreferred answer (MRDA).
Example	French "Si"; Danish "Jo"; Dutch: "Toch niet" and "Toch wel" ; German: "Doch"

D4.1.3 Commissive functions

Broader concept	/promise/
Conceptual domain	/offer/
Definition	/addressRequest/ Communicative function of a dialogue act by which the sender, S, commits himself unconditionally to perform a certain action in the manner or with the frequency described by the semantic content. S believes that the addressee, A, prefers the action to be performed rather than not be performed.
– Source	Searle (1969)
– Note	Related terminology in other schemes: Commit (DAMSL, COCONUT, Verbmobil, Maltus); Commitment (MRDA); Inform Intent (SPAAC).
Example	"I will look that up for you"

Conceptual domain	/offer/
Definition	/promise/ Communicative function of a dialogue act by which the sender, S, commits himself to perform a certain action, described by the semantic content, conditional on the consent of addressee A that S do so. S assumes that the addressee, A, prefers the action to be performed rather than not be performed.
– Source	Commonplace
Example	"Shall I start?"; "Would you like to have some coffee?"

Conceptual domain	/addressRequest/
Definition	/acceptRequest/ /declineRequest/ Communicative function of a dialogue act by which the sender, S, commits himself to perform an action that he was requested to perform, possibly with certain restrictions or conditions concerning manner or frequency of performance.
– Source	DIT
Explanation	The addressRequest function covers a range of possible responses to a request. If the response does not express a condition, then the sender commits himself unconditionally to perform the requested action; this is the special case of /acceptRequest/. If the condition is specified that the action be performed zero times, then the sender in fact declines to perform the requested action (as he commits himself to not perform the action). See also the data categories for the qualifiers /conditional/ and /partial/.
– Note	Related terminology in other schemes: Assess (AMI).
Example	A: "Please give me the gun." S: "If you push the bag to me."

Definition	/acceptRequest/ Communicative function of a dialogue act by which the sender, S, commits himself to perform an action that he has been requested to perform.
– Source	LIRICS
– Note	Related terminology in other schemes: Accept (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	"Sure"

Definition	/declineRequest/ Communicative function of a dialogue act by which the sender, S, indicates unwillingness to perform an action that he has been requested to perform.
– Source	LIRICS
– Note	Related terminology in other schemes: Reject (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	"Not now"

Definition	/addressSuggest/ Communicative function of a dialogue act by which the sender, S, commits himself to perform an action that was suggested to him, possibly with certain restrictions or conditions concerning manner or frequency of performance.
– Source	DIT
– Note	Related terminology in other schemes: Assess (AMI).
Example	A: "Let's go there together." S: "Only if we're in full agreement about the way to proceed when we get there."

Definition	/acceptSuggest/ Communicative function of a dialogue act by which the sender, S, commits himself to perform an action that was suggested to him.
– Source	LIRICS
– Note	Related terminology in other schemes: Accept (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	"Let's do that"

Definition	/declinSuggest/ Communicative function of a dialogue act by which the sender, S, indicates unwillingness to perform an action that was suggested to him.
– Source	LIRICS
– Note	Related terminology in other schemes: Reject (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	"I don't think so"

D4.1.4 Directive functions

Conceptual domain	/request/ /instruct/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to create a commitment for the addressee, A, to perform a certain action in the manner or with the frequency described by the semantic content, which S assumes A is able to perform, conditional on A's consent.
– Source	DIT
Example	"Please turn to page five"; "Please don't do this ever again"; "Please drive very carefully".

Broader concept	/instruct/
Conceptual domain	/request/ /addressOffer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to create a commitment for the addressee, A, to carry out a named action in the manner or with the frequency described; S assumes that A is able and willing to carry out the action.
– Source	DIT; HCRC Map Task
– Note	Related terminology in other schemes: Action-directive (DAMSL, SWBD-DAMSL, COCONUT); Command (MRDA).
Example	"Go right round until you get to just above that."; "Take three of these pills a day, for the next two weeks"; "Do not enter!"
–Source	HCRC MapTask corpus

Broader concept	/addressOffer/
Conceptual domain	/instruct/ /acceptOffer/ /declinOffer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S agrees to A performing the action that A has offered to perform.
– Source	DIT
– Note	Related terminology in other schemes: Assess (AMI).
Example	"Yes please"; French: "Je vous en prie"

Definition	/suggest/ Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, aware that a certain action, described by the semantic content, is potentially promising for achieving a certain goal, which is either named explicitly or contextually salient.
– Source	DIT
– Note	Related terminology in other schemes: Open-option (DAMSL, SWBD-DAMSL, COCONUT).
Example	"Let's wait for the speaker to finish."

Broader concept	/acceptOffer/ /addressOffer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S agrees to A performing the action that A has offered to perform.
– Source	LIRICS
– Note	Related terminology in other schemes: Accept (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	"Yes please"; French: "Je vous en prie"; Dutch: "Graag"; German: "Bitte"

Broader concept	/declineOffer/ /addressOffer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S does not agree to A performing the action that A has offered to perform.
– Source	LIRICS
– Note	Related terminology in other schemes: Reject (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	English: U: "No thank you"; Danish: "Nej tak"; French: "Non merci".

D.5.2 Feedback functions

Definition	/autoPositive/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S believes that S's processing of the previous utterance(s) was successful.
– Source	LIRICS
– Note	Related terminology in other schemes: Signal-Understanding (DAMSL), Acknowledge (HCRC MapTask, SWBD-DAMSL) Ack (TRAINS) and Feedback_Positive (Verbmobil). This type of feedback may be further broken down into specific levels of processing (dealing with the sender's attention, perception, interpretation, evaluation and execution), as exemplified in the DIT and SLSA schemes.
Explanation	Feedback mostly concerns the processing of the last utterance from the addressee, but sometimes, especially in the case of <i>positive</i> feedback, it concerns a longer stretch of dialogue.
Example	"Uh-huh"; "Okay"; Nonverbally: nodding; "Yes"

Definition	/alloPositive/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S believes that A's processing of the previous utterance(s) was successful.
– Source	LIRICS
– Note	This type of feedback may be further broken down into more specific levels of processing, as exemplified in the DIT and SLSA schemes).
Example	"Correct!"

Definition	/autoNegative/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S believes that S's processing of the previous utterance encountered a problem.
– Source	LIRICS
– Note	Related terminology in other schemes: Signal-Non-Understanding (DAMSL) and Feedback_Negative (Verbmobil). This type of feedback may be further broken down into more specific levels of processing, as is exemplified in the DIT schema.
Example	English: "Sorry?"; "What?"; Spanish: "Que?"; Italian, Portuguese: "Como?"

Definition	/alloNegative/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S believes that A's processing of the previous utterance encountered a problem.
– Source	LIRICS
– Note	This type of feedback may be broken down into more specific levels of processing.
Example	"No no no no no"

Definition	/feedbackElicitation/ Communicative function of a dialogue act performed by the sender, S, in order to know whether A's processing of the previous utterance was successful.
– Source	LIRICS
– Note	Feedback elicitation could be further broken down into specific levels of processing.
Example	English: "Okay?"; Italian: "Capisce?"; Dutch: "Ja?"

D.5.3 Turn management functions

Definition	/turnAccept/ Communicative function of a dialogue act performed by the sender, S, in order to signal his willingness to take the speaker role, as requested by the previous speaker.
– Source	Common in literature on turn taking in conversation.
– Note	Related terminology in other schemes: Take-Turn (TRAINS).
Example	Nonverbally: nodding; A: "Would you like to say something at this point?" S: "Certainly."

Definition	/turnAssign/ Communicative function of a dialogue act performed by the sender, S, in order to signal that he wants the addressee, A, to take the turn.
– Source	Common in literature on turn taking in conversation.
– Note	Occurs especially in multiparty dialogue. Related terminology in other schemes: Turn Give DIT), Assign-Turn (TRAINS).
Example	"Adam?", characteristically accompanied by the speaker directing his gaze to Adam, possibly also nodding or pointing in his direction and raising the eyebrows.

