5.3.1 Reference number of working document: **ISO/TC 37/SC 4 NXXX**

Date: 2019-09-05

ISO/DIS 24617-2, Second Edition

ISO/TC 37/SC 4/WG 2

Secretariat: KATS

Language resources management —Semantic annotation framework (SemAF) — Part 2: Dialogue acts

Gestion de ressources linguistiques — Cadre d'annotation sémantique — Partie 2: Actes de dialogue

Document type: Draft International standard
Document subtype:
Document stage: DIS
Document language: E

Macintosh HD:Users:harrybunt:Desktop:ISO-DA Revision:DIS:DIS 24617-2_2019-09-11-HB.docx Basic template BASICEN3 2002-06-01

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*

— *Part 2: Dialogue acts*

— *Part 4: Semantic roles*

— *Part 6: Principles of semantic annotation*

— *Part 7: Spatial information*

— *Part 8: Semantic relations in discourse*

— *Part 9: Reference annotation framework*

— *Part 11: Measurable quantitative information*

— *Part 12: Quantification*

Introduction

Since its publication in 2012, ISO 24617-2 has been used in a number of annotation efforts as well as in the development of language-based interactive systems. These experiences have brought to light (1) that the standard allowed dialogue act annotations that are slightly inaccurate in some respects, (2) that some applications would benefit from the availability of mechanisms for customizing the set of concepts defined in the standard, and (3) that certain use cases require the representation of functional dialogue act information to be extended with semantic content information. This second edition seeks to remedy the noted inaccuracies, and to provide mechanisms (a) for customizing the set of defined concepts, and (b) for extending the information types in dialogue act annotations.

The improved accuracy of this second edition concerns the annotation of semantic dependence relations of dialogue acts and their scopes, and of rhetorical relations between dialogue acts. The mechanisms for extending and customizing the standard for a specific application concern most notably the annotation of information about the (domain-specific) semantic content of dialogue acts, the introduction of application-specific dialogue act types, the addition of communicative functions for fine-grained specification of feedback, and the annotation of speaker emotions.

This second edition is downward compatible with the original ISO 24617-2:2012 in the sense that every annotation made with the original version is a valid annotation according to the second edition. Existing annotations do not need to be revised in order to be compliant with the second edition.

Language resources management —Semantic annotation framework (SemAF) — Part 2: Dialogue acts

1 Scope

This international standard 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. The standard 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. The standard specifies data categories for reference sets of communicative functions and dimensions of dialogue analysis, and provides mechanisms for customizing these sets of concepts, extending them with application- or domain-specific concepts, or selecting relevant coherent subsets of them. Additionally, the standard provides guidelines for annotators and annotated examples. This standard 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.

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

ISO 24612:2011 *Language resource management – Linguistic annotation framework.*

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

ISO 24617-6:2015 *Language resources management —Semantic annotation framework (SemAF) — Part 6: Principles of semantic annotation.*

3 Terms and definitions

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

3.1
addressee
dialogue (3.6) **participant** (3.14) oriented to by the **sender** (3.20) in a manner to suggest that his **utterance**s (3.23) are particularly intended for this participant, and that some response is therefore anticipated from this participant, more so than from the other participants.

NOTE Source: Goffman (1981). 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".

[1] In this document, ‘he’, ‘him’ and ‘his’ are used in a generic sense, without implying any gender-related distintinctions.

3.2
allo-feedback act
feedback act (3.8) where the **sender** (3.20) elicits information about the **addressee**'s (3.1) processing of an **utterance** (3.23) that the sender contributed to the **dialogue** (3.6), 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 1. A: Now move up.
2. B: Slightly northeast you mean?
3. A: Slightly yeah
With utterance 3, A performs an allo-feedback act signalling that he thinks B understood utterance 1 correctly.

3.3
auto-feedback act
feedback act (3.8) where the **sender** (3.20) provides information about his own processing of an utterance contributed to the **dialogue** (3.6) by another **participant** (3.14)

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 A communicative function may be ‘qualified', i.e, one or more **qualifiers** (3.15) may be associated with it. For example, an answer may be qualified as ‘uncertain', and the acceptance of a request may be ‘conditional'. See Clause 10.3 for explanation and examples.

3.5
context
synonym for **information state** (3.13)

3.6
dialogue
exchange of **utterances** (3.23) between two or more persons or artificial conversational systems

3.7
dialogue act
communicative activity of a **dialogue** (3.6) **participant** (3.14), interpreted as having a certain **communicative function** (3.4) and **semantic content** (3.18)

NOTE A dialogue act may additionally also have certain **functional dependence relations** (3.10), **rhetorical relations** (3.17) and **feedback dependence relation**s (3.9) with other units in a **dialogue** (3.6).

3.8
dimension
class of **dialogue acts** (3.6) that are concerned with a particular aspect of communication, corresponding to a particular category of semantic content

NOTE Examples are (1) dialogue acts advancing the task or activity that motivates the dialogue (the ‘Task' dimension) ; (2) dialogue acts providing and eliciting feedback (the Auto- and Allo-Feedback dimensions); (3) dialogue acts for allocating the speaker role (the Turn Management dimension).

See clauses 5, 7, and 9 for discussion and more examples.

3.9
feedback act
dialogue act (3.6) which provides or elicits information about the **sender**'s (3.20) or the **addressee**'s (3.1) processing of something that was uttered in the dialogue

NOTE Two classes of feedback are distinguished in this standard: **auto-feedback acts** (3.2) and **auto-feedback acts** (3.3).

3.10
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.

3.11
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 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.

3.12
functional segment
minimal stretch of communicative behaviour that has one or more **communicative functions** (3.4)

NOTE 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.

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.

A consequence of this definition is that functional segments may be discontinuous, may overlap or be embedded, and may contain parts from more than one turn.

3.13
information state
context
the totality of a **dialogue** (3.5) **participant**'s (3.14) beliefs, assumptions, expectations, goals, preferences, hopes, and other attitudes that may influence the participant's interpretation and generation of communicative behaviour

3.14
participant
person or artificial agent involved in the exchange of **utterance**s (3.23)

3.15
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'. See Clause 10.3 for more examples.

3.16
reference segment
stretch of communicative behaviour that a **feedback dependence relation** (3.11) refers to and that is not a **functional segment** (3.11)

3.17
rhetorical relation
discourse relation
semantic or pragmatic relation between two **dialogue acts** (3.6) or their **semantic content**s (3.20)

NOTE 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 e.g. Hovy and Maier, 1992; Lascarides & Asher, 2007; 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.

EXAMPLE In the following example, the statement in the second utterance 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 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

3.18
semantic content
information, situation, action, event, or objects that a stretch of communicative behaviour refers to

3.19
semantic content category
semantic content type
kind of information, situation, action, event, or objects that form the **semantic content** (3.20) of a **dialogue act** (3.7)

EXAMPLES The various dimensions (see 3.8) defined in this standard 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 the category of information about the allocation of the speaker role, and so forth.

3.20
sender
dialogue (3.5) **participant** (3.14) who produces a **dialogue act** (3.7)

3.21
speaker
sender (3.20) of a **dialogue act** (3.6) in the form of speech, possibly combined with nonverbal communicative behaviour

NOTE A dialogue participant may say something while another participant occupies the **speaker role** (3.22), therefore the term ‘speaker' is not synonymous with ‘participant who occupies speaker role'.

3.22
speaker role
role occupied by a **dialogue** (3.5) **participant** (3.14) who has temporary control of the **dialogue** (3.6) and speaks for some period of time

NOTE Source: DAMSL Revised Manual.

3.23
turn unit
stretch of communicative activity produced by one **participant** (3.14) who occupies the **speaker role** (3.21), bounded by periods where another participant occupies the speaker role

3.23
utterance
anything said, written, keyed, gesticulated, or otherwise expressed

NOTE An utterance is mostly a part of what a sender contributes in a turn unit.

4 Purpose, justification, and use cases

4.1 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); of the Map Task studies of human-human dialogue (Anderson et al., 1991); of the Verbmobil speech translation project (Alexandersson et al., 1998), and of the ICSI-MRDA and AMI/AMIDA meeting browsing technology projects (Shriberg et al., 2004; Carletta et al., 2005). 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, 1994) 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 schemes into a comprehensive general-purpose annotation scheme. The annotation scheme of the NICT Kyoto corpus (Ohtake and Mizukami, 2017) is a variant of the DIT++ scheme.

In the European project LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems, Romary et al., 2007) a set of dialogue acts defined in the DIT++ taxonomy was selected and redefined in the form of data categories, following ISO standard 12620 for concept definitions. 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 standard.

The main purpose of this standard is to define a reference set of domain-independent basic concepts for dialogue act annotation, plus a formal language for representing such annotations. 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.

4.2 Use cases

The set of concepts defined in ISO 24617-2:2012 is based on the DIT++ taxonomy, which was originally developed to serve a double purpose: on the one hand for the articulate functional description of communicative activity in natural human dialogue, and on the other hand to provide a basis for the design of dialogue management modules in interactive systems. As part of the ISO Semantic Annotation Framework (SemAF), , the a strong focus in ISO 24617-2:2012 came to lie on its use for annotation. Still, the concepts of ISO 4617-2, like those of DIT++, have multiple use cases, which can be grouped into four types:

- UC1: Manual annotation of spoken, written, or multimodal human-human or human-computer dialogue.
- UC2: Automatic annotation of spoken, written, or multimodal human-human or human-computer dialogue starting from transcriptions or recordings of raw spoken or multimodal communicative user behaviour.
- UC3: Recognition of dialogue acts in spoken, written, nonverbal, or multimodal communicative user behaviour in human-computer interaction.
- UC4: Generation of dialogue acts by the dialogue manager component of a dialogue system.

The different use cases bring different requirements and desiderata:

- UC1: Manual dialogue act annotation has the advantage of producing annotations of the highest quality if performed by experts, but has the drawback of being very costly and only feasible for limited amounts of data. Expert manual annotation delivers the highest quality of annotations since expert human annotators are not only skilled in recognizing the relevant features of communicative behaviour, but also have a wealth of context information, general world knowledge, and common-sense reasoning abilities to infer speaker beliefs and intentions. Expert annotators are therefore able to assign fine-grained characterizations to segments of dialogue behaviour with high accuracy. In order to support manual annotation, the annotation scheme should therefore include fine-grained concepts with the level of detail that expert annotators can use.
- UC2: Automatic annotation of human-human dialogue, or of the user's contributions in a human-computer dialogue, typically lacks the general world knowledge and the skills of expert human annotators, and typically has access to context information only as far as represented in the dialogue history. Automatic annotation therefore in general cannot reliably characterize dialogue behaviour with the same level of detail as expert human annotation. To effectively support automatic annotation, the annotation scheme should therefore contain concepts that are more coarse-grained than those needed for manual annotation.
- UC3: The automatic recognition of dialogue acts in user behaviour in an interactive system is a very similar task as automatic dialogue act annotation, except that in an interactive system the semantic contents of dialogue acts play a prominent role, often determined by structural properties of the application domain. For a given application, it may be beneficial to have a tight coupling between communicative functions and semantic content, and to define application-specific functions for specific types of content. For effectively supporting this use case, it may be beneficial to extend the (application-independent) annotation scheme with application-specific concepts.
- UC4: The generation of dialogue acts in an interactive system concerns the decision how to continue a dialogue when it is the system's turn, and this is the main task of the system's dialogue manager component. This is typically a two-stage process, where the first stage is to decide on the communicative functions and semantic contents of one or more possible dialogue acts, and the second is to decide on an appropriate realization in linguistic, nonverbal, or multimodal form. In contrast with human dialogue participants, who may be somewhat vague or unspecific about their beliefs and intentions, a system's dialogue manager typically works with precise beliefs and goals, and generates, in the first of these two stages, dialogue acts with fine-grained communicative functions, possibly even more fine-grained than human participants commonly use. This happens for instance for feedback acts, since the system may report a processing problem with great accuracy. This calls for the annotation scheme to include very fine-grained functions, however not more fine-grained than humans can understand.

