BS ISO 24617-2:2012

BSI Standards Publication

Language resource management — Semantic annotation framework (SemAF)

Part 2: Dialogue acts

National foreword

This British Standard is the UK implementation of ISO 24617-2:2012.

The UK participation in its preparation was entrusted to Technical Committee TS/1, Terminology.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Published by BSI Standards Limited 2013.

ISBN 978 0 580 66372 7

ICS 01.020

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 January 2013.

Amendments issued since publication

Date	Text affected

BS ISO 24617-2:2012

INTERNATIONAL STANDARD

ISO 24617-2

First edition
2012-09-01

Language resource management — Semantic annotation framework (SemAF) —

Part 2: Dialogue acts

Gestion des ressources langagières — Cadre d'annotation sémantique (SemAF) —

Partie 2: Actes de dialogue

Reference number
ISO 24617-2:2012(E)

COPYRIGHT PROTECTED DOCUMENT

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

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 24617-2 was prepared by Technical Committee ISO/TC 37, *Terminology and other language and content resources*, Subcommittee SC 4, *Language resource management*.

ISO 24617 consists of the following parts, under the general title: *Language resource management — Semantic annotation framework:*

— *Part 1: Time and events (SemAF-Time, ISO-TimeML)*

— *Part 2: Dialogue acts*

The following parts are under preparation:

— *Part 3: Named entities (SemAF-NE)*

— *Part 4: Semantic roles (SemAF-SRL)*

— *Part 5: Discourse structure (SemAF-DS)*

— *Part 6: Principles of semantic annotation (SemAF-Basics)*

— *Part 7: Spatial information (ISO-Space)*

— *Part 8: Semantic relations in discourse (SemAF-DRel)*

INTERNATIONAL STANDARD **ISO 24617-2:2012(E)**

Language resource management — Semantic annotation framework (SemAF) —

Part 2: Dialogue acts

1 Scope

This part of ISO 24617 provides a set of empirically and theoretically well-motivated concepts for dialogue annotation, a formal language for expressing dialogue annotations — the dialogue act markup language (DiAML) — and a method for segmenting a dialogue into semantic units. This allows the manual or automatic annotation of dialogue segments with information about the communicative actions which the participants perform by their contributions to the dialogue. It supports multidimensional annotation, in which units in dialogue are viewed as having multiple communicative functions. The DiAML language has an XML-based representation format and a formal semantics which makes it possible to apply inference to DiAML representations.

This part of ISO 24617 specifies data categories for reference sets of communicative functions and dimensions of dialogue analysis and provides principles and guidelines for extending these sets or selecting coherent subsets of them. Additionally, it provides guidelines for annotators and annotated examples. It is applicable to spoken, written and multimodal dialogues involving two or more participants.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 12620:2009, *Terminology and other language resources — Specification of data categories and management of a Data Category Registry for language resources*

ISO 24610-1:2006, *Language resource management — Feature structures — Part 1: Feature structure representation*

ISO 24612:2011, *Language resource management — Linguistic annotation framework*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.[1)]

1) In this document, “he”, “him” and “his” are used in a generic sense, without implying any gender-related distinctions.

3.1
addressee
dialogue (3.5) *participant* (3.13) oriented to by the *sender* (3.18) in a manner to suggest that his *utterances* (3.22) are particularly intended for this participant and that some response is therefore anticipated from this participant, more so than from the other participants

Note to entry: This definition is a *de facto* standard in the linguistics literature. It has been slightly modified here, in replacing “speaker” by “sender” and avoiding the use of ambiguous pronouns. Goffman's original definition says: “dialogue participant oriented to by the speaker in a manner to suggest that his utterances are particularly intended for him and that some response is therefore anticipated from him/her, more so than from the other participants”.

[SOURCE: Goffman (1981).]

3.2
allo-feedback act
feedback act (3.8) where the *sender* (3.18) elicits information about the *addressee*'s (3.1) processing of an *utterance* (3.22) that the sender contributed to the *dialogue* (3.5) or where the sender provides information about his perceived processing by the addressee of an utterance that the sender contributed to the dialogue before

EXAMPLE A: Now move up.

B: Slightly northeast you mean?

A: Slightly yeah.

A performs an allo-feedback act signalling that he thinks B understood his first utterance correctly.

3.3
auto-feedback act
feedback act (3.8) where the *sender* (3.18) provides information about his own processing of an utterance (3.22) contributed to the *dialogue* (3.5) by another *participant* (3.13)

EXAMPLE B's utterance in the example dialogue fragment in (3.2) signals that he is uncertain whether he understood the previous utterance correctly.

3.4
communicative function
property of certain stretches of communicative behaviour, describing how the behaviour changes the *information state* (3.12) of an understander of the behaviour

Note to entry: A communicative function may be “qualified”, i.e. one or more **qualifiers** (3.14) may be associated with it. For example, an answer may be qualified as “uncertain” and the acceptance of a request may be “conditional”. See 10.3 for explanation and examples.

3.5
dialogue
exchange of *utterances* (3.22) between two or more persons or artificial conversational systems

3.6
dialogue act
communicative activity of a *dialogue* (3.5) *participant* (3.13), interpreted as having a certain *communicative function* (3.4) and *semantic content* (3.16)

Note to entry: A dialogue act may also have certain *functional dependence relations* (3.10), *rhetorical relations* (3.15) and *feedback dependence relations* (3.9) with other units in a *dialogue* (3.5).

3.7
dimension
class of *dialogue acts* (3.6) that are concerned with a particular aspect of communication, corresponding to a particular category of semantic content

EXAMPLE Dialogue acts advancing the task or activity that motivates the dialogue (the Task dimension), dialogue acts providing and eliciting feedback (the Auto- and Allo-Feedback dimensions) and dialogue acts for allocating the speaker role (the Turn Management dimension).

Note to entry: See Clauses 5, 7 and 9 for discussion and more examples.

3.8
feedback act
dialogue act (3.6) which provides or elicits information about the *sender*'s (3.18) or the *addressee*'s (3.1) processing of something that was uttered in the dialogue

Note to entry: Two classes of feedback are distinguished in this part of ISO 24617: *allo-feedback acts* (3.2) and *auto-feedback acts* (3.3).

3.9
feedback dependence relation
relation between a *feedback act* (3.8) and the stretch of communicative behaviour whose processing the act provides or elicits information about

EXAMPLE In the example that accompanies definition 3.2, both the allo-feedback act expressed by utterance 3 and the auto-feedback act expressed by utterance 2 have a feedback dependence relation to utterance 1.

3.10
functional dependence relation
relation between a given *dialogue act* (3.6) and a preceding dialogue act on which the semantic content of the given dialogue act depends due to its *communicative function* (3.4)

EXAMPLE The relation between an answer and the corresponding question, such as between utterance 3 and utterance 2 in the example accompanying definition 3.2; or the relation between the acceptance of an offer and the corresponding offer.

Note to entry: A dialogue act, A2, may also depend on another dialogue act, A1, occurring earlier in a dialogue because of relations between their semantic contents, e.g. because A2 contains a reference to an element occurring in A1. This is not a functional dependence relation, since it is not due to A2's communicative function.

3.11
functional segment
minimal stretch of communicative behaviour that has one or more *communicative functions* (3.4)

EXAMPLE The functional segment corresponding to the answer given by S in the following dialogue fragment does not include the parts *"Just a moment please"* and *".... let me see..."* but only the parts "*the first train to the airport on Sunday morning is"* and *"at 5:45"*:

1. U: What time is the first train to the airport on Sunday morning please?
2. S: Just a moment please... the first train to the airport on Sunday morning is .... let me see... at 5:45.

Note 1 to entry: A consequence of this definition is that functional segments may be discontinuous, may overlap or be embedded and may contain parts contributed by different participants.

Note 2 to entry: The condition of being "minimal" ensures that functional segments do not include material that does not contribute to the expression of a communicative function that identifies the segment.

3.12
information state
context
totality of a *dialogue* (3.5) *participant*'s (3.13) beliefs, assumptions, expectations, goals, preferences, hopes and other attitudes that may influence the participant's interpretation and generation of communicative behaviour

3.13
participant
person or artificial agent involved in the exchange of *utterance*s (3.22)

3.14
qualifier
predicate that can be associated with a *communicative function* (3.4)

EXAMPLE A: Would you like to have some coffee?

B: Only if you have it ready.

B's utterance accepts A's offer under a certain condition; this can be described by qualifying the communicative function Accept Offer with the predicate “conditional”.

Note to entry: See 10.3 for more examples.

3.15
rhetorical relation
relation between two *dialogue acts* (3.6), indicating a pragmatic connection between the two or between their *semantic content*s (3.16)

EXAMPLE 1 The statement in the second utterance which follows provides a *motivation* for the question in the first utterance:

A: Can you tell me what flights there are to Sydney on Saturday? I'd like to attend my mother's 80th birthday.

EXAMPLE 2 A rhetorical relation between the semantic contents of two dialogue act occurs in the following, where the content of B's statement mentions a *cause* for the content of A's statement:

A: I can never find these stupid remote controls

B: That's because they don't have a fixed location

Note to entry: Relations such as *elaboration, explanation, justification, cause* 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 or propositions, described in text segments. See, for example, Hovy and Maier, 1992, Lascarides & Asher, 2007 or Mann & Thompson, 1988. Many of these relations also occur in dialogue, either as relations between dialogue acts or between the semantic contents of dialogue acts.

3.16
semantic content
information, situation, action, event or objects that a stretch of communicative behaviour refers to

3.17
semantic content category
semantic content type
kind of information, situation, action, event or objects that form the *semantic content* (3.16) of a *dialogue act* (3.6)

EXAMPLE The various *dimensions* (3.7) defined in this part of ISO 24617 correspond to categories of semantic content. In particular, the Task dimension corresponds to the category of task-specific actions and information; the Allo- and Auto-Feedback dimensions correspond to the categories of information about the processing by the current speaker or by the addressee, respectively, of something that was said before; the Turn Management dimension corresponds to the category of information about the allocation of the speaker role and so forth.

3.18
sender
dialogue (3.5) *participant* (3.13) who produces a *dialogue act* (3.6)

3.19
speaker
sender (3.18) of a *dialogue act* (3.6) in the form of speech, possibly combined with nonverbal communicative behaviour

Note to entry: A dialogue participant may say something while another participant occupies the *speaker role* (3.20), therefore the term “speaker” is not synonymous with “participant who occupies the speaker role”.

3.20
speaker role
role occupied by a *dialogue* (3.5) *participant* (3.13) who has temporary control of the *dialogue* and speaks for some period of time

[SOURCE: DAMSL Revised Manual.]

3.21
turn unit
stretch of communicative activity produced by one *participant* (3.13) who occupies the *speaker role* (3.20), bounded by periods where another participant occupies the speaker role

3.22
utterance
anything said, written, keyed, gesticulated or otherwise expressed

Note to entry: An utterance is mostly a part of what a sender contributes in a turn unit.

4 Purpose and justification

The notion of a dialogue act plays a key role in the analysis of spoken and multimodal dialogue, as well as in the design of spoken dialogue systems and embodied conversational agents. These activities all depend on the availability of dialogue corpora, annotated with dialogue act information.

Over the years a variety of dialogue act annotation schemes have been developed, such as those of the TRAINS human-computer dialogue project (Allen et al., 1994), the Map Task studies of human-human dialogue (Carletta et al., 1996) and of the Verbmobil speech translation project (Alexandersson et al., 1998). These schemes were developed for specific purposes and application domains. They contain overlapping sets of concepts and make use of often mutually inconsistent terminology, sometimes employing different terms for the same concept or the same term for different concepts.

The multidimensional DIT scheme (Bunt, 1984) was developed for information-seeking dialogues without depending on a particular domain. The DAMSL scheme (Dialogue Act Markup using Several Layers, Allen and Core,1997; Core et al., 1998) constitutes an application-independent multidimensional annotation scheme. The DIT^{++} scheme (Bunt, 2006; 2009) combines the DIT scheme with concepts from DAMSL and other more recent schemes into a comprehensive general-purpose annotation scheme.

In the EU-funded project LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems, Romary et al., 2007) a reference set of dialogue acts, taken from the DIT^{++} taxonomy, was defined in the form of data categories, following ISO 12620. This set of concepts has been tested for its usability and coverage a) in the manual annotation of spoken dialogues in English, Dutch and Italian and b) in the automatic annotation of spoken and multimodal dialogue in English and forms a significant part of the background of this part of ISO 24617.

The main purpose of this part of ISO 24617 is to define a reference set of domain-independent basic concepts for dialogue act annotation, plus a formal language, based on XML, for representing such annotations. Guidelines are provided for how to use the defined concepts and the annotation language, supported by extended examples. This formal language, the Dialogue act markup language (DiAML) has a formal semantics, which makes it possible to apply techniques for automatic reasoning to DiAML annotations.

Guidelines and principles are also provided for extending the set of concepts defined in this part of ISO 24617, for example, with domain-specific concepts, as well as for selecting coherent subsets.

5 Basic concepts and metamodel

The term “dialogue act” is often used rather loosely in the sense of a 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 modelling of linguistic and nonverbal communicative behaviour in dialogue.

Dialogue acts offer a way of characterizing the meaning of communicative behaviour in terms of update operations, to be applied to the information states of participants in the dialogue; this approach is commonly known as the “information-state update” or “context-change” approach — see e.g. Bunt (1989; 2000a); Traum and 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, it is understood that the speaker is reproaching 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 conten*t, which describes the objects, properties, relations, situations, actions or events that the dialogue act is about. The communicative function of a dialogue act specifies how an addressee should update his information state with the information expressed in the semantic content when he understands the dialogue act.

A dialogue act as defined in this part of ISO 24617 (3.6) is a semantic unit of communicative behaviour. Dialogue act annotation is the marking up of stretches of dialogue with information about the dialogue acts performed in these stretches and is often limited to assigning communicative function tags. A dialogue act being a semantic unit in communicative behaviour, the question arises as to which 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 therefore not very useful units of behaviour for assigning communicative functions to. Communicative functions can be assigned more accurately to smaller units, which are called *functional segments* and which are defined as the minimal stretches of communicative behaviour that are functionally relevant. See Clause 8 for more details about dialogue segmentation.

Inherent to the notion of a dialogue act is that there is an agent who produces the dialogue act, called the “sender” and one or more agents who are addressed, called “addressees”. Dialogue studies often focus on two-person dialogues, in which case the dialogue acts have only one addressee. Besides sender and addressee(s), there may be various types of side-participants who are present but do not or only marginally participate (see Clark, 1996).

Dialogue act annotation is often limited to assigning communicative functions to dialogue segments, which corresponds intuitively to indicating the type of communicative action that is performed. A semantically more complete characterization also provides information about the *type of semantic content.* The DAMSL annotation scheme distinguishes three categories of semantic content: 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. The DIT^{++} scheme distinguishes a number of subcategories of communication-related information, such as feedback information, turn allocation information and topic progression information. The various categories of semantic content are also called “dimensions” and are discussed in more detail in Clause 7.

Some types of dialogue acts are inherently dependent for their full meaning 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 being answered and for the acceptance or rejection of offers, suggestions, requests and apologies. The following example illustrates this, where the meaning of (1.3) clearly depends very much on whether it is an answer to the question (1.1) or to the question (1.2).

EXAMPLE 1

(1.1) B: Do you know who's coming tonight?

(1.2) B: Which of the project members d'you think will be there?

(1.3) A: I'm expecting Jan, Alex, Claudia and David, and maybe Olga and Andrei.

As an answer to (1.1), it says that nobody else is expected to come than the people that are mentioned, but as an answer to (1.2) it leaves open the possibility that other people will come, who are not members of "the project".

For dialogue acts which have such a dependence on other dialogue acts, due to their responsive character, the marking up of the links to these "antecedent" dialogue acts allows the annotation not just to express e.g. that the 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 shown in the following example:

EXAMPLE 2

(2.1) A: It ties you on in terms of the technology and the complexity that you want

(2.2) A: like for example voice recognition

(2.3) A: because you might need to power a microphone and other things

(2.4) A: so that's one constraint there

In this example[2)] we see a sequence of four functional segments contributed by the same participant. Segment (2.2) is related to the initial statement through an *Exemplification* relation and (2.3) through an *Explanation* relation, while (2.4) is related to the preceding three segments through a *Summarization* relation. Such relations are known as *rhetorical relations*. In view of the wide diversity of the sets of rhetorical relations that have been proposed (see, e.g., Mann and Thompson, 1988; Hovy and Maier, 1993; Sanders et al., 1992), this part of ISO 24617 does not propose any specific set of such relations, but only provides a conceptual category for which a particular set of relations may be specified.

Feedback-providing and eliciting acts also relate to what happened earlier in the dialogue, but in a different way. They are concerned with the processing of what was said before — such as its perception or its interpretation:

EXAMPLE 3

(3.1) A: Is this flight also available on Thursday?

(3.2a) B: On Thursday you said?

(3.2b) B: The twelfth you mean?

With utterance (3.2a), B checks whether he heard correctly what A said. This is a response to A's *utterance*, rather than to the dialogue act that the utterance expresses; with utterance (3.2b), by contrast, B checks whether he has correctly interpreted what A said. Both types of dependence are called a *feedback dependence relation*.

Note that nonverbal feedback, for instance in the form of nodding or vocal backchannels like "*uh-huh"*, *"um"*, "*huh*", "*m-hm"*, may have a feedback dependence relation to what is being said at that moment, rather than to what was said before. 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 above also illustrates another phenomenon that is frequently found in dialogue, namely that speakers may have incomplete or uncertain information. The use of "*maybe*" in (1.3) expresses that A is uncertain about part of the information that he provides.

2) From the AMI corpus, see http://corpus.amiproject.org.

In addition, speakers may express a certain sentiment about 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:

EXAMPLE 4

(4.1) A: Would you like to have some coffee?

(4.2) B: That would be great, thank you!

(4.3) B: Only if you have it ready.

For the annotation of conditions, uncertainty and sentiment, this part of ISO 24617 makes use of so-called *function qualifiers*, which can be attached to communicative functions — see 10.3 for more detail.

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 in Figure 1:

a) sender, addressee and participants in other roles (side-participants);

b) functional segment;

c) dialogue act, communicative function, communicative function qualifier and semantic content category (or "dimension");

d) functional dependence relation, rhetorical relation and feedback dependence relation.

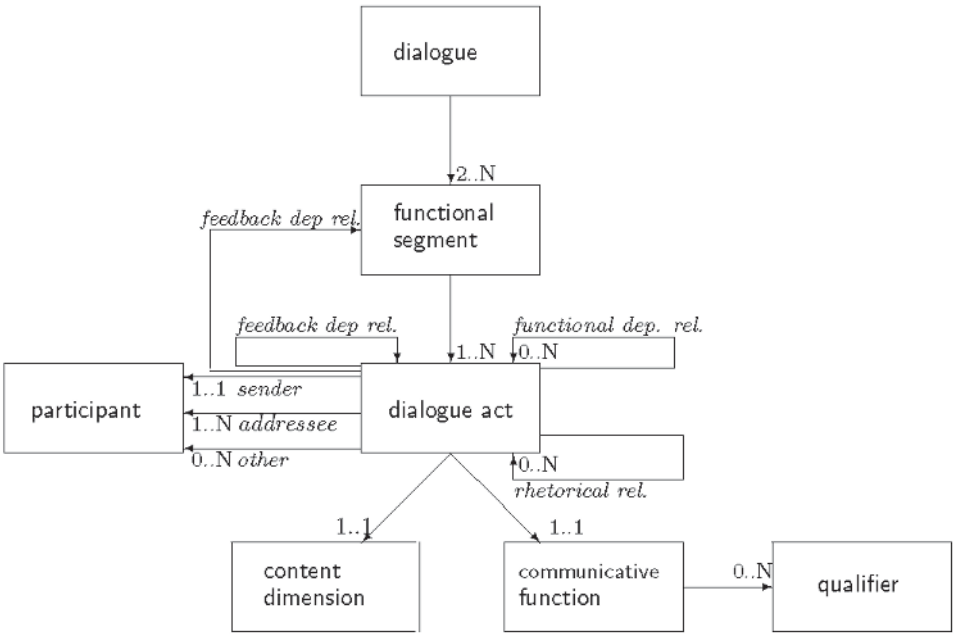

Figure 1 — Metamodel for dialogue act annotation

6 Definition of communicative functions

Existing dialogue act annotation schemes use 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. Defining a communicative function by its linguistic form has the advantage that its recognition can be straightforward, but runs into the problem that the same linguistic form can be used to express different functions. For example, the utterance "*Why don't you start?"* has the form of a question and can be intended as such, but can also be used to invite or encourage 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 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; however, that would only say something about the *form* of the behaviour, nothing about its communicative function. A repetition for instance often has a feedback function, as in (5.2a) but it can also have other functions, as in (5.3), where it is used as a confirmation in response to a check question:

EXAMPLE 5

(5.1) S: There are evening flights at seven-fifteen and eight-thirty

(5.2a) C: Seven-fifteen and eight-thirty

(5.2b) C: And that's on Sunday too

(5.3) S: And that's on Sunday too

This part of ISO 24617 follows a strictly semantic approach to the definition of communicative functions. But while linguistic form is taken not to be part of the definition of a communicative function, a requirement for introducing a communicative function is that there are ways in which a sender can indicate that his behaviour should be understood as having that particular function, by 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. Questions of the form *Do you know* [*X*] are illustrative: while an utterance of this form would standardly seem to ask an addressee whether he possess the knowledge [*X*], it is more often used to request the addressee to provide the information [*X*], if possible. This makes such a question a conditional request.

The full complexity of the phenomenon of indirect speech acts is beyond the scope of this part of ISO 24617, but an important class of indirect speech acts can be covered by qualifying them as conditional — see 10.3.

7 Annotation schemes

7.1 Structure of annotation schemes

Existing dialogue act annotation schemes can be divided into one-dimensional and multidimensional schemes. One-dimensional schemes have a set of mutually exclusive tags and are used for coding stretches of dialogue with a single tag. Multidimensional schemes, on the other hand, are intended for encoding stretches of dialogue with multiple tags. Schemes of the latter kind typically have a relatively large tag set. There are several advantages to the structuring of such a tag set into clusters of communicative functions tags:

— 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.

— The tags within a cluster are usually mutually exclusive; this has the advantage that, once a particular tag has been assigned, the rest of the tags within that cluster do not need to considered any further. If a cluster is hierarchically organized, as is the case in this part of ISO 24617, 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). Communication is thus a complex, multi-faceted activity and this is reflected in the multifunctionality that dialogue utterances often exhibit.