Definition	/turnGrab/ Communicative function of a dialogue act performed by the sender, S, in order to take the speaker role away from the participant who currently occupies it.
– Source	Common in literature on turn taking in conversation.
– Note	Related terminology in other schemes: Grabber (MRDA); Turn Grabber (Maltus, Primula); Interruption (SLSA).
Example	"Hold on"; nonverbally: sticking up a hand as a stop signal

Definition	/turnKeep/ Communicative function of a dialogue act performed by the sender, S, in order to keep the speaker role.
– Source	Common in literature on turn taking in conversation.
– Note	Related terminology in other schemes: Turn maintain (DAMSL, SWBD-DMSL); Holder (MRDA); Hold (SPAAC, Chiba); Turn holder (Maltus, Primula); Turn holding (SLSA). Note: utterances used for turn keeping often also have a stalling function.
Example	"Ehm" not in turn-initial position

Definition	/turnRelease/ Communicative function of a dialogue act performed by the sender, S, in order to give other dialogue participants the opportunity to occupy the speaker role.
– Source	Common in literature on turn taking in conversation
– Note	Related terminology in other schemes: Turn closing (SLSA).
Example	Sender uses declining intonation towards the end of a contribution and subsequently pauses.

Definition	/turnTake/ Communicative function of a dialogue act performed by the sender, S, in order to have the speaker role, which is available at that moment.
– Source	:Common in literature on turn taking in conversation.
– Note	Related terminology in other schemes: Take-Turn (TRAINS)
Example	"Ehm..." as a turn-initial segment

D.5.4 Time management functions

Definition	/stalling/ Communicative function of a dialogue act performed by the sender, S, in order to have a little extra time to construct his contribution.
– Source	DIT
– Note	Related terminology in other schemes: Hold (SPAAC, MRDA); Stall (AMI); Delay (DAMSL, SWBD-DAMSL, COCONUT).
Example	"Let me see...", "Ehm..."; Nonverbally: slowing down
–Note	Turn-initial segments with a Stalling function often also have a Turn Take or Turn Accept function; segments inside a turn which have a Stalling function often also have a Turn Keep function.

Definition	/pausing/ Communicative function of a dialogue act performed by the sender, S, in order to suspend the dialogue for a short while.
– Source	DIT
– Note	Related terminology in other schemes: Pause (Alparon); Please wait (C-Star); Hold before answers (MRDA).
Explanation	Pausing occurs either in preparation of continuing the dialogue, or because something else came up which is more urgent for the sender to attend to.
Example	English: "Just a moment"; Danish: "Lige et øjeblik"; Dutch: "een ogenblikje"; French: "Veuillez patienter".

D.5.5 Own and partner communication management functions

Definition	/completion/ Communicative function of a dialogue act performed by the sender, S, in order to assist the addressee, A, by completing or adding to the expression that A is constructing.
– Source	Commonplace
– Note	Related terminology in other schemes: Complete (SPAAC); Collaborative completion (MRDA).
Example	A: "which should leave us plenty of time to uhhh... uhhh" S: "get to Corning"
– Source	TRAINS corpus

	/correctMisspeaking /
Definition	Communicative function of a dialogue act performed by the sender, S, in order to correct (part of) an utterance by the addressee, A, assuming that A made a speaking error.
– Source	DAMSL; DIT
– Note	Related terminology in other schemes: Correction suggestion (TRAINS).
Example	A: "second engine E3 is going to uhh Corning to pick up the bananas, back to Avon, drop..." S: "to pick up the oranges" A: "sorry, pick up the oranges"
–Source	TRAINS corpus

	/signalSpeakingError/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to signal to the addressee, A, that he (S) has made a mistake in speaking.
– Source	DIT
Example	S: "so you want to leave at eight o'clock in the morning?" U: "yes oh sorry no..."

	/retraction/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to withdraw something that he just said within the same turn.
Example	"then we're going to g– "
– Source	HCRC Map Task corpus

	/selfCorrection/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to correct an error that he just made, or to improve on an infelicitous formulation that he just used, within the same turn.
– Source	Levelt, 1983
– Note	Related terminology in other schemes: Speech repair (DAMSL, MRDA, TRAINS); Correct-self (SPAAC).
Example	"then we're going to g– ... turn straight back "
– Source	HCRC Map Task corpus

D.5.6 Discourse structuring functions

	/interactionStructuring/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to explicitly indicate to the addressee, A, the function or topic of his next contribution(s).
– Source	LIRICS
– Note	The function "Interaction structuring" covers a range of phenomena related to the structure of a dialogue, such as topic introduction, dialogue act announcement and topic closing.
Examples	English: "A question"; Dutch: "vraagje"

	/opening/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S is ready and willing to engage in a dialogue with A.
– Source	Commonplace
Example	"okay," (especially in multi-party dialogue).

D.5.7 Social obligations management functions

	/initialGreeting/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that S is present and aware of A's presence; S puts pressure on A to acknowledge this.
– Source	DIT
– Note	Related terminology in other schemes: Greet (Verbmobil).
Explanation	Greetings usually come in initiative-response pairs within a dialogue; this data category corresponds to the first element of such a pair.
Example	"Hello!"; "Good morning"

	/returnGreeting/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to acknowledge that S is aware of the presence of the addressee, A, and of A having signalled his presence to S.
– Source	DIT
– Note	Related terminology in other schemes: Greet (Verbmobil).
Explanation	Greetings usually come in initiative-response pairs within a dialogue; this data category corresponds to the second element of such a pair.
Example	"Hello!"; "Good morning"

	/initialSelfIntroduction/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make himself known to the addressee, A; S puts pressure on A to acknowledge this.
– Source	Commonplace
Explanation	Introductions usually come in initiative-response pairs within a dialogue; this data category corresponds to the first element of such a pair.
Example	"I'm Jack"

	/returnSelfIntroduction/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make himself known to the addressee, A in response to a self-introduction by A.
– Source	DIT
Explanation	Introductions usually come in initiative-response pairs within a dialogue; this data category corresponds to the second element of such a pair.
Example	"And I'm Jill"

	/apology/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to signal that he wants the addressee, A, to know that S regrets something; S puts pressure on A to acknowledge this.
– Source	Commonplace
– Note	Related terminology in other schemes: Apologize (C-Star); Polite (Verbmobil).
Example	"Sorry about that."

	/acceptApology/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to mitigate, the feelings of regret that the addressee, A, has expressed.
– Source	Commonplace
Example	"No problem."

Definition	/thanking/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S is grateful for some action performed by A; S puts pressure on A to acknowledge this.
– Source	Commonplace
– Note	Related terminology in other schemes: Thank (Verbmobil).
Explanation	Utterances used for thanking often also indicate that the sender wants to end the dialogue.
Example	English: "Thanks a lot."; Portuguese: "Muito obrigado"; Swedish: "Tack so mycket"; Greek: "Evcharisto"

Definition	/acceptThanking/ Communicative function of a dialogue act performed by the sender, S, in order to mitigate to the feelings of gratitude which the addressee, A', has expressed.
– Source	Commonplace
Example	English: "Don't mention it"; Spanish: "De nada"; Greek: "parakalo".

Definition	/initialGoodbye/ Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S intends the current utterance to be his final contribution to the dialogue; S puts pressure on A to acknowledge this.
– Source	DIT
– Note	Related terminology in other schemes: Bye (Verbmobil).
Explanation	Goodbyes usually come in initiative-response pairs within a dialogue; this data category corresponds to the first element of such a pair. Initial and return goodbyes are commonly used to close a dialogue.
Example	S: "Bye bye, see you later" A: "Bye bye, see you."