ISO 24617-2 was originally designed with the main purpose of supporting interoperable dialogue act annotation, i.e., the use cases UC1 and UC2. The use cases UC3 and UC4 have been found to be potentially of equally great interest, however. Some of the modifications described in this second edition are aimed at providing effective support for these use cases.

5 Modifications compared to First Edition (ISO 24617-2:2012)

5.1 Overview

The first edition of ISO 24617-2 was published in 2012. It has been applied in annotation efforts, including the development of the DialogBank (Bunt et al., 2019), and its concepts have been used for language understanding, dialogue management, and output generation in spoken and multimodal interactive systems.[2] These applications have brought to light certain inaccuracies and limitations of the standard which are addressed in this second edition.

The development of this standard has also contributed to the specification of a framework for defining other standards for semantic annotation. This framework was established as an ISO standard in 2016 (ISO 24617-6, Principles of semantic annotation). The second edition of ISO 24617-2 follows the requirements and recommendations of this framework in improving its specification.

5.2 Downward compatibility and optional elements

An important issue in revising an annotation scheme concerns the compatibility between annotations according to the original and the revised version. Unless the revision corrects serious errors in the original scheme, it is in general desirable that 'old' annotations are still valid according to the revised version, and do not require to be re-annotated (or converted). In other words, the revised standard should preferably be *'downward compatible'* with the original version.

Designing a revised version of the ISO 24617-2 standard in a downward compatible way is greatly facilitated by the *extensibility* of the original version, which means that it allows its stock of concepts to be extended with additional concepts. ISO 24617-2 is extensible in four respects:

1. Dimensions: Due to the orthogonality of the set of dimensions, additional dimensions may be introduced as long as they are orthogonal to the already existing dimensions and to each other.
2. Communicative functions: The taxonomy of communicative functions expresses semantic relations between functions: dominance relations express different degrees of specialization; and sister relations express mutually exclusivity of functions. Communicative functions may be added to the taxonomy as long as they respect these relations.
3. Qualifiers: Like dimensions, due to the orthogonality of the qualifier attributes and their values.
4. Rhetorical relations: The ISO standard does not specify a particular set of relations, but allows any such set to be plugged in.

The extensibility of ISO 24617-2 is in turn facilitated by the optionality of some of its components. Following the ISO Principles of semantic annotation, three types of optionality can be distinguished:[3]

- Type I, semantic optionality: a component that a certain type of annotation structure may contain, but does not have to. If it does, then this provides extra information. Example: the specification of a set of `other participants' for a dialogue act.
- Type II, syntactic optionality: a component that may be but does not need to be specified in annotation representations, since it has a default value in the abstract syntax. Example: the polarity in the annotation of an event by means of an <event> element in ISO-TimeML (default "positive").
- Type III, uninterpreted optionality: a component that may be specified in annotation representations but that does not encode anything in the abstract syntax. It thus has no semantic interpretation, but it may be useful during an annotation process, or for other purposes. Example: the indication of the part of speech of an event description in ISO-TimeML.

The document specifying ISO 24617-2:2012 mentions in a number of places an element of the standard as being 'optional', without making clear in what sense the optionality should be taken. In this second edition, the mentioning of 'optional' elements is made unambiguous in this respect.

---

[2] For applications in interactive systems see Malchanau (2019) and Malchanau et al. (2019).

[3] See Bunt et al. (2018) for details.

5.3 Inaccuracies in the first edition

5.3.1 Dependence relations

ISO 24617-2:2012 distinguishes three types of semantic relations in dialogue: functional dependence relations, feedback dependence relations, and rhetorical relations (a.k.a. discourse relations).

The definitions of functional and feedback dependence relations in ISO 24617-2:2012 were not optimally clear, which occasionally caused confusion for annotators. In this second edition the definitions have been made clearer, with explanations and illustrations of their use.

According to ISO 24617-2:2012, feedback acts have a feedback dependence relation to the preceding dialogue act(s) or dialogue segment(s) that they provide or elicit feedback about. The ISO 24617-2:2012 first edition does not make clear how one should choose between these two possible types of 'antecedent'. Upon closer inspection, it turns out that this issue should be analysed slightly differently, leading to the introduction of so-called 'reference segments' which allow more accurate annotations. This is explained in Clause 6.2.

5.3.2 Rhetorical relations

In view of the scope of this standard, the annotation of rhetorical relations was not considered to be a core issue in ISO 24617-2:2012, but rather an optional addition (as an instance of semantic optionality as defined in Clause 5.1), the more so since a separate standard was envisaged specifically for the annotation of rhetorical relations. However, experience shows that rhetorical relations in dialogue are often very important to take into account, especially in the use cases UC3 and UC4, mentioned in Clause 4.2.

The possibility of annotating rhetorical relations between dialogue acts in ISO 24617-2:2012 was limited in three respects: (1) no particular set of relations was specified; (2) there was no possibility to indicate the roles of the arguments; (3) it was not possible to distinguish between relations at the level of dialogue acts and relations at the level of their semantic contents. Since the publication of this standard, ISO 24617-8:2015 (DR-core) was published, which defines an annotation scheme for rhetorical relations. This second edition provides an option for annotating rhetorical relations in dialogue in a more fine-grained manner by importing concepts of the DR-core annotation scheme. This is discussed in Clause 6.3.

5.3.3 Sentiment and emotions

ISO 24617-2:2012 did not make a distinction between the annotation of sentiment and that of emotion. While the term 'sentiment' is sometimes used in a very broad sense, including emotions, it is more common to restrict 'sentiment' to a positive or negative feeling about something, often as shared by a group of people. This second edition follows that direction and redefines the meaning of the "sentiment" qualifier attribute accordingly. On the other hand, it is sometimes felt important also to be able to annotate emotions like happiness, regret, irritation, etc. To that end, this second edition opens the possibility to import elements from the W3C recommendation EmotionML. This is discussed in Clause 8.5 and Annex E.

5.4 Extension compared to the first edition

5.4.1 Dimensions

The Contact Management dimension, known from the DIT++ annotation scheme, has been found to be missing when applying the ISO 24617-2:2012 scheme in annotation efforts and has been added, along with a few communicative functions that are specific for this dimension.

The Task Management dimension, known from DAMSL, has also been considered for inclusion in the ISO scheme. In DAMSL, this dimension was introduced for "talking about the task" in task-oriented dialogues. This includes utterances that involve coordinating the activities of the two speakers (e.g. *"Are you keeping track of the time?"*), or discussing the status of the task (e.g. *"Shall we start?"* or *"Are we done?"*). A case where a great number of task management acts have been annotated as such is the DBOX corpus (Petukhova et al., 2014), where the task consists of participating in a quiz with the system in the role of quiz master; these dialogues consist of two parts: a first part in which the quiz master explains the rules of the game, and a second part in which the game is played. Rather than distinguishing two dimensions (Task and Task Management) in which utterances can have parallel functions, it seems more accurate to treat the two parts of these dialogues as concerning different tasks. This analysis is supported by the observations that (1) Task Management acts do not have other communicative functions than acts in the Task dimension, and (2)

functional segments that have a function in the Task dimension do not have a Task Management function, and vice versa. In view of these considerations, Task Management has not been added as a dimension to the ISO 24617-2:2012 annotation scheme.

5.4.2 Plug-ins

ISO 24617-2 has the explicit purpose of defining a reference set of domain-independent basic concepts for dialogue act annotation (see Clause 4.1). As a consequence, this standard focuses on generic dialogue act types which may be found in every kind of dialogue and every application context. This is on the one hand a strength of the standard, but on the other a limitation when it comes to optimal applicability in specific applications, in particular in the use cases UC3 and UC4. In order to overcome this limitation, a general mechanism is introduced in this second edition which allows the standard to be extended with application- or domain-specific concepts. This mechanism, the 'tripartite annotation scheme plug-in' with 'plug-in interface', is explained in Clause 9.

Annex E specifies how to use this mechanism for enriching DiAML annotations with:

- Semantic content information;
- Application-specific communicative functions;
- Additional communicative functions for annotating casual talk, e.g. in the initial phase of a task-related dialogue;
- Fine-grained communicative functions for feedback, especially for generation by a dialogue system (use case UC4);
- Rhetorical relations with argument roles;
- Emotions.

6 Basic concepts and metamodel

6.1 Dalogue acts

The term 'dialogue act' is often used rather loosely in the sense of speech act used in dialogue. Indeed, the idea of interpreting communicative behaviour in terms of actions, such as questions, promises, and requests goes back to speech act theory (Austin, 1962; Searle, 1969). But where speech act theory is primarily an action-based approach to meaning within the philosophy of language, dialogue act theory is an empirically-based approach to the computational modeling of 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 updates his information state with the information expressed in the semantic content when he understands the dialogue act.

This approach to the definition of communicative functions is strictly semantic, in contrast to approaches based on linguistic form. For example, the behaviour of a speaker who repeats something that was said by someone else may be characterised as a 'repetition' (which is a communicative function in some annotation schemes); however, this only says something about the *form* of the behaviour compared to the repeated behaviour, not about its function. A repetition often has a feedback function, as in (1.2), but it can also have other functions, as in (1.4), where it is used as a confirmation in response to a check question:

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

2. C: Seven-fifteen and eight-thirty
3. C: And that's on Sunday too
4. S: And that's on Sunday too

A form-related requirement for introducing a communicative function is however that there are observable features of communicative (linguistic and/or nonverbal) behaviour which are indicative for that function in the context in which the behaviour occurs. This requirement puts all communicative functions on an empirical basis.