Multifunctionality comes in a variety of forms. Allwood (1992) distinguishes between *sequential* and *simultaneous* multifunctionality and provides the following example as an illustration:

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. In Example 6 we see A's utterance containing five functional segments, with communicative functions such as *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 are considered as markables.

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

EXAMPLE 7

(7.1) A: Do you know what date it is?

(7.2) B: Today is the fifteenth.

(7.3) A: Thank you.

A's utterance (7.3) 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 participant in an unsuccessful dialogue may just want to terminate the interaction in a polite way. The feedback function of the thanking in (7.3) can be inferred along the following lines: By saying *"Thank you",* A expresses his gratitude to B. This can only be for what B just said; this would constitute a reason for being grateful if A considers B's utterance as relevant and useful, which means that A accepted B's utterance as an answer to his question. The feedback function in such a case can be viewed as a conversational implicature (Grice, 1979), i.e. as a contextually plausible consequence which the addressee is intended to infer.

The implication relation between thanking and positive feedback is different from that between a propositional answer (*"yes"* or "*no"*) and a confirmation, where the relation is one of *entailment*, i.e. an implication which is logically valid. (Every confirmation by its very nature is also an answer.) Entailment relations occur when the definition of one communicative function is a special case of that of another.

It may be argued that such cases should not be considered as instances of multifunctionality, e.g. a speaker who wants to issue a confirmation can hardly have the intention of *additionally* giving 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 dialogue fragment:

EXAMPLE 8

(8.1) A: Is that your opinion too?

(8.2) B: Uh,.. well,... I guess so.

In (8.1), speaker A asks a question to B and assigns the turn to 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 what to say, indicates that he accepts the turn. So the segment *"Uh,.. well,..."* is multifunctional, having both a stalling function and a turn-accepting function. Note that A's utterance is also multifunctional: it asks a question about B's opinion and it assigns the turn to B (due to its intonation, in combination with A looking at B and raising his eyebrows).

The design of a dialogue act annotation schema can reflect the multifunctional view of utterances in two ways: 1) by structuring the tag set into clusters (see below); 2) by accompanying instructions to annotators for how to apply multiple tags. 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 apply multiple tags, 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 a conceptual similarity of certain 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 and 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".

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; 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 other than those for advancing the task. Dialogue participants share information not only about the task that is pursued 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. This part of ISO 24617 uses the term "dimension" to refer to these various semantic content categories or to the communicative activities concerned with these content categories. This leads to dimensions such as feedback, turn management, time management and contact management in addition to the dimension formed by the task that motivates the dialogue. Clause 9 describes the set of dimensions defined in this part of ISO 24617.

7.4 Dimension- specific and general-purpose functions

Not every grouping of communicative functions qualifies as a dimension in the sense of this part of ISO 24617. For example, the group of information-giving acts (statements, warnings, answers, confirmations and so on) does not form a dimension, since information can be given about any aspect of the dialogue, such as the underlying task, feedback, change of topic or contact. Information-giving acts are thus not specifically related to a particular category of information. The same is true of information-seeking acts (open questions, check questions, menu questions and so on) and of the commissive and directive acts (request, suggest, instruct, offer, promise and so on), which can be about any kind of action. These clusters of functions therefore do not qualify as dimensions; since these functions can be combined with any kind of information or

action, they are called *general-purpose communicative functions*. When combined with a semantic content of a certain category, they form a dialogue act addressing the dimension corresponding to that kind of content. These functions are discussed further in Clause 10.1; Table 1 provides examples of general-purpose functions used in some of the dimensions defined in the LIRICS annotation scheme.

There are also communicative functions which, in contrast with the general-purpose functions, can be used only 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 10.2.

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

Communicative function	Dimension	Example expressions
Propositional question	Task	*"Is there an earlier possibility?"*
Set question	Task	*"What time does the meeting start?"*
Check question	Auto-Feedback	*"On Thursday, you said?"*
Disconfirm	Auto-Feedback	*"On Tuesday"*
Inform	Social Obligations Management	*"I'm very grateful for your help"*
Confirm	Auto-Feedback	*"Slightly yeah, very slightly"*
Offer	Discourse Structuring	*"Shall I repeat the connection?"*
Decline offer	Discourse Structuring	*"No thank you"*
Instruct	Time Management	*"We're going to turn east"*
Request	Turn Management	*"Peter, would you please go on?"*
Accept request	Turn Management	*"Yes, I'd like to say something at this point"*

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

Dimension	Communicative function	Example expressions
Auto-feedback	Auto-Positive	*"Okay"; "Uh-huh"*
	Auto-Negative	*"Huh?"; "I beg your pardon?"*
Turn management	Turn Keep	utterance-final pitch rise
	Turn Grab	hold gesture, with raised hand
	Turn Assign	*"Peter?"*, looking at Peter, raising eyebrows
Time management	Stalling	slowing down speech; fillers
	Pausing	*"Just a minute", "Hold on"*
Contact management	Contact Check	*"Hello?"*
Discourse structuring	Topic Introduction	*"Concerning the windows,.."*
	Interaction Structuring	*"I repeat:.."*
Social obligations management	Apology	*"Sorry"*
	Thanking	*"Thank you"*
	Greeting	*"Hello!", "Good morning"*

8 Dialogue segmentation

The multifunctionality of dialogue behaviour, discussed in 7.2, is optimally accounted for when communicative functions are assigned to all those segments of behaviour that correspond to a dialogue act. These segments are called *functional segments*, defined more precisely as a *minimal stretch of communicative behaviour that has a communicative function*, not excluding the possibility of having more than one function (see also definition 3.12). The condition of being "minimal" ensures that functional segments do not include material that does not contribute to the expression of the communicative function(s) that identify the segment. A consequence of this definition is that functional segments may be discontinuous, may overlap or be embedded and may contain parts contributed by different speakers.

Consider the segmentation of the turn unit contributed by S in (9.2):

EXAMPLE 9

(9.1) U: What time is the first train to the airport on Sunday morning please?

(9.2) S: The first train to the airport on Sunday morning is .... let me see... at 5:45.

This turn unit contains three functional segments: 1) the discontinuous segment *"The first train to the airport on Sunday morning is at 5:45*", which expresses an answer in the Task dimension; 2) the embedded segment "*The first train to the airport on Sunday morning",* which provides positive feedback by displaying S's understanding of what U said; 3) the segment *"let me see",* which has the function of stalling for time. The identification of these functional segments can be viewed as segmenting the turn unit in each dimension in which parts of it have a communicative function:

EXAMPLE 10

Dimension	*Segmentation*
Task	**The first train to the airport on Sunday morning is** [... let me see...] **at 5:45**
Auto-feedback	**The first train to the airport on Sunday morning** / is ... let me see... at 5:45
Time management	The first train to the airport on Sunday morning is / **.. .let me see…** / at 5:45

In the Task dimension, the turn unit is segmented into the discontinuous functional segment *"The first train to the airport on Sunday morning is at 5:45"* and the intervening stretch *"... let me see...",* which does not have a communicative function in this dimension. In the Auto-Feedback dimension the turn unit is segmented into the functional segment *"The first train to the airport on Sunday morning"* and the contiguous stretch *"is ... let me see... at 5:45",* which is not a functional segment. In the Time-Management dimension the turn unit is segmented into the stretch *"The first train to the airport on Sunday morning is",* which does not have a communicative function; the functional segment *"... let me see..."* and the stretch *"at 5:45",* which does not have a communicative function in this dimension. The segmentation in the Task dimension illustrates the possible discontinuity of a functional segment; comparing this segmentation with the one in the Auto-Feedback dimension shows that two functional segments may overlap (in particular, one may be embedded in another).

Example 11 illustrates the possibility of a dialogue act to spread over multiple turns. A asks a question, the answer to which consists of a list of items which B communicates one by one:

EXAMPLE 11

A: Could you tell me what departure times there are for flights to Frankfurt on Saturday?

B: Certainly. There's a flight in the morning leaving at 08:15,

A: yes,

B: and a KLM at 08:50,

A: yes,

B: and a flight at 10:30,

A: yes,

B: ...

Segments of verbal behaviour have a natural delineation in terms of their constituent words. For nonverbal communicative behaviour this is less obvious; still, the various forms of nonverbal behaviour (hand gestures, head gestures, facial expressions, etc.) do have their own morphology (see e.g. Kendon, 2004), which can be used to identify their beginning and end. The definition of a functional segment as a "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 the use of 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. See Annex B for examples.

9 Dimensions

As noted in 7.4, not every grouping of related communicative functions makes a dimension. In order to identify dimensions for multidimensional dialogue act annotation, Petukhova and Bunt (2009a,b) formulate and test the following five criteria.

First, only dimensions should be considered which are observed in communicative behaviour. This places the notion of a dimension on an empirical basis.

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

Third, each dimension should be recognizable with acceptable precision by human analysts as well as by automatic dialogue understanding and dialogue annotation systems in order to be useful.

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 stipulates that each dimension in a multidimensional system can be addressed by dialogue acts independent of addressing other dimensions. More precisely, for every dimension, *D*, there should be forms of communicative behaviour which express a dialogue act that is concerned with information of the kind characteristic for *D*, without necessarily 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 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 practical consideration, making explicit that an annotation standard should capitalize on existing good practices.

In sum, the following criteria and considerations help make a well-motivated choice of the dimensions in a multidimensional dialogue act annotation schema.

Each dimension in a dialogue act annotation schema shall be

a) empirically observed in the functions of dialogue utterances,

b) theoretically justified, forming a well-established and well-studied aspect of communication,

c) addressable independently of the other dimensions,

d) recognizable with acceptable precision by human annotators and by automatic annotation systems, and

e) present in existing dialogue act annotation schemes.

In their study, Petukhova and Bunt (2009a) survey the literature and analyse the content of 18 existing annotation schemes in order to verify the requirements b) and e) 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 kinds. 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 results of these tests and surveys are summarized in Annex G.

This study confirms that the following nine dimensions fulfil all five of the above-listed requirements a) to e) and qualify as dimensions in a standard dialogue act annotation schema.

9.1 *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, participating in a game and so on. The Task dimension is formed by those dialogue acts that are intended to advance the underlying task or activity.

9.2 *Auto-Feedback*

The term “feedback” in dialogue is most often used to refer to the activity of 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 larger 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 and Fiske (1977); Allwood et al. (1993); Clark and 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 (2000a) distinguish several feedback levels: *attention; perception; understanding; evaluation*; *execution*. The term “auto-feedback” is used here in order to make a distinction with “allo-feedback” (see next subclause).[3)]

9.3 *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 is called *allo-feedback*; examples are: *“Is this clear enough?”, “That's what I meant”* and *“You got me wrong”.*

9.4 *Turn Management*

Turn Management acts are concerned with the allocation of the speaker role, also called the “floor” (Sacks et al., 1974). Allwood (1997) defines turn management as the distribution of the right to occupy the speaker role in dialogue. He argues that this is rather a normative notion than a behavioural unit.[4)] Accordingly, the

3) The terms “allo-feedback” and “auto-feedback” (Bunt, 1995) have their origin in the Greek words *allos* and *autos*, meaning “other” and “self” and referring to the participant whose processing the speaker is considering.

4) The corresponding behavioural unit is what in this part of ISO 24617 is called a “turn unit”; see definition 3.22.

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 (e.g. *"uh-huh"*), nonverbal sounds and other nonverbal activity. (Backchannels and nonverbal dialogue acts are contributions made by a participant without occupying the speaker role.) In multi-party dialogue one may find multiple simultaneous speakers (Campbell, 2008) and the conversation may effectively split up into sub-conversations involving subgroups of participants.

9.5 *Time Management*

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

9.6 *Discourse Structuring*

A dialogue participant may perform a dialogue act in order to indicate the intention to close the discussion of a certain topic or to focus on a new one. Such dialogue acts are based on the speaker's view of the state of the underlying task or on the development of a plan that he may have for organizing the dialogue and on assumptions that arise concerning the need to structure the interaction in order for the dialogue to proceed successfully.

9.7 *Social Obligations Management*

Participating in a dialogue is a social activity, where one is supposed to act in accordance with norms and conventions of 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).

Bunt (2000b) 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 mark the end of a conversation.

9.8 *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 occurrence of speech-editing acts dialogue acts such as (self-)repairs and restarts.

9.9 *Partner Communication Management*

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

10 Core dialogue acts

The various annotation schemes for dialogue acts that have been proposed share a number of communicative functions which are of obvious importance in virtually any type of dialogue. Traum and Hinkelman (1992) use the term “core dialogue acts” to refer to those acts that are 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 (*promise, offer, request, propose,...),* the “reportative” speech acts used for stating facts (*assert, conclude*) and the “expressive” acts for expressing psychological states (*apologise, thank, congratulate*). In this part of ISO 24617, 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 communicative functions are included in this part of ISO 24617. 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 act types 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 criteria similar to those for the choice of dimensions. The criterion of *empirical validity* requires that for every communicative function there are linguistic or nonverbal means which a speaker can use to indicate that his behaviour has that function. 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 part of ISO 24617 requires precise definitions in terms of intended information state updates.

Another empirical requirement is that of *coverage*. For example, the phenomenon that conversational analysts have called “adjacency pairs” means that if an annotation schema includes one element of such a pair, then it should also contain the other. A thanking act is, for instance, often responded to by a “downplayer”, and an annotation schema which contains a tag for encoding thankings should also contain one for downplayers.

In order to be appropriate as elements in an annotation standard, each communicative function should be recognizable with acceptable precision by humans and by machines, and must commonly occur in existing annotation schemes.

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 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*, can choose from the set of functions available for *D* the most specific one for which there is sufficient evidence. For example, in Example 12, B's utterance forms an information-providing act in response to A's check question:

EXAMPLE 12

A: And that's the first flight tomorrow, right?

B: That's right.

This means (see Figure 2) that the choice is between the functions *Inform, Agreement, Disagreement, Correction, Answer, Confirm* and *Disconfirm*. Of these, the functions *Disagreement, Correction* and *Disconfirm* do not apply here 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*.

A multidimensional annotation scheme with orthogonal dimensions and semantically connected sets of communicative functions allows annotators to follow the strategy of always marking up segments with the most specific communicative function for which there is sufficient evidence, so that a functional segment has at most as many functions as there are dimensions.

All in all, the communicative functions included in this part of ISO 24617 satisfy the following six requirements and desiderata:

Every communicative function shall be

— empirically observed in features of communicative behaviour in dialogue,

— theoretically validated as an update operation on information states (i.e. has a clear semantics),

— relevant for obtaining a good coverage of the phenomena in the dimensions considered,

— recognizable by humans and machines,

— a member of a semantically connected set of functions, and

— present in existing annotation schemes.

The definition of communicative functions in this part of ISO 24617 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 all the core dialogue act functions are or will 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 registration procedures. This part of ISO 24617 includes only small numbers of domain-independent core communicative functions for the various dimensions:

a) **General-purpose functions**:

— 5 information-seeking functions;

— 7 information-providing functions;

— 8 commissive functions;

— 6 directive functions.

b) **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 concerned with 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 the analysis of existing annotation schemes summarized in Annex G (see Tables G.3 to G.15).

The functions in the information-seeking class are questions of various kinds. Many schemes 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 many languages in the distinction of different sentence types. In this part of ISO 24617 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 list of 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 of bringing 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 with the information that is provided, 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 being 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*, where the speaker orders 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 requires his consent to do 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 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 can be made and more specific types of each of the functions mentioned here developed: for instance, some taxonomies distinguish different answer types, such as WH-answer and YN-answer; these and other more specific functions may be added as refinements of the taxonomy defined in this part of ISO 24617, depicted in Figure 2. The mother–daughter relation in this taxonomy reflects increasing specialization going from mother to daughter; 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 their defining characteristic that they can be used to build a dialogue act in any dimension by combining the function with a semantic content of the category of the dimension (see

also Clause 11), The definitions of the core general-purpose functions are provided in Annex E in the form of data categories.

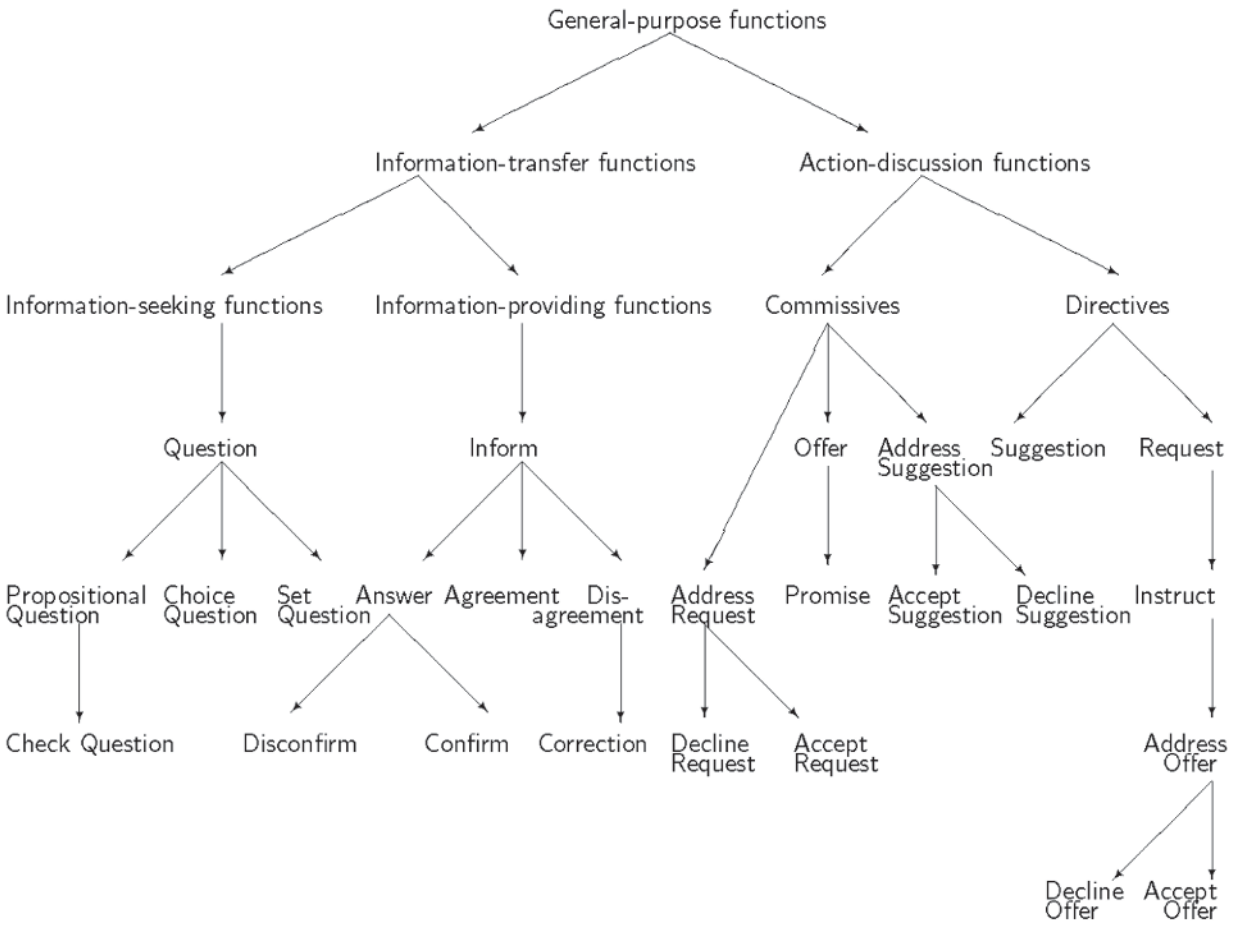

Figure 2 — General-purpose functions

10.2 Dimension-specific functions

Dimension-specific functions, which can be used in only one of the dimensions, mostly have no or only marginal 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, such as greetings and goodbyes, likewise do not require a semantic content; others, such as expressions of thanks or apologies, may have a semantic content if the speaker wants to indicate what he is thankful for or apologises for.

The following subclauses describe the core communicative functions identified for each of the nine core dimensions. Their precise definitions are specified in Annex E in the form of data categories.

10.2.1 Task dimension

Dimension-specific communicative functions for the Task dimension are specific for communication about a particular task domain. For example, 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 its domain-independence, this part of ISO 24617 does not include any such 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 cupping a hand behind an ear (*"I didn't hear you"*) or by blinking.

Feedback-providing acts fall apart into positive and negative ones. In the Auto-Feedback dimension, positive acts signal that the sender successfully processed a previous utterance; negative ones that a problem was encountered. In the Allo-Feedback dimension, positive acts signal that the sender believes that the addressee processed a previous utterance successfully, negative ones that the sender believes that the addressee was unsuccessful. Feedback elicitation acts express that the speaker wants to know whether the addressee was successful in processing a previous utterance.

Some annotation schemes distinguish various levels of processing to which feedback acts may refer; see the DIT^{++} scheme, Bunt (2009), which distinguishes the levels of *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 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. For human annotators as well as for automatic annotation systems, it is often impossible to reliably indicate a specific level of processing for feedback messages, therefore this part of ISO 24617 does not include feedback functions for specific levels of processing, but only the more general functions expressing positive and negative feedback.

10.2.3 Turn management

The turn-management functions in this part of ISO 24617 are defined as the activities that a dialogue participant undertakes for obtaining, maintaining or giving up the speaker role. Turn-management functions can be divided into *turn-initial* ones, which only occur at the beginning of a turn and which are concerned with obtaining the speaker role and *turn-final* ones, which occur only within or at the end of a turn and which are concerned with keeping the speaker role or making it available. A functional segment may thus have both a turn-initial and a turn-final turn-management function.

10.2.4 Time management

Stalling for time is a widespread phenomenon in spoken interaction and may occur for a variety of reasons. It is typically indicated by slowing down and using fillers like *"uh", "let me see", "you know", "well".* Fillers and slowing down can be used when the speaker needs just a few seconds (rather than several minutes or even more). The communicative function characterizing this behaviour is called "*stalling"*. A speaker who needs more time then just a few seconds to, for instance, look up something or because he is interrupted by something urgent should do something else. This is where expressions like *"just a minute", "hold on", "momentito", "un instant", "veuillez patienter"* are used, which signal that the speaker is briefly suspending his contribution to the dialogue but intends to resume shortly. This is called "*pausing"*.

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. When the discourse structure is addressed explicitly by dialogue acts, this is done most often using a general-purpose function, as in *"Peter, will you introduce the next item?"*

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.

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 fulfil 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 "downplayer" (*"De nada"; "Pas de quoi"*).

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 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:

EXAMPLE 13

(13.1) A: Can I offer you some coffee?

(13.2) B: Only if you have it ready.

Suggestions and requests can be accepted conditionally and with certain modalities. Information-providing acts may also express the speaker's awareness that he possesses uncertain information:

EXAMPLE 14

(14.1) A: Do you know who'll be coming tonight?