Definition	/returnGoodbye/ Communicative function of a dialogue act performed by the sender, S, in order to acknowledge his awareness that the addressee, A, has signalled his final contribution to the dialogue and S signals in return his agreement to end the dialogue; S has been pressured to respond to an initialGoodbye by A.
– Source	DIT
– Note	Related terminology in other schemes: Bye (Verbmobil).
Explanation	Goodbyes usually come in initiative-response pairs within a dialogue; this data category corresponds to the second element of such a pair. Initial and return goodbyes are commonly used to close a dialogue
Example	A: "Bye bye, see you later" S: "Bye bye, see you."

D.6 Qualifiers

D.6.1 Certainty

Definition	/certainty/ Class of predicates which can be associated with a communicative function to express whether the sender of a dialogue act with that function is certain or uncertain about the correctness of the information that he provides, or about the commitment that he takes on to perform a certain action .
-------------------	--

Definition	/uncertain/ Predicate which can be associated with a communicative function to express that the sender of a dialogue act with that function is uncertain about the correctness of the information that he provides, or about the commitment that he takes on to perform a particular action .
– Source	AMI (2005)
Example	"That might be a good idea."

Definition	/certain/ Predicate which can be associated with a communicative function to express that the sender of a dialogue act with that function is certain about the correctness of the information that he provides, or about the commitment that he takes on to perform a particular action .
Example	"I definitely don't support that."
– Source	DIT

D.6.2 Conditionality

Definition	/conditionality/ Class of predicates which can be associated with an action-discussion function to express whether the sender of a dialogue act with that function is considering the performance of the action under discussion subject to conditions or not.
-------------------	--

Definition	/conditional/ Predicate which can be associated with an action-discussion function, or with the communicative function of a dialogue act that responds to an action-discussion act, to express that the sender of a dialogue act with that function is considering the performance of the action under discussion subject to certain conditions.
Example	"If you're ready maybe you can start the presentation"
– Source	DIT
Example	A: "Can we just go over that again" B: "We have no time, unless you do it very quickly"
– Source	AMI corpus

Definition	/unconditional/ Predicate which can be associated with an action-discussion function, or with the communicative function of a dialogue act that responds to an action-discussion act, to express that the sender of a dialogue act with that function is considering the performance of the action under discussion without any conditions.
Example	A: "I'll come no matter what."

D.6.3 Partiality

Definition	<p>/partiality/</p> <p>Class of predicates which can be associated with a responsive general-purpose function or a feedback providing function to express whether the sender of a dialogue act with that function is responding to the entire semantic content of the act that is responded to or only to part of it (in the case of a responsive general-purpose function), or whether the sender provides feedback to the entire segment that the feedback refers to or only to part of it.</p>
-------------------	--

Definition	<p>/partial/</p> <p>Predicate which can be associated with a general-purpose function to express that the sender of a dialogue act with that function is responding to only part of the semantic content of another dialogue act.</p>
– Source	AMI (2005)
Example	<p>A: "I don't think that would be practical, and it's expensive too." B: "It's certainly expensive."</p>

Definition	<p>/full/</p> <p>Predicate which can be associated with a responsive general-purpose function or a feedback providing function to express that the sender of a dialogue act with that function is responding to the entire semantic content of the act that is responded to (in the case of a responsive general-purpose function), or that the sender provides feedback to the entire segment that the feedback refers to.</p>
-------------------	--

D.6.4 Sentiment: Emotion and Attitude

Definition	<p>/sentiment/</p> <p>Class of predicates which can be associated with a communicative function to express an emotional stance of the sender of a dialogue act with that function towards the semantic content of the dialogue act, or to express a mental attitude towards the addressee.</p>
-------------------	---

Annex E (informative)

Data categories for non-core communicative functions and dimensions

E.1 Non-core dimensions

Definition	/contactManagement/
– Source	Category of those dialogue acts which are performed by a dialogue participant for establishing or ensuring contact with other participants. DIT

E.2 Non-core communicative functions

Broader concept	/examQuestion/
Definition	/question/ Dialogue act where the sender, S, wants to know whether the addressee, A, possesses certain information which S possesses. S puts pressure on A to provide this information
– Source	Commonplace

Broader concept	/lie/
Definition	/inform/ Dialogue act where the sender, S, wants the addressee, A, to believe that a certain proposition is true which S believes to be false.
– Source	Commonplace

Definition	/contactIndication/
– Source	Dialogue act where the sender, S, wants to make it known to the addressee, A, that S is ready to send messages to, and receive messages from, A. DIT
Example	"Yes?"

Definition	/contactCheck/
– Source	Dialogue act where the sender, S, wants to establish whether the addressee, A, is ready to receive messages from, and send messages to, S. DIT
Example	"Hello?!"

Annex F (informative)

A survey and analysis of dimensions and communicative functions in existing annotation schemas

As part of the project to establish the present standard, a detailed study was conducted in order to provide theoretical and empirical arguments for identifying core dimensions and communicative functions (Petukhova & Bunt, 2009a). The study included a survey of the literature on dialogue analysis and of the use of functions and dimensions in 18 existing annotation schemas. Moreover, a number of statistical and machine-learning tests were carried out in order to identify dependencies among potential dimensions. The following criteria for identifying core dimensions were investigated:

(59) Each dimension should be:

- a) theoretically justified, in the sense of forming a well-established and well-studied aspect of communication;
- b) empirically observed in the functions of dialogue utterances;
- c) addressable independently of the other dimensions.
- d) recognizable with acceptable precision by human annotators and by automatic annotation systems;
- e) present in a significant number of existing dialogue act annotation schemes.

The independence of dimensions was investigated by calculating the co-occurrences of communicative functions across dimensions, by calculating the phi coefficient to measure semantic relatedness between dimensions, by determining for a range of candidate dimensions the frequencies of occurrence of functional segments addressing only those dimensions, and by checking the occurrences of dimension pairs in sequences of functional segments.

This study was published as Technical Report TR 2009-003 of the Centre for Creative Computing at Tilburg University (Petukhova & Bunt, 2009a).¹¹⁾ The tables 1 – 9 in this annex are copied from this publication, together with the conclusions. Table 1 shows the relative frequencies of functional segments in 10 dimensions (9 of which are core dimensions in the present standard) for three different dialogue corpora. The variation between the corpora is worth noting. Table 2 shows the relative frequencies of functional segments addressing only one dimension. From this table it may be concluded that the 10 dimensions considered in the table are all independently addressable. The tables 3-9 show the occurrence of dimension-specific communicative functions in various dimensions in 18 existing annotation schemas. The tables 11 – 15 show the occurrence of general-purpose functions in these annotation schemas.

The conclusions reached in this study are the following.

Eight dimensions, namely **Task**, **Feedback**, **Turn Management**, **Social Obligations Management**, **Own Communication Management**, **Discourse Structuring**, **Partner Communication Management** and **Time Management** fulfil all five criteria, and can be considered as 'core' aspects of dialogue communication. Our conclusion with respect to Feedback is moreover that a distinction should be made at least between **Feedback giving** and **Feedback eliciting** aspects, since dialogue participants not only report about successes and failures of their own processing of previous utterances, but also constantly evaluate the partner's cognitive state, message processing, and degree of involvement in the communication, and may elicit information about these aspects.

11) A highly condensed version was presented at the 2009 NAACL-HLT conference (Boulder, Colorado, May 2009); see Petukhova & Bunt (2009b).

Making only the distinction between feedback-giving and feedback-eliciting acts, however, does not do justice to the fact that feedback-giving acts can report not only on the speaker's own processing of previous dialogue but also on the speaker's beliefs about the addressee's processing - a distinction which is semantically important and which is captured by the distinction between **Auto-** and **Allo-Feedback**. Note also that the phi coefficient (-0.3) indicates that Auto- and Allo-Feedback are not very closely related. These arguments support the suggestion to distinguish the two as separate dimensions.