Dialogue act annotation is the marking up of stretches of dialogue with information about the dialogue acts they contain. 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 can be quite long and complex, and are therefore not the most useful units of behaviour to assign 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. Clause 7.3 discusses 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 the 'addressee(s)'. 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 additionally provides information about the category 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 advances the task which underlies the dialogue, or discusses how to perform the task, or concerns the communication process. The DIT++ scheme distinguishes a number of subcategories of communication-related information, such as feedback information, turn allocation information, and speech management information. The ISO 24617-2 scheme inherits the DIT++ categories of semantic content, also called 'dimensions'; these are discussed in Clause 7.

Example (2) illustrates the use of the key attributes of a dialogue act in the DiAML-XML annotation of a task-related yes-no question addressed by speaker 'a' to addressee 'b', expressed by the functional segment 'm1':

(2) <dialogueAct xml:id="da1" target="#m1" sender="#a" addressee="#b" dimension="task"
communicativeFunction="propositionalQuestion"/>

6.2 Dependence relations

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 that is being answered, and also for the acceptance or rejection of offers, suggestions, requests, and apologies. This is illustrated in example (3), where the meaning of the answer in turn 3 depends on whether it is an answer to the question in turn 1 or to the one in turn 2.

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

As an answer to the question in 1, A's answer says that nobody else is expected to come than the people that are mentioned, but as an answer to the question in 2 it leaves open the possibility that other people will come, who are not members of 'the project'.

This kind of semantic dependence, which is due to the responsive character of some communicative functions, is called a *functional dependence relation*. Marking up this relation between a dialogue act with a responsive communicative function and its 'antecedent' dialogue acts allows the annotation to not just indicate e.g. that an utterance has the function of an answer, but also to indicate *to which question* it is an answer, as illustrated in (4). Clause 8.3.4 lists the responsive communicative functions defined in this standard.

(4) a. B: Which of the project members do you think will be there?
A: I'm expecting Jan, Alex, Claudia, and David, and maybe Olga and Andrei.

b. <dialogueAct xml:id="da1" target="#m1" sender="#b" addressee="#a" dimension="task" communicativeFunction="setQuestion"/>
<dialogueAct xml:id="da2" target="#m2" sender="#a" addressee="#b" dimension="task" communicativeFunction="answer" functionalDependence="#da1"/>

The property of 'responsiveness' is closely related to what in the literature is called 'backward-looking'; for example, in the DAMSL annotation scheme the communicative functions are divided over two categories: forward-looking and backward-looking. Backward-looking functions are defined as those functions that indicate how the current utterance relates to the previous discourse. These include not only answers and other dialogue acts whose semantic content is co-determined by antecedent dialogue acts, but also feedback acts and other acts concerned with speech editing.

Positive and negative feedback-providing acts depend for their interpretation also on 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. This is illustrated by the examples in (5).

(5) 1. A: The flight on Tuesday would suit me really well.
B: Okay.

2. A: The flight on Tuesday would suit me really well.
B: On Tuesday?

In the first example B indicates that he has correctly understood A's remark; in the second he checks whether he heard (or remembers) correctly what A said. This relation between a positive or negative feedback act ant its 'antecedent' is called a *feedback dependence relation.*

A feedback dependence relation indicates one or more preceding *dialogue acts* if the feedback concerns high-level processing, such as understanding, and it indicates a *dialogue segment* in the case of low-level processing, such as hearing what was said. In the latter case, ISO 24617-2:2012 stipulates that the feedback dependence relation should refer to the smallest functional segment containing the segment that the feedback act is about. This way of annotating feedback dependence relations is not quite accurate, since feedback about a stretch of communicative behaviour smaller than a functional segment is not about the entire segment. For example, negative feedback that signals a problem in hearing certain words may imply positive feedback about the rest of the segment. Similarly for feedback-eliciting acts and for dialogue acts in the Own Communication Management (OCM) dimension or in the Partner Communication Management (PCM) dimension. In particular, Self-Corrections and Partner Corrections frequently refer to a single word or phrase which does not form a functional segment. To make more accurate annotation possible, this second edition introduces a 'reference segment' as being a stretch of communicative behaviour that is the object of a feedback dependence relation and that is not a functional segment.

6.3 Rhetorical relations

Dialogue acts may also be semantically and pragmatically related through other relations, known as *rhetorical relations* or *discourse relations*, as in the examples shown in (5).

(5) 1. A: It ties you on in terms of the technology and the complexity that you want
2. A: like for example voice recognition
3. A: because you might need to power a microphone and other things
4. A: so that's one constraint there

In this example[4] we see a sequence of four functional segments contributed by the same participant. The segments in lines 2-4 are all related to the dialogue act expressed in the first segment. Segment 2 is related to

[4] From the AMI corpus, see http://corpus.amiproject.org.

the initial statement through an *Exemplification* relation, segment 3 through a *Cause* relation, and segment 4 through a *Restatement* relation.

A wide diversity of sets of rhetorical relations has been proposed (see e.g. Hobbs, 1979; Mann and Thompson, 1988; Lascarides and Asher, 1991, Hovy and Maier, 1993; Prasad et al., 2008; Sanders et al., 1992), which has inspired a great deal of discussion, comparisons, and attempts to specify mappings between various sets (Benamara and Taboada, 2015; Bunt and Prasad, 2016; Schefler and Stede, 2016; Demberg et al., 2017; Sanders et al., 2018). In view of this situation, the first edition of this standard did not propose the use of any specific set of relations, but only provided a conceptual category for which a set of relations may be specified. In DiAML-XML this provision plays out in the definition of an element called '<rhetoricalLink>' which has attributes referring to two dialogue acts and an attribute for whose value a rhetorical relation can be specified. Example (6) illustrates the use of this provision for indicating a causal relation between two dialogue acts (da2 and da3).

(6a) A: Have you seen Pete today?
B: He didn't come in; he has the flu.

(6b)
```
<dialogueAct xml:id="da1" target="#fs1" sender="#a" addressee="#b"
      dimension="task" communicativeFunction="propositionalQuestion"/>
<dialogueAct xml:id="da2" target="#fs2" sender="#b" addressee="#a"
      dimension="task" communicativeFunction="answer" functionalDependence="#da1"/>
<dialogueAct xml:id="da3" target="#fs3" sender="#b" addressee="#a"
      dimension="task" communicativeFunction="inform"/>
<rhetoricalLink dact="#da3" rhetoAntecedent="#da2" rhetoRel="cause"/>
```

ISO standard 24617-8:2015 (DR-core) for annotating rhetorical relations defines a set of 'core' rhetorical relations. These relations have been used in DiAML-XML annotations as values of the @rhetoRel attribute in several annotation efforts (see e.g. Petukhova et al., 2014 and Bunt et al., 2019). The <rhetoricalLink> element was found to be rather coarse-grained, however, for the two limitations already mentioned in Clause 5.4: (1)) it is not possible to indicate the roles of the arguments; and (2) it is not possible to distinguish between relations at the level of dialogue acts and relations at the level of their semantic contents. For example, the annotation in (6b) expresses the existence of a causal relation involving two dialogue acts, but does not say which is the reason and which is the result, nor if and how their semantic contents are involved.

The distinction between using a rhetorical relation to relate the semantic content of two dialogue acts and to relate one dialogue act as a whole to another or to the semantic content of another act, is often described as using a 'semantic' and a 'pragmatic' variant of the relation. The distinction is illustrated by (6a) and (7). Where in (6a) having the flu is the cause of not coming i('semantic cause'), in (7) beating his wife is not the cause of Jim being an idiot, but is a reason for the speaker *to say* that Jim is an idiot, so the causal relation is between the semantic content of the second dialogue act and the performance of the first ('pragmatic cause'),.

(7) Jim is an idiot. He beats his wife.

Using devices from DR-core, the bottom line of (6b) can be replaced by the more fine-grained representation in (6c), which specifies argument roles.

(6c)
```
<drLink arg1="#da2" arg2="#da3" rel="cause">
      <argRole arg="#da2" role="result"/>"
      <argRole arg="#da3" role="reason"/>"
</drLink>
```

The involvement of the semantic contents of the dialogue acts could be indiicated in annotations by introducing an attribute (say '@relVariant') with values like 'semantic' and 'pragmatic', but such annotation would not be semantically interpretable since no semantic content is available in the DiAML semantics. Adding content information to DiAML annotations can be achieved by means of 'plug-ins', which are introduced in Clause 10.2. Annex E describes alternative possible plug-ins for semantic content, and a plug-in for rhetorical relations on top of a content plug-in.

In this second edition of ISO 24617-2 the constructs <drLink> and <argRole> are introduced in the DiAML-XML concrete syntax, and the conceptual structures that they encode are added to the DiAML abstract syntax with their semantics. Semantically the <drLink> structure is similar to the <rhetoricalLink> element; the

interpretation consists of an update operation that inserts the semantic relation in an addressee's information state, adding a specification of the argument roles. Both link structures form an optional part of the annotation scheme (Type I, semantic optionality): they can be omitted from annotation structures without affecting their well-formedness; their presence just gives additional information. They are both to be used in combination with a plug-in that specifies a set of rhetorical relations and, for using the <drLInk> structure, also a set of argument roles. While the <drLink> structure allows more fine-grained annotations than <rhetoricalLink> structures, the latter possibility remains an option that can be used in situations where fine-grained annotation of rhetorical relations is not considered important. Like the first edition, the second edition of ISO 25617-2 does not define any particular set of rhetorical relations. Annex E specifies a plug-in with a set of relations based on DR-core and indicates how other sets of rhetorical relations can be introduced.

6.4 Qualifiers

The examples in (8) illustrate another phenomenon that is frequently found in dialogue, namely that speakers may are uncertain about the information they provide, as in B's utterance in (8a), or about their commitment to the performance of an action, as in (8b1). Speakers may also express a certain sentiment about the information or event that is being discussed, as in (8.b3), or express a reservation in the form of a condition, as in (8.b2), where an offer is conditionally accepted:

(8) a. A: Do you know what time the meeting starts?
B: At 4 p.m. I think.

b. A: Would you like to have some coffee?
1. B: Maybe later.
2. B: Only if you have it ready.
3. B: Yes please!

For the annotation of conditions, uncertainty, and sentiment, this standard makes use of so-called *qualifiers*, - see Clause 8.5.

The use of qualifiers is 'syntactically optional' in DiAML; if no qualifier values are specified in DiAML representations, then default values are assumed in the underlying abstract annotation structures.