(14.1) B: I have a hunch that Mary won't come.

(14.1) B: Peter, Alice and Bert will probably come.

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:

EXAMPLE 15

(15a) A: Can you tell me what time is the first flight tomorrow?

B: The first flight tomorrow morning is at seven-thirty.

A: Perfect!

(15b) A: What about a fresh cup of coffee?

B: Ah, you're wonderful!

In the first example, A's positive feedback expresses that B is very satisfied with the information obtained; in the second, B's acceptance of A's offer additionally expresses B's positive feelings toward A.

In order to be able to represent such phenomena, this part of ISO 24617 includes certain *qualifiers* that may be associated with a communicative function. A corpus-based study of these phenomena (Petukhova and Bunt, 2010) indicates that uncertainty and conditionality can be captured by means of binary distinctions (certain/uncertain, conditional/unconditional); therefore, two binary-valued attributes, "certainty" and "conditionality", are defined. The certainty values "uncertain" and "certain" can be associated with information-providing functions in order to represent the speaker's expression of certainty about the correctness of the

information that he provides. The default value of this attribute is “certain”. The conditionality values “conditional” and “unconditional” can be used with action-discussion functions, which have in common that the participant whose action is under discussion is able and willing to perform that action. 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”.

For representing a speaker's sentiment, a wide variety of descriptors has been proposed in the literature, ranging from six basic emotions (Ekman, 1972) to several hundred possible values. This part of ISO 24617 includes two binary attributes for representing conditional and uncertain variants of dialogue acts and one attribute (“sentiment”) with an open class of values which may be associated with any communicative function. The possible values of this attribute can be chosen as appropriate for a given domain, task or interactive setting.

11 Dialogue act markup language (DiAML)

DiAML has been designed in accordance with the linguistic annotation framework (LAF), as specified in ISO 24612, which 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; “representation” refers to the format in which an annotation is rendered, independent of its content. According to LAF, *annotations* are the proper level of standardization, rather than *representations*. This distinction is implemented in the DiAML definition by a syntax specification that defines, in addition to a class of XML-based *representation structures*, a class of more abstract *annotation structures*. These components are called *concrete* and *abstract syntax*, respectively. Annotation structures are set-theoretical structures, consisting of concepts of the types that populate the metamodel shown in Figure 1. The concrete syntax defines a rendering of annotation structures in XML. Section 11.1 gives a very brief outline of the abstract syntax; more details are given in Annex C.

NOTE For the formal semantics of the abstract syntax and more about the design of alternative representation formats, see Bunt (2010; 2011b) and Ide & Bunt (2010).

11.1 Abstract syntax

The abstract syntax of DiAML consists of

a) a specification of the elements from which annotation structures are built up, called a “conceptual inventory”, and

b) a specification of the possible ways of combining these elements to form annotation structures.

The conceptual inventory of DiAML consists of sets of dialogue participants, dimensions, communicative functions, functional segments, qualifiers and rhetorical relations.

An annotation structure is a set of *entity structures* and *link structures.* Entity structures contain semantic information about a functional segment; link structures describe semantic relations between functional segments.

An entity structure in DiAML contains a characterization of a dialogue act, in a so-called “dialogue act structure” (see below) and a specification of which functional segment it is anchored to and how it relates to other acts in the dialogue. Formally, an entity structure is a quadruple <*s, α, E, δ*> consisting of a functional segment, *s*, a “dialogue act structure”, α, a set *E* of entity structures containing dialogue acts upon which α depends, and a specification, δ, of the type of dependence (functional or feedback). If *E* is empty, *E* and δ may, for simplicity, be omitted.

A *dialogue act structure* contains the information that characterizes a single dialogue act. This includes a specification of the sender, the addressee(s) and the communicative function. For dialogue acts with a general-purpose communicative function, the dimension of the semantic content is another important component; for dialogue acts with a dimension-specific function, the dimension does not need to be specified, since it is inherent in the definition of the function. General-purpose functions may additionally have one or

more qualifiers. A dialogue act structure is therefore either a triple, consisting of a sender, *S*, a (set of) addressee(s), *A*, and a dimension-specific function, f_d, or a quintuple having a general-purpose function, *g*, instead of a dimension-specific one and containing additionally a dimension, *d*, and a list, *q*, of zero or more function qualifiers (if *q* is empty, it may for simplicity be omitted). In accordance with the metamodel shown in Figure 1, a dialogue act may also have "other participants" in addition to a sender and addressees; this is reflected in the abstract syntax by allowing an additional (set of) other participant(s), *H.* Formally, a dialogue act structure is either a triple <*S, A,* f_d> or a quintuple <*S, A, d, g, q*> or a quadruple <*S, A, H,* f_d> or a sixtuple <*S, A, H, d, g, q*>.

A link structure is a triple <ε, *E,* ρ> consisting of an entity structure, ε, a set, *E*, of one or more entity structures and a rhetorical relation, ρ, which relates the dialogue act in ε to those in *E*.

11.2 Concrete syntax

The concrete syntax is defined in accordance with the methodology for defining semantic annotation languages described in Bunt (2010; 2012). This methodology includes the notion of an *ideal representation format*, defined as one which is 1) "complete", in the sense that every annotation structure defined by the abstract syntax can be represented, and 2) "unambiguous", in the sense that every representation defined by the concrete syntax represents one and only one annotation structure defined by the abstract syntax. Since the semantics of DiAML are defined for the structures defined by the *abstract* syntax, any two representation formats which are "ideal" in this sense are semantically equivalent and every representation in one such format can be converted by a meaning-preserving mapping into any other such format.

The DiAML concrete syntax specifies names of XML tags, attributes and values corresponding to the various ingredients in the conceptual inventory and defines the possible ways of combining these elements in XML representation structures. In particular, XML elements are defined for entity structures and link structures. The dimensions, communicative functions and function qualifiers that can be used in DiAML are defined as data categories, in accordance with ISO 12620, in Annex E.

Entity structures are represented by an XML element called dialogueAct, which has the following attributes:

— xml:id, whose value is a unique identifier of a dialogue act structure;

— target, whose value refers to a functional segment;

— sender, addressee and otherParticipant, whose values refer to dialogue participants, identified in the metadata of the annotated primary data, with the attribute otherParticipant being optional;

— dimension, whose value names one of the nine dimensions defined in this part of ISO 24617;

— communicativeFunction, whose value names one of the communicative functions defined in this part of ISO 24617;

— certainty, conditionality and sentiment, whose values is one of the communicative function qualifiers defined in this part of ISO 24617, the attributes being optional.

— functionalDependence and feedbackDependence, whose values refer to one or more dialogue acts that the given dialogue act has a dependence relation with, and both of which are optional.

Link structures are represented by an XML element called rhetoricalLink, which has the following attributes:

— dact, whose value refers to a given dialogue act that is rhetorically related to other dialogue acts in the annotated dialogue;

— rhetoRelatum, whose value refers to one or more dialogue acts that the given dialogue act is rhetorically related to;

— rhetoRel, whose value names a rhetorical relation.

The formal specification of the DiAML concrete syntax using XML is provided in Annex D.

Example 16 shows the representation of the annotation of the dialogue fragment given in Example 17.

EXAMPLE 16

(16.1) P1: What time does the next train to Utrecht leave?

(16.2) P2: The next train to Utrecht leaves I think at 8:32.

Annotations may be attached to primary dialogue data in a variety of ways: directly to stretches of speech, defined by temporal begin and end points, or to structures at lower levels of description such as the output of a tokenizer. Here it is assumed that the relevant functional segments are identified at another level of XML representation, according to ISO 24612. P2's utterance is segmented into two overlapping functional segments: fs2.1 in the Auto-Feedback dimension (reflecting the interpretation that the repetition of a large part of the question signals positive feedback on understanding that question) and fs2.2 in the Task dimension. Following the guidelines of the text encoding initiative (TEI P5, 2010), the prefix "#" is used to indicate that the prefixed value is identified either in the metadata of the primary data or in another layer of annotation or elsewhere within the same representation. With these assumptions, the DiAML representation of Example 16 is as shown following the example below.

EXAMPLE 17

1. P1: What time does the next train to Utrecht leave?

   Task fs1: What time does the next train to Utrecht leave?

2. P2: The next train to Utrecht leaves I think at 8:32.

   AutoFB fs2.1: The next train to Utrecht leaves

   Task 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"/>
<dialogueAct xml:id="da2" target="#fs2.1" sender="#p2" addressee="#p1"
   communicativeFunction="autoPositive" feedbackDependence="#da1"/>
<dialogueAct xml:id="da3" target="#fs2.2" sender="#p2" addressee="#p1"
   communicativeFunction="answer" certainty="uncertain" dimension="task"
   functionalDependence="#da1"/>
</diaml>
```

12 Principles for extending and restricting the standard

The limited number of dimensions, communicative functions, qualifiers and relations among dialogue units defined in this part of ISO 24617 cannot be expected to be adequate for every kind of dialogue analysis, for every task domain, for every kind of dialogue or for every annotation purpose. The general design principles underlying this part of ISO 24617 should, however, be useful also for defining extensions, modifications or restrictions as needed for particular applications. This clause summarizes the main design principles which should be respected in extensions and restrictions and formulates corresponding guidelines.

12.1 Main design principles

The main design principles underlying this part of ISO 24617 are the following.

a) Dialogue act annotation requires a multidimensional approach, since interactive behaviour in dialogue is multifunctional, i.e. each stretch of communicatively meaningful behaviour may have more than one communicative function. The annotation scheme defined in this part of ISO 24617 is designed to support the assignment of multiple communicative functions to units in dialogue.

b) Dimensions can be defined as distinct types of communicative activity, such as giving feedback, turn management, pursuing the underlying task or activity and taking care of social obligations. Each of these types of activity is concerned with a particular category of information (processing of utterances, allocation of participant roles, task/activity and social obligations, respectively). Dimensions can therefore also be defined as categories of semantic content.

c) Communicative functions are most accurately assigned to *functional segments*, defined as minimal stretches of communicative behaviour that have a communicative function (possibly more than one). Functional segments may be discontinuous, overlap, spread over multiple turns and include parts contributed by different participants. Segmenting a dialogue into functional segments is most accurately achieved in a "multidimensional" way, identifying for each dimension the stretches of communicative behaviour that have a communicative function in that dimension.

d) Communicative functions are defined semantically in terms of how they can be combined with a semantic content to define an update operation changing the information states of dialogue participants.

e) All dimensions are

    1) empirically observed,

    2) theoretically justified,

    3) recognizable by human annotators and by automatic annotation systems,

    4) addressable independently from other dimensions ("orthogonal"), and

    5) present in existing annotation schemes.

f) The set of communicative functions is divided into sets of *dimension-specific* functions, one for each 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. A dimension-specific communicative function can only be combined with semantic content of the category that is characteristic for that dimension.

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 by humans and machines, and

    5) present in existing annotation schemes.

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. For each dimension, the set of dimension-specific communicative functions for that dimension is also semantically connected.

i) The semantic connectedness of the communicative functions that can be used in any given dimension has the advantage that a functional segment never needs to be annotated with more than one function per dimension, assuming that for each dimension in which the segment has a communicative function, the most specific function is chosen for which there is sufficient evidence. Given the orthogonality of the dimensions, this has the consequence that a functional segment is annotated with maximally as many functions as there are dimensions.

12.2 Schema extension

The design specified by this part of ISO 24617 is easily extensible in the following ways.

a) **Addition of dimensions:** Dimensions can freely be added as long as the requirements of 12.1 e) are met. For specific purposes or domains, new dimensions may be added for which not all of these requirements are met, because some of them are objects of investigation. For example, an additional dimension does not need to be theoretically justified *a priori*, since the purpose may be to investigate dialogue phenomena which have not yet been studied extensively. A property that is particularly important for an additional dimension is that of being orthogonal to the dimensions already present, in order to avoid redundancy and ambiguity in annotations. For example, Contact Management, which is on of the dimensions of the DIT^{++} annotation scheme, was noted by Petukhova and Bunt (2009a; b) as a possible additional dimension, being orthogonal to the nine dimensions presented in this part of ISO 24617, theoretically justified, empirically observed and recognizable with acceptable precision by human annotators and automatic annotation programs.

b) **Addition of communicative functions:** Communicative functions may be added to the core functions defined in this part of ISO 24617, provided that they satisfy the requirements 12.1 g) and h), where, as in the case of adding a dimension, some of these requirements may be waived because they form an object of study. The DIT^{++} and LIRICS taxonomies contain several examples of communicative functions that satisfy these requirements and that could be added to the core functions. This part of ISO 24617 is intended to be domain-independent and therefore does not define any dimension-specific communicative functions for the Task dimension. Such functions may freely be added, provided they satisfy the requirements of 12.1, g), 1) to 4), and (h).

c) **Addition of communicative function qualifiers:** For the sentiment qualifier attribute, values may freely be introduced. Additional qualifier attributes and values may be introduced provided that they 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 ensure that a uniquely determined most specific value can always be chosen for the attribute.

d) **Specification of rhetorical relations:** Rhetorical relations may freely be added to the design specified in this part of ISO 24617, but, to avoid ambiguity and redundancy, the set of specified rhetorical relations should preferably be semantically “connected”, in the sense that any two relations should be either mutually exclusive or one a special case of the other.

12.3 Scheme restriction

Subschemes of this annotation standard scheme 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 omitted. By virtue of the orthogonality of the set of dimensions, whether or not a particular dimension is included has no influence on the remaining dimensions.

— Communicative functions for which there is a less specific function present in the annotation scheme may freely be omitted, since in that case the remaining set of communicative functions is still semantically connected.

— It is not recommended that a communicative function be omitted for which the scheme contains more specific functions while maintaining the more specific functions, since this limits the possibilities for an annotator to use a less specific functional tag in the case of lack of evidence for a more specific one.

— Communicative functions may be omitted which are considered irrelevant for a particular purpose, if this does not have undesirable limiting effects on the desired coverage of dialogue phenomena.

— Communicative function qualifiers may freely be omitted. For qualifier attributes for which a default value is defined (such as certainty and conditionality), omitting a value is semantically equivalent to using the

default value; for qualifier attributes for which no default value is defined (such as sentiment), omitting a value is equivalent to leaving that aspect underspecified.

Annex A
(informative)

Annotation guidelines

This annex first considers some general issues in dialogue act annotation (see A.1). The segmentation of a dialogue into functional segments is discussed in A.2. In A.3 guidelines are provided for the use of DiAML and the annotation schema defined in this part of ISO 24617. The examples in this annex focus on specific issues in dialogue act annotation; examples of fully annotated dialogue fragments can be found in Annex B.

A.1 General issues in dialogue act 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. This part of ISO 24617 is intended to apply to dialogues in a 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 such as highlighting, blinking and beeping in the case of computer systems.

The prototypical setting of human dialogue is that of face-to-face communication, where speech is combined with other vocal sounds (laughs, sighs, heavy breathing, etc.), facial expressions, gaze direction and physical activities including head-, hand-, arm- and shoulder gestures, forms of touching (stroking, caressing, hugging, shaking hands, patting on the shoulder, …) 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.

While this part of ISO 24617 has an emphasis on its use for creating interoperable language resources, it has also been successfully applied to a range of nonverbal and multimodal behaviours. (See, e.g. Petukhova and Bunt, 2009d, 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 spoken interaction, with or without visual contact or involving several people in a setting where they can see and hear each other. In either type of situation there is much of the time one participant who occupies the speaker role, i.e. who “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. These participants are the addressees of the dialogue acts performed by the speaker.

In certain formal settings 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 cabinet member or group of representatives.

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 in the studio or for the viewers at home.

In a conversational setting with multiple participants, it is also quite common that the speaker addresses one of the participants more than the others. In such a case, it is best to use the attribute @addressee to

designate the participant that is addressed primarily and to use the attribute @otherParticipant to designate the other participants.

A.1.3 Annotation purposes and information situations

This part of ISO 24617 is intended both for use by human annotators and for use by automatic annotation systems. It has been tested for usefulness 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 available sources of information. For a multimodal dialogue, where speech is used in combination with nonverbal behaviour, this means that not only the recorded speech should be available, but also a video recording of the nonverbal behaviour or at least 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 the other of the following reasons:

a) because it has linguistic or nonverbal properties which, in the context in which the segment occurs, are indicators of that function;

b) because the function is an implication of another function which the segment has, typically for reason a).

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*.

EXAMPLE A: Would you like to have some coffee?

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 function 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_2, is *logically entailed* by the communicative function, F_1, if F_1 is a special case of F_2. This happens in hierarchies of communicative functions such as the general-purpose functions of this part of ISO 24617, where, for instance, *Confirm* is a special case of *Answer* and *Correction* is a special case of *Disagreement*, which in turn is a special case of *Inform*. Another type of entailment exists between a dialogue act that responds to a previous dialogue act and feedback about the processing of that previous utterance. For example, an *Accept Offer* entails positive auto-feedback about the processing of the utterance which expressed the *Offer*.

b) A communicative function, F_1, may have another function, F_2, as a *conversational implicature*, i.e. in most situations where a functional segment has function F_1, it also has function F_2, assuming that the dialogue participants behave cooperatively. For example, a thanking act such as *"Thank you"* will normally be understood as also being 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 an explicit function. It is therefore recommended implicated functions be annotated. 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 need not be annotated; in the second case, it should.

NOTE For more details about types of implicit functions and strategies for how to deal with them, see Bunt (2011a).

Standard speech act theory mostly regards indirect speech acts as just another *form* of the same communicative act as the direct form. By contrast, this part of ISO 24617 incorporates the view that indirect forms signal packages of beliefs and intentions subtly different than direct ones and thus expressing a slightly different communicative act. 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 question 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 can be interpreted as the conditional request *"Please tell me what time it is, if you know/can".*

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 such a case the annotator may annotate the utterance as having a qualified *Request* function, with the attribute "conditionality" having the value "conditional". This is represented in DiAML as follows:

```
<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 (or to a "functional segment", to be precise), put yourself in the position of a participant to whom the utterance was addressed and imagine that you try to understand what the speaker wants 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 an actual addressee and like a real addressee, try to understand the speaker's communicative behaviour. (As mentioned in A.1.3, depending on the purpose of the annotation, it may also be an option for you to look ahead in the dialogue.)

b) ***Think functionally, not formally.*** The linguistic form of an utterance often provides vital clues for choosing an annotation tag, but such clues can be misleading; in choosing your tags you should of course use the available clues to your advantage, but 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* in *"Which books did you read on your vacation?"* or *where* in *"Where do you live?"* In other languages this is different. Moreover, 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, like *"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; *"Would you like some coffee?"* is most likely an offer, rather than a question and *"Shall we go?"* a suggestion.

c) ***Be specific*** The communicative functions that you can choose from differ in specificity, corresponding to their relative positions in hierarchical subsystems of the taxonomy. 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 assumption 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 and you don't really have evidence for choosing the more specific one, then use the less specific one.

A.2 Segmentation

According to this part of ISO 24617, dialogue acts correspond to *functional segments* as defined in Clause 8. In this definition, a functional segment is characterized 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, not to unnecessarily large stretches. Consider Example 1 (from a Map Task dialogue).

EXAMPLE 1 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 (explaining the term "moon lander"). In order to do this 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 assign the *Instruct* function to segment fs1 and the *Inform* function to fs2, rather than assigning both of them to the entire utterance. Fine-grained segmentation also allows us in this example to indicate that the *Inform* in fs2 is an explanation of something in the *Instruct* in fs1, as represented in the following:

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

There are cases where the identification of a stretch of behaviour that corresponds to a functional segment is not obvious, in particular when a longer stretch which could be said to express a certain type of dialogue act has a part which expresses that same type of dialogue act. Example 2 illustrates this (from a Map Task dialogue).

EXAMPLE 2 E: and then you go up and around that, a little to the right.

A: slightly northeast?

E: yeah, slightly northeast.

E's second utterance as a whole could be said to constitute a *Confirm* act, with the semantic content expressed by A in his utterance, and each of the parts "*yeah"* and *"slightly northeast"* could also be said to constitute such acts. To avoid the introduction of unnecessarily many annotation tags, in such cases a "maximal" approach may be advisable, choosing the longer stretch as a functional segment.

When working with a given pre-segmented transcription of a spoken dialogue, the segmentation given in the transcript is not necessarily perfect or not quite as one would like it to be. First, there may be cases where one would prefer a given segment to be split 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. 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 together. In such a case it is recommended that all these parts be annotated with the same tags.

Third, a given segmentation may disallow discontinuous segments, which causes a problem when a speaker interrupts a contribution which has a certain communicative function by a part that has a different communicative function, as in Example 3.

EXAMPLE 3 Can you tell me what time the train to *uh*,... Viareggio leaves?