Time Management acts co-occur frequently with Turn Management acts, since speakers often need a bit of time to formulate their contribution when they take (or have and want to keep) the turn. This consideration applies only to *stallings* under certain context conditions, however; *pausing*, by contrast, does not imply that the speaker wants to keep the turn. It should be also noticed that *stallings* do not always imply that the speaker wants to keep the turn; extensive amounts of protraction accompanied by certain non-verbal behaviour may indicate that the speaker needs assistance. It was noticed by Butterworth (1980) that an excessive amount of gaze aversion may also lead a listener to infer that the speaker is having difficulty formulating a message. Moreover, as Clark (1996) shows, time delays are not always used for turn-keeping purposes, because even in monologues where speakers do not need to keep the turn, time delays are frequently used. Time and Turn Management are therefore better kept apart rather than considered as one dimension.

Another view on **Time Management** acts is that they are produced unintentionally, *stallings* in particular. They should therefore perhaps not be regarded as dialogue acts. An act that is not consciously intentional may still be relevant, however; for example, humans produce a lot of facial expressions unconsciously, but they display the emotional or cognitive state of the dialogue participant, which is obviously important for dialogue analysis. In other words, they affect the information states of dialogue participants if they have shared encoded meaning. Goffman (1963) points out that the receiver is always responsible for the interpretation of an act as intentional or not. Kendon (2004) also notices that whether an action is deemed to be intended or not is something that is dependent entirely upon how that action appears to others. So this does not provide a good argument against viewing Time Management as a dimension of dialogue communication.

Contact Management could be considered as an 'optional' dimension, since this aspect of communication is not reflected in most existing dialogue act annotation schemes (6 out of 18). It was noticed, however, that for some types of dialogues, e.g. phone conversations or tele-conferences (as in the OVIS corpus), this aspect may be important." The standard is open to the addition of Contact Management and other dimensions, provided that they meet the requirement of being addressable independently of the other dimensions.

Dimension <i>Corpus</i>	<i>AMI</i>	<i>DIAMOND</i>	<i>OVIS</i>
Task	33.0	47.7	48.8
Auto-Feedback	20.0	14.0	18.0
Allo-Feedback	0.7	3.8	39.0
Turn Management	15.0	14.0	1.0
Social Obligation Management	0.3	5.0	3.8
Discourse Structuring	2.2	2.3	2.4
Own Communication Management	8.7	0.7	0.3
Time Management	16.8	10.7	0.6
Partner Communication Management	0.3	0.3	0.1
Contact Management	0.1	1.3	12.3

Table 1 — Distribution of functional segments across dimensions for analysed dialogue corpora in (%).

Dimension <i>Corpus</i>	<i>AMI</i>	<i>OVIS</i>	<i>DIAMOND</i>
Task	28.8	37.9	29.9
Auto-Feedback	14.2	16.3	20.9
Allo-Feedback	0.7	4.1	6.8
Turn Management	7.4	0.9	8.5
Time Management	0.3	0.4	0.7
Contact Management	0.1	0.3	0.7
Discourse Structuring	1.9	1.8	2.7
Own Communication Management	0.5	0.8	2.7
Partner Communication Management	0.2	3.1	0.4
Social Obligation Management	0.3	6.4	0.7

Table 2 — Distribution of functional segments addressing a single dimension for three dialogue corpora in (%).

<i>Schema</i>	<i>Positive auto-feedback function</i>				
DIT	Positive attention	Positive perception	Positive interpretation	Positive evaluation	Positive execution
LIRICS	Positive Auto-Feedback				
DAMSL	Signal understanding		Acknowledgment		
SWBD-DAMSL	Signal understanding		Repeat-rephrase	Acknowledgment	Summarize-reformulate
MRDA	Signal understanding		Acknowledgment Appreciation	Assessment	
Coconut	Signal understanding		Acknowledgment Repeat-rephrase		
AMI	Comment-about-understanding POS			Assess	Inform POS
HCRC MapTask			Acknowledgment		
Verbmobil	Backchannel			Acknowledge	Positive feedback
SLSA	Pos.contact	Pos.perception	Pos.understanding	Pos. acceptance/attitude	
TRAINS	Acknowledgement			Pos.evaluation	
SPAAC	Echo		Acknowledge	Appreciate	
MALTUS	Pos. attention	Repeat-rephrase		Appreciation	
Chiba	Follow up: pos. understand			Pos. response	
Alparon			Acknowledgement		
C-Star			Acknowledgement		

Table 3 — Positive Auto-feedback functions in different dialogue act taxonomies.

<i>Schema</i>	<i>Negative auto-feedback function</i>				
DIT	Negative attention	Negative perception	Negative interpretation	Negative evaluation	Negative execution
LIRICS	Negative Auto-Feedback				
DAMSL	Signal-non-understanding				
SWBD-DAMSL	Signal-non-understanding				
MRDA	Signal-non-understanding		Understanding Check		
Coconut	Signal-non-understanding		Clarification Check		
AMI	Comment-about-understanding NEG				Inform NEG
HCRC MapTask	Check				
Verbmobil	Request clarify				Neg.feedback
SLSA	Neg. contact	Neg. perception	Neg. understanding	Neg. attitude	
TRAINS			Neg. evaluation		
SPAAC	Pardon				
MALTUS	Neg.attention				
Chiba	Follow up: understand			Neg. response	
Alparon					
C-Star					

Table 4 — Negative Auto-feedback functions in different dialogue act taxonomies.

<i>Schema</i>	<i>Turn Management function</i>					
DIT	Turn take	Turn grab	Turn accept	Turn keep	Turn assign	Turn release
LIRICS	Turn take	Turn grab	Turn accept	Turn keep	Turn assign	Turn release
DAMSL				Turn maintain		
SWBD-DAMSL			Hold before answers	Turn maintain	Turn exit	
MRDA	Regain turn	Grabber	Hold before answers	Holder		
Coconut				Turn maintain		
SLSA	Turn take	Interruption	Turn opening	Turn holding	Turn closing	
TRAINS	Turn take			Turn keep	Turn assign	Turn release
SPAAC				Hold		
MALTUS		Turn grabber		Turn holder		Back-channel
Primula		Turn grabber		Turn holder		Back-channel
Chiba				Hold		

Table 5 — Turn Management functions in different dialogue act taxonomies.

<i>Schema</i>	<i>Social Obligation Management function</i>				
DIT	Greeting/ return greeting	Self-introduction/ return self- introduction	Goodbye/ return goodbye	Apology/ accept apology	Thanking/ accept thanking
LIRICS	Greeting/ return greeting	Self-introduction/ return self- introduction	Goodbye/ return goodbye	Apology/ accept apology	Thanking/ accept thanking
DAMSL	Greeting		Goodbye		
SWBD- DAMSL	Greeting			Apology/ downplayer	Thanking/ downplayer
MRDA				Downplayer/ sympathy	Thanking
Coconut	Greeting		Goodbye		
AMI	Be-positive/be-negative				
Verbmobil	Greet	Introduce	Bye	Polite (apologies and compliments)	Thank
SLSA	Greet				
TRAINS	Greet				
MALTUS	Politeness				
Primula	Politeness; face-threatening/face-saving				
Alparon	Greet		Bye		
C-Star	Greeting	Self-introduction		Apologize	Thanking

Table 6 — Social Obligation Management functions in different dialogue act taxonomies.

<i>Schema</i>	<i>Discourse Structuring function</i>				
DIT	Opening	Pre-closing	Topic introduction	Topic shift	Topic shift announcement
LIRICS	Interaction structuring				
DAMSL	Opening	Closing			
SWBD-DAMSL	Opening	Closing			
MRDA	Topic change				
Coconut	Opening	Closing		Topic	
AMI	Argument structure and topic segmentation schemes				
HCRC MapTask	Ready (for topic shifts)				
Verbmobil		Task close	Task initiate	Digress	
LinLin	Opening	Closing	Topic layer		
SLSA	Opening	Closing	Opening	Continuation	
SPAAC				Initiate: release issue	Topic
MALTUS	Topic change				
Primula	Opening	Closing	Topic opening	Topic closing/change	
Chiba	Opening	Closing	Topic break		
C-Star		Closing	Introduce topic		

Table 7 — Discourse Structuring functions in different dialogue act taxonomies.