6.5 Metamodel

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:

- sender and addressee;
- participant in another role, or 'side-participant' (not necessarily present);
- markable: functional segments and reference segments;
- dialogue act, communicative function, and dimension (semantic content category);
- communicative function qualifier (syntactically optional);
- functional dependence relation and feedback dependence relation (only for certain communicative functions);
- rhetorical relation (semantically optional)

7 Multifunctionality, segmentation and multidimensionality

7.1 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, 1994). Communication is thus a 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:

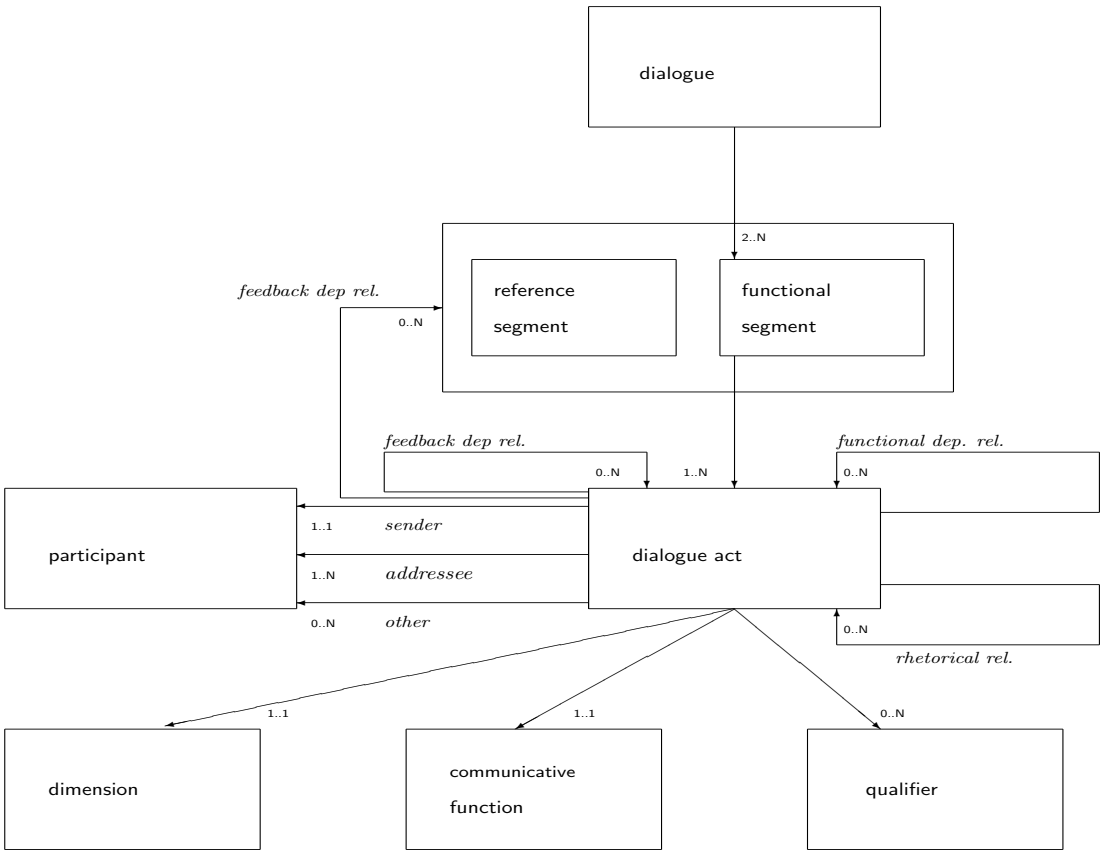

Figure 1. Metamodel for dialogue act annotation.

(9) 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. A's utterance in (9) is a sequence of 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 Clause 7.3), 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:

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

A's utterance in (10.3) has the function of thanking, and will most likely be taken to imply that A has understood and accepted the information provided in (10.2) - i.e., as having a positive feedback function. 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. But *"Thank you"* does not *always* express positive feedback; it may also be used as a polite way to end an unsuccessful dialogue.

The implication relation between thanking and positive feedback is different from that between a confirmation and a positive answer, where the relation is one of *entailment*, i.e. an implication which is logically valid. Entailment relations occur when one communicative function is a special case of another. Entailed communicative functions should not be considered as instances of multifunctionality.

Turn-initial hesitations are a source of multifunctionality, as the following dialogue fragment illustrates:

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

In the first turn of (11), speaker A asks a question to B (Bert) and assigns the turn to B. In the second turn B performs a stalling act in order to buy some time for deciding what to say. The fact that he starts speaking before having decided what to say, indicates that he accepts the turn. So the segment *"Uh,.. well,..."* has both a stalling function and a turn-accepting function. Note that A's utterance is also multifunctional: it asks a question and assigns the turn to B.

While stalling and turn-accepting are two functions that often go together, many other combinations of communicative functions cannot co-occur. For example, a functional segment cannot have both a turn-accepting function and a turn-giving function, or a request-accepting and a request-refusing function. The structure of the ISO 24617-2 annotation schema reflects the possible multifunctionality of functional segments. For example, the management of turn taking can be described by means of six mutually exclusive communicative functions: Turn Accept, Turn Take, Turn Grab, Turn Keep, Turn Release and Turn Assign. This cluster of functions forms the Turn Management dimension.

7.2 Multidimensionality and dimensions

Dialogue act annotation schemes often make use of a clustering of communicative functions that is determined by the intuitive relatedness of certain functions or by similarities in the way they are expressed. 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 annotation schema (Core and Allen, 1997) is organised 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 clusters of communicative functions; the tags in the other layers are concerned with different 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; and (6) rhetorical roles.

Bunt (2005; 2006) proposes to structure a dialogue act tag set into multiple dimensions by basing the notion of 'dimension' on the observation that dialogue participants share information not only about the task they pursue 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, in addition to those for advancing a certain task or performing a certain activity. Each of these types of activity is concerned with a different category of information. This standard uses the term 'dimension' to refer to these categories and the corresponding activities.. This leads to dimensions such as *feedback, turn management, time management*, and *contact management*, besides the dimension formed by the task that motivates the dialogue. Clause 8.2 describes the dimensions defined in this standard.

Not every grouping of communicative functions qualifies as a dimension in the sense of this standard. For example, the group of information-giving acts (statements, warnings, answers, corrections, and so on) does not form a dimension since they are not concerned with a particular category of information: information can be given about *any* aspect of the dialogue, such as the underlying task, the understanding of an utterance,,a change in topic, or the need to have a break. 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. 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 in the dimension corresponding to that category. These functions are discussed further in Clause 8.3.2

In contrast with the general-purpose functions, some 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. This class of functions is discussed in Clause 8.3.3.

7.3 Segmentation

The multifunctionality of dialogue behaviour is optimally accounted for when communicative functions are assigned to all those segments of behaviour that expresses a dialogue act. Such segments are called *functional segments*, defined more precisely as a *minimal stretch of communicative behaviour that has a communicative function* (possibly more than one, see also definition 3.12 in Clause 3). The condition of being 'minimal' ensures that functional segments do not include material that does not contribute to the expression of its communicative function(s). A consequence of this definition is that functional segments may be discontinuous, may overlap, may be embedded in another functional segment, may spread over multiple turns, and may contain parts contributed by different speakers.

For example, consider the segmentation of the turn unit contributed by S in (12).

(12) 1. U: What time is the first train to the airport on Sunday morning?
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 to U's question 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 recognition of what U said; and (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; see (13).

(13)

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

Example (14) 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.

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

B: Certainly. There's a Lufthansa flight in the morning leaving at 08:15,
A: yes,
B: and a KLM flight at 08:50,
A: yes,
B: and a Garuda flight at 10:30,
A: yes,
B: ...

Segments of verbal behaviour have a natural delineation in terms of their constituent words. Nonverbal forms of communicative behaviour, such as hand gestures, head gestures, and facial expressions, do have their own morphology (see e.g. Kendon, 2004), which can be used to identify nonverbal functional segments. The definition of a functional segment as “minimal stretch of communicative behaviour that has a communicative function" thus applies not only to verbal behaviour but also to nonverbal and multimodal communicative behaviour (see also Petukhova and Bunt, 2012).

In multimodal dialogue, participants combine the use of different modalities including speech and nonverbal elements 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 a head gestures. See Annex B for examples.

8 Specification of the annotation scheme

8.1 Overview

Following the methodological standard ISO 24617-6 (Principles of semantic annotation), this specification consists of four parts:

1. A metamodel, providing a schematic overview of the concepts that may occur in annotations, and the relations between them.
2. An abstract syntax, providing a formal specification of the inventory of the concepts from which annotations are built up and of the possible ways of combining them, using set-theoretical operations, to form conceptual structures called ‘annotation structures’.
3. A concrete syntax, defining a representation format for annotation structures.
4. A semantics, defining an interpretation of annotation structures (and their representations).

The metamodel was introduced in Clause 6 and is visualized in Figure 1. The clauses 8.2 - 8.5 describe in some detail the most important ingredients of the metamodel: the dimensions (8.2), the communicative functions (8.3), the qualifiers (8.4) and the relations (8.5). Clause 9 specifies the abstract and concrete syntax of annotations and their semantics. Together, these components define the DIAML markup language. The concrete syntax specifies a reference representation format, using XML (‘DiAML-XML’); other representation formats have also been used in applications of the annotation scheme, and are equally valid as long as they define a complete and unambiguous representation system.[5] From a syntactic point of view, DiAML-XML is just a compact form of XML; its importance is that it defines a class of XML expressions that have a formal semantics.

8.2 Dimensions

Not every grouping of related communicative functions makes a dimension in the sense of this standard, as noted in Clause 7.2. In order to identify proper dimensions for dialogue act annotation, Petukhova and Bunt (2009a,b) formulate a number of criteria, the following of which are the most important.

1. Empirical validity: each dimension corresponds to a distinct class of well-studied communicative activities that dialogue participants perform, such as turn taking, contact management, and feedback.

[5] A representation format is complete and unambigous for a given abstract syntax if it defines a representation for each annotation structure defined by the abstract syntax, and if each representation unambiguously represents one such annotation structure. For discussion and applications see Bunt (2010); Pustejovsky et al. (2017). For alternative, tabular representation formats see Bunt et al. (2019).

2. Orthogonality, or *independence:* each dimension can be addressed by dialogue acts independent of other dimensions. More precisely, for every dimension *D* there should be forms of communicative behaviour which express a dialogue act that is concerned with the activities and the kind of information characteristic for *D*, without necessarily also expressing a dialogue act in one of the other dimensions.
3. Recognisability: each dimension should be recognizable by human ananotators as well as by automatic understanding and annotation systems.

Petukhova and Bunt (2009a) survey the literature, analyse 18 existing annotation schemes, and apply statistical and machine-learning tests in order to verify these requirements for a range of proposed dimensions. The tests concern co-occurrence relations among dialogue acts and dimensions, tests of orthogonality, measures of semantic relatedness, and data on human and machine recognition of dimensions. This study, which is summarized in Annex G, shows that the following ten candidates qualify as dimensions for ISO 24617-2.

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.

2. ***Auto-Feedback and Allo-Feedback***

The term ‘feedback' in dialogue is most often used to refer to the signalling of understanding, perception, and evaluation of what was said. Feedback is an essential part 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 have been studied extensively for their linguistic properties, their non-verbal expression, and their durational, temporal and prosodic properties 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, interpretation (understanding), evaluation*, and *execution*.