Here we see a *Set Question* interrupted by a segment that does not contribute to the question and has a *Stalling* function. The preferred segmentation would distinguish in this case one functional segment in the Task dimension, namely fs1 = “Can you tell me what time the train to Viareggio leaves?” and one in the Time-Management dimension, namely fs2 = *“uh,...”,* leading to the following representation in DiAML:

```
<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*.

EXAMPLE 4 A: There are two flights early in the morning, at 7.45 and at 8.15.

B: Yes.

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 A's utterances 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 (*Answer*, 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 part of an answer to the same question.

A.3 Representing annotations in DiAML

According to the abstract syntax of DiAML specified in Clause 11, a DiAML annotation structure is formally a set of entity structures and link structures. An entity structure contains information about a dialogue act expressed by a given functional segment; a link structure contains information about rhetorical relations between two or more dialogue acts.

In order to be compliant with LAF in accordance with ISO 24612, the XML-based 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, and

c) the dialogue act annotation associated with a functional segment.

Functional segments can be identified by means of a 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 D.

According to the metamodel in Figure 1, a dialogue act has a sender, at least one addressee, possibly other participants, a semantic content category, a communicative function (which may have one or more qualifiers), and possibly functional and feedback dependence relations and rhetorical 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 optional attributes whose values represent qualifiers, functional relations and feedback relations, while rhetorical relations correspond to relational XML elements which may 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 A.3.1 and A.3.2. For assigning dimensions, the decision to be made is which kind of information or action is addressed. Does it concern

a) the underlying task/activity,

b) the speaker's processing of previous utterances,

c) the addressee's processing of previous utterances,

d) the allocation of the speaker role,

e) the time needed to continue the dialogue,

f) the editing of what the speaker is saying,

g) the editing of what the addressee is currently saying,

h) the structure of the dialogue, or

i) social obligations?

A.3.1 Encoding general-purpose functions

A.3.1.1 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 some aspect of 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 functions in Figure 2 as decision trees, going systematically from left to 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. At that point choose the function that dominates the functions at that level.

A.3.1.2 Action-discussion functions

All action-discussion functions have in common that their semantic content describes 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 some aspect of the interaction or actions for dealing with social obligations.

This class of communicative functions falls apart into the classes of Commissives and Directives, familiar from speech act theory. Commissive acts all have as their common property that the sender expresses a commitment to perform an action, while directive acts are characterized 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 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

Dimension-specific functions can often be recognized by their use of particular fixed forms and formulaic expressions.

A.3.2.1 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 feedback. 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 recognize. Some of the frequently used fixed forms for negative auto-feedback are *"Huh?", "What?", "I beg your pardon"* (and similar 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.

EXAMPLE A: I would like to travel next Saturday, in the afternoon.

B: Next Saturday in the afternoon I have a flight leaving at 16:10.

B: On Saturday Ma8 8 after 12.m. I have a flight leaving at 16:10.

In his first utterance, B literally repeats part of A's question, thereby displaying what he perceived A said. In his second utterance, 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",* which is a source of ambiguity).

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 *Auto-Positive*.

It may be worth noting that there is a systematic relation between auto- and allo-feedback acts, for the following reason. A dialogue act in the Allo-Feedback dimension concerns 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 he is speaking about his own processing of a previous utterance, hence the response is an act in that participant's Auto-Feedback dimension. In general, the response to an allo-feedback act is an auto-feedback act. The reverse is also true. When participant A encounters a processing problem and tries to resolve it, e.g. using the auto-feedback question A: *"Do you mean this Saturday?",* then a response such as B's *"That's right",* speaks about the addressee's processing; hence this is an act in the Allo-Feedback dimension.

A.3.2.2 Turn management

Turn-management functions are characterized by the sender having the goal to obtain, to keep or to hand over the speaker role. Consider the case of a question-answer pair.

EXAMPLE 1 A: Do you know what time it is?

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. B's primary aim is to answer A's question and in order to do so he has to have the speaker role; this suggests that B did not have a separate goal to have the speaker role himself. Similarly, does A, by asking a question, express that he wants B to occupy the speaker role next? This does not seem to be the case, since A can continue for a while occupying the speaker role after asking a question.

EXAMPLE 2 A: Do you know what time it is? I should take the twelve seventeen train. Maybe it's too late already

B: It's twelve fifteen.

Does A's continuing to speak after asking a question indicate that he has the goal to keep the turn? If A's behaviour is interpreted in that way, then as a consequence one should perhaps assign a turn-keeping function to nearly everything that a speaker says. (By the same token, one might assign a turn-taking function any time a participant starts speaking and a turn-release function any time a participant stops speaking.)

A recommendation for when to assign a turn-management function is to assign such a function only to those stretches of behaviour which have the *sole* or *main* purpose to obtain, keep or 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, turn-initial and turn-final, which could be considered as separate dimensions. Usually only the first segment in a turn has a turn-initial function and only the last a turn-final one. The non-final utterances in a turn have a turn-keeping function when the speaker signals (e.g. by using a rising intonation or a filled pause) that he wants to continue.

When a speaker accepts the speaker role 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*").

A.3.2.3 Time management

Time-management functions are concerned with the sender buying some time. This part of ISO 24617 distinguishes two cases:

a) the speaker is unable to express immediately what he intended to say (*stalling*);

b) the speaker suspends the dialogue for a while (*pausing*).

Each of these cases may occur for several reasons. *Stalling* may occur, for example, because the speaker is looking for the right words to express what he wants to convey, because he hasn't quite made up his mind as to which information to convey or because he needs a little time to look up something. *Pausing* may occur, for example, because the speaker is aware that collecting/computing the relevant information requires considerable time or because something more urgent came up. Still other reasons can be imagined in both cases.

Stalling acts often take the form of filled pauses (e.g. "*um, let me see, well,...*"), 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 such as "*Please wait while I check the flight status*" should not be marked as *pausing* acts, but rather as requests in the Time-Management dimension, using the general-purpose function *Request*.

A.3.2.4 Own and partner communication management

In own communication management (OCM) acts, the speaker is editing his own speech. The speaker interrupts himself, noting that he said something wrong or retracts something that he just said *"Oh sorry no,..."; "No wait,.."*) or corrects himself by replacing something he just said ("*I want to travel on Tuesday THURSday*").

Partner communication management (PCM) acts similarly edit what is said by the addressee, who is at that moment occupying the speaker role. Two important cases are the correction of what is being said (*correct misspeaking*), used to correct what is perceived as a slip of the tongue and the completion of what the addressee/current speaker is struggling to say (*completion*).

A.3.2.5 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, signalling positive feedback concerning the entire preceding dialogue. Dialogue acts that have the sole function of closing a dialogue do not seem to exist; conventionally, a dialogue is considered closed when the participants have exchanged farewell greetings. This part of ISO 24617 therefore does not include a separate *Closing* function.

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*, of the *Question* or of whatever dialogue act is performed which addresses this new topic. Only explicitly signalled topic shifting should be annotated as such.

A.3.2.6 Social obligations management (SOM)

Welcome and farewell greetings that play a role in starting and ending a dialogue are domain-independent, 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 in every language. They 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 *("It was nothing"; "My pleasure"*); and a farewell greeting puts pressure on the addressee to produce a response farewell greeting.

SOM acts can also be constructed by using a general-purpose function. 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, having the same effect as a thanking and a farewell greeting.

It may be noted 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 and conditionality, 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.3.3.1 Certainty

The sender of a dialogue act can express certainty or uncertainty about the correctness of the information provided in an information-providing act. This is illustrated in Example 1, below, 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 Example 2, the resulting sentences no longer contain any indication that the speaker is uncertain about the correctness of what he says. The default case of certainty, corresponding to the unmarked expression, is therefore *certain*.

EXAMPLE 1 A: Do you know who'll be coming tonight?

B: I have a hunch that Mary won't come.

B: Peter, Alice and Bert will probably come.

B: I heard that Tom and Anne might come.

B: I'm not sure if Bill will come.

EXAMPLE 2 A: Do you know who'll be coming tonight?

B: Mary won't come.

B: Peter, Alice and Bert will come.

B: I heard that Tom and Anne will come.

B: Bill will come.

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

EXAMPLE 3 1. Mary will definitely not come.

2. Peter, Alice and Bert will come for sure.

3. I certainly agree with that.

Certainty and the lack thereof are not only indicated by verbal expressions, but also by prosody, by gaze direction and by several types of gestures. Prominent nonverbal expressions of uncertainty include gaze aversion, head waggles, rotating hand, lip pouting, lowering eyebrows and self-touching.

IMPORTANT — 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, as illustrated by the following.

EXAMPLE 4 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 A.1 can be used.

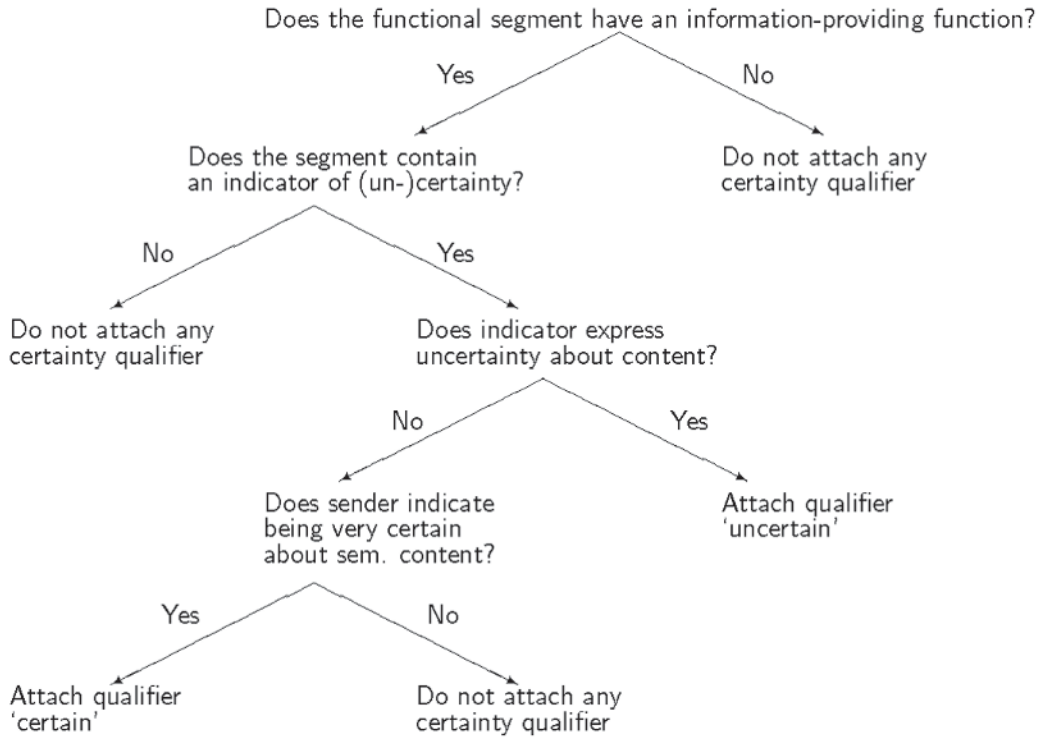

Figure A.1 — Decision tree for applying certainty qualifiers

A.3.3.2 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 illustrate this phenomenon.

EXAMPLE 5

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 Example 5 a) we see the conditional acceptance of an offer, in b) a conditional request with a conditional acceptance, in c) a conditional promise, in d) two conditional acceptances of a request, and in e) 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 case is *unconditional* and does not need to be marked up. Explicit expressions of "unconditionality" are hard to find, other than the adverb "*unconditionally"*, which is hardly ever used in natural dialogue.

Conditional dialogue acts can often be recognized by the use of conditional expressions such as *"if ..."*, "*unless*" and "*just"* — as in d), above — but just as in the case of certainty, these expressions can also be understood as part of the semantic content rather than as qualifiers. For deciding whether to add a conditionality qualifier to the annotation of a communicative function, the decision tree can be used, as shown in Figure A.2.

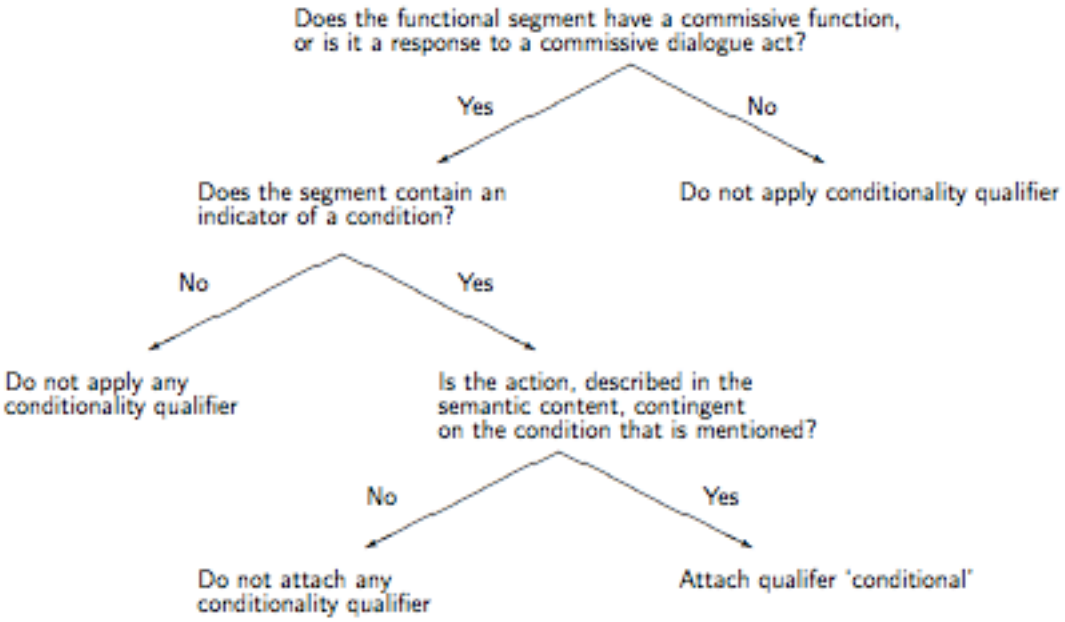

Figure A.2 — Decision tree for applying conditionality qualifiers

A.3.3.3 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. Examples of verbal expressions of sentiment are *"That would be great"* [Example (4.2) in Clause 5), *"Perfect!"* [Example (15a) in 10.3] and "*Ah, you're wonderful!"* [Example (15b) in 10.3]. 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, since the class of sentiment qualifiers is not specified in this part of ISO 24617.

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

A.3.4.1 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, contributed by another participant. This is the case for the following communicative functions defined in this part of ISO 24617:

— 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 referring to the functional antecedent by the value of the @functionalDependence attribute.

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 the following example, or

b) responds to an implicit dialogue act.

EXAMPLE U: Can you tell me what time there are trains from Harwich to York?

S: What day would you like to travel?

U: Tomorrow morning.

S: On Tuesday morning there are trains at 6:45, 70:30,...(etc.)

In the above example, S's second utterance forms a functional segment with the function *Answer*, which responds to the question formed by the dialogue acts expressed by U's first and second utterances together. In such a case, it is recommended marking functional dependence relations to each of those dialogue acts which together make up the question — in this example, to both the dialogue acts expressed in the first and third utterances.

NOTE In a complete annotation, the dialogue act in the third utterance is marked as an answer act with as its functional antecedent the question in the second, which itself asks for a clarification of the question in the first.

```
<dialogueAct xml:id="da4" target="#fs4" speaker="#s" addressee="#u" dimension="task"
communicativeFunction="answer" functionalDependence="#da1 #da3"/>
```

The case of responding to an implicit dialogue act is illustrated by B.2.6 (see 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. It is best to follow the same strategy as for multi-act antecedents and mark up functional dependences.

A.3.4.2 Feedback dependence

Every auto- or allo-feedback act is about the processing of one or more previous contributions to the dialogue and therefore has a feedback relation to these. 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 using the value of the feedbackDependence attribute to refer to the dialogue acts that the feedback is about. For feedback acts with an *Auto-Positive* or *Allo-Positive* function the feedback is usually about the last utterance from the previous speaker, but positive feedback is

sometimes more global. Feedback about several utterances can be represented by multiple values of the feedbackDependence attribute, in the same way as for functional dependences in (A.19).

A.3.4.3 Rhetorical relations

Many of the relations that 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 part of ISO 24617 does not specify any particular set of such relations and therefore does not provide guidelines for their encoding.

Annex B
(informative)

Annotated dialogue examples

B.1 Overview

This annex specifies the DiAML annotation of example dialogues and dialogue fragments. B.2 explains the annotation of some very short dialogue fragments, such as question-answer pairs. B.3 contains the annotations of a complete spoken human-computer dialogue (from a *Wizard-of-Oz* experiment) and of an extended fragment of a multimodal human-human dialogue.

B.2 Short dialogue fragments

B.2.1 This example is a two-turn dialogue fragment, where each turn constitutes a single functional segment in the Task dimension. For the anchoring of DiAML annotations to the primary text, it is assumed that the functional segments are defined at another level of analysis (see Annex D) 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. In DiAML, the association of dialogue information with a functional segment is represented by the value of the @target attribute, following the TEI guidelines for text encoding (TEI, 2010). The identification of the participants in the dialogue may similarly be assumed to be part of the metadata of the primary data (externally defined identifiers p1 and p2). This example shows the annotation of a functional dependence relation; participant P2 responds to P1's question by providing the requested information, hence this segment should be annotated as an *Answer* which has a functional dependence relation to P1's question.

a)
1. P1: Where I should check in for Munich?
2. P2: For Munich go to counters 31 to 40.

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

It may be argued that an answer to a question will always entail positive feedback, since a question can only be answered successfully if it has been understood. It is not necessary to annotate functional segments with communicative functions which they have by implication; since they can be inferred, they can be automatically added if their markup would be useful for some purpose. In this example, one might argue that the repetition *"for Munich"* in the answer is an explicit feedback signal, showing that P2 understood that P1 said *"for Munich".* Eliminating *"for Munich"* from P2's answer would seem rather awkward, however, so in this case no explicit feedback act has been annotated.

B.2.2 This example is again a question-answer pair. P1 again asks a question, but he does so in an indirect way. It might seem that P1 is asking whether P2 possesses the information when the next train to Utrecht leaves, but what P1 really wants to know is the departure time of that train. As opposed to the direct question *" What time the next train to Utrecht leave?",* which carries the assumption that the addressee is able to provide this information, the indirect formulation does not carry this assumption; it questions it. Such indirect questions are interpreted in this part of ISO 24617 as conditional requests, since they are semantically equivalent to

"Please tell me what time the next train to Utrecht leaves, if you know". P1's question is therefore annotated as in b), a dialogue act with the communicative function *Request* with the qualifier "conditional".

The second turn is segmented into two overlapping functional segments. The first part of P2's utterance, *"The next train to Utrecht leaves"* repeats most of P1's question and may be considered as a feedback signal; hence this part forms a functional segment in the Auto-Feedback dimension. A feedback dependence relation is annotated to indicate that this feedback concerns the dialogue act in the first turn. P2's utterance as a whole expresses the answer to P1's (indirect) task-related question and therefore constitutes a functional segment in the Task dimension. This segment constitutes an answer to the question in the first turn and is qualified as "uncertain" since the speaker signals his uncertainty about the correctness of the answer he provides.

a)
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?
2. P2: The next train to Utrecht leaves I think at 8:32.
   AuFB fs2.1: The next train to Utrecht leaves.
   TA fs2.2: The next train to Utrecht leaves I think at 8:32.

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

B.2.3 This example is a three-turn fragment of a dialogue from the HCRC Map Task corpus (Carletta et al., 1996) and illustrates the use of general-purpose functions for addressing another dimension than that of the task. In turn 2, participant P2 checks that he understood the previous instruction correctly, producing a *Check Question* in the Auto-Feedback dimension. In turn 3, participant P1 confirms P2's understanding, thus addressing P2's processing of that same instruction, i.e. performing a *Confirm* act in the Allo-Feedback dimension.

Turn 3 has been segmented into two functional segments. The first (*"Yeah"*) is considered as answering the question in the previous turn, the second as providing the additional information *"very slightly*", i.e. as an Inform act which elaborates the short answer, "*Yeah".* This is expressed in the annotation by a rhetorical relation to the preceding confirmation. Note that, since this part of ISO 24617 does not define a specific set of rhetorical relation, the value "elaborate" in b) is merely indicative of how a rhetorical relation can be annotated, given a set of relations such as those of the Penn Discourse Treebank (Prasad et al., 2008) or those discussed in Hovy and Maier (1993).

a)
1. P1: Move up
   TA fs1: Move up
2. P2: Slightly northeast?
   AuFB fs2: Slightly northeast?
3. P1: Yeah very slightly.
   AlloFB fs3.1: Yeah
   AlloFB fs3.2: very slightly

b)
```
<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"
    communicativeFunction="checkQuestion" dimension="autoFeedback"
    feedbackDependence="da1"/>
<dialogueAct xml:id="da3" target="#fs3.1"
    sender="#p1" addressee="#p2"
    communicativeFunction="confirm" dimension="alloFeedback"
    functionalDependence="#da2"/>
<dialogueAct xml:id="da4" target="#fs3.2"
    sender="#p1" addressee="#p2"
    communicativeFunction="inform" dimension="alloFeedback"/>
<rhetoricalLink dact="#da4" rhetoRelatum="#da2" rhetoRel="elaborate"/>
</diaml>
```

B.2.4 This example shows a two-turn fragment of a dialogue from the TRAINS corpus (Allen et al., 1994), which shows the use of a dimension-specific function (*Correct Misspeaking*) in the dimension of Partner Communication Management (PCM). Notice that a PCM act refers to something that is being said at that moment, as opposed to an allo-feedback act, which refers 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 the same "feedback dependence" relation may be used to indicate this relation.

a)
1. P1: engine E3 is going to pick up the bananas, back to Avon, dro... [fs1]
2. P2: to pick up the oranges [fs2]

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

B.2.5 This example, taken from a dialogue in the AMI multi-party dialogue corpus, shows the use of general-purpose functions (*Offer* and *Accept Offer*) in the Turn Management dimension and the annotation of a functional dependence relation.

a)
1. P1: Would you like to say something at this point? [fs1]
2. P2: Certainly. [fs2]

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

B.3 Annotated dialogues

B.3.1 Annotated human-computer dialogue

The dialogue which is analysed here is from the OVIS corpus of dialogues over the telephone between a user and the experimental OVIS information system of the Dutch railways.[5)]

In B.3.1.1, the transcription of the dialogue is shown with its segmentation into functional segments and with an informal indication of its communicative function(s) and dimension(s). Segment boundaries are indicated by "|". Horizontal lines have been added as turn separators for improved readability. The full annotation of the functional segments in DiAML is shown in B.3.1.2.

As in the example given in B.2.3, the annotation makes use of the rhetorical relation "elaborate", which as such is not part of this part of ISO 24617, to show how rhetorical relations can be annotated in DiAML.

B.3.1.1 User–System dialogue with OVIS information system:

1. S: good morning |
   fs1.1 SOM Initial Greeting
   this is the automatic information system of Public Transport Travel Information
   fs1.2 SOM Self-Introduction
   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 Set Question

---

2. U: from Harwich to York.
   fs2 Task Answer

---

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 Set Question
   fs3.2 AutoFb Inform

---

4. U: uh... | 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 Check Question

---

6. U: uh | yes | Monday evening
   fs6.1 TimeM Stalling
   fs6.2 = "uh yes"
   fs6.2 AlloFb Answer [uncertain]
   fs6.3 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...
   fs8.1 AlloFb Accept Request
   fs8.2 Task Inform
   April twenty-six.
   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"

---

5) http://www.let.rug.nl/vannoord/Ovis/ The dialogue was originally in Dutch.

fs9.1 Task Set Question
fs9.2 AutoFb Inform

10. U: at |uh... | around eight o'clock
fs10.1 = “at [..] around eight o'clock”
fs10.1 Task Answer
fs10.2 TimeM Stalling

11. S: so you want to leave at eight o'clock in the morning?
fs11 AutoFb Check Question

12. U: oh yes | sorry | eight o'clock in the evening.
fs12.1 AutoFb Inform
fs12.2 SOM Apology
fs12.3 AlloFb Disconfirm

13. S: so you want to leave at twenty o'clock?
fs13 AutoFb Check Question

14. U: yes
fs14 AlloFb Confirm

15. S: I have found the following connection.
fs15.1 DiscourseStructuring Interaction Structuring
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 = “no thank you”
fs16.1 AutoFb Decline Offer
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 Propositional Question

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 Propositional Question

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 Interaction structuring
goodbye
fs23.2 SOM Initial Goodbye

B.3.1.2 DiAML annotation of the dialogue in B.3.1.1:

```
<diaml xmlns:“http://www.iso.org/diaml/”>
<dialogueAct xml:id=“da1” target=“#fs1.1”
      sender=“#s” addressee=“#u”
      communicativeFunction=“initGreeting”/>
<dialogueAct xml:id=“da2” target=“#fs1.2”
      sender=“#s” addressee=“#u”
      communicativeFunction=“selfIntroduction”/>
```

```
<dialogueAct xml:id="da3" target="#fs1.3"
      sender="#s" addressee="#u"
      communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da3" rhetoRelatum="#da2" rhetoRel="elaborate"/>
<dialogueAct xml:id="da4" target="#fs1.4"
      sender="#s" addressee="#u"
      communicativeFunction="setQuestion" dimension="task"/>
```

```
<dialogueAct xml:id="da5" target="#fs2"
      sender="#u" addressee="#s"
      communicativeFunction="answer" dimension="task"
      functionalDependence"#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="inform" feedbackDependence="#da5"/>
```

```
<dialogueAct xml:id="da8" target="#fs4.1"
      sender="#u" addressee="#s"
      communicativeFunction="stalling"\>
<dialogueAct xml:id="da9" target="#fs4.2"
      sender="#u" addressee="#s"
      communicativeFunction="answer" dimension="task"
      functionalDependence="#da6"/>
```

```
<dialogueAct xml:id="da10" target="#fs5"
      sender="#s" addressee="#u"
      communicativeFunction="checkQuestion" dimension="autoFeedback"/>
```

```
<dialogueAct xml:id="da11" target="#fs6.1"
      sender="#u" addressee="#s"
     communicativeFunction="stalling"/>
<dialogueact xml:id="da12" target="#fs6.2"
     sender="#u# addressee="#s"
     communicativeFunction="answer" dimension="alloFeedback"
     certainty="uncertain" functionalDependence="#da10"/>
<dialogueAct xml:id="da13" target="#fs6.3"
     sender="#u" addressee="#s"
     communicativeFunction="inform" dimension="alloFeedback"/>
<rhetoricalLink dact="#da13" rhetoRelatum="#da12" rhetoRel="elaborate"/>
```

```
<dialogueAct xml:id="da14" target="#fs7.1"
      sender="#s" addressee="#u"
      communicativeFunction="inform" dimension="autoFeedback"/>
<dialogueAct xml:id="da15" target="#fs7.2"
      sender="#s" addressee="#u"
      communicativeFunction="request" dimension="autoFeedback"/>
```

```
<dialogueAct xml:id="da16" target="#fs8.1"
      sender="#u" addressee="#s"
      communicativeFunction="acceptRequest" dimension="alloFeedback"
      functionalDependence="#da15"/>
<dialogueAct xml:id="da17" target="#fs8.2"
      sender="#u" addressee="#s"
      communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da18" target="#fs8.3"
      sender="#u" addressee="#s"
      communicativeFunction="inform" dimension="task"
<rhetoricalLink dact="#da18" rhetoRelatum="#da17" rhetoRel="elaborate"/>
```

```
<dialogueAct xml:id="da19" target="#fs9.1"
      sender="#s" addressee="#u"
      communicativeFunction="setQuestion"  dimension="task"/>
<dialogueAct xml:id="da20" target="#fs9.2"
```

```
      sender="#s" addressee="#u"
      communicativeFunction="inform" feedbackDependence="#fda17 #da18"/>
<dialogueAct xml:id="da21" target="#fs10.1"
      sender="#u" addressee="#s"
      communicativeFunction="answer" dimension="task"
      functionalDependence="#da19"/>
<dialogueAct xml:id="da22" target="#fs10.2"
      sender="#u" addressee="#s"
      communicativeFunction="stalling"/>
<dialogueAct xml:id="da23" target="#fs11"
      sender="#s" addressee="#u"
      communicativeFunction="checkQuestion" dimension="autoFeedback"
      feedbackDependence="#da21"/>
<dialogueAct xml:id="da24" target="#fs12.1"
      sender="#u" addressee="#s"
      communicativeFunction="inform" dimension="autoFeedback"/>
<dialogueAct xml:id="da25" target="#fs12.2"
      sender="#u" addressee="#s"
      communicativeFunction="apology"/>
<dialogueAct xml:id="da26" target="#fs12.3"
      sender="#u" addressee="#s"
      communicativeFunction="disconfirm" dimension="alloFeedback"
      functionalDependence="#da23"/>
<dialogueAct xml:id="da27" target="#fs12.3"
      sender="#u" addressee="#s"
      communicativeFunction="correction" dimension="task"
      functionalDependence="#da23"/>
<dialogueAct xml:id="da28" target="#fs13"
      sender="#s" addressee="#u"
      communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<dialogueAct xml:id="da29" target="#fs14"
      sender="#u" addressee="#s"
      communicativeFunction="confirm" dimension="alloFeedback"
      functionalDependence="#da28"/>
<dialogueAct xml:id="da30" target="#fs15.1"
      sender="#s" addressee="#u"
      communicativeFunction="interactionStructuring"/>
<dialogueAct xml:id="da31" target="#fs15.2"
      sender="#s" addressee="#u"
      communicativeFunction="answer" dimension="task"
      functionalDependence="#da5 #da9 #da27"/>
<dialogueAct xml:id="da32" target="#fs15.3"
      sender="#s" addressee="#u"
      communicativeFunction="offer" dimension="alloFeedback"/>
<dialogueAct xml:id="da33" target="#fs16.1"
      sender="#u" addressee="#s"
      communicativeFunction="declineOffer" dimension="autoFeedback"
      functionalDependence="#da32"/>
<dialogueAct xml:id="da34" target="#fs16.2"
      sender="#u" addressee="#s"
      communicativeFunction="thanking"/>
<dialogueAct xml:id="da35" target="#fs17"
      sender="#s" addressee="#u"
      communicativeFunction="offer" dimension="task"/>
<dialogueAct xml:id="da36" target="#fs18"
      sender="#u" addressee="#s"
      communicativeFunction="propositionalQuestion" dimension="task"/>
<dialogueAct xml:id="da37" target="#fs19.1"
      sender="#s" addressee="#u"
```

```
      communicativeFunction="inform" dimension="discourseStructuring"/>
<dialogueAct xml:id="da38" target="#fs19.2"
      sender="#s" addressee="#u"
      communicativeFunction="answer" dimension="task"
      qualifier="uncertain" functionalDependence="#da36"/>
<dialogueAct xml:id="da39" target="#fs20"/>
      sender="#u" addressee="#s"
      communicativeFunction="propositionalQuestion" dimension="task"/>
<dialogueAct xml:id="da40" target="#fs21.1"
      sender="#s" addressee="#u"
      communicativeFunction="apology"/>
<dialogueAct xml:id="da41" target="#fs21.2"
      sender="#s" addressee="#u"
      communicativeFunction="answer" dimension="task"
      functionalDependence="#da39"/>
<dialogueAct xml:id="da42" target="#fs22"
      sender="#u" addressee="#s"
      communicativeFunction="thanking"/>
<dialogueAct xml:id="da43" target="#fs23.1"
      sender="#s" addressee="#u"
      communicativeFunction="interactionStructuring"
      dimension="discourseStructuring"/>
<dialogueAct xml:id="da44" target="#fs23.2"
      sender="#s" addressee="#u"
      communicativeFunction="initGoodbye"/>
</diaml>
```

B.3.2 Annotated human-human dialogue

B.3.2.1 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_inbreath and VOC_outbreath, respectively.

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 a 1) verbal segment, where the sender says "*um"* 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 Clause 8, that a functional segment in general has several components, consisting of sender behaviour in various communicative channels, together making up a multimodal unit.

While the transcription ("encoding") of multimodal dialogue behaviour and its segmentation is not within the scope of this part of ISO 24617, the encoding in B.3.2.2 is a plausible extension of the text encoding defined by the Text Encoding Initiative (TEI P5, 2010). In this particular example, the vocal (but nonverbal) behaviour and the characterization of lip gestures are described simply by named values; in other cases, such as head gestures, the representation will be more complex and involve the representation of several features such as duration, direction, speed and number of repetitions. See Annex D for TEI-compliant encoding of functional segments and the anchoring of dialogue acts in primary data.

B.3.2.2 Fragment of multimodal human-human Map Task dialogue:

1. P1: okay, | starting off, | we are .. above .. a caravan park
   - fs1.1 TurnM: Turn Take
   - DS: Opening
   - fs1.2 DS: Interaction Structuring
   - fs1.3 Task: Inform
2. P2: mmhmm
   - fs2 AutoFb Auto-Positive
3. P1: we are going to go due south | NONVOC_noise ... |

fs3.1 Task: Instruct
fs3.2 TimeM: Stalling
TurnM: Turn Keep
straight south | ... and NONVOC_noise ... |
fs3.3 OCM: Self-Correction
fs3.4 TimeM: Stalling
TurnM: Turn Keep
then we're going to g— turn
fs3.5 OCM: Self-Correction
turn straight back round and head north ... past an old mill ... on the right ... hand side
fs3.6: Task: Instruct

4. P2: VOC_outbreath ... | B due south and then back up again
fs4.1 TurnM: Turn Take
fs4.1 TimeM: Stalling
fs4.2 AutoFb: Check Question

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 [Elaborate]

6. P2: right okay
fs6 AutoFb: Auto-Positive

7. P1: okay | and then we're going to turn ... VOC_inbreath east
fs7.1 = “okay”
fs7.2 = “and then we're going to turn [...] east”
fs7.3 = “ ... VOC_inbreath”
fs7.1 AutoFb: Auto-Positive
TurnM: Turn Grab
fs7.2 Task: Instruct
fs7.3 TimeM: Stalling
TurnM: Turn Keep

8. P2: mmhmm
fs8 AutoFb: Auto-Positive

9. P1: not ... straight east ... slightly sort of northeast | ...
fs9.1 Task: Inform
VOC_outbreath ...
fs9.2 TimeM: Stalling
TurnM: Turn Keep

10. P2: s-- | slightly northeast
fs10.1 TurnM: Turn Grab
fs10.2 AutoFb: Check Question

11. P1: slightly slightly yeah | very slightly | VOC_inbreath ...
fs11.1 Task: Confirm
fs11.2 Task: Inform
fs11.3 TimeM: Stalling
TurnM: Turn Keep
and we're going to continue straight along ... GES_lipsmack VOC_inbreath ... um ...
quite a wee distance
fs11.4 = “we're going to continue straight along [...] quite a wee distance on that course and then we're going to turn north again”
fs11.4 Task: Instruct
fs11.5 = “... GES_lipsmack VOC_inbreath ... um ...”
fs11.5 TimeM: Stalling
TurnM: Turn Keep

12. P2: right | mmhmm
fs12.1 AutoFb: Auto-Positive
fs12.2 AutoFb: Auto-Positive

13. P1: and ... | immediat-- | well |
    - fs13.1 TurnM: Turn Take
      TimeM: Stalling
    - fs13.2 OCM: Retraction
    - fs13.3 TurnM: Turn Keep

    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 a distance
    - fs13.5 TimeM: Stalling
      TurnM: Turn Keep
    - fs13.6 Task: Instruct

14. P2: okay
    - fs14 AutoFb: Auto-Positive

15. P1: okay | so we've turned | and we're going up north again
    - fs15.1 AutoFb: Auto-Positive
    - fs15.2 Task: Inform
    - fs15.3 Task: Instruct

B.3.2.3 DiAML annotation of the dialogue in (B.9):

```
<diaml xmlns:“http://www.iso.org/diaml/”>
<dialogueAct xml:id=“da1” target=“#fs1.1”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“turnTake”/>
<dialogueAct xml:id=“da2” target=“#fs1.1”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“opening”/>
<dialogueAct xml:id=“da3” target=“#fs1.2”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“interactionStructuring”/>
<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” feedbackDependence=“#da4”/>
<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”/>
<dialogueAct xml:id=“da8” target=“#fs3.2”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“stalling”/>
<dialogueAct xml:id=“da9” target=“#fs3.3”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“selfCorrection”/>
<dialogueAct xml:id=“da10” target=“#fs3.4”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“stalling”/>
<dialogueAct xml:id=“da11” target=“#fs3.4”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“turnKeep”/>
<dialogueAct xml:id=“da12” target=“#fs3.5”
     sender=“#p1” addressee=“#p2”
     communicativeFunction=“selfCorrection”/>
<dialogueAct xml:id=“da13” target=“#fs3.6”
     sender=“#p1” addressee=“#p2”
```

```
     communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da14" target="#fs4.1"
     sender="#p2" addressee="#p1"
     communicativeFunction="turnTake"/>
<dialogueAct xml:id="da15" target="#fs4.2"
     sender="#p2" addressee="#p1"
     communicativeFunction="stalling"/>
<dialogueAct xml:id="da16" target="#fs4.3"
     sender="#p2" addressee="#p1"
     communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<dialogueAct xml:id="da17" target="#fs5.1"
     sender="#p1" addressee="#p2`'
     communicativeFunction="confirm" dimension="alloFeedback"/>
<dialogueAct xml:id="da18" target="#fs5.2"
     sender="#p1" addressee="#p2"
     communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da19" target="#fs5.3"
     sender="#p1" addressee="#p2"
     communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da19" rhetoRelatum="#da18" rhetoRel="elaborate"/>
<dialogueAct xml:id="da20" target="#fs5.4"
     sender="#p1" addressee="#p2"
     communicativeFunction="instruct" dimension="task"/>
<rhetoricalLink dact="#da20" rhetoRelatum="#da18" rhetoRel="elaborate"/>
<dialogueAct xml:id="da21" target="#fs6"
     sender="#p2" addressee="#p1"
     communicativeFunction="autoPositive" feedbackDependence="#da20"/>
<dialogueAct xml:id="da22" target="#fs7.1"
     sender="#p1" addressee="#p2"
     communicativeFunction="autoPositive" feedbackDependence="#da21"/>
<dialogueAct xml:id="da23" target="#fs7.1"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnGrab"/>
<dialogueAct xml:id="da24" target="#fs7.2"
     sender="#p1" addressee="#p2"
     communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da25" target="#fs7.3"
     sender="#p1" addressee="#p2"
     communicativeFunction="stalling"/>
<dialogueAct xml:id="da26" target="#fs7.3"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da27" target="#fs8"
     sender="#p2" addressee="#p1"
     communicativeFunction="autoPositive" feedbackDependence="#da24"/>
<dialogueAct xml:id="da28" target="#fs9.1"
     sender="#p1" addressee="#p2"
     communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da29" target="#fs9.2"
     sender="#p1" addressee="#p2"
     communicativeFunction="stalling"/>
<dialogueAct xml:id="da30" target="#fs9.2"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da31" target="#fs10.1"
     sender="#p2" addressee="#p1"
     communicativeFunction="turnGrab"/>
<dialogueAct xml:id="da32" target="#fs10.2"
     sender="#p2" addressee="#p1"
```

```
     communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<dialogueAct xml:id="da33" target="#fs11.1"
     sender="#p1" addressee="#p2"
     communicativeFunction="confirm" dimension="alloFeedback"
     functionalDependence="#da32"/>
<dialogueAct xml:id="da34" target="#fs11.2"
     sender="#p1" addressee="#p2"
     communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da34" rhetoRelatum="da33" rhetoRel="elaborate"/>
<dialogueAct xml:id="da35" target="#fs11.3"
     sender="#p1" addressee="#p2"
     communicativeFunction="stalling"/>
<dialogueAct xml:id="da36" target="#fs11.3"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da37" target="#fs11.4"
     sender="#p1" addressee="#p2"
     communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da38" target="#fs11.5"
     sender="#p1" addressee="#p2"
     communicativeFunction="stalling"/>
<dialogueAct xml:id="da39" target="#fs11.5"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnKeep"/>
 <dialogueAct xml:id="da40"   target="#fs12.1"
     sender="#p2" addressee="#p1"
     communicativeFunction="autoPositive" feedbackDependence="#da38"/>
<dialogueAct xml:id="da41"   target="#fs12.2"
     sender="#p2" addressee="#p1"
     communicativeFunction="autoPositive" feedbackDependence="#da38"/>
<dialogueAct xml:id="da42" target="#fs13.1"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnTake"/>
<dialogueAct xml:id="da43" target="#fs13.1"
     sender="#p1" addressee="#p2"
     communicativeFunction="stalling"/>
<dialogueAct xml:id="da44" target="#fs13.2"
     sender="#p1" addressee="#p2"
     communicativeFunction="retraction"/>
<dialogueAct xml:id="da45" target="#fs13.3"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da46" target="#fs13.4"
     sender="#p1" addressee="#p2"
     communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da47" target="#fs13.5"
     sender="#p1" addressee="#p2"
     communicativeFunction="stalling"/>
<dialogueAct xml:id="da48" target="#fs13.5"
     sender="#p1" addressee="#p2"
     communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da49" target="#fs13.6"
     sender="#p1" addressee="#p2"
     communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da50" target="#fs14"
     sender="#p2" addressee="#p1"
     communicativeFunction="autoPositive" feedbackDependence="#da49"/>
<dialogueAct xml:id="da51" target="#fs15.1"
     sender="#p1" addressee="#p2"
```

```
     communicativeFunction="autoPositive" feedbackDependence="#da50"/>
<dialogueAct xml:id="da52" target="#fs15.2"
     sender="#p1" addressee="#p2"
     communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da53" target="#fs15.3"
     sender="#p1" addressee="#p2"
     communicativeFunction="instruct" dimension="task"/>
</diaml>
```

Annex C
(normative)

Formal definition of DiAML

C.1 Overview

The dialogue act markup language (DiAML) has been designed in accordance with the linguistic annotation framework (LAF), see ISO 24612, which draws a distinction between the concepts of *annotation* and *representation*. The term "annotation" refers to the linguistic information that is added to regions of primary data, independent of the format in which the information is represented; "representation" refers to the format in which an annotation is rendered, independent of its content. According to LAF, *annotations* are the proper level of standardization, rather than *representations*. This distinction is implemented in the DiAML definition using a methodology for defining annotation languages developed by Bunt (2010), according to which a syntax specification defines, besides a class of XML-based *representation structures*, also a class of more abstract *annotation structures*. These components are called *concrete* and *abstract* syntax, respectively. Annotation structures are set-theoretical structures consisting of concepts of the kind that populate the metamodel shown in Figure 1. The concrete syntax defines a rendering of annotation structures in XML.

C.2 Abstract syntax

The abstract syntax of DiAML consists of

a) a specification of the elements from which annotation structures are built up, called a "conceptual inventory", and

b) a specification of the possible ways of combining these elements to form annotation structures.

C.2.1 Conceptual inventory

The conceptual inventory of DiAML consists of the following finite sets:

— *DP:* dialogue participants;

— *Dim:* dimensions;

— *CF:* communicative functions;

— *FS:* functional segments (provided by the segmentation of the primary data);

— *QV:* a set of finite sets $Q_1,.. Q_k$, of qualifiers (the sets Q_i are called "*qualification aspects*");

— *RR:* rhetorical relations.

C.2.2 Annotation structures

An annotation structure is a set of *entity structures* and *link structures.* Entity structures contain semantic information about a functional segment; link structures describe semantic relations between functional segments.

Entity structures: An entity structure in DiAML contains a characterization of a dialogue act, in a so-called "dialogue act structure" (see below) and a specification of which functional segment it is anchored to and how

it relates to other acts in the dialogue. Formally, an entity structure is a quadruple <*s, α, E, δ*> consisting of a functional segment *s*, a "dialogue act structure" α, a set *E* of entity structures which contain dialogue acts that α depends on and a specification (*δ*) of the type of dependence (functional or feedback). If the set *E* is empty, *E* and δ may for simplicity be omitted.

A *dialogue act structure* contains the information that characterizes a single dialogue act. This includes a specification of the sender, the addressee(s) and the communicative function. For dialogue acts with a general-purpose communicative function, the dimension of the semantic content is another important component; for dialogue acts with a dimension-specific function no dimension needs to be specified, since it is inherent in the definition of the function. General-purpose functions may additionally have one or more qualifiers. According to the metamodel shown in Figure 1, a dialogue act may also have "other participants" in addition to a sender and addressees; this is reflected in the abstract syntax by allowing an additional (set of) other participant(s) *H.* A dialogue act structure is therefore one of the following structures:

— a triple <*S, A,* f_d>, consisting of a sender *S*, a (set of) addressee(s) *A* and a dimension-specific communicative function f_d,;

— a quadruple <*S, A, H,* f_d*>,* additionally containing a set *H* of other participants;

— a quintuple <*S, A, d, g, q*>, having a general-purpose function *g* instead of a dimension-specific one and containing additionally a dimension *d* and a list *q* of zero or more function qualifiers (if *q* is empty, it may for simplicity be omitted);

— a sixtuple <*S, A, H, d, g, q*>, with additionally a set *H* of other participants.

Link structures: A link structure is a triple <*ε, E, ρ*> consisting of an entity structure ε, a set *E* of one or more entity structures and a rhetorical relation *ρ*, which relates the dialogue act in *ε* to those in *E*.

C.3 Concrete syntax

C.3.1 General

The concrete syntax is defined in accordance with the methodology for defining semantic annotation languages described in Bunt (2010). This methodology includes the notion of an *ideal representation format*, defined as one which is 1) "complete", in the sense that every annotation structure defined by the abstract syntax has a representation defined by the concrete syntax, and 2) "unambiguous", in the sense that every representation defined by the concrete syntax represents one and only one annotation structure defined by the abstract syntax. Since the semantics of DiAML is defined for the structures defined by the *abstract* syntax, any two representation formats which are "ideal" in this sense are semantically equivalent and every representation in one such format can be converted by a meaning-preserving mapping into any other such format.[6)]

The DiAML concrete syntax specification consists of a vocabulary, specifying names of XML tags, attributes and values for the various ingredients in the conceptual inventory and a specification of the possible ways of combining these elements in XML representation structures, specifying XML elements for entity structures and link structures and defining an XML representation of the anchoring of dialogue act structures in functional segments. The vocabulary includes the names of dimensions, communicative functions and function qualifiers, which are defined as data categories in Annex E.

Mirroring the definition of entity structures and link structures in the abstract syntax, the concrete syntax defines XML *entity representation structures* and *link representation structures*:

6) See Bunt (2010; 2012) for formal definitions and proofs and Ide and Bunt (2010) for applying this to the GrAF framework for linguistic annotation (Ide and Suderman, 2007).

C.3.2 Entity structure representations

An entity structure representation is an XML element called dialogueAct, with the following attributes:

— the obligatory attribute @xml:id, whose value is a unique identifier of an entity structure representation;

— the obligatory attribute @target, whose value refers to a functional segment;

— the obligatory attribute @sender, whose value refers to a dialogue participant (identified in the metadata of the annotated primary data);

— the obligatory attribute @addressee, whose values refer to a set of dialogue participants;

— the optional attribute @otherParticipant, whose values refer to a set of dialogue participants;

— the obligatory attribute @communicativeFunction, whose value names one of the communicative functions defined in this part of ISO 24617;

— the attribute @dimension, which is obligatory for those structures where the value of the @communicativeFunction attribute is a general-purpose function and which is optional for structures where this attribute has a dimension-specific function as its value. Its value names one of the nine dimensions defined in this part of ISO 24617;

— the optional attributes @functionalDependence and @feedbackDependence, whose values refer to one or more entity structure representations;

— the optional attributes @certainty, @conditionality and @sentiment, whose values specify one of the communicative function qualifiers defined in this part of ISO 24617.

C.3.3 Link structure representation

An XML element called rhetoricalLink is defined, for representing rhetorical relations among dialogue acts, which has the attributes @dact, @rhetoRelatum and @rhetoRel for representing respectively the currently annotated dialogue act, the one that it has a rhetorical relation to and the particular rhetorical relation.

The representation of the annotation of the dialogue fragment follows the example it represents.

EXAMPLE 1. P1: What time does the next train to Utrecht leave?

Taskfs1: What time does the next train to Utrecht leave?

2. P2: The next train to Utrecht leaves I think at 8:32.

AutoFB fs2.1: The next train to Utrecht leaves

Taskfs2.2: 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"/>
<dialogueAct xml:id="da2" target="#fs2.1" sender="#p2" addressee="#p1"
    communicativeFunction="inform" feedbackDependence="#da1"/>
<dialogueAct xml:id="da3" target="#fs2.2" sender="#p2" addressee="#p1"
    communicativeFunction="answer" certainty="uncertain" dimension="task"
    functionalDependence="#da1"/>
</diaml>
```

C.4 DiAML semantics

C.4.1 Semantics of dialogue acts and communicative functions

A fundamental requirement for semantic annotations is that they have a well-defined semantics (Bunt and Romary, 2002), because, among other reasons, they should enable inferencing. The DiAML language has a formal semantics defined for its abstract syntax, in terms of information-state updates; see Bunt (2011b). The details of such a semantics depend on the precise definition of information states. In this subclause, a semantics is outlined which makes no further assumptions than that an information state has a number of components representing different kinds of information, an assumption which is shared between all proposals for information states[7)] and that an information state has a part (called the "pending context") for buffering update information that needs to be checked for consistency before being added to the rest of the information state. The details of an information-state update semantics also depend on whether a single addressee is considered or multiple addressees and on whether only the information states of addressees are considered to be updated by dialogue contributions or also that of the sender. To simplify matters, only the update of a single addressee's information state is considered here, which is the basis for approaches involving multiple information states.