<i>Schema</i>	<i>Own Communication Management</i>			<i>Partner Communication Management</i>	
DIT	Error signalling	Retraction	Self-correction	Correct-misspeaking	Completion
LIRICS	Error signalling	Self-correction		Correct-misspeaking	Completion
DAMSL		Speech repair		Correct-misspeaking	Completion
SWBD-DAMSL		Speech repair		Correct-misspeaking	Completion
MRDA		Speech repair		Correct-misspeaking	Collaborative completion
Coconut		Correct assumption; Speech repair		Correct-misspeaking	Completion
SLSA	Change				
TRAINS		Repair			
SPAAC		Correct-self		Correct	Complete
MALTUS		Restated info with repetition/correction		Restated info with correction	

Table 8 — Own and Partner Communication Management functions in different dialogue act taxonomies.

<i>Schema</i>	<i>Time Management</i>		<i>Contact Management</i>	
DIT	Stalling	Pausing	Contact check	Contact indication
LIRICS	Stalling	Pausing	Contact check	Contact indication
DAMSL	Communication management: delay		Communication channel	
SWBD-DAMSL	Stalling; delay; Hold before answers		Communication channel	
MRDA	Hold before answers			
Coconut	delay		Communication channel	
AMI	Stall			
Verbmobil	Deliberate		Refer-to-settings	
SLSA	Choice			
TRAINS	Keep			
SPAAC	Hold			
Alparon		Pause		
C-Star		Please wait		

Table 9 — Time and Contact Management functions in different dialogue act taxonomies.

DIT	LIRICS	DAMSL	SWBD-DAMSL	MRDA	Coconut
Inform	Inform	(Re-)assert;	Statement-non-opinion;	Statement	(Re-)assert;
Uncertain Inform		Other Statement	Statement- opinion		Other Statement
Agreement	Agreement	Agreement:accept	Accept	Accept;Affirmative answer	Accept
-	-	Accept-part	Accept-part	Partial Accept	Accept-part
-	-	Maybe	Maybe	Maybe	Maybe
Disagreement	Disagreement	Reject	Reject; Dispreferred responses	Reject; Dispreferred responses	Reject
-	-	Reject-part	Reject-part	Partial Reject	Reject-part
-	-	hold	hold	-	hold
Correction	Correction	-	-	-	Correct Assumption
Inform Elaborate	-	-	-	Repeat; Summary; Defending/Explanation; Elaboration	ElaborateItem; informational relations: condition, consequence and other Info
Answer (Uncertain) Set answer	Answer Set answer		Answers-to-non-yes-no-questions	Answer	
Uncertain Prop. Answer	-		Maybe	Maybe	
Unc. confirm	-		Maybe	Maybe	
Unc. disconfirm	-		Maybe	Maybe	
Prop. answer	Prop. answer	Answer	Yes-answer; no-answer; affirmative non-yes answer; negative non-no answers; other answers	Affirmative answer; negative answer	Answer
Confirm	Confirm		Yes-answer; no-answer	Affirmative answer	
Disconfirm	Disconfirm		Yes-answer; no-answer; Dispreferred responses	Dispreferred answer	

Table 10 — Information providing functions in DIT, LIRICS, DAMSL, SWBD-DAMSL, MRDA and Coconut.

DIT	AMI	Verbmobil	TRAINS	HCRC	SPAAC	Maltus
Uncertain Inform	Inform Uncertain Inform	- Inform	- Inform	- Statement: Explain	Express possibility Inform; Express opinion; express wish	- Statement
Agreement	Inform Positive	Accept	Accept	Reply-y	Accept	Positive answer
Disagreement	Inform Negative	Reject	Reject	Reply-no	Negate	Negative answer
Correction	others		-	-	correctSelf	Restated info correct
Inform Elaborate	Inform	Clarify; Give_reason	Support Inform; Argumentation Acts: elaborate, summarize, clarify	Explain; clarify	answElab (Elaboration of an Answer)	
Answer	Inform Positive	Inform			Answer	
(Uncertain) Set Answer	Inform Positive	Inform		Reply-w	Answer	
(Uncertain) Prop. Answer	Inform Positive	Feedback Positive or Negative	Evaluation	Reply-y/ Reply-n	Answer	Positive/ negative answer
(Uncertain) Confirm	Inform Positive	Confirm		Reply-y	Confirm	Positive Answer
(Uncertain) Disconfirm	Inform Negative	Disconfirm; Reject		Reply-n		Negative answer

Table 11 — Information providing functions in DIT, AMI, Verbmobil, TRAINS, HCRC MapTask, SPAAC and Maltus.

DIT	LIRICS	DAMSL	SWBD-DAMSL	MRDA	Coconut
Question	Question	Forward-looking	Forward-looking/ directive	Question	Forward-looking: directive
Set question Propos. Question Check: Posi/Nega-Check	Set question Propos. Question Check Question		WH-question YN-question Declarative ques- tions Tags-Questions	Wh-question Y/N Question Checks: follow me; understanding check; Declarative Ques- tion; Tag Question	
Alternative ques- tion Ind. Set Question Ind. Prop.Question Ind. Alt. Question	Alternative ques- tion Ind. Set Question Ind. Prop.Question Ind. Alt. Question	Info-Request	OR-question/Or- clause Open-ended Questions Rhetorical ques- tions	OR-question/Or- clause Open-ended Questions Rhetorical ques- tions	Info-Request

Table 12 — Information seeking functions in DIT, LIRICS, DAMSL, SWBD-DAMSL, MRDA and Coconut taxonomies.

DIT	AMI	Verbmobil	TRAINS	HCRC	SPAAC	Maltus
(Indirect) Set Question	Elicit Inform	Request Sug- gest	WHQ	Query-w		
(Indirect) Prop.Question Check: Posi/Nega- Check (Indirect) Alternative Question	Elicit-offer-or- suggestion	Request Comment Request Clar- ify	YN-Question Check	Query-yn Check/ Align Query-w	Req.Direct ReqModal	Question Attention Question

Table 13 — Information seeking functions in DIT, AMI, Verbmobil, TRAINS, HCRC MapTask, SPAAC and Maltus.

DIT	LIRICS	DAMSL	SWBD-DAMSL	MRDA	Coconut
Offer	Offer	Offer	Offer	Suggestion	Offer
Promise	Promise	Commit		Commitment	Commit
Address Request	-	Maybe; accept-part; reject-part	Maybe; accept-part; reject-part	Maybe; Partial Accept; Partial Reject	Maybe; accept-part; reject-part
Accept Request	Accept Request	Commit	Commit	Commitment	Commit
Decline Request	Decline Request	Reject	Reject; Dispreferred responses	Reject; Dispreferred answer	Reject
Address Suggestion	-	Maybe; accept-part; reject-part	Maybe; accept-part; reject-part	Maybe; Partial Accept; Partial Reject	Maybe; accept-part; reject-part
Accept Suggestion	-	Accept	Accept	Accept	Accept
Decline Suggestion	-	Reject	Reject; Dispreferred responses	Reject; Dispreferred answer	Reject
Other commissives	-	Other forward-looking functions	Other forward-looking functions	-	Other forward-looking functions
Ind. Request; Request	Request	Action-Directive	Action-Directive	Command; repetition request	Action-Directive; clarification Request; Request
Instruct	Instruct	Action-Directive	Action-Directive	Command	Action-Directive
Address Offer	-	Maybe; accept-part; reject-part	Maybe; accept-part; reject-part	Maybe ; Partial Accept; Partial Reject	Maybe; accept-part; reject-part
Accept Offer	Accept Offer	Accept	Accept	Accept	Accept
Decline Offer	Decline Offer	Reject	Reject; Dispreferred responses	Reject; Dispreferred answer	Reject
Suggestion	Suggestion	Open-Option	Open-Option	Suggestion	Open-Option
Other directives	-	Other forward-looking functions	Other forward-looking functions	-	Other forward-looking functions

Table 14 — Action discussion functions in DIT, LIRICS, DAMSL, SWBD-DAMSL, MRDA and Coconut.