Dialogue participants do not only discuss and report on their own processing of dialogue utterances, 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). The terms ‘auto-feedback' and ‘allo-feedback', borrowed from DIT++, are used here to distinguish between the discussion of the speaker’s own processing own of what was said and that of the addressee’s processing.[6] Examples of utterances expressing allo-feedback acts are: *“Is this clear enough?", “That's what I meant"* and *“You got me wrong".*

3. ***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.[7] Accordingly, the decision to take the next turn or to offer the next turn to the partner(-s) depends on the speaker's needs, motivations, and beliefs, and on the rights and obligations in a conversational situation.

In dialogues with two or three participants, normally only one participant is speaking at any given moment, while the other participants express their involvement through backchannels (like *”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

[6] The terms ‘*allo-feedback'* and ‘*auto-feedback'* (Bunt, 1995) have their origin in the Greek words ‘allos' and ‘autos', meaning ‘other' and ‘self', respectively, referring to the participant whose processing the speaker is considering.

(Campbell, 2008), and the conversation may effectively split up into sub-conversations involving subgroups of participants.

4. ***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 prolongued, when the speaker performs a *pausing* act to suspend the dialogue for a while.

5. ***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.

6. ***Social Obligations Management***

Participating in a dialogue is a social activity, where one is supposed to act in accordance with norms and conventions for social behaviour. Dialogue participants have ethical tasks and obligations, and perform dialogue acts to fulfill 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 management (SOM) acts are often not just 'social'; they are also used for making the dialogue more transparent. 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.

7. ***Own and Partner 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' 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.

Partner Communication Management 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, believing that a speaking error was made (*correct-misspeaking*).

8. ***Contact Management***

Contact Management acts are concerned with establishing and monitoring contact between speaker and addressee(s). Such acts are particularly important in situations where the participants do not have visual contact, such as in telephone conversations or internet chats. "Hello", "Allô", and "Moshi moshi" are examples in English, French, and Japanese.

8.3 Communicative functions

8.3.1 Overview

The various dialogue act annotation schemes that have been proposed share a number of communicative functions which are important in almost any type of dialogue Traum and Hinkelman (1992) used the term 'core dialogue acts' to refer to those acts that are familiar from traditional speech act theory. These 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,*

[7] The corresponding behavioural unit is what in this standard is called a 'turn unit'; see definition 3.22 in Clause 3.

congratulate). In this standard the terms ‘core dialogue act' and ‘core communicative function' are used to refer to the types of dialogue acts and their communicative functions that are most commonly found in dialogue and that are not specifically related to particular task domains; data categories specifying names and definitions of these communicative functions are part of this standard. These include the most common commissive, directive, and reportative acts known from speech act theory and some of the expressive ones, plus a set of other 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 similar criteria as the choice of dimensions. The criterion of *empirical validity* requires that for every communicative function there exist linguistic or nonverbal means which can be used to indicate this function. A good *coverage* is also an empirical requirement For example, a consequence of the occurrence of what conversational analysts have called ‘adjacency pairs' is 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 that contains a tag for encoding thankings should also contain one for downplayers.

The criterion of *recognisability* reinforces that of empirical validity, and moreover requires that every communicative function has a precise definition, which clearly distinguishes it from other functions. In particular, the semantic approach taken in this standard requires precise definitions in terms of information state updates.

It is moreover 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 a given dimension are either mutually exclusive or one is a specialization of the other. A scheme with 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 (15) B's utterance forms an information-providing act in response to A's check question. Given the set of information-providing functions shown in Figure 2, this means 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 typically 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*.

(15) A: And that's the first flight tomorrow, right?
B: That's right.

A multidimensional annotation scheme with orthogonal dimensions and semantically connected sets of communicative functions allows annotators who 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.

This standard includes only rather small numbers of domain-independent core communicative functions:

- general-purpose functions:
  - 6 information-seeking functions;
  - 7 information-providing functions;
  - 8 commissive functions;
  - 6 directive functions.
- dimension-specific functions:
  - 2 auto-feedback functions;
  - 3 allo-feedback functions;
  - 2 time management functions;
  - 6 turn management functions;

- 3 discourse structuring functions;
- 3 own communication management functions;
- 2 partner communication management functions;
- 2 contact management functions;
- 13 social obligation management functions.

The possibility of adding other communicative functions is considered in Clause 10.

8.3.2 General-purpose functions

The general-purpose functions defined in this standard are concerned with obtaining or providing certain information or discussing (communicative or other) actions. 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); see Figure 2.

The functions in the information-seeking class are questions of various kinds. Many annotation 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 standard a distinction is made between *propositional questions*, where the speaker wants to know the truth of a given proposition (also known as 'yes/no questions'); *check questions*, which are propositional questions where the speaker expects the answer to be positive; *set questions*, where the speaker wants to know which elements of a given set of entities have a certain property (also known as 'WH-questions'); and *choice questions* (also known as 'multiple-choice questions', 'menu questions', or 'alternatives-questions'), where the speaker wants to know which one of a list of alternatives applies. Special question types are so-called 'test questions' (a.k.a. 'exam questions'), where the speaker wants to know whether the addressee knows the answer, which the speaker knows. Rhetorical questions only look like questions, and are not included in the class of question types.

The most obvious case of an *information-providing* function is the *Inform*, which in various annotation schemes also goes by the names *statement* and *assertion*, and which is the function of a dialogue act where the speaker has the aim to bring certain information to the addressee's attention. More specific functions are *Agreement* and *Disagreement*, where the speaker believes that the addressee agrees or disagrees, respectively, 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 is that in the case of a promise this commitment is unconditional, whereas in the case of an offer the commitment occurs only if the addressee accepts the offer.

The prototypical case of a *directive* function is the *Instruct*, where the speaker puts pressure on the addressee to perform a certain action. The *Request* is a conditional directive, which puts pressure on an addressee to perform the requested action if he agrees 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. 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*.

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 structure as a decision tree -- see Annex A.

The general-purpose functions have as defining characteristic that they can be used to build a dialogue act in any dimension by combining the function with a semantic content of the category of that dimension, The definitions of the core general-purpose functions are provided in Annex F in the form of ISO data categories.

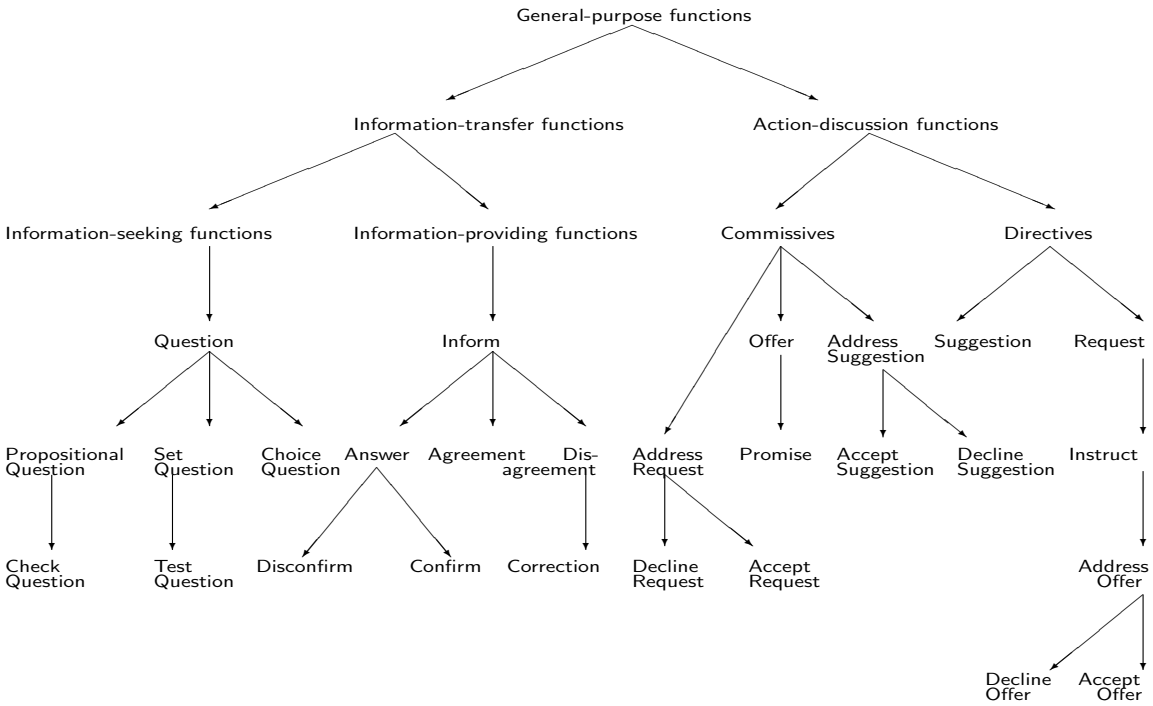

Figure 2. General-purpose functions.

8.3.3 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, like 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 what he apologises for.

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

1. *Function in the Task dimension*

Dimension-specific communicative functions for the Task dimension are specific for communication about a particular task domain. For example, specialised 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, the present standard does not include any such functions. However, a mechanism to introduce domain-specific communicative functions is described in Annex E.

2. *Functions in the feedback dimensions*

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 can be divided 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 processing 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 one or more previous utterances.

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 the present standard does not include feedback functions for specific levels of processing, but only the more general functions expressing positive and negative feedback. However, a mechanism for importing the fine-grained feedback functions from DIT++ is described in Annex E.

3. *Functions in the Turn Management dimension*

The turn management functions in this standard 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 end 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.

4. *Functions in the Time Management dimension*

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, for instance to 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 participation in the dialogue but intends to resume shortly. This is called '*pausing*'.

5. *Functions for 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?"*

6. *Functions in the Own and Partner Communication Management dimensions*

Own communication management (OCM) acts occur when a speaker edits his own speech, and most commonly take the form of self-corrections (also called 'repairs') and retractions. The most common forms of Partner Communication Management (PCM) acts are the correction of a partner's speaking error and the completion of an utterance which the partner is struggling to complete.

7. *Functions for social obligations management*

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

8. *Functions for contact management*

In certain circumstances it may happen that one is not sure that the person(s) one wanted to talk to is/are actually present and available for communicating with. *"Anybody home?"* and *"Are you still there?"* are much-used expressions to check this, and "*Hello*" when picking up the phone is similarly an indication of one's presence and availability.

8.3.4 Responsive communicative functions

As mentioned in Clause 5, some dialogue acts by the very nature of their communicative function respond to previous dialogue acts, and are for the determination of their meaning dependent on the semantic content of these previous dialogue acts. The following communicative functions defined in this standard are responsive:

Answer, Confirm, Disconfirm, Correction, Agreement, Disagreement
Accept Offer, Decline Offer, Address Offer
Accept Request, Decline Request, Address Request
Accept Suggestion, Decline Suggestion, Address Suggestion

8.4 Functional and feedback dependences

Dialogue acts with a responsive communicative function depend on a previous dialogue act for their semantic content and thus have a 'functional dependence relation', as discussed in Clause 6.2. A feedback dependence relation connects a feedback act, an OCM act, and a PCM act to the stretches of preceding communicative behaviour whose processing they are concerned with. The latter characterization leaves open the possibility that a feedback act with a responsive communicative function, such as an answer to a feedback question.