The most important kind of structure defined by the DiAML abstract syntax is the dialogue act structure, which is a functional characterization of a dialogue act. It does not correspond to a full-blown dialogue act representation, since it does not include the semantic content but only a dimension which classifies the semantic content. The semantics of a full-blown dialogue act is obtained by combining the interpretation of a dialogue act structure with a semantic content. Formally, this is accomplished by applying the interpretation of an entity structure $<s, \alpha, E, \delta>$, containing a dialogue act structure α, to the semantic content $\kappa(s)$ of the functional segment in which the dialogue act is expressed. The result will be an information state update operation that represents the meaning of that dialogue act. This is shown in Formula (C.1) for the case that the dialogue act has no functional dependences to other dialogue acts. (The semantics of dependence relations is considered below).

$$I_a(<s,\alpha>) = I_a(\alpha)(\kappa(s)) \quad \text{(C.1)}$$

The interpretation $I_a(\alpha)$ of a dialogue act structure, α, occurring in the right-hand side of Formula (C.1), is defined in Formula (C.2) for those structures which have no (or an empty set of) communicative function qualifiers (the case of functions with qualifiers is considered separately below):

$$I_a(<S,A,H,f_d>) = I_a(<S,A,f_d>) = I_a(f_d)(I_a(S), I_a(A))$$

$$I_a(<S,A,H,d,f>) = I_a(<S,A,d,f>) = I_a(f)(I_a(S), I_a(A), I_a(d)) \quad \text{(C.2)}$$

i.e. the interpretation of a dialogue act structure is the interpretation of its communicative function, applied to the interpretations of its sender, its addressee and, if present, its dimension; no interpretation is given here to the possible presence of a set, H, of "other participants". The result of this will be a function which can be applied to a semantic content.

C.4.2 Dialogue acts as update operations

The semantics of an annotation structure as a whole, consisting of the entity structures $\{e_1,\ldots, e_n\}$ and the link structures $\{L_1,\ldots, L_k\}$, is defined as the sequential application of the update functions corresponding to the constituent entity and link structures ordered by the textual order of their functional segments, where the updates of two textually coinciding entity (or link) structures are unified ($\cup$), rather than sequenced (;). The notation *";/∪"* is used to indicate this formally: *"α ;/∪ β"* means that the operation α is followed by the operation β if α is textually ordered before β and is unified with β if the two textually coincide:

$$I_a(\{e_1,.., e_n, L_1,.., L_k\}) = I_a(e_1)\ ;/\cup\ ..;/\cup\ I_a(e_n)\ ;/\cup..\ ;/\cup\ I_a(L_1)\ ;/\cup\ ..\ ;/\cup\ I_{a(}L_k) \quad \text{(C.3)}$$

7) See e.g. Poesio and Traum (1998); Bunt (2000); Ahn (2001); Cooper (2004); Keizer et al. (2011); Petukhova (2011).

The semantic relatedness between dialogue acts, as, for example, visualized in Figure 2, is brought out in their interpretation as information state updates. Compare, for example, a *Confirm* act and an *Answer*. According to their definitions in E.5.1.2, an *Answer* with semantic content *p* tells an addressee, A, that 1) the sender, S, wants to make information *p* available to A, 2) that S believes that A wanted to know *p*, and 3) that S assumes that *p* is true. A *Confirm* act with the same semantic content does all that as well and additionally tells A that 4) S believes that A already thought *p,* but was uncertain about it. Relations like the one between *Confirm* and *Answer* are captured in the DiAML semantics by defining the interpretation of a communicative function as a combination of *elementary update functions*, each of which takes care of one single update. The semantics of the *Answer* function is thus the combination of three elementary update functions and that of the *Confirm* function is the combination of these three elementary functions and a fourth one, which expresses the difference between *Confirm* and *Answer*. The combination of elementary update functions is formally defined as follows.

For two functions *f* and *g*, which are identical in the overlap of their domains, the "union" $f \cup g$ is defined as follows:

for any argument x, if *f(x)* is defined then

$$(f \cup g)(x) = f(x) \tag{C.4}$$

or, if *g(x)* is defined then

$$(f \cup g)(x) = g(x) \tag{C.5}$$

or $(f \cup g)(x)$ is undefined.

Elementary update functions are defined as parameterized schemes with parameters for a sender, an addressee and an information state component, such as the following ones:

$U_{10}(X,Y,D_i,p)$: (C.6)
add to component D_i of *Y*'s pending context the information that participant *X* wants to know whether *p*;

$U_{11}(X,Y,D_i,p)$: (C.7)
add to component D_i of *Y*'s pending context the information that participant *X* assumes participant *Y* to know whether *p*.

These two schemes can be used to specify the semantics of the communicative function *Propositional Question* as (25):

$$I_a(\text{Propositional Question}) = \lambda X.\ \lambda Y.\ \lambda D_i.\ \lambda z.\ U_{10}(X,Y,D_i,z) \cup U_{11}(X,Y,D_i,z) \tag{C.8}$$

The function specified in the right-hand side can be inserted in the interpretation of a dialogue act annotation structure, as defined in (C.4). When applied to two participants *Sys* and *Usr* and to a task-related question, the result is the update function (C.9), in which Sys_{TaskC} denotes the component of the system's pending context where task-related information is buffered:

$$I_a(\textit{Prop. Question})(\textit{Usr, Sys, TaskC}) = \lambda p.\ U_{10}(\textit{Usr, Sys, Sys}_{\textit{TaskC}}, p) \cup U_{11}(\textit{Usr, Sys, Sys}_{\textit{TaskC}}, p) \tag{C.9}$$

Applying the function defined in (C.8) to a propositional semantic content (i.e. a value for *p*) results in the specification of how to update *Sys*'s pending context. For example, when *Usr* asks *Sys* whether flight KLM flight 476 departs at 19:15, formalised as Dep(KL476)=19:15, then if *Sys* understands *Usr* correctly, the component Sys'_{TaskC} of the system's pending context (the component for buffering task-related information) is extended with two beliefs:

a) according to the update U_{10}(*Usr, Sys, TaskC*, Dep(KL476)=19:15), *Sys* believes that *Usr* wants to know whether Dep(KL476) is 19:15;

b) according to the update U_{11}(*Usr, Sys, TaskC*, Dep(KL476)=19:15), *Sys* believes that *Usr* assumes that *Sys* knows whether Dep(KL476) is 19:15.

C.4.3 The semantics of function qualifiers and dependence relations

C.4.3.1 Qualifiers

Communicative function qualifiers make the information state updates of the communicative functions that they qualify more specific. Qualifiers come in two varieties, “restrictive” and “additive” ones (see Bunt, 2011b). Restrictive qualifiers make the preconditions of a communicative function more specific, for instance specifying for an answer that there is some uncertainty about the correctness of its content. Additive qualifiers, by contrast, enrich a communicative function with additional information, for instance adding that an offer is accepted *happily*. The “certainty” and “conditionality” qualifiers of this part of ISO 24617 are restrictive; “sentiment” is an additive qualifier.

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

a. $I_a(<f,q_r>) = I_a(f)(I_a(q_r))$

b. $I_a(<f,q_a>) = \lambda S.\ \lambda z.\ [I_a(f)(S,z) \cup I_a(q_a)(S,z)]$ (C.10)

Clause (C.11) shows how this semantics of function qualification is used to obtain the semantic interpretation of a dialogue act structure with qualifiers:

$I_a(<S,A,d,f,q>) = I_a(<f,q>)(I_a(S),\ I_a(A),\ I_a(d))$ (C.11)

The following example of an uncertain *Inform* act illustrates this.

a. S: The KL 476 departs I think at 19:15.

b. $V_a(\text{Inform},\textit{uncertain}) = [\lambda s.\ \lambda A.\ \lambda B.\ \lambda C_i.\ \lambda p.\ V_a(\text{Inform})(A,B,C_i,p,s)](V_a(\textit{uncertain}))$

$= \lambda A.\ \lambda B.\ \lambda C_i.\ \lambda p.\ U_1(A,B,C_i,p,\ weak) \cup U_2(A,B,C_i,p,weak)$ (C.12)

where the update schemes U_2 and U_2 are defined as follows:

$U_1(X,Y,D_i,\ p)$: (C.13)
add to component D_i of Y's pending context the information that participant X wants participant Y to believe that p.

$U_2(X,Y,D_i,\ p,s)$: (C.14)
add to component D_i of Y's pending context the information that participant X believes that p, with belief strength s.

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 update scheme U_2, which occurs in the semantics of the *Inform* function, leaves the strength of the speaker's belief unspecified.)

C.4.3.2 Dependence relations

The semantics of an entity structure with dependence relations is defined as follows, where s_k is the functional segment of entity structure e_k and f_a the communicative function of a; κ_2 is a function which combines the semantic contents of sets of dependent dialogues acts and γ is a similar combination function for feedback acts:

a. $I_a(<s,a,E,\ \text{functional}>) = I_a(a)(\kappa_2(\kappa_1(s),\ \{\kappa_1(s_i)) | e_i \in E\},\ f_a)$

b. $I_a(<s,a,E,\ \text{feedback}>) = I_a(a)\ (\gamma(s),\ \{e_i | e_i \in E\},\ f_a)$ (C.15)

For example, for an *Answer* act *a,* expressed in the functional segment *s*, which is functionally dependent on the *Question* act *b*, we have an entity structure <*s,a,*{*b*}>, the semantic interpretation according to (C.15a) is:

$$I_a(<s,a,\{b\},\text{functional}>) = I_a(a)(\kappa_2(\kappa_1(s), \{\kappa_1(s_b)\}, Answer)) \quad \text{(C.16)}$$

C.4.3.3 Rhetorical relations

The semantics of a link structure $<\varepsilon,E,\rho>$ is an update operation which creates a link in the addressee's information state between the related dialogue acts. The creation of such links requires information states to include representations of the dialogue acts that occurred earlier in the dialogue, a “dialogue history”. This assumes that the dialogue acts that occur in a dialogue are represented as such in an information state, an assumption that is shared by virtually all proposals for dialogue context modelling. More specifically, it is commonly assumed that an information state has a part called the “dialogue history”, where a record is kept of the communicative events in the dialogue, typically in the form of transcriptions of what each participant says (and does); to these representations, an interpretation is attached in terms of dialogue acts. The updates corresponding to link structures then come down to the addition of rhetorical links between these representations.

Annex D
(normative)

DiAML technical schema

D.1 Overview

This annex introduces the technical scheme for the dialogue act markup language (DiAML) for the concrete representation of annotations of dialogue data with dialogue act information using XML.

This representation relies on a three-level architecture:

a) a primary source, which may correspond to a speech recording, textual transcription or any lower-level annotation thereof (e.g. a tokenization or a morphosyntactic annotation according to ISO 24611);

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. The attributes of this element and their values have been specified in Annex C.

Functional relations between dialogue acts, like the relation between a question and an answer or between an offer and its acceptance, are represented by the values of the @functionalDependence attribute; the relation between a dialogue act with a feedback function and the preceding dialogue act(s) that it provides or elicits feedback about, is likewise represented by the values of the @feedbackDependence attribute.

Rhetorical relations among dialogue acts are represented by rhetoricalLink elements, which have an attribute @rhetoRel for specifying a particular rhetorical relation. The possible values of this attribute are not fixed by this part of ISO 24617, but would for example include such relations as *elaborate*, *justify*, *exemplify*, *clarify*.

Functional segments are identified by means of the functionalSegment element, which groups together the components of multimodal communicative behaviour that constitute a multimodal functional segment. The verbal component of a multimodal functional segment can be identified in terms of the words in a transcription of the sender's spoken contribution, following joint TEI-ISO standard 24610-1 and TEI P5 for referring to the corresponding stretch of text 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 a (possibly discontinuous) verbal segment or to a nonverbal or multimodal stretch of dialogue behaviour.

D.2 Example

The following excerpt exemplifies how the three levels mentioned above 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 ("... *um*..."), which is accompanied by a frowning expression and a head gesture and followed by lip smacking and a sigh, before the verbal contribution continues:

P1: Do you know where I should check in for Munich?
P2: For Munich go to … um [+frown +waggle] [lip smack] [sigh] counters 31 to 40.

The utterance by P2 is segmented into two functional segments: the discontinuous verbal segment *"For Munich go to counters 31 to 40",* in which P2 expresses an answer to the preceding question and the multimodal segment containing the frown, waggle, lip smack and sigh, plus the word "*um*"; in this segment P2 performs both a *Stalling* act and a *Turn Keep* act. 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 11.2.

The transcription of spoken or multimodal dialogue is not included in of this part of ISO 24617, but the example shows how dialogue act annotations can be linked to XML representations of multimodal functional segments (see Petukhova and Bunt, 2012 for further discussion of the issues involved). This example shows, for the sake of illustrating the possibilities, the XML representation of a multimodal segment that 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 gestures, can be added in similar ways.

The TEI header contains metadata that include the identities of the dialogue participants.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="http://www.tei-c.org/release/xml/tei/custom/schema/relaxng/-
tei_all.rng"
schematypens="http://relaxng.org/ns/structure/1.0"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>DiAML annotation example</title>
      </titleStmt>
      <publicationStmt>
        <p>...</p>
      </publicationStmt>
      <sourceDesc>
        <p>...</p>
      </sourceDesc>
    </fileDesc>
    <profileDesc>
      <particDesc>
        <person xml:id="p1">
          <p>the first participant</p>
        </person>
        <person xml:id="p2">
          <p>the second participant</p>
        </person>
      </particDesc>
    </profileDesc>
  </teiHeader>
  <text>
    <timeline unit="ms">
      <when xml:id="T1" absolute="192725"/>
      <when xml:id="T2" absolute="328377"/>
      <when xml:id="T3" absolute="357722"/>
      <when xml:id="T4" absolute="468737"/>
      <when xml:id="T5" absolute="488614"/>
      <when xml:id="T5" absolute="567512"/>
      <when xml:id="T6" absolute="715836"/>
      <when xml:id="T7" absolute="729126"/>
      <when xml:id="T8" absolute="761223"/>
      <when xml:id="T9" absolute="789264"/>
```

```
 <when xml:id="T10" absolute="881926"/>
 <when xml:id="T11" absolute="902804"/>
 <when xml:id="T12" absolute="1279207"/>
</timeline>
<body>
  <div>
    <head>Simple dialogue fragment</head>
     <u xml:id="u1" who="#p1" start="#T1" end="#T2">Do you know where I
        should check in for Munich</u>
     <u xml:id="u2a" who="#p2" start="#T3" end="#T4">For Munich go to</u>
     <u xml:id="u2b" who="#p2" start="#T5" end="#T6">um</u>
     <u xml:id="u2c" who="#p2" start="#T11" end="#T12">counters 31 to 40</u>
  </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">For</w>
      <w xml:id="w12">Munich</w>
      <w xml:id="w13">go</w>
      <w xml:id="w14">to</w>
      <w xml:id="w15">um</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>The nonverbal communicative behaviour of each of the participants,
          segmented and time-stamped)</head>
    <kinesic type="headMove" subtype="headGesture" xml:id="hmv1" who="#p2"
             start="#T5" end="#T6" ana="#gestDesc1#heg1"/>
    <kinesic type="browMove" subtype="frown" xml:id="bmv1" who="#p2"
             start="#T5" end="#T6"/>
    <kinesic type="lipMove" subtype="lipsmack" xml:id="lmv1" who="#p2"
             start="#T7" end="#T8"/>
    <vocal xml:id="voc1" who="#p2" type="outbreath" start="#T9" end="#T10"/>
    <kinesic type="headGesture" xml:id="heg1" ana="#gestDesc1"/>
    <fs xml:id="gestDesc1">
      <f name="direction">
        <symbol value="leftright"/>
      </f>
      <f name="velocity">
        <symbol value="slow"/></f>
    </fs>
```

```
</div>
<div>
  <head>Identification of functional segments</head>
  <fs type="verbalSegment" xml:id="ves1">
    <f name="segParts" fVal="#u1"/>
  </fs>
  <fs type="verbalSegment" xml:id="ves2">
    <f name="segParts" fVal="#u2a" "#u2c"/>
  </fs>
  <fsspanGrp type="functionalSegment" xml:id="fs1" ana="#da1">
    <f namespan type="verbalComponent" fVal="#ves1"xml:id="ts1" from="w1"
       to="w10"/>
  </fs>
  </spanGrp>
  <fs spanGrp type="functionalSegment" xml:id="fs2" ana="#da2">
    <f name="verbalComponent" fVal="#ves2"/>
  </fs>
  <fs type="functionalSegment" xml:id="fs3" ana="#da3 #da4" >
    <span type="verbalComponent" xml:id="ts2.1" from="w11" to="w19"/>
    <f namespan type="vocalComponent" fValfrom="#voc1"/>
    <f namespan type="headComponent" fValfrom="#hmv1"/>
    <f namespan type="lipComponent" fValfrom="#lmv1"/>
    <f namespan type="browComponent" fValfrom="#bmv1"/>
  </fs spanGrp>
</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"/>
    </f>
  </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>
    <f name="functionalDependence" fVal="#da1"/>
  </fs>
  <fs type="dialogueAct" xml:id="da3" target="#fs3">
    <f name="sender" fVal="#p2"/>
    <f name="addressee" fVal="#p1"/>
    <f name="communicativeFunction">
       <symbol value="stalling"/></f>
  </fs>
  <fs type="dialogueAct" xml:id="da4" target="#fs3">
    <f name="sender" fVal="#p2"/>
    <f name="addressee" fVal="#p1"/>
```

```
          <f name="communicativeFunction">
            <symbol value="turnKeep"/></f>
        </fs>
      </div>
    </body>
  </text>
</TEI>
```

An alternative, direct XML encoding of DiAML would look as follows, assuming the same representation of metadata 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="#da1"/>
</diaml>
```

Annex E
(normative)

Data categories for core concepts

E.1 Overview

This annex contains data categories for the core concepts of this part of ISO 24617, namely the concepts of *sender* and *addressee*, *functional segment*, the nine *dimensions*, the *communicative functions*, and the function *qualifiers* that have been introduced.

A data category, as defined by ISO 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, e.g. 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 a *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 which helps to understand it and place it in perspective, and an *Example*, which has an optional *Source* attribute for indicating its origin.

E.2 Dialogue participants

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

/addressee/	
Definition	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, Interlocutor.

E.3 Functional segments

/functionalSegment/	
Definition	Minimal stretch of communicative behaviour that has a communicative function.
— Source	Geertzen et al., 2007
— Note	A functional segment may have more than one communicative function. A functional segment may be discontinuous, may be part of or overlap with another functional segment, may have parts contributed in more than one speaking turn and may have parts contributed by different speakers.
Explanation	A functional segment is “minimal” 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 s1 of communicative behaviour has a communicative function F, then any larger segment s2 which includes s1 could also be said to have that function. The minimality constraint thus helps to avoid considering spurious functional segments.

E.4 Dimensions

/task /	
Definition	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 and Task Management (DAMSL), Activity (GBG-IM), Task/Activity (DIT)
Explanation	The notion of a “task” is intended in a very broad sense here, including any activity which can be said to aim at achieving a goal. Such a goal may be quite specific, such as knowing the arrival time of a particular train or more general, such as creating a pleasant atmosphere.

/autoFeedback/	
Definition	Category of dialogue acts by which the sender discusses or reports on his processing of previous dialogue contributions.
— Source	Bunt, 1995
— Note	Related terminology in other schemes: Feedback (GBG-IM); Backchannel (common). Feedback in GBG-IM includes the class of feedback elicitation acts which forms part of the /alloFeedback/ category.