DIT	AMI	Verbmobil	TRAINS	HCRC Map-Task	SPAAC	Maltus
Offer	Offer	Offer	Offer	-	Offer	-
Promise	Offer	Commit	Promise		Inform Intent	Commit
Address Request	Inform Positive/ Partial/ Uncertain	Feedback	-		-	Other answer
Accept Request	Inform Positive	Accept	Accept		Accept	Positive answer
Decline Request	Inform Negative	Reject	Reject		Refuse	Negative answer
Address Suggestion	Inform Positive/ Partial/ Uncertain	Feedback	-		-	answer
Accept Suggestion	Inform Positive	Accept	Accept		Accept	Positive answer
Decline Suggestion	Inform Negative	Reject	Reject		Refuse	Negative answer
Other commissives	others	-	-		-	DO
Ind. Request	Elicit-offer-or-suggestion	Request: Request	Request		Direct	DO
Instruct	Inform	Request Commit		Command: Instruct		DO
Address Offer	Inform Positive/ Partial/ Uncertain	Feedback	-	-	-	Other answer
Accept Offer	Inform Positive	Accept	Accept		Accept	Positive answer
Decline Offer	Inform Negative	Reject	Reject		Refuse	Negative answer
Suggestion	Suggest	Suggest	Suggest		Suggest	Suggest
Other directives	others					- DO

Table 15 — Action discussion functions in DIT, AMI, Verbmobil, TRAINS, HCRC MapTask, SPAAC and Maltus.

Annex G

(informative)

Editors, contributors and meetings

G.1 Editors and contributors

International Standard 24617-2 was prepared by Technical Committee ISO/TC 37, *Terminology and Other Language Resources*, Subcommittee 4, *Language Resource Management*, Working Group 2, *Representation schemas*, following up on the EU-supported project LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems) in collaboration with TC 37/SC 4 ad-hoc Thematic Domain Group 3, Semantic content, and the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information. The LIRICS project was coordinated by Laurent Romary, INRIA-LORIA; the ACL SIGSEM Working Group is headed by Harry Bunt, Tilburg University.

This document has been produced by Harry Bunt with support from members of the editorial project group and the external consultancy group.

The editorial project group for this project consisted of:

- Jan Alexandersson (DFKI, Saarbrücken, Germany)
- Harry Bunt (Tilburg University, The Netherlands)
- Jean Carletta (University of Edinburgh, Scotland)
- Jae-Woong Choe (Korea University, Seoul, Korea)
- Alex Chengyu Fang (City University of Hong Kong)
- Koiti Hasida (AIST, Tokyo, Japan)
- Volha Petukhova (Tilburg University, The Netherlands)
- Andrei Popescu-Belis (IDIAP Research Institute, Martigny, Switzerland)
- Claudia Soria (Istituto di Linguistica Computazionale, Pisa, Italy)
- David Traum (University of Southern California, Marina del Rey, USA)

Authors of background LIRICS documents and publications include Amanda Schiffrin, Jeroen Geertzen, Volha Petukhova and Laurent Romary; participants in LIRICS project work include Nicoletta Calzolari, Anna Joan Casademont, Tomaso Casselli, Monica Monachini, and Valeria Quocchi. Through participation in joint TDG 3/SIGSEM WG meetings many other people have directly or indirectly contributed to this work, including Chu-Ren Huang, Nancy Ide, Dafydd Gibbon, Jerry Hobbs, Simon Keizer, James Pustejovsky and Thorsten Trippel.

The 'Expert Consulting Group' of the project consisted of:

- James Allen
- Jens Allwood
- Nick Campbell
- Roberta Catizone
- Thierry Declerk

- Anna Esposito
- Raquel Fernández
- Giacomo Ferrari
- Gil Francopoulo
- Dirk Heylen
- Julia Hirschberg
- Kristiina Jokinen
- Maciej Karpinski
- Staffan Larsson
- Kiyong Lee
- Oliver Lemon
- Carlos Martinez-Hinarejos
- Paul Mc Kevitt
- Michael McTear
- David Novick
- Tim Paek
- Patrizia Paggio
- Catherine Pelachaud
- Massimo Poesio
- German Rigau
- Laurent Romary
- Nicla Rossini
- Milan Rusko
- Candice Sidner
- Marieke van Erp
- Ielka van der Sluis
- Pavel Smrz
- Kristinn Thorisson
- Yorick Wilks
- Aesun Yoon

G.2 Meetings and Workshops

ISO TC 37/SC 4/TDG 3 had its inaugural meeting in Lisbon, Portugal on May 24-25, 2004. The following meetings have taken place since:

- January 10-11, 2005 in Tilburg, The Netherlands.
ISA-1: Joint workshop with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information, in conjunction with the 6th International Conference on Computational Semantics (IWCS-6).
 - September 22-23, 2005, at INRIA-LORIA in Nancy, France during the Nancy Inference Week.
ISA-2 Joint workshop with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information.
 - January 20-22, 2006 at Jeju island, Korea (as part of TC 37/SC 4 meeting).
 - April 20-22, 2006, Marina del Rey, California, Institute for Information Sciences (ISI).
Invitation-only joint workshop with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information.
 - August 22-24, 2006, Beijing (at ISO TC 37 annual meeting).
 - October 26-28, 2006, Brandeis University, Boston, USA.
Meeting to start the work on temporal annotation in the form of ISO TC 37/SC 4 project "Semantic Annotation Framework (SemAF) Part 1, Time and Events".
 - January 8-9, 2007, Tilburg, The Netherlands.
ISA-3: Joint workshop with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information in conjunction with the 7th International Conference on Computational Semantics (IWCS-7).
 - May 7-9, 2007, Paris, France, at AFNOR, in conjunction with a meeting of the LIRICS project.
 - January 12-13, 2008, Hong Kong, City University (in conjunction with ISO TC 37/SC 4 meeting and ICGL conference).
 - September 29 - October 1, 2008, Pisa, Italy, CNR Istituto di Linguistica Computazionale.
First meeting of the project editorial group.
 - January 5-6, 2009, Tilburg, The Netherlands.
ISA-4: 4th International Workshop in Interoperable Semantic Annotation. Joint workshop with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information, in conjunction with the 8th International Conference on Computational Semantics (IWCS-8), January 7-9.
 - May 29-31, 2009, Boulder, Colorado, USA
ISO TC 37/SC 4/WG 2 meeting, including meeting of the project editorial group and External Consultancy Group.
 - September 15-16, 2009, Tilburg, The Netherlands.
ISO TC 37/SC 4/WG 2 meeting, including meeting of the project editorial group and External Consultancy Group.
 - January 15-17, 2010, Hong Kong
ISA-5, Fifth Joint ISO-SIGSEM Workshop on Interoperable Semantic Annotation, including meeting of the project editorial group.
-