Dialogue acts with a feedback-specific communicative function are expressed by utterances like *"Okay", "Uh-huh", "Yes"* (positive auto-feedback), *"No no no*" (negative allo-feedback), "*Right*" (positive allo-feedback), *"Huh?", "What?"* (negative auto-feedback), and "*Okay*?" (feedback elicitation) - typically with a particular intonation and accompanying facial expressions. Such utterances by themselves contain little semantic information. The same is true for OCM- and PCM-acts with a dimension-specific communicative function. For their semantic content, these acts depend on the stretches of communicative behaviour that they provide or elicit information about. Feedback dependence relations enable this.

Feedback acts with a general-purpose communicative function are either responsive or non-responsive. For a responsive one the semantic content is determined by the semantics of the functional segment that expresses it combined with the semantics of the functional segment(s) of the dialogue act(s) that they functionally depend on; for non-responsive ones the semantic content is locally determined by the functional segment by which they are expressed. Feedback acts with a general-purpose communicative function therefore do not have feedback dependence relations. The same goes for OCM- and PCM-acts. In conclusion, a dialogue act either has no

dependence relation at all or it has either a functional or a feedback dependence relation, as determined by its communicative function.

8.5 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, emotions and attitudes. For example, it is customary to distinguish only two possible responses to an offer: acceptance and refusal. An offer may however be responded to in less clear-cut ways, and can for instance be accepted conditionally, as in (16.2a), or declined with uncertainty, as in (16.2b).

(16) 1. A: Can I offer you some coffee?
2. a. B: Only if you have it ready.
b. B: Maybe later.

Suggestions and requests can also be accepted and declined conditionally and with uncertainty. Similarly, information-providing acts may express the speaker's awareness that he possesses uncertain information, as illustrated in (17):

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

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

(18) a. A: Can you tell me what time is the first flight tomorrow?
B: The first flight tomorrow morning is at seven-thirty.
A: Perfect!

b. A: What about a fresh cup of coffee?
B: Ah, you're wonderful!

In order to be able to represent such phenomena, this standard includes certain *qualifiers* that may be associated with a communicative function. A corpus-based study (Petukhova and Bunt, 2010) indicates that uncertainty and conditionality can be captured by means of binary distinctions (certain/uncertain; conditional/ unconditional); the standard therefore defines two binary-valued attributes, @certainty and @conditionality. The certainty values ‘uncertain' and ‘certain' can be associated with information-providing functions in order to represent the speaker's expression of (un)certainty about the correctness of the information that he provides, and with commissive functions to indicate unvertainty about the speaker’s commitment to perform the action under discussion. This attribute has the default value ‘certain'. The conditionality values ‘conditional' and ‘unconditional' can be used with action-discussion functions, and concern the ability and willingness of the participant whose action is under discussion. This attribute has the default value ‘unconditional'.

For representing a speaker's attitude or emotional stance the present standard follows the commonly made distinction between ‘sentiment’ and ‘emotion’, where the former is understood as a view or opinion that is held (and is often shared by a group), while the latter is a mental state that occurs as a result of arousal by an internal stimulus (a thought) or an external one (an event). Sentiment is often characterized by a polarity (positive or negative), possibly with an intensity, see e.g. Roman et al. (2015). For emotions a wide variety of descriptors has been proposed in the literature, ranging from six basic emotions (Ekman, 1972) to several hundred possible values and complex structural descriptions. This standard includes in its first edition the option of annotating a sentiment or an emotion by means of a qualifier as the value of the attribute @sentiment for which any set of values may be chosen depending on the domain, annotation goals, or interactive setting, or on theoretical preferences. Experiences in using the standard have shown that it may be useful to characterize participants’ sentiment w.r.t. something in the dialogue as ‘positive’ or ‘negative’; in this second edition these values are stipulated for the @sentiment attribute. The annotation of emotions, by contrast, requires more articulate structures. Annex E describes a plug-in mechanism for importing concepts from the W3C recommendation ‘EmotionML’ in order to enrich DiAML annotations with emotion-related information.

9 DiAML: Dialogue Act Markup Language

The Dialogue Act Markup Language DiAML has been designed in accordance with ISO 24617-6 (Principles of semantic annotation), which implements the distinction made in the ISO Linguistic Annotation Framework (LAF,

ISO 24612:2009) between *annotations* and *representations*. 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 presented; ‘representation' refers to the format in which an annotation is rendered. According to LAF, *annotations* are the proper level of standardization, rather than *representations*. Following ISO 24617-6, this distinction is implemented in the DiAML definition by a syntax specification that defines, besides a class of XML-based *representation structures*, also a class of more abstract *annotation structures*. These specifications are called *concrete* and *abstract syntax*, respectively. Annotation structures are set-theoretical structures, consisting of concepts of the classes that populate the metamodel (Fig. 1). The concrete syntax defines a reference format for rendering annotation structures in XML. Alternative representation formats for DiAML annotation structures are discussed in Bunt et al. (2019).

The following subclauses outline the abstract syntax, the concrete syntax, and the semantics of DiAML annotations; Annex C contains a formal specification of the abstract syntax and the concrete DiAML-XML syntax. For a detailed specification of the semantics of DiAML annotation structures see Bunt (2014).

9.1 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, and qualifiers.

An annotation structure is a set of *entity structures* and *link structures.* Entity structures contain semantic information about a dialogue segment; link structures describe semantic relations between entity stuctures. Entity structures are always of the general form ⟨m,z⟩, where ‘m’ is a markable and ‘z’ designates a structure that describes some linguistic information. Link structures are typically of the form ⟨e1, e2, R⟩, consisting of two entity structures and a relation.

The entity structure of central interest in DiAML is a pair ⟨m,α⟩, of which the linguistic information ‘da’ is a so-called ‘dialogue act structure’. A dialogue act structure contains the information that characterizes a single dialogue act. This includes minimally 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 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. For a dialogue act which depends semantically on (the interpretation of) one or more previous dialogue segments, a sixth component is a set *E* of elements that the act depends on through functional or feedback dependence relations. In a setting in which other participants than the sender and the addressees should be taken into account, an additional element is a set *H* of ‘other participants’ (see Fig. 1). A dialogue act structure is therefore in the simplest case a triple ⟨*S, A,* f_d⟩, consisting of a sender *S*, a (set of) addressee(s) *A*, and a dimension-specific function f_d, and in the most complex case a 7-tuple as in (19), with a general-purpose function *f* a dimension *d,*, a set *q* of one or more qualifiers, and a set *E* of one or more dialogue units that the act depends on.

(19) α = ⟨*S, A, H, f, d, q, E*⟩

A link structure in DiAML 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*.

9.2 Concrete syntax The DiAML concrete syntax is defined in accordance with the CASCADES methodology for defining semantic annotation languages, described in Bunt (2013) and ISO 24617-6. 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 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.

The DiAML concrete syntax specifies a reference representation format based on XML, called DiAML-XML. This specification lists 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 structures. In particular, XML elements are defined for entity structures and link structures. The dimensions, communicative functions, and qualifiers that can be used in DiAML are defined as data categories, following ISO 12620:2009. Their specification can be found in Annex F.

Entity structures for dialogue acts 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; the attribute otherParticipant is optional;
- @dimension, whose value names one of the dimensions defined in this standard;
- @communicativeFunction, whose value names one of the communicative functions defined in this standard;
- @certainty, @conditionality, and @sentiment, whose values is one of the qualifiers defined in this standard. The attributes are optional;.
- @functionalDependence, whose values refer to one or more dialogue acts that the given dialogue act has a functional dependence relation with. This attribute has a value only for dialogue acts with a responsive communicative funcdtion.
- @feedbackDependence, whose values refer to one or more dialogue acts or reference segments that the given dialogue act has a feedback dependence relation with. This attribute has a value only for certain feedback acts and for dialogue acts in the Own Communication Management or Partner Communication Management dimensions, as specified in Clause 8.4.

Link structures are represented either by the XML element <rhetoricalLink> or by the element <drLink>. The <rhetoricalLink> element has the following attributes:

- @dact, whose value refers to a dialogue act that is rhetorically related to other dialogue acts;
- @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 <drLink> element has the following attributes:

- @arg1 and @arg2, whose values refer to two rhetorically related dialogue acts or their semantic content, if a plug-in for semantic content is used (see Clause 10 and Annex E);
- @rel, whose value is a rhetorical relation;

and makes use of embedded <argRole> elements, which have an attribute @arg, whose value identifies a dialogue act, and an attribute @role, whose value names an argument role.

Example (20c-d), shows the abstract annotation structure and its DiAML-XML representation of the dialogue fragment in (20a), segmented as shown in (20b).

(20a) P1: *What time does the next train to Utrecht leave?*
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; they may be attached 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 functional segments are identified at another level of XML representation, following ISO standard 24610-1. P2's utterance is segmented into two overlapping functional segments: m2 in the Auto-Feedback dimension (reflecting that the repetition of a large part of an

utterance signals positive feedback on understanding it) and m3 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. Note that the abstract annotation structure in (20c) is a set of three elements, corresponding to the three dialogue acts in this fragment, where the second and the third element both have the first element embedded, indicating their dependence on the first dialogue act.

(20b) *Segmentation of the exchange in (20a):*
m1 = *What time does the next train to Utrecht leave?* (Task dimension)
m2 = *The next train to Utrecht leaves* (Auto-Feedback dimension)
m3 = "*The next train to Utrecht leaves I think at 8:32.*" (Task dimension).

(20c) *Annotation structure according to DiAML abstract syntax:*
{⟨m1,⟨p1,p2,setQuestion,Task⟩⟩,
⟨m2,⟨p2,p1,autoPositive,{⟨m1,⟨p1,p2,setQuestion,Task⟩⟩}⟩⟩,
⟨m3,⟨p2,p1,aswer,Task,{uncertain},{⟨m1,⟨p1,p2,setQuestion,Task⟩⟩}⟩⟩⟩}

(20c) *DiAML-XML annotation representation:*

```
<diaml xmlns:"http://www.iso.org/diaml/">
  <dialogueAct xml:id="da1" target="#m1" sender="#p1" addressee="#p2"
    communicativeFunction="setQuestion" dimension="task"/>
  <dialogueAct xml:id="da2" target="#m2" sender="#p2" addressee="#p1"
    communicativeFunction="autoPositive" feedbackDependence="#da1"/>
  <dialogueAct xml:id="da3" target="#m3" sender="#p2" addressee="#p1"
    communicativeFunction="answer" certainty="uncertain" dimension="task"
    functionalDependence="#da1"/>
</diaml>
```

9.3 Semantics

DiAML annotation structures have a semantics in terms of information-state updates. The most important kind of structure defined by the DiAML abstract syntax, the dialogue act structure, is a *functional* characterization of a dialogue act. It does not correspond to a complete dialogue act, since it does not include the semantic content (but only a semantic content category, a 'dimension'). The semantics of a complete dialogue act is obtained by combining the interpretation of a dialogue act structure with a semantic content. This is accomplished by applying the interpretation $I_a(\langle s,\alpha\rangle)$ of an entity structure which contains a dialogue act structure α, to the semantic content $\kappa(s)$ of the functional segment that expresses the dialogue act. The result is an information state update operation as shown in (21) for a dialogue act that has no functional dependences to other dialogue acts.