/alloFeedback/	
Definition	Category of dialogue acts in which the sender discusses the addressee's processing of previous dialogue contributions.
— Source	Bunt, 1995

/turnManagement/	
Definition	Category of dialogue acts whose performance is intended to regulate the allocation of the speaker role.
— Source	Allwood et al., 1993

— Note	In the literature often referred to as the “turn-taking system”.
/timeManagement/	
Definition	Category of dialogue acts which concern the allocation of time to the participant occupying the speaker role.
— Source	DIT

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

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

/partnerCommunicationManagement/	
Definition	Category of dialogue acts which are performed by a dialogue participant who does not have the speaker role and by which he edits the speech of the participant who currently has the speaker role.
— Source	DIT^{++}

/socialObligationsManagement/	
Definition	Category of dialogue acts performed for dealing with social obligations such as greeting, thanking and apologising.
— Source	DIT

E.5 Communicative functions

E.5.1 General-purpose functions

E.5.1.1 Information-seeking functions

/question/	
Conceptual domain	/setQuestion/ /propositionalQuestion/ /choiceQuestion/ /checkQuestion/
Definition	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 it, nor does it include rhetorical questions, which from a semantic point of view are not questions at all but rather the expression of an opinion.

Example	“And so?”
— Source	DIAMOND corpus
/propositionalQuestion/	
Conceptual domain	/checkQuestion/
Broader concept	/question/
Definition	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 the addressee, A, knows whether the proposition is true and puts pressure on A to provide this information.
— Source	LIRICS
— Note	Related terminology in other schemes: YN-Question (TRAINS), Qyery-yn (HCRC Map Task); info-request (DAMSL)
Explanation	A propositional question corresponds to what is commonly termed a YN-question in the linguistic literature. This part of ISO 24617 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 actually not the case.
Example	“Does the meeting start at ten?”

/setQuestion/	
Broader concept	/question/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to know which elements of a given set have a certain property specified by the semantic content. S puts pressure on the addressee, A, to provide this information, which S assumes that A possesses. S believes that at least one element of the set has that 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?”

/checkQuestion/	
Broader concept	/propositionalQuestion/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to know whether a proposition, which forms the semantic content, is true. S holds the uncertain belief that it 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: Check (DIT, HCRC MapTask, TRAINS), Tag Question (SWBD-DAMSL), Request_Comment (Verbmobil)
Example	“The meeting starts at ten, right?”

/choiceQuestion/	
Broader concept	/question/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to know which one from a list of alternative propositions, specified 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	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”.
Example	“Should the telephone cable go in the telephone line slot or in the external line slot?
— Source	DIAMOND corpus

E.5.1.2 Information-providing functions

/inform/	
Conceptual domain	/agreement/ /disagreement/ /answer/ /confirm/ /disconfirm/ /correction/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the information contained in the semantic content available 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, MALTUS).
Explanation	The inform function may also have more specific rhetorical functions such as: explain, elaborate, exemplify and justify; this is treated in this part of ISO 24617 by means of rhetorical relations.
Example	“The 6.34 to Breda leaves from platform 2.”

/agreement/	
Broader concept	/inform/
Definition	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!”

/disagreement/	
Conceptual domain	/correction/
Broader concept	/inform/
Definition	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 assumes to be true.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Reject (DAMSL, COCONUT, TRAINS, MRDA, Verbmobil); Denial (TRAINS)
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	J: “do you know where to find ink savings?” S: “uh… oh I think to the left of the ink cartridge” J: “uh… no”
— Source	DIAMOND corpus

/correction/	
Broader concept	/disagreement/
Definition	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 that instead the information that S provides is correct.
— Source	Commonplace
Example	“To Montreal, not to Ottawa”

/answer/	
Conceptual domain	/confirm/ /disconfirm/
Broader concept	/Inform/
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

/confirm/	
Broader concept	/answer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that the proposition which forms the semantic content is true. S believes that A holds a weak belief that this proposition is true and that A wants to know for certain whether it is; S assumes that it is.
— Source	Commonplace
Example	“Indeed”

/disconfirm/	
Broader concept	/answer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A that he proposition which forms the semantic content is false. S believes that A holds a weak belief that this proposition is true and that S wants to know for certain whether it is; S assumes that it is false.
— 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”

E.5.1.3 Commissive functions

/offer/	
Conceptual domain	/promise/
Definition	Communicative function of a dialogue act by which the sender, S, commits himself to perform the action, specified by the semantic content, in the manner or with the frequency that may be specified, conditional on the consent of the addressee that S do so.
— Source	Commonplace
— Note	Related terminology in other schemes:
Example	“Shall I begin?”; “Would you like to have some coffee?”

/promise/	
Broader concept	/offer/
Definition	Communicative function of a dialogue act by which the sender, S, commits himself to perform the action, specified in the semantic content, in the manner or with the frequency that may be specified. S believes that this action would be in the interest of the addressee.
— 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”

/addressRequest/	
Conceptual domain	/acceptRequest/ /declineRequest/
Definition	Communicative function of a dialogue act by which the sender, S, indicates that he considers performing an action that he was requested to perform.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Assess (AMI)
Explanation	The addressRequest function covers a range of possible responses to a request. If the response does not contain the expression of 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).
Example	A: “Give me the gun.” S: “If you push the bag to me.”

/acceptRequest/	
Broader concept	/addressRequest/
Definition	Communicative function of a dialogue act by which the sender, S, commits himself to performing an action that he was requested to perform.
— Source	LIRICS
— Note	Related terminology in other schemes: Accept (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil)
Example	A: “Could you close the door please?” B: “Sure.”

/declineRequest/	
Broader concept	/addressRequest/
Definition	Communicative function of a dialogue act by which the sender, S, commits himself to not perform an action that he was requested to perform.
— Source	LIRICS
— Note	Related terminology in other schemes: Reject (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil)
Example	“Not now.”

/addressSuggest/	
Conceptual domain	/acceptSuggest/ /declineSuggest/
Definition	Communicative function of a dialogue act by which the sender, S, indicates that he considers to perform an action that was suggested to him, possibly depending on certain conditions that he makes explicit.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Assess (AMI)
Example	A: “Let's go together.” S: “Only if we're in full agreement about how to proceed when we get there.”

/acceptSuggest/	
Broader concept	/addressSuggest/
Definition	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	LIRICS
— Note	Related terminology in other schemes: Accept (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil)
Example	A: “Shall we go and have a look around?” B: “Let's do so.”

/declineSuggest/	
Broader concept	/addressSuggest/
Definition	Communicative function of a dialogue act performed by which the sender, S, indicates that he will not perform an action that was suggested to him, possibly depending on certain conditions that he makes explicit.
— Source	LIRICS
— Note	Related terminology in other schemes: Reject (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	“I'd rather not.”

E.5.1.4 Directive functions

/request/	
Conceptual domain	/instruct/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, feel obliged to perform a certain action in the manner or with the frequency described by the semantic content, conditional on A's consent to perform the action. S assumes that A is able to perform this action.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Influence-addressee-future-action (DAMSL); Request Commit (Verbmobil)
Example	“Please turn to page five”; “Don't do this ever again, please”.

/instruct/	
Broader concept	/request/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, feel obliged to perform a certain action which is described in or can be inferred from the semantic content, in the manner or with the frequency described by the semantic content. S assumes that A is able to perform this action.
— Source	DIT^{++}; HCRC Map Task
— Note	Related terminology in other schemes: Action-directive (DAMSL, SWBD-DAMSL, COCONUT); Command (HCRC Map Task); Direct (SPAAC); Do (MALTUS)

Example	“Go right round until you get to just above that.”
— Source	HCRC Map Task corpus
/suggest/	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, consider the performance of a certain action, specified by the semantic content. S believes that this action is in A's interest and assumes that A is able to perform the action.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Open-option (DAMSL, SWBD-DAMSL, COCONUT).
Example	“Let's wait for the speaker to finish.”

/addressOffer/	
Broader concept	/instruct/
Conceptual domain	/acceptOffer/ /declineOffer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to indicate that he is considering the possibility that A performs the action that A has offered to perform, possibly with certain conditions that he makes explicit.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Assess (AMI).
Example	“Maybe later”

/acceptOffer/	
Broader concept	/addressOffer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S would like A to perform the action that A has offered to perform, possibly with certain conditions that he makes explicit.
— 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”

/declineOffer/	
Broader concept	/addressOffer/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S does not want A to perform the action that A has offered to perform, possibly depending on certain conditions that he makes explicit.
— Source	LIRICS
— Note	Related terminology in other schemes: Reject (DAMSL, SWBD-DAMSL, TRAINS, Verbmobil).
Example	English: “No thank you”; Danish: “Nej tak”; French: “ Non merci”.

E.5.2 Dimension-specific functions

E.5.2.1 Feedback functions

/autoPositive/	
Definition	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, MRDA), Acknowledgement (HCRC MapTask, TRAINS, SPAAC, C-Star), Backchannel (Verbmobil). 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 positive feedback, it concerns a longer stretch of dialogue.
Example	“Uh-huh”; “Okay”; Nonverbally: nodding; “Yes”

/autoNegative/	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S's processing of the previous utterance(s) encountered a problem.
— Source	LIRICS
— Note	Related terminology in other schemes: Signal-Non-Understanding (DAMSL, Coconut, MRDA), Pardon (SPAAC), Feedback-Negative (Verbmobil). This type of feedback may be further broken down into more specific levels of processing, as is exemplified in the DIT and SLSA schemes.
Example	English: “I beg your pardon”; Portuguese: “Como?”

/alloPositive/	
Definition	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 in the DIT^{++} and SLSA schemes.
Example	“Correct”; “Right”

/alloNegative/	
Definition	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) encountered a problem.
— Source	LIRICS
— Note	This type of feedback may be broken down into more specific levels of processing, as is done in the DIT^{++} scheme.
Example	“No no no no no”

/feedbackElicitation/	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to know whether A's processing of the previous utterance(s) was successful.
— Source	Allwood et al., 1993
— Note	Related terminology in other schemes: Request Clarify (Verbmobil), Understanding Check (MRDA), Clarification Check (COCONUT), Check (HCRC Map Task), Question Attention (MALTUS).
Example	English: “Okay?”; Italian: “Capisce?”; Dutch: “Ja?”

E.5.2.2 Turn-management functions

/turnAccept/	
Broader concept	/tunTake/
Definition	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	Occurs especially in multiparty dialogue. Related terminology in other schemes: Take-Turn (TRAINS), Turn Opening (SLSA).
Example	A: “What do you say, Craig?” C: “OK, let me see.”
— Source	AMI corpus

/turnTake/	
Conceptual domain	/turnAccept/
Definition	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: Turn-Take (TRAINS), Regain Turn (MRDA).
Example	““Uh...” as a turn-initial segment
— Source	

/turnGrab/	
Definition	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

/turnAssign/	
Broader concept	/turnRelease/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to pass the speaker role to a designated other participant.
— Source	Common in literature on turn taking in conversation.
— Note	Related terminology in other schemes: Turn Give (DIT), Assign-Turn (TRAINS).
Example	A: “Craig?”, characteristically accompanied by the speaker directing his gaze to Craig, possibly also nodding or pointing in his direction and raising the eyebrows.
— Source	AMI corpus

/turnRelease/	
Conceptual domain	/turnAssign/
Definition	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.

/turnKeep/	
Definition	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	“Uh” not in turn-initial position
— Source	

E.5.2.3 Time-management functions

/stalling/	
Definition	Communicative function of a dialogue act performed by the sender in order to have a little extra time to construct his contribution.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Hold (SPAAC); Stall (AMI); Delay (DAMSL, SWBD-DAMSL, COCONUT). 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.
Example	“Let me see...”, “Uh...”; speaking slowly; repeating something (“We .. we went to…”)
— Source	

/pausing/	
Definition	Communicative function of a dialogue act performed by the sender 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 order to prepare a continuation of the dialogue (e.g. the sender needs to look up something) 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”

E.5.2.4 Discourse-structuring functions

/interactionStructuring/	
Conceptual domain	/opening/
Definition	Communicative function of a dialogue act performed in order to explicitly structure the interaction.
— Source	LIRICS
— Note	The function “Interaction structuring” covers a range of activities which explicitly structure the dialogue, such as the introduction of a new topic, the announcement of a certain type of dialogue act and the closing of a topic.
Example	“And the windows, we had to replace all the windows”
— Source	Switchboard corpus

/opening/	
Broader concept	/interactionStructuring/
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	DAMSL
— Note	Related terminology in other schemes: Task Initiate (Verbmobil)
Example	“Okay” at the start of a (multi-party) dialogue
— Source	AMI corpus

E.5.2.5 Own- and partner-management functions

/selfError/	
Conceptual domain	/selfCorrection/
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^{++}
— Note	Related terminology in other schemes: Repaired (TRAINS); Change (SLSA)
Example	S: “so you want to leave at eight o'clock in the morning?” U: “yes oh sorry no...”
— Source	OVIS corpus

/retraction/	
Conceptual domain	/selfCorrection/
Broader concept	/selfError/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to withdraw something he just said within the same turn.
— Source	DIT^{++}
— Note	Related terminology in other schemes: Speech Repair (DAMSL, MRDA, TRAINS), Repair (TRAINS), Correct-Self (SPAAC)
Example	“then we're going to g— ”
— Source	HCRC Map Task corpus

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

/completion/	
Definition	Communicative function of a dialogue act performed for assisting the addressee in the completion of an utterance.
— Source	Commonplace
— Note	Related terminology in other schemes: Complete (SPAAC); Collaborative completion (MRDA).
Example	A: “which should leave us plenty of time to uh... uh” 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
— Note	Related terminology in other schemes: Correct Misspeaking (DIT); Correction suggestion (TRAINS).
Example	A: “second engine E3 is going to uh 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

E.5.2.6 Social obligations management functions

/initGreeting/	
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: Greeting (DAMSL, SWBD-DAMSL, COCONUT, C-Star), Greet (Verbmobil, SLSA, TRAINS, Alparon).
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”; “How are you?”

/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: Greeting (DAMSL, SWBD-DAMSL, COCONUT, C-Star), Greet (Verbmobil, SLSA, TRAINS, Alparon).
Explanation	Greetings usually come in initiative-response pairs within a dialogue; this data category corresponds to the second element of such a pair.
Example	I: “Schiphol Information, good morning.” C: “Good morning”.

/initSelfIntroduction/	
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
— Note	Related terminology in other schemes: Introduce (Vermobil).
— Explanation	Introductions usually come in initiative-response pairs within a dialogue; this data category corresponds to the first element of such a pair.
Example	“Schiphol Information.”

/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	Commonplace
— Note	Related terminology in other schemes: Introduce (Vermobil).
— Explanation	Introductions usually come in initiative-response pairs within a dialogue; this data category corresponds to the second element of such a pair.
Example	I: “Schiphol Information, good morning.” C: “Good morning, this is De Bruin in Arnhem.”
— Source	

/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: Apologise (C-Star); Polite (Verbmobil).
Example	A: “second engine E3 is going to uh 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

/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
— Note	Related terminology in other schemes: Downplayer (SWBD-DAMSL, MRDA)
Example	“No problem.”

/thanking/	
Definition	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).
Example	English: “Thanks a lot.”; Portuguese: “Muito obrigado”; Swedish: “Tack so mycket”, Greek: “Evcharisto”

/acceptThanking/	
Definition	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
— Note	Related terminology in other schemes: Downplayer (SWBD-DAMSL)
Example	English: “Don't mention it”; Spanish: “De nada”; Greek: “parakalo”.

/initGoodbye/	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to signal the current utterance is his last contribution to the dialogue; S pressures the addressee, A, to respond with a returnGoodbye act.
— Source	Commonplace
— Note	Related terminology in other schemes: Goodbye (DAMSL, COCONUT), Bye (Verbmobil, Alparon).
Example	S: “Bye bye, see you later” (A: “Bye bye, see you.”)

/returnGoodbye/	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to acknowledge his awareness that the addressee, A, has made his last contribution to the dialogue and to signal his agreement to end the dialogue.
— Source	Commonplace
— Note	Related terminology in other schemes: Bye (Verbmobil).
Example	(S: “Bye bye, see you later”) A: “Bye bye, see you.”

E.6 Qualifiers

E.6.1 Conditionality

/conditionality/	
Definition	Class of predicates which can be associated with most action-discussion functions to express whether the sender of a dialogue act with that function is considering the performance of the action under discussion subject to certain conditions.
— Source	Petukhova and Bunt (2009a)
/conditional/	
Definition	Predicate which can be associated with most action-discussion functions 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.
— Source	Petukhova and Bunt (2009a)
Example	“If you're ready maybe you can start the presentation”
— Source	AMI corpus
Example	A: “Can we just go over that again” B: “I'm afraid we don't have time, unless you do it very quickly”
— Source	AMI corpus

/unconditional/	
Definition	Predicate which can be associated with an action-discussion function to express that the sender of a dialogue act with that function is considering the performance of the action under discussion without any conditions.
— Source	Petukhova and Bunt (2009a)
Example	“I'll come tomorrow, no matter what.”

E.6.2 Certainty

/certainty/	
Definition	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.
— Source	DIT^{++}

/uncertain/	
Definition	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.
— Source	Petukhova and Bunt (2009a)
Example	“That might be a good idea.”
— Source	AMI corpus

/certain/	
Definition	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.
— Source	Petukhova and Bunt (2009a)

E.6.3 Sentiment: Emotion and Attitude

/sentiment/	
Definition	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.
— Source	Petukhova and Bunt (2009a)
— Note	In the absence of a widely agreed set of sentiment values, this part of ISO 24617 does not define any data categories for sentiment values.

Annex F
(informative)

Examples of possible additional data categories

F.1 Overview

This annex contains some examples of data categories that are not included in this part of ISO 24617, but that illustrate the possibilities of extending the standard. These examples include the dimension called Contact Management and a few communicative functions. The main reason for not including them is that they are not very commonly found in existing annotation schemes (see Annex G).

F.2 Dimensions

/contactManagement/	
Definition	Category of dialogue acts which are performed by a dialogue participant for establishing or ensuring contact with other participants..
— Source	DIT^{++}

F.3 Communicative functions

/examQuestion/	
Broader concept	/question/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to know whether the addressee, A, possesses the requested information which S does possess. S puts pressure on A to provide the requested information.
— Source	Commonplace
— Note	Exam questions have the same form as ordinary questions; their occurrence in a particular setting with participants in the roles of examiner and examinee makes a question recognizable as an exam question.

/rhetoricalQuestion/	
Broader concept	/inform/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to inform the addressee, A, that S believes that it would be absurd to think that the proposition, expressed in the semantic content, is true. S believes that this proposition is false. S puts pressure on A to respond as if S asked a propositional question with the same content.
— Source	Commonplace
— Example	“Do you think I'm crazy?”; “Was Rome built in a day?”
— Note	A rhetorical question looks exactly like a propositional question, but is in fact a way of stating the denial of that proposition.

/lie/	
Broader concept	/inform/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, believe that a certain proposition is true which S believes to be false.
— Source	Commonplace

/contactCheck/	
Broader concept	/question/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to verify that the addressee, A, is ready to communicate with S.
— Source	DIT^{++}
— Example	“Yes?”; “Hello?”

/contactIndication/	
Broader concept	/question/
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make it known to the addressee, A, that S is ready to communicate with A.
— Source	DIT^{++}
— Example	“Yes”; “Oh hi”

/completionElicitation/	
Definition	Communicative function of a dialogue act performed for eliciting help from the addressee in the completion of the current utterance.
— Source	DIT^{++}
Example	A: “which should leave us plenty of time to *uh... uh*” S: “get to Corning”
— Source	TRAINS corpus

Annex G
(informative)

Concepts in existing schemes

As part of the project to establish this part of ISO 24617, a detailed study was conducted in order to provide theoretical and empirical arguments for identifying core dimensions and communicative functions (Petukhova and 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 schemes. 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:

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, and

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 and Bunt, 2009a).[8)] Tables G.1 to G.9 are from this publication, together with the conclusions. Table G.1 shows the relative frequencies of functional segments in ten dimensions (nine of which are core dimensions in this part of ISO 24617) for three different dialogue corpora. The variation between the corpora is worth noting. Table G.2 shows the relative frequencies of functional segments addressing only one dimension. From this table it may be concluded that the ten dimensions considered in the table are all independently addressable.

Tables G.3 to G.9 show the occurrence of dimension-specific communicative functions in various dimensions in 18 existing annotation schemes. Tables G.11 to G.15 show the occurrence of general-purpose functions in these annotation schemes.

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.

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

With respect to Feedback a distinction should be made 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. Making only the distinction between feedback-giving and feedback-eliciting acts, however, does not to 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 speakers 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 noted 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. Butterworth (1980) noted 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 nonetheless display the emotional or cognitive state of the dialogue participant, which is obviously important for dialogue analysis and which may 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 this aspect is important for some types of dialogues, e.g. for telephone conversations, as in the OVIS corpus — see Table G.1, which shows the distribution of functional segments over the ten dimensions considered in (Petukhova and Bunt, 2009a). The table shows that these dimensions, which are distinguished in the DIT^{++} and LIRICS annotation schemes are all empirically justified.

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

Dimension / Corpus	AMI	DIAMOND	OVIS
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
Time Management	16,8	10,7	0,6
Social Obligations Management	0,3	5,0	3,8
Discourse Structuring	2,2	2,3	2,4
Own Communication Management	8,7	0,7	0,3
Partner Communication Management	0,3	0,3	0,1
Contact Management	0,1	1,3	12,3

Another criterion for distinguishing a dimension being that of independent addressability: for each of the ten dimensions occurring in Table G.1, the question was studied as to whether functional segments occur which express a dialogue act addressing a dimension without also expressing a dialogue act in another dimension. Table G.2 shows the results for the AMI, OVIS and DIAMOND corpora.