Bibliography

- [1] Ahn, R. (2001) *Agents, Objects and Events. A computational approach to knowledge, observation, and communication*. PhD Thesis, Eindhoven University of Technology.
- [2] Ahrenberg, L., N. Dahlback & A. Jonsson (1995) Coding Schemes for Studies of Natural Language Dialogue. In: *Proceedings from the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, Stanford, pp. 8–13.
- [3] Allen, J. & M. Core (1997) DAMSL: Dialogue Act Markup in Several Layers (Draft 2.1). Technical Report, Multiparty Discourse Group, Discourse Resource Initiative, September/October 1997.
- [4] Allen, J. & R. Perrault (1979) Plans, Inferences, and Indirect Speech Acts. In *Proceedings of the 17th Annual Meeting of the ACL*.
- [5] Allen, J. and L. Schubert and G. Ferguson and P. Heeman and C.H. Hwang and T. Kato and M. Light and N. Martin and B. Miller and M. Poesio and D. Traum (1994) , The TRAINS project: A case study in defining a conversational planning agent. Technical Report 532, Department of Computer Science, University of Rochester, New York.
- [6] 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. :DFKI, Saarbrücken.
- [7] Allwood J. (1992) On dialogue cohesion. *Gothenburg papers in Theoretical Linguistics* 65, Gothenburg University, Department of Linguistics.
- [8] Allwood J. (1994) Obligations and options in dialogue. *THINK Quarterly* 3(1): 9–18.
- [9] Allwood J. (1997) A critical look at speech act theory. In: O. Dahl (ed.) *Logic, Pragmatics, and Grammar*.
- [10] Allwood, J. (2000). An activity-based approach to pragmatics. In: H. Bunt & W. Black, eds. *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics*. John Benjamins, Amsterdam, pp. 47–80.
- [11] Allwood, J., J. Nivre, & E. Ahlsén (1992) *On the Semantics and Pragmatics of Linguistic Feedback*. *Journal of Semantics*, 9(1): 1–26.
- [12] Allwood, J., J. Nivre & E. Ahlsén (1994) *Semantics and Spoken Language Manual for Coding Interaction Management*. Report from the HSFR project *Semantik och talsprak*.
- [13] AMI (Augmented Multiparty Interaction) Consortium (2005) *Guidelines for Dialogue Act and Addressee Annotation Version 1.0*, October 13, 2005. Unpublished manuscript..
- [14] Appelt, D. (1982) *Planning Natural Language Utterances to Satisfy Multiple Goals*. SRI International AI Center Technical Note 259.
- [15] Austin, J.L. (1962) *How to do things with words*. Clarendon Press, Oxford, UK.
- [16] Bach, K. and R. Harnish (1979) *Linguistic Communication and Speech Acts*. MIT Press, Cambridge, Ma.
- [17] Bales, R.F. (1951) *Interaction process analysis: a method for the study of small groups*. Addison-Wesley, Cambridge.
- [18] Bunt, H. (1995) Dynamic Interpretation and Dialogue Theory. in M. Taylor, D. Bouwhuis & F. Néel (eds.) *The Structure of Multimodal Dialogue*, Vol. 2. Amsterdam: Benjamins, 139-166.
- [19] Bunt, H. (2000) Dialogue Pragmatics and Context Specification. In: Harry Bunt & William Black (eds.) *Abduction, Belief and Context in Dialogue*. Benjamins, Amsterdam: 81–150.
- [20] Bunt, H. (2004) A Framework for Dialogue Act Specification. 4th Joint ISO-SIGSEM Workshop on Semantic Annotation, Tilburg, January 2005.

- [21] Bunt, H. (2006) Dimensions in Dialogue Act Annotation. In Proceedings of the 5th International Conference on Language Resources and Systems (LREC 2006), Genova, May 2006, pp. 919–924.
- [22] Bunt, H. (2007a) Multifunctionality and multidimensional dialogue act annotation. In: E. Ahlsén et al., eds., *Communication - Action - Meaning*. Gothenburg University, pp. 237-259.
- [23] Bunt, H. (2007b) The Semantics of Semantic Annotation. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation (PACLIC-21)*, Seoul, November 2007, pp. 13-28.
- [24] Bunt, H. (2009) Multifunctionality and multidimensional dialogue semantics. In *Proceedings of the Workshop on the Semantics on the Semantics and Pragmatics of Dialogue (DiaHolmia)*, Stockholm, July 2007, pp. 3–14.
- [25] Bunt, H. (2010a) A methodology for designing semantic annotation languages. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, Hong Kong, January 2010, pp. 29–46.
- [26] Bunt, H. (2010b) Multifunctionality in dialogue. *Computers, Speech and Language, special issue on dialogue modelling*, Yorick Wilks, editor.
- [27] Bunt, H. & Y. Girard (2005) Designing an open, multidimensional dialogue act taxonomy. In C. Gardent & B. Gaiffe (eds.) DIALOR'05, Proceedings of the Ninth Workshop on the Semantics and Pragmatics of Dialogue, Nancy, June 2005, 37-44.
- [28] Bunt, H., R. Morante and S. Keizer (2007) An empirically based computational model of grounding in dialogue. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, September 2007, pp. 283–290.
- [29] Bunt, H. & L. Romary (2002) Towards multimodal content representation. In *Proceedings LREC 2002 Workshop on International Standards of Terminology and Language Resources*, Las Palmas, May 2002. ELRA, Paris, pp. 54-60.
- [30] Bunt, H. & L. Romary (2004) Standardization in Multimodal Content Representation: Some Methodological Issues. Proceedings 4th International Conference on Language Resources and Systems (LREC 2004), Lisbon, May 2004, pp. 2219-2222.
- [31] Bunt, H. & A. Schiffrin (2006) Methodological aspects of semantic annotation. In *Proceedings 5th International Conference on Language Resources and Systems (LREC 2006)*, Genova, May 2006, pp. 1444-1449.
- [32] Campbell, N. (2008) Individual Traits of Speaking Style and Speech Rhythm in a Spoken Discourse. In *Verbal and non-verbal features of human-human and human-machine interaction*. Revised papers from COST Action 2102 International Conference, Patras, Greece.
- [33] Carletta, J., A. Isard, S. Isard, J.Kowtko & G. Doherty-Sneddon (1996) HCRC dialogue structure coding manual. Technical Report HCRC/TR-82.
- [34] Clark, H. (1996) *Using language*. Cambridge University Press, Cambridge, UK.
- [35] Clark, H., & J. Fox Tree (2002) *Using 'uh' and 'um' in spontaneous speech*. *Cognition*, 84, pp. 73–111.
- [36] Clark, H. & M.A. Krych (2004) *Speaking while monitoring addressees for understanding*. *Journal of Memory and Language*, 50: 62–81.
- [37] Clark, D. & A. Popescu-Belis (2004) Multi-level Dialogue Act Tags. In Proceedings of SIGDial 2004, 5th SIGDial Workshop on Discourse and Dialogue, Cambridge, MA, pp. 163–170.
- [38] Cooper, R. (2004) A type-theoretic approach to information state update in issue-based dialogue management. Invited paper at CATALOG'04, Eighth Workshop on the Semantics and Pragmatics of Dialogue, Barcelona. Submitted for publication in a collection in memory of Jon Barwise.
- [39] Core, M., M. Ishizaki, J. Moore, C. Nakatani, N. Reithinger, D. Traum & S. Tutiya (1998) The Report of The Third Workshop of the Discourse Resource Initiative, Chiba University, Chiba, Japan.
- [40] Core, M. & J. Allen (1997) Coding Dialogs with the DAMSL Annotation schema. *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, MA.

- [41] Di Eugenio, B., P.W. Jordan & L. Pylkkänen (1998) The COCONUT project: dialogue annotation manual. ISP Technical Report 98-1, December 1998, University of Pittsburgh.
- [42] Duncan, S. & D. Fiske (1977) *Face-to-face interaction: research, methods, and theory*. Lawrence Erlbaum Associates, New Jersey.
- [43] Ekman, P. (1972) *Universals and cultural differences in facial expressions of emotion*. In J/ Cole (ed.) *Nebraska Symposium on Motivation*, 1971. Lincoln, Neb., University of Nebraska Press, pp. 207–283.
- [44] Geertzen, J., V. Petukhova & H. Bunt (2007) *A multidimensional approach to utterance segmentation and dialogue act classification*. ÅIn *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, September 2007, pp. 140–149.
- [45] Goffman, E. (1963) *Behavior in Public Places*. Basic Books, New York.
- [46] Goldman-Eisler, F. (1968) *Psycholinguistics: experiments in spontaneous speech*. New York, Academic Press.
- [47] Heeman, P. & J. Allen (1999) Speech repairs: intonational phrases and discourse markers: Modellig speakers' utterances in spoken dialogue. *Computational linguistics* 1293):1–45.
- [48] Hovy, E. & E. Maier (1993) Parsimonious or Profligate: How Many and Which Discourse Structure Relations? Unpublished manuscript.
- [49] Ichikawa, A., M. Araki, M. Ishizaki, S. Itabashi, T. Itoh, H. Kashioka, K. kato, H. Kikuchi, T. Kumagai, A. Kurematsu, H. Koiso, M. Tamoto, S. Tutiya, S. Nakazato, Y. Horiuchi, K. Maekawa, K. Yamashita, T. Yoshimura (1998) Standardizing Annotation Schemes for Japanese Dicourse. In: *Proceedings of the First International Conference on Language resources and Evaluation*, Madrid.
- [50] Ide, N. & H. Bunt (2010) Anatomy of Annotation Schemes: Mapping to GrAF. In *Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV)*, Uppsala, Sweden.
- [51] Ide, N. & L. Romary (2004) International Standard for a Linguistic Annotation Framework. *Natural Language Engineering* 10:221-225.
- [52] [Ide, N. & K. Suderman (2007) GrAF: A graph-based format for linguistic annotations. In *Proceedings of the First Linguistic Annotation Workshop (LAW-I)*, Prague, Czech Republic, pp. 1–8.
- [53] ISO 24610-1:2006 Language resource management: Feature structure, Part 1: Feature structure representation. ISO, Geneva.
- [54] ISO 24612:2009 Language resource management: Linguistic annotation framework (LAF). ISO, Geneva.
- [55] Jekat, S., A. Klein, E. Maier, I. Maleck, M. Mast & J.J. Quantz (1995) Dialogue acts in Verbmobil. Verbmobil Report 65, April 1995. DFKI, Saarbrücken.
- [56] Jurafsky, D., E. Shriberg, and D. Biasca (1997) Switchboard SWBD-DAMSL Shall-Discourse-Function Annotation Coders Manual.
- [57] Kendon, A. (2004) *Gesture: Visible action as utterance*. Cambridge University Press.
- [58] Kipp, M. (2001) Anvil - A Generic Annotation Tool for Multimodal Dialogue. In *Proceedings of the Seventh International Conference on Speech Communication and Technology (Eurospeech)*, Aalborg, pp. 1367–1370. See also <http://www.anvil-software.de>
- [59] Klein, M. (1998) Supported Coding Schemes. EU Project LE4-8370 MATE Deliverable 1.1.
- [60] Klein, M. (1999) Standardization efforts on the level of dialogue acts in the MATE project. ÅProceedings of ACL 1999 Workshop on Standards and Tools for Discourse Tagging, pp. 35–41.
- [61] Klein, M. & C. Soria (1998) Dialogue acts. In: M. Klein (ed.) ÅSupported Coding Schemes. EU Project LE4-8370 MATE Deliverable 1.1.
- [62] Larsson, S. (1998) Coding schemas for dialog moves. Unpublished manuscript, available at <http://www.ling.gu.se/~sl>.

- [63] Lascarides, A. & N. Asher (2007) Segmented Discourse representation Theory: Dynamic Semantics with Discourse Structure. In Harry Bunt and Reinhard Muskens (eds) *Computing Meaning*, Volume 3, pp. 87–124.
- [64] Levelt, W. (1983) Monitoring and Self repair in Speech. *Cognition* 14, 41–104.
- [65] Levelt, W. (1989) *Speaking: From Intention to Articulation*. MIT Press, Cambridge, Massachusetts.
- [66] Levinson, S. (1983) *Pragmatics*. Cambridge University Press, Cambridge, UK.
- [67] LIRICS D4.2 (2006) Preliminary set of semantic data categories. Deliverable D4.2 of eContent project LIRICS.
- [68] LIRICS D4.3 (2007) Documented compilation of semantic data categories. Deliverable D4.3 of eContent project LIRICS.
- [69] LIRICS D4.4 (2007) Multilingual test suites for semantically annotated data. Deliverable D4.4 of eContent project LIRICS.
- [70] Mann, W. & S. Thompson (1988) *Rhetorical Structure Theory: Toward a functional theory of text organization*. MIT Press, Cambridge, MA.
- [71] McNeill, D. (2005) *Gesture and Thought*. Chicago University Press.
- [72] Petukhova, V. (2005) Multidimensional interaction of multimodal dialogue acts in meetings. MA thesis, Tilburg University.
- [73] Petukhova, V. & Bunt, H. (2009a) Towards a Multidimensional Semantics of Discourse Markers in Spoken Dialogue, In *Proceedings of the Eighth International Workshop on Computational Semantics (IWCS-8)*, Tilburg, the Netherlands, pp. 157–168
- [74] Petukhova, V. & H. Bunt (2009b) Dimensions in dialogue act annotation schemes: a survey. TiCC, Tilburg Centre for Creative Computing Technical Report TR 2009-003, Tilburg University.
- [75] Petukhova, V. & H. Bunt (2009c) The independence of dimensions in multidimensional dialogue act annotation. in *Proceedings of Joint NAACL-HLT Conference*, Boulder, Colorado, May 2009.
- [76] Petukhova, V. & H. Bunt (2009d) Who's next? Speaker selection mechanisms in multiparty dialogue. In *Proceedings of the Workshop on the Semantics on the Semantics and Pragmatics of Dialogue (DiaHolmia)*, Stockholm, pp. 19–26.
- [77] Petukhova, V. & H. Bunt (2009e) Grounding by nodding. In *Proceedings of GESPIN 2009: Gestures and Speech in Interaction*, Poznań, September 2009.
- [78] Petukhova, V. & H. Bunt (2010) Introducing communicative function qualifiers. In *Proceedings Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, Hong Kong, January 2010, pp. 123–131.
- [79] Poesio, M. & D. Traum (1998) Towards an Axiomatization of Dialogue Acts. In *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues*, Enschede, pp. 207–222.
- [80] Popescu-Belis, A. (2005) Dialogue Acts: One or More Dimensions? ISSCO Working Paper 62, ISSCO, Geneva. November 2005.
- [81] Popescu-Belis, A. (2008) Dimensionality of Dialogue act Tagsets: An Empirical Analysis of large Corpora. *Language resources and Evaluation* 42(1): 99-107.
- [82] Sanders, T., (1992) Semantic and pragmatic sources of coherence: On the categorization of coherence relations in context. *Discourse Processes* 24(1): 119–147.
- [83] Searle, J.R. (1969) *Speech Acts*. Cambridge, UK: Cambridge University Press.
- [84] Searle, J.R. (1975) Indirect Speech Acts. In P. Cole and J. Morgan (eds.) *Syntax and Semantics, Volume 3: Speech Acts*. Academic Press, New York, pp. 59–82.

- [85] Sacks, H., E. Schegloff & G. Jefferson (1974) A simplest systematics for the organization of turn-taking for conversation. *Language* 53: 696–735.
- [86] Smith, V.L., & H. Clark (1993) *On the course of answering questions*. *Journal of Memory and Language*, 32: 25–38.
- [87] Soria, C. & V. Pirrelli (2003) A multi-level annotation meta-schema for dialogue acts. In A. Zampolli, N. Calzolari & L. Cignoni (eds.), *Computational Linguistics in Pisa. Linguistica Computazionale, Special Issue, XVIII-XIX*, Pisa-Roma: IEPI, pp. 865–900.
- [88] Traum, D. (1999) Speech Acts for Dialogue Agents. In: M. Wooldridge & A. Rao (eds.) *Foundations of rational agency*, Kluwer, Dordrecht, 169–201.
- [89] Traum, D. (2000) 20 Questions on Dialogue Act Taxonomies. *Journal of Semantics* 17(1): 7–30.
- [90] Traum, D. & E. Hinkelman (1992) Conversation acts in task-oriented spoken dialogue. *Computational Intelligence* 3(8): 575-599.
- [91] Traum, D. & S. Larsson (2003) The Information State Approach to Dialogue Act Management. In J. van Kuppevelt & R. Smith (eds.) *Current and new Directions in Discourse and Dialogue*, Kluwer, Dordrecht.