(21) $I_a(\langle s,\alpha\rangle) = I_a(\alpha)(\kappa(s))$

The interpretation $I_a(\alpha)$ of a dialogue act structure α is defined as follows for those structures without qualifiers:

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

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 its dimension. Annex C provides more information about the DiAML semantics; a full specification can be found in Bunt (2014).

10 Extension and customization

The annotation scheme defined in this standard is more comprehensive than other existing dialogue annotation schemes. For some, relatively simple application domains it may be unnecessarily complex, while on the other hand the domain-independence of the scheme has the effect that it may lack some communicative functions or other concepts that are important for specific applications. The sets of communicative functions, qualifiers, and relations among dialogue units defined in this standard cannot be expected to be all that is needed for every

application, for every task domain, for every type of interaction, and for every annotation purpose. The structural properties of the annotation scheme are however useful also for simplifying the annotation scheme for relatively simple applications and for defining extensions to customize the scheme to a certain application, in particular for the use cases UC3 and UC4 (see Clause 4.2).

Simplified versions of the annotation scheme are easily defined thanks to the structural properties of the scheme, which result from a careful design and the optionality of parts of it; this is discussed in Clause 10.1.

Extensions of the annotation scheme affect the interoperability of the annotations that result from its application in the use cases UC1 and UC2; whether this is desirable or undesirable depends on the purpose of the annotations. This second edition of the 24617-2 standard makes use of *tripartite annotation scheme plug-ins*, a powerful mechanism for enriching DiAML-annotations. This is discussed in Clause 10.2

10.1 Simplifying the annotation scheme: options and selections

The structural properties of the annotation scheme defined in this standard which contribute to its customisability are mainly the following:

a. The dimensions are orthogonal: a functional segment expressing a dialogue act in one dimension does not necessarily have a communicative function in another dimension..
b. Each communicative function has a semantic definition in terms of how it defines an update operation on the information states of dialogue participants when combined with a semantic content.
c. The set of general-purpose communicative functions is *semantically connected* (any two functions are either mutually exclusive or one is a specialization of the other). The general-purpose communicative functions are semantically connected, and for each dimension the set of dimension-specific communicative functions is semantically connected.
d. Due to the semantic connectedness property, a functional segment never needs to be annotated with more than one communicative function in each dimension where it has a function. A functional segment thus has maximally as many functions as there are dimensions.

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

- By virtue of the orthogonality of the dimensions, any dimension and the corresponding set of dimension-specific communicative functions may be left out of consideration.
- Communicative functions for which there is a less specific function in the annotation scheme may be left out, since the remaining set of functions is still semantically connected.
- Due to their optionality, there's no obligation to use qualifiers and rhetorical relations.

10.2 Extending the annotation scheme: tripartite plug-ins and interfaces

In software, a 'plug-in' is an addition to an existing computer program that adds some feature to it, typically in order to customize it. Similarly, annotation tools often come with 'plug-ins' in order to customize them for a particular annotation scheme. Annotation schemes sometimes come with 'plug-ins' in the sense of a vocabulary of terms that may be used in annotations; this is for example the case for the EmotionML annotation scheme (Burkhardt et al., 2014). This subclause describes the notions of a ***triplartite annotation plug-in*** and a ***plug-in interface,*** as introduced in Bunt (2019).

According to the SemAF principles of semantic annotation, as laid down in ISO 24617-6: 2015, a semantic annotation scheme has a three-part definition, consisting of (1) an abstract syntax that specifies the well-formed '*annotation structures'* as set-theoretical constructs, such as pairs and triples; (2) a semantics that specifies the meanings of the annotation structures; (3) a concrete syntax that specifies a representation format for annotation structures. Formally, the definition of an annotation scheme is thus a triple $L_a = \langle AS_a, CS_a, Sm_a \rangle$, formed by specifications of an abstract syntax (AS_a), a concrete syntax (CS_a), and a semantics (Sm_a). Each of these components is further structured:

- The abstract syntax specification AS_a is a pair $\langle CI_a, AC_a \rangle$, consisting of the conceptual inventory (CI_a) and the specification of entity structures and link structures (AC_a); together, these define the class of well-formed annotation structures.

- The concrete syntax specification CS_a is a triple $\langle V_a, CC_a, F_a \rangle$, where V_a is a vocabulary, CC_a is the specification of a class of syntactic structures, and F_a is an encoding function that maps AS_a-annotation structures to CS_a-representations. Together, V_a and CC_a define the class of well-formed representations.

- The semantic specification Sm_a is a pair $\langle M_a, I_a \rangle$, consisting of a model and an interpretation function.

Altogether, the definition of an annotation scheme is thus a nested triple:

(23) $L_a = \langle AS_a, CS_a, Sm_a \rangle$

$= \langle \langle CI_a, AC_a \rangle, \langle V_a, CC_a, F_a \rangle, \langle M_a, I_a \rangle \rangle$

An ***tripartite annotation plug-in*** is a annotation scheme that can be added on to a host annotation scheme, in general requiring a ***plug-in interface*** for allowing annotations that combine elements from the two schemes (Bunt, 2019). Being an add-on annotation scheme, an annotation plug-in has the same tripartite structure, introducing (1) additional entity structures and link structures in the abstract syntax, (2) the semantics of these structures, and (3) their encodings.

An interface for a given host annotation scheme and plug-in defines link structures that relate entity structures of the two schemes. It has again the tripartite structure of an annotation scheme, but it does not introduce new entity structures but only new link structures.

Added on to a host annotation scheme, a plug-in in general defines additions to all three parts of the host scheme (see (23)): to the host's concrete syntax but also to the abstract syntax and semantics, making sure that the extended annotation scheme complies with the ISO 24617-6 principles of semantic annotation. A plug-in therefore has the same three-part structure, specifying additional concepts, entity structures and link structures, their concrete encodings, and their semantics. To emphasize their highly structured character, very different from a plug-in in the form of a vocabulary, such plug-ins are referred to as 'tripartite plug-ins'. In the rest of this document, 'plug-in' is used as short for 'tripartite plug-in'. A plug-in interface, defining link structures that relate entity structures of the two schemes, is again a structured specification of abstract structures, their concrete encodings, and their semantics, but an interface is typically simpler than a plug-in since it only introduces structures for 'linking' the structures of the host annotation scheme and the plug-in.

Note that plug-ins for the annotation scheme of this standard must respect the structural properties listed in the previous subclause. In particular, any plug-in that introduces new dimensions or communicative functions is required to ensure that the resulting set of dimensions is still orthogonal and that the resulting set of communicative functions is semantically connected.

Annex E describes tripartite plug-ins and plug-in interfaces for adding semantic content information to dialogue acts, domain-specific communicative functions, rhetorical relations, emotions, more fine-grained feedback functions, and additional functions for social obligations management,.

Bibliography

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

Ahrenberg, L., N. Dahlbäck and A. Jönsson (1995) Coding schemas for Studies of Natural Language Dialogue. In: *Working Notes from the AAAI Spring Symposium, Stanford.*

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.

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.

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.

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

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

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.

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

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

Augmented Multiparty Interaction Consortium (AMI) (2005) Guidelines for Dialogue Act and Addressee Annotation.

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

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

Benamara, F. and M. Taboada (2015) Mapping different rhetorical relation annotations: A proposal. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pp. 147-152.

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.

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

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.

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.

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.

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

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

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.

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.

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.

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

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

Bunt, H. (2014) A context-change semantics for dialogue acts. In *Computing Meaning, Vol. 4*, Harry Bunt, Johan Bos and Stephen Pulman, editors. Springer, Dordrecht, pp. 177-201.

Bunt, H. (2018) *Semantic Annotation of Quantification in Natural Language -- preparatory study for developing an ISO standard TiCC* Technical Report 2018-15, Tilburg Center for Cognition and Communication and Department of Cognitive Science and Artificial Intelligence, Tilburg University.

Bunt (2019) An annotation scheme for quantification. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS 2019)*, Gothenburg, Sweden.

Bunt, H. (2019) Plug-ins for content annotation of dialogue acts. In *Proceedings of the 15th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-15),* Gothenburg, Sweden, pp. 34-45.

Bunt, H., E. Gilmartin, C. Pelachaud, V. Petukhova, L. Prévot and M. Theune (2018) Downward Compatible Revision of Dialogue Act Annotation. In *Proceedings 14th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-14),* Santa Fé, New Mexico, pp. 21-34.

Bunt, H., J. Pustejovsky and K. Lee (2018) Towards an ISO Standard for the Annotation of Quantification. In *Proceedings 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018.

Bunt, H., V. Petukhova, Andrei Malchanau, Alex Chengyu Fang, and Kars Wijnhoven (2019)
The DialogBank: Dialogues with Interoperable Annotations. *Language Resources and Evaluation* 53:213-249.

Bunt, H. and R. Prasad (2016) ISO DR-Core (ISO 24617-8): Core Concepts for the Annotation of Discourse Relations. *Proceedings 12th Joint ACL-ISO Workshop on Interoperable Semantic Annotation,* Portoroz, Slovenia, pp. 45-54.

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.

Burkhardt, F., Pelachaud, C., Schuller, Bj., and Zovato, E. (2017) EmotionML. In D. Dahl (ed.) *Multimodal Interaction with W3C Standards*. Springer, Cham (Switzerland), pp. 65-80.

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.

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.

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

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

Clark, H., and Fox Tree, J. (2002). Using ‘uh' and ‘um' in spontaneous speech. *Cognition*, 84: 73--111.

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

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.

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

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. Available at http://cogsci.l.chiba-u.ac.jp/$\sim$tutiya/Publications/DRI98Report.pdf

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.

Demberg, V., F. Torabi Asr and M. Scholman (2017) How consistent are our discourse annotations? Insights from mapping RST-DT and PDTB annotations. *Dialogue & Discourse*, 10(1), 87-135.

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

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

Ekman, P. (1972). Universals and Cultural Differences in Facial Expressions of Emotion. In J. Cole (Ed.), *Nebraska Symposium on Motivation* (Vol. 19, pp.207-282). University of Nebraska Press.

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.

Gilmartin, Emer, Christian Saam, Brendan Spillane, Maria O’Reilly, Ketong Su, Arturo Calvo, Loredana Cerrato, Killian Levacher, Nick Campbell, Vincent Wade (2018) The ADELE Corpus of Dyadic Social Text Conversations:Dialog Act Annotation with ISO 24617-2. Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC-2018).

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

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.

Hobbs, J.R. (1979) Coherence and Coreference. *Cognitive Science* 3 (1):67-90.

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

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

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

ISO 24610-1:2006 Language resource management: Feature structures, Part 1: Feature structure representation. ISO, Geneva.

ISO 24612:2009 Language resource management: Linguistic annotation framework (LAF). ISO, Geneva.

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

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

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

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

Klein, M. (1999) Standardization efforts on the level of dialogue acts in the MATE project. Available at http://acl.lcd.upenn.edu/W/W99/W99-0305.pdf

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

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

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.

Lascarides, A. and N. Asher (1991) *Discourse relations and defeasible knowledge.* Proceedings 29th Annual Meeting of the Association for Computational Linguistics, pp. 55-62.

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

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

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

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

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

Malchanau, A., V. Petukhova and H. Bunt (2019) Towards Integration of Cognitive Models in Dialogue Management: Designing the Virtual Negotiation Coach Application*. Dialogue and Discourse 9(2), 35-79.*

Malchanau, A. (2019) *Cognitive Architecture for Multimodal Multidimensional Dialogue Management.* Ph.D. thesis, University of Saarland, Saarbrücken.

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

Ohtake, Kiyonori and Etsuo Mizakumi 2017. The NICT Kyoto Dialogue Corpus. In Nancy Ide and James Pustejovsky (eds) *Handbook of Linguistic Annotation*, Springer, Berlin, pp. 1265-1286.

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

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/TR2009003.pdf

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

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

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

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.

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.

Petukhova, V., Gropp, M., Klakow, D., Eigner, G., Topf, M., Srb, S., Moticek, P., Potard, B., Dines, J., Deroo, O., Egeler, R., Meinz, U., Liersch, S., and A. Schmidt (2014) The DBOX Corpus Collection of Spoken Human-Human and Human-Machine Dialogues. In *Proceedings 9th International Conference on Language Resources and Evaluation (LREC 2014}*.

Volha Petukhova, Christopher A. Stevens, Harmen de Weerd, Niels Taatgen, Fokie Cnossen, Andrei Malchanau (2016) Modelling Multi-issue Bargaining Dialogues: Data Collection, Annotation Design and Corpus. In *Proceedings 10th International Conference on Language Resources and Systems (LREC 2016)*.

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.

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

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

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.

Roman, N, P.Piwek, A. Carvalho and A. Alvares (2015) Sentiment and behaviour annotation ina corpus of dialogue summaries. *Journal of Universal Computer Science* 21 (4), 561-586.

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

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

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

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

Sanders, T.,V. Demberg, J. Hoek, M. Scholman, S. Zufferey and J. Evers-Vermeul (2018) Unifying dimensions in coherence relations: How various annotation frameworks are related. *Corpus Linguistics and Linguistic Theory.* DOI*:* https://doi.org/10.1515/cllt-2016-0078

Schefler, T. and M. Stede (2016) Mapping PDTB-style connective annotation to RST-style annotation. In Proceedings 13th Conference on Natural Language Processing (KONVENZ 2016), pp. 242-247.

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.

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

Traum, D. (1999) Speech Acts for Dialogue Agents. In: M. Wooldridge and A. Rao (eds.) {\it Foundations of rational agency}, Kluwer, Dordrecht, 169--201.

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

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

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.

Annex A: Annotation guidelines
(normative)

This annex provides practical information for using the ISO 24617-2 standard, second edition, for dialogue act annotation. Section A.1 discusses some general issues in dialogue act annotation. Section A.2 explains the segmentation of a dialogue into meaningful segments. Section A.3 contains guidelines for applying the concepts of DiAML annotation: dimensions, communicative functions, qualifiers, functional dependence relations, feedback dependence relations, and rhetorical relations.

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. The present standard is intended to apply to dialogues in a wider sense, where the participants do not only use natural language but also nonverbal means, such as gestures and facial expressions, in the case of human participants and embodied conversational agents, and means like highlighting, blinking, and beeping in the case of computer systems such as virtual assistants and helpdesks.

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.

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 may happened that the speaker addresses one of the participants more than the others. In such a case, it is recommended 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 standard is intended for use by human annotators and by automatic annotation systems. It has been tested for being useful for both these purposes (see e.g. Petukhova, 2011; Petukhova et al., 2014; Chowdhury et al, 2016, Petukhova et al., 2016, Ngo et al., 2017, Gilmartin et al., 2018, Bunt et al., 2019).

If the primary aim 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

The communicative function of a dialogue segment may be recognisable in one of two ways: (1) through its linguistic or nonverbal properties which, in the context in which the segment occurs, are indicators of that function: or (2) through inference, because its function is implied by another function which the segment has (typically for the reason (1)). In the first case it is common to say that the segment has that communicative function *explicitly*; in the second case that it has that function *implicitly*. The following example illustrates this:

(A1) 1. A: Would you like to have some coffee?
2. B: Some coffee would be great, thanks.

A's utterance may be understood as an *Offer*; due to the fact that expressions of the form "*Would you like*…" are commonly used for that purpose; B's response is an *Accept Offer* by virtue of its linguistic form and the fact that it occurs immediately after an *Offer*. Since an offer can only be accepted when it has been understood, B's response by implication also has a positive auto-feedback function.

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

a. A communicative function F_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 like the general-purpose functions of these standard, where for instance a *Confirm* is a special case of an *Answer*, and a *Correction* is a special case of a *Disagreement*, which in turn is a special case of an *Inform*.

b. Another class of implied functions has to do with turn-taking. Every time someone starts speaking, this could be interpreted as the performance of a turn-taking act; every time someone stops speaking this could be interpreted as a turn-release act; and every time a speaker goes on speaking this could be interpreted as a turn-keeping act. These phenomena are discussed below in Section A.1.3

c. 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 the function F_1 it also has the function F_2, assuming that the dialogue participants behave cooperatively. For example, a thanking act like *"Thank you"* will normally be understood as being also a signal of positive feedback.

Should implicit communicative functions be annotated? Annotating logically entailed functions would be redundant, since by their very nature such functions can be inferred from explicit functions. For conversationally implicated functions the situation is a little different, since these functions do not necessarily follow from an explicit function. The annotation of all implicated positive feedback functions would be very impractical, and the annotation of all implied turn management functions would even be impossible for implicit turn-keeping functions. Entailed or implicated communicative functions should therefore *not* be annotated either. If the purpose of an annotation campaign is such that it would be important to recognise implicated functions, then this can be done post-hoc by checking whether the dialogue context allows the inference of these functions and add these functions accordingly. An annotator running into the situation where a functional segment has an explicitly expressed communicative function and an implied function, should decide whether the implied function is a logical consequence or a matter of what is plausible in the given context. In the first case the implied function should definitely not be annotated; in the second case it may. For more details about types of implicit functions and strategies for how to deal with them see Bunt (2011).

Standard speech act theory mostly regards indirect speech acts as just another *form* of the same act as the direct form. By contrast, this standard takes the view that indirect forms signal subtly different packages of

beliefs and intentions than direct ones, and thus express 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 (or 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".*

The latter 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:

(A2) <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 Clause 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, rather than 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

A.2 Segmentation

According to this standard, dialogue acts correspond to *functional segments* as defined in Clause 8: 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 the following example:

(A3) 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 expression of the question, and has a *Stalling* function. The preferred segmentation would distinguish in this case one discontinuous 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:

(A4) <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"/>

It may also 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*:

(A5) 1. A: There are two flights early in the morning, at 7.45 and at 8.15
2. B: Yes
3. A: and two more in the evening, at 7.15 and at 8.30

The utterances 1 and 3 in this example form a single discontinuous functional segment.

In an actual annotation process, the identification of functional segments and communicative functions typically go hand in hand. For annotating dialogue acts with a (auto- or allo-)feedback-specific function, or an OCM- or a PCM-specific function, the annotation also goes hand in hand with the identification of the relevant reference segments.

A.3 DiAML-annotations

A.3 DiAML-annotations

Once a functional segment has been identified, the assignment of sender and addressee roles in dialogue act annotations is usually a trivial matter. For deciding on the assignment of communicative functions, see below. For assigning dimensions, the decision to be made is which kind of information or action is addressed. Does it concern (1) the underlying task/activity; or (2) the speaker's processing of previous utterances; or (3) the addressee's processing of previous utterances; or (4) the allocation of the speaker role; or (5) the time needed to continue the dialogue; or (6) the editing of what the speaker is saying; or (7) the editing of what the addressee is currently saying; or (8) the structure of the dialogue; (9) ensuring contact; or (10) social obligations?

A.3.1 Encoding general-purpose functions

Information transfer functions

All dialogue acts with an information transfer function have the main purpose of making certain information available to the addressee (acts with an *Inform* function or a function dominated by *Inform* in the hierarchy shown in Fig. 2) or of the speaker obtaining certain information (the Information-seeking functions in Fig. 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 Fig. 2 as decision trees, going systematically left-right through the functions at the next level down and checking the defining conditions that distinguish each of these functions from their ancestor and from each other. Since the functions at one level in a subtree are mutually exclusive, at most one of them applies. If one is found that applies, then go down one level to the functions dominated by this function, and repeat the process. Keep doing this until hitting a level where none of the functions apply. At that point choose the function that dominates the functions at that level.

Action discussion functions

All action discussion functions have in common that their semantic content describes 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 characterised by the sender having the goal that the addressee commits himself to performing an action. In order to decide whether a segment of dialogue has a commissive or a directive function, an annotator should decide whether the segment has the purpose of expressing or trying to impose 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 recognised by their use of particular fixed forms and formulaic expressions.

Auto- and Allo-Feedback

Feedback acts have the purpose of providing or eliciting information about the processing of utterances in dialogue. Both auto- and allo-feedback providing functions are divided into positive and negative ones. Positive feedback is very often expressed implicitly, and should in such a case most probably not be encoded, as argued in A.1.4. Negative feedback is virtually always explicit, and as such easy to recognise. Some of the frequently used fixed forms for negative auto-feedback are *“Huh?”, “What?”, “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. For example:

(A6) 1. A: I would like to travel next Saturday, in the afternoon.
2. B: Next Saturday in the afternoon I have a flight leaving at 16:10.
3. B: On Satuday 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 what A said. In utterance 3, by contrast, B paraphrases parts of A's question, and this can be taken to indicate not only what B heard but also how B interpreted what A said (which in this example may be particularly relevant for the interpretation of *“next”,* 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. This is 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 a 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 like B: *"That's right",* speaks about the addressee's processing, hence this is an act in the Allo-Feedback dimension.

Turn Management

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

(A7) 1. A: Do you know what time it is?
2. B: It's nearly twelve fifteen.

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

(A8) 1. A: Do you know what time it is? I should have the twelve seventeen train. Maybe it's too late already
2. 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, to keep, or to get rid of the speaker role. Just starting to