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

Dimension Corpus	AMI	DIAMOND	OVIS
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
Social Obligations Management	0,3	6,4	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
Contact Management	0,1	0,3	0,7

Table G.3 — Positive auto-feedback functions in existing annotation schemes

Schema	Positive auto-feedback function				
DIT^{++}	Positive attention	Pos. perception	Pos. interpretation	Pos. evaluation	Pos. execution
LIRICS	Positive auto-feedback				
DAMSL	Signal-understanding		Acknowledgement		
SWBD-DAMSL	Signal-understanding		Acknowledgement	Summarize-reformulate	
MRDA	Signal-understanding		Acknowledgement Appreciation	Assessment	
COCONUT	Signal understanding	Acknowledgement Repeat-rephrase			
AMI	Comment-about-understanding POS			Assess	Inform POS
HCRC Map Task			Acknowledgement		
Verbmobil	Backchannel			Acknowledge	Feedback Positive
SLSA	Positive contact	Pos. perception	Pos. understanding	Pos. acceptance/ attitude	
TRAINS	Acknowledgement			Positive Evaluation	
SPAAC	Echo		Acknowledge	Appreciate	
MALTUS	Positive attention	Repeat-rephrase		Appreciation	
Chiba				Positive response	
Alparon			Acknowledgement		
C-Star			Acknowledgement		

Table G.4 — Negative auto-feedback functions in existing annotation schemes

Schema	Negative auto-feedback function				
DIT^{++}	Neg. attention	Neg. perception	Neg. interpretation	Neg. evaluation	Neg. execution
LIRICS	Negative auto-feedback				
DAMSL	Signal-non-understanding				
SWBD-DAMSL	Signal-non-understanding				
MRDA	Signal-non-understanding		Understanding Check		
COCONUT	Signal non-under-standing		Clarification Check		
AMI	Comment-about-understanding NEG				Inform NEG
HCRC Map Task			Check		
Verbmobil	Request Clarify				Feedback Negative
SLSA	Negative contact	Neg. perception	Neg. understanding	Neg. attitude	
TRAINS			Negative evaluation		
SPAAC	Pardon				
MALTUS	Negative attention				
Chiba	Follow-up: understand			Negative response	
Alparon					
C-Star					

Table G.5 — Turn-management functions in existing annotation schemes

Schema	Turn-management function					
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 answer	Turn Maintain	Turn Edit	
MRDA	Regain Turn	Grabber		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		Backchannel
Primula		Turn Grabber		Turn Holder		Backchannel
Chiba				Hold		

Table G.6 — Social obligations management functions in existing annotation schemes

Scheme	Social obligations management functions				
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
COCONUT	Greeting		Goodbye		
MRDA				Sympathy; Downplayer	
AMI	Be-positive; Be-negative				
Verbmobil	Greet	Introduce	Bye	Polite (apology and compliment)	Thank
SLSA	Greet				
TRAINS	Greet				
MALTUS	Politeness				
Primula	Politeness; Face-threatening; Face-saving				
Alparon	Greet		Bye		
C-Star	Greeting	Self-Introduction		Apologize	Thanking

Table G.7 — Discourse structuring functions in existing annotation schemes

Scheme	Discourse structuring functions				
DIT^{++}	Opening	Pre-Closing	Topic Introduction	Topic Shift	Topic Shift Announcement
LIRICS	Interaction Structuring				
DAMSL	Opening	Closing			
SWBD-DAMSL	Opening	Closing			
COCONUT	Opening	Closing	Topic		
MRDA			Topic Change		
AMI	Argument structure and topic segmentation schemes				
HCRC Map Task			Ready (for topic shifts)		
Verbmobil	Task Initiate	Task Close		Digress	
SLSA	Opening	Closing	Opening	Continuation	
LinLin	Opening	Closing	Topic Layer		
SPAAC			Initiate	Topic	
MALTUS				Topic Change	
Primula	Opening	Closing	Topic Opening	Topic Closing/Change	
Chiba	Opening	Closing	Topic Break		
C-Star		Closing	Introduce Topic		

Table G.8 — Own and partner communication management functions in existing annotation schemes

Scheme	Own communication management			Partner communication management	
DIT^{++}	Self-Error	Retraction	Self-Correction	Correct-Misspeaking	Completion
LIRICS	Self-Error	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		Speech Repair; Correct Assumption		Correct-Misspeaking	Completion
SLSA	Change				
TRAINS	Repair				
SPAAC		Correct-Self		Correct	Complete
MALTUS		Restated info with repetition/correction			

Table G.9 — Time and contact management functions in existing annotation schemes

Scheme	Time management		Contact management	
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 G.10 — General-purpose information-providing functions in DIT^{++}, LIRICS, DAMSL, SWBD-DAMSL, MRDA and COCONUT

DIT^{++}	LIRICS	DAMSL	SWBD-DAMSL	MRDA	Coconut
Inform	Inform	Assert; Re-assert;	Statement Opinion; Statement Non-opinion	Statement	Assert; Re-assert;
Uncertain Inform		Other Statement			Other Statement
Agreement	Agreement	Agreement; Accept	Accept	Accept; Affirmative Answer	Accept
Disagreement	Disagreement	Reject	Reject; Dispreferred Response	Reject; Dispreferred Response	Reject
Correction	Correction				Correct Assumption
Uncertain Answer			Maybe	Maybe	
Answer	Answer; Set Answer Propositional Answer	Answer	Answers-to-non-yes-no-questions; Yes-Answer; No-Answer Affirmative-Non-No Answer; Negative-Non-No Answer Dispreferred Response	Answer; Affirmative Answer; Negative Answer Dispreferred Answer	Answer
Confirm	Confirm				
Disconfirm	Disconfirm				

Table G.11 — General-purpose information-providing functions in DIT^{++}, AMI, Verbmobil, TRAINS, HCRC Map Task, SPAAC and MALTUS

DIT^{++}	AMI	Verbmobil	TRAINS	HCRC	SPAAC	MALTUS
Inform	Inform	Inform	Inform	Statement	Inform; Express-wish/ opinion	Statement
Uncertain Inform	Uncertain Inform				Express possibility	
Agreement	Inform Positive	Accept	Accept	Reply-y	Accept	Positive Answer
Disagreement	Inform Negative	Reject	Reject	Reply-n	Negate	Negative Answer
Correction						
Inform Elaborate/ Motivate/…		Clarify; Give-Reason	Support Inform; Argumentation Acts: Elaborate, Summarize, Clarify	Explain; Clarify	answElab	
Answer	Inform Positive/ Negative	Inform	Evaluation		Answer	Positive/ Negative Answer
Set-Answer				Reply-w		
Propositional Answer				Reply-y; Reply-n		
Confirm		Confirm		Reply-y	Confirm	Positive Answer
Disconfirm	Inform Negative	Disconfirm		Reply-n		Negative Answer

Table G.12 — General-purpose information-seeking functions in DIT^{++}, LIRICS, DAMSL, SWBD-DAMSL, MRDA and COCONUT

DIT^{++}	LIRICS	DAMSL	SWBD-DAMSL	MRDA	Coconut
Question	Question	Info-Request		Question	Info-Request
Set-Question	Set-Question		WH-Question	WH-Question	
Propositional Question	Propositional Question		YN-Question	YN-Question	
Check-Question	Check- Question		Declarative Question; Tag Question	Declarative Question; Tag-Question Check: Follow-Me; Understanding Check	
Choice Question	Choice Question		OR-Question/ OR-Clause		
			Open-Ended Question		
			Rhetorical Question		

Table G.13 — General-purpose information-seeking functions in DIT^{++}, AMI, Verbmobil, TRAINS, HCRC Map Task, SPAAC and MALTUS

DIT^{++}	AMI	Verbmobil	TRAINS	HCRC	SPAAC	MALTUS
Question	Elicit Inform; Elicit-Offer-or-Suggestion	Request Suggestion; Request Comment			Request Direct; Request Modal	Question
Set-Question			WHQ	Query-w		
Propositional Question			YN-Question	Query-yn		
Check-Question			Check	Check/ Align		
Choice Question				Query-w		

Table G.14 — General-purpose action-discussion functions in DIT^{++}, LIRICS, DAMSL, SWBD-DAMSL, MRDA and COCONUT

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	Reject	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 Suggestion	Accept	Accept	Accept	Accept
Decline Suggestion	Decline Suggestion	Reject	Reject; Dispreferred Response	Reject; Dispreferred Response	Reject
		Other Forward-Looking Functions	Other Forward-Looking Functions		Other Forward-Looking Functions
Request	Request				
Instruct	Instruct				
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	Reject	Reject
Suggestion	Suggestion			Suggestion	
		Other Backward-Looking Functions	Other Backward-Looking Functions		Other Backward-Looking Functions

Table G.15 — General-purpose action-discussion functions in DIT^{++}, AMI, Verbmobil, TRAINS, HCRC Map Task, SPAAC and MALTUS

DIT^{++}	AMI	Verbmobil	TRAINS	HCRC	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		Reject	Negative Answer
Address suggestion	Inform Positive/Partial/Uncertain					Other Answer
Accept Suggestion	Inform Positive	Accept	Accept		Accept	Positive Answer
Decline Suggestion	Inform Negative	Reject	Reject		Reject	Negative Answer
Request	Elicit-Offer-or-Suggestion; Inform	Request Commit	Request		Direct	Do
Instruct				Command: Instruct		
Address Offer	Inform Positive/Partial/Uncertain					Other Answer
Accept Offer	Inform Positive	Accept	Accept		Accept	Positive Answer
Decline Offer	Inform Negative	Reject	Reject		Reject	Negative Answer
Suggestion	Suggest	Suggest	Suggest		Suggest	Suggest

Bibliography

[1] Ahn, R. (2001) *Agents, Objects and Events: A computational approach to knowledge, observation and communication*. PhD Thesis, Eindhoven University of Technology, pp. 8-13

[2] Ahrenberg, L., N. Dahlbäck and A. Jönsson (1995) Codings schemas for Studies of Natural Language Dialogue. In: *Working Notes from the AAAI Spring Symposium, Stanford*, pp. 8-13

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

[4] Allen, J. and M. Core (1997) DAMSL: Dialogue Act Markup in Several Layers (Draft 2.1). Technical Report, Multiparty Discourse Group, Discourse Resource Initiative, September/October 1997

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

[6] Allwood J. (1992) On dialogue cohesion. *Gothenburg papers in Theoretical Linguistics* 65, Göteborgs Universitet, Department of Linguistics

[7] Allwood J. (1994) Obligations and options in dialogue. *THINK Quarterly* 3(1): 9–18

[8] Allwood J. (1997) A critical look at speech act theory. In: Õ. Dahl (ed.) *Logic, Pragmatics and Grammar*. Gothenburg: Department of Linguistics, Göteborgs Universitet, pp. 53-69

[9] Allwood, J. (2000). An activity-based approach to pragmatics. In: Harry Bunt and William Black, eds. *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics*. Amsterdam: Benjamins, pp. 47–80

[10] Allwood, J., Nivre, J. and Ahlsén E. (1993). On the Semantics and Pragmatics of Linguistic Feedback. *Journal of Semantics*, 9-1: 1–26

[11] Allwood, J., J. Nivre and E. Ahlsén (1994) Semantics and Spoken Language Manual for Coding Interaction Management. Report from the HSFR project *Semantik och talspråk*

[12] Augmented Multiparty Interaction Consortium (AMI) (2005) Guidelines for Dialogue Act and Addressee Annotation

[13] Austin, J.L.(1962) *How to do things with words*. Clarendon Press, Oxford, UK

[14] Bales, R.F. (1951) *Interaction process analysis: a method for the study of small groups*. Addison-Wesley, Cambridge

[15] Bunt, H. (1989) Information Dialogues as Communicative Action in Relation to User Modelling and Information Processing. In M. Taylor, D. Bouwhuis and F. Néel (eds.) *The Structure of Multimodal Dialogue*, Amsterdam: North-Holland Elsevier, pp. 47–74

[16] Bunt, H. (1994) Context and Dialogue Control. *THINK Quarterly* 3(1): 19-31

[17] Bunt, H. (1995) Dynamic Interpretation and Dialogue Theory. In M. Taylor, D. Bouwhuis and F. Néel (eds.) *The Structure of Multimodal Dialogue, Vol. 2*. Amsterdam: Benjamins, 139-166

[18] Bunt, H. (2000a) Dialogue pragmatics and context specification. In Harry Bunt and William Black (eds.) *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics*. Amsterdam: Benjamins, pp. 81-150

[19] Bunt, H. (2000b) Non-problems and social obligations in human-computer conversation. In Y. Wilks (ed.) *Proceedings of the 3rd International Workshop on Human-Computer Conversation*, Bellagio

[20] Bunt, H. (2005) A Framework for Dialogue Act Specification. *4th Joint ISO-SIGSEM Workshop on Semantic Annotation*, Tilburg, January 2005. http://sigsem.uvt.nl/wg/discussionnotes4.htm

[21] Bunt, H. (2006) Dimensions in Dialogue Act Annotation. In *Proceedings 5th International Conference on Language Resources and Systems* (*LREC 2006*), Genoa, pp. 919–924

[22] Bunt, H. (2007) The Semantics of Semantic Annotation. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation* (*PACLIC-21*), Seoul, pp. 13–28

[23] Bunt, H. (2009) The DIT++ taxonomy for functional dialogue markup. In D. Heylen, C. Pelachaud, R. Catizone and D. Traum (eds) *Proceedings of AAMAS 2009 Workshop "Towards a Standard Markup Language for Embodied Dialogue Acts* (*EDAML 2009*), Budapest, pp. 13–24

[24] Bunt, H. (2010) A methodology for designing semantic annotation languages. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources* (*ICGL 2010*), Hong Kong, pp. 29-45

[25] Bunt, H. (2011a) Multifunctionality in dialogue. *Computers, Speech and Language* 25 (2): 222–245

[26] Bunt, H. (2011b) The Semantics of Dialogue Acts. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, Oxford, pp. 1-13

[27] Bunt, H. (2012) Defining languages for semantic annotation with an abstract syntax and a formal semantics. Forthcoming in *Language Resources and Evaluation*

[28] Bunt, H. and L. Romary (2002). Towards multimodal content representation. In *Proceedings LREC 2002 Workshop on International Standards of Terminology and Language Resources*, Las Palmas, May 2002. Paris: ELRA, pp. 54–60

[29] Butterworth, B. (1980). Evidence from pauses in speech. In: *Language Production: Speech and Talk, Vol.1*, B.Butterworth (ed.). London: Academic Press, pp. 155–177

[30] 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. Berlin: Springer, pp. 107-120

[31] Carletta, J., A. Isard, S. Isard, J.Kowtko and G. Doherty-Sneddon (1996) HCRC dialogue structure coding manual. Technical Report HCRC/TR-82

[32] Clark, H.H. (1996) *Using language*. Cambridge University Press, Cambridge, UK

[33] Clark, H. and Fox Tree, J. (2002). Using "uh" and "um" in spontaneous speech. *Cognition*, 84: 73–111

[34] Clark, H. and Krych, M.A. (2004). Speaking while monitoring addressees for understanding. *Journal of Memory and Language* 50: 62–81

[35] Clark, D. and A. Popescu-Belis (2004) Multi-level Dialogue Act Tags. In *Proccedings of SIGDial 2004, 5th SIGDial Workshop on Discourse and Dialogue*, Cambridge, MA, 163–170

[36] Cooper, R. (2004) Information states, attitudes and dependent record types. In: L. Cavedon, P. Blackburn, N. Braisby and A. Shimojima (eds.) *Logic, Language and Computation*, vol. 3. Stanford: CSLI Publications, pp. 85-106

[37] Core, M., M. Ishizaki, J. Moore, C. Nakatani, N. Reithinger, D. Traum and S. Tutiya (1998) The Report of The Third Workshop of the Discourse Resource Initiative, Chiba University

[38] Core, M. and J. Allen (1997) Coding Dialogs with the DAMSL Annotation schema. *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, MA, pp. 28-35

[39] Di Eugenio, B., P.W. Jordan and L.Pylkkanen (1998) The COCONUT project: dialogue annotation manual. ISP Technical Report 98-1, University of Pittsburgh

[40] Duncan, S. and Fiske, D. (1977) *Face-to-face interaction: research, methods and theory*. New Jersey: Lawrence Erlbaum Associates

[41] Ekman, P. (1972) Universals and cultural differences in facial expressions of emotion. In J. Cole (ed.) *Nebraska Symposium on Motivation*, Lincoln, pp. 207–283

[42] Geertzen, Jeroen; Petukhova, Volha and Bunt, Harry (2007), A multidimensional approach to utterance segmentation and dialogue act classification. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, pp. 140–149

[43] Goffman, E. (1963) *Behavior in Public Places*. New York: Basic Books

[44] Grice, P. (1975) Logic and Conversation. In: P. Cole and J. Morgan (eds.) *Syntax and Semantics*, *vol. 3: Speech Acts*. New York: Seminar Press, pp. 43-58

[45] Hovy, E. and E. Maier (1992) Parsimonious or profligate: how many and which discourse structure relations? ISI research report, Information Sciences Institute, University of Southern California, Marina del Rey

[46] 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 Discourse. In *Proceedings of the First International Conference on Language Resources and Evaluation*, Madrid, pp. 731-735

[47] Ide, N. and H. Bunt (2010) Anatomy of Annotation Schemes: Mapping to GrAF. In *Proceedings of the 4th Linguistic Annotation Workshop* (*LAW IV*), Uppsala

[48] Ide, N. and L. Romary (2004) International Standard for a Linguistic Annotation Framework. *Natural Language Engineering* 10: 221–225

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

[50] Keizer, S., H. Bunt and V. Petukhova (2011) Multidimensional dialogue management. In A. van den Bosch and G. Bouma (eds.) *Interactive Multimodal Question Answering*. Berlin: Springer, pp. 57–86

[51] Kendon, A. (2004) *Gesture: visible action as utterance*. Cambridge, UK: Cambridge University Press.

[52] Klein, M. (1998) Supported Coding Schemes. EU Project LE4-8370 MATE Deliverable 1.1

[53] Klein, M. (1999) Standardization efforts on the level of dialogue acts in the MATE project. In *Proceedings of the ACL Workshop "Towards Standards and Tools for Discourse Tagging*", pp. 35-41

[54] Klein, M. and C. Soria(1998) Dialogue acts. In: M. Klein (ed.) Supported Coding Schemes. EU Project LE4-8370 MATE Deliverable 1.1

[55] Larsson, S. (1998) Coding schemas for dialog moves. Unpublished paper, available at http://www.ling.gu.se/sl

[56] Lascarides, A. and N. Asher (2007) Segmented Discourse representation Theory: Dynamic Semantics with Discourse Structure. In Harry Bunt and Reinhard Muskens (eds) *Computing Meaning, Volume 3*, Berlin: Springer, pp. 87–124

[57] Levinson, S. (1983) Pragmatics. *Cambridge University Press*, Cambridge, UK

[58] LIRICS D4.2 (2006) Preliminary set of semantic data categories. Deliverable D4.2 of eContent project LIRICS. Available from http://lirics.loria.fr

[59] LIRICS D4.3 (2007) Documented compilation of semantic data categories. Deliverable D4.3 of eContent project LIRICS. Available from http://lirics.loria.fr

[60] LIRICS D4.4 (2007) Multilingual test suites for semantically annotated data. Deliverable D4.4 of eContent project LIRICS. Available from http://lirics.loria.fr

[61] LIRICS D7.5 (2007) LIRICS Project Final Public Report. Deliverable D7.7 of eContent project LIRICS. Available from http://lirics.loria.fr

[62] Mann, W. and S. Thompson (1988) *Rhetorical Structure theory: Toward a functional theory of text organization*. MIT Press, Cambridge, MA

[63] Petukhova, V. (2011) Multidimensional Dailogue Modelling. Ph.D. Thesis, Tilburg University

[64] Petukhova, V. and H. Bunt (2009a) Dimensions in dialogue act annotation schemes: a survey. Technical Report TR 2009-003, Tilburg Centre for Cognition and Communication (TiCC), Tilburg University. Avaliable at http://www.tilburguniversity.edu/research/institutes-and-research-groups/ticc/research-programs/cc/technical-reports/

[65] Petukhova, V. and H. Bunt (2009b) The independence of dimensions in multidimensional dialogue act annotation. In *Proceedings Joint NAACL-HLT Conference*, Boulder, Colorado, pp. 197-200

[66] Petukhova, V. and H. Bunt (2009c) Who's next? Speaker selection mechanisms in multiparty dialogue. In *Proceedings of the 13th Workshop on the Semantics and Pragmatics of Dialogue* (*DiaHolmia*), Stockholm, pp 19-26

[67] Petukhova, V. and H. Bunt (2009d) Grounding by nodding. In *Proceedings of GESPIN 2009: Gestures and Speech in Interaction*, Poznán

[68] Petukhova, V. and H. Bunt (2010) Introducing communicative function qualifiers. In *Proceedings Second International Conference on Global Interoperability for Language Resources* (*ICGL 2010*), Hong Kong, pp. 123-131

[69] Petukhova, V. and H. Bunt (2012) The coding and annotation of multimodal dialogue acts. In *Proceedings 8th International Conference on Language Resources and Systems* (*LREC 2012*), Istanbul, pp 1293-1300

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

[71] Popescu-Belis, A. (2005) Dialogue Acts: One or More Dimensions? ISSCO Working Paper 62, ISSCO, Geneva. November 2005. http://www.issco.unige.ch/publicaitons/working-papers/papers/apb-issco-wp62b.pdf

[72] Popescu-Belis, A. (2008) Dimensionality of Dialogue act Tagsets: An Empirical Analysis of large Corpora. *Language Resources and Evaluation* 42(1): 99-107

[73] Prasad, R., N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi and B. Webber (2008) The Penn Discourse TreeBank 2.0. In *Proceedings 6th International Conference on Language Resources and Systems* (*LREC 2008*), Marrakech

[74] Romary, L., G. Budin, H.Bunt, N. Calzolari, T. Declerck, G. Francopoulo, L. Gillam (2007) LIRICS Project Final Public Report. Deliverable D7.7 of eContent project LIRICS. Available from http://lirics.loria.fr

[75] Searle, J.R. (1969) *Speech Acts*. Cambridge, UK: Cambridge University Press

[76] Sacks, H., E. Schegloff and G. Jefferson (1974) A simplest systematics for the organization of turn-taking for conversation. *Language* 53:696–735

[77] Sanders, T., W. Spooren and L. Noordman (1992) Toward a taxonomy of coherence relations. *Discourse Processes* 15 (1), 1–35

[78] Soria, C. and V. Pirrelli (2003) A multi-level annotation meta-schema for dialogue acts. In A. Zampolli, N. Calzolari and L. Cignoni (eds.) *Computational Linguistics in Pisa. Linguistica Computazionale, Special Issue*, XVIII-XIX, Pisa-Roma: IEPI, 865–900

[79] TEI (2010) *P5: Guidelines for Electronic Text Encoding and Interchange*. Text Encoding Initiative Consortium. Available at http://www.tei-c.org/Guidelines/P5/

[80] Traum, D. (1999) Speech Acts for Dialogue Agents. In: M. Wooldridge and A. Rao (eds.) *Foundations of rational agency*, Kluwer, Dordrecht, 169–201

[81] Traum, D. (2000) 20 Questions on Dialogue Act Taxonomies. *Journal of Semantics* 17(1): 7-30

[82] Traum, D. and E. Hinkelman (1992) Conversation acts in task-oriented spoken dialogue. *Computational Intelligence* 3 (8): 575-599

[83] Traum, D. and S. Larsson (2003) The Information State Approach to Dialogue Act Management. In J. van Kuppevelt and R. Smith (eds.) *Current and New Directions in Discourse and Dialogue*. Dordrecht: Kluwer

ICS 01.020
Price based on 104 pages

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

bsi.

...making excellence a habit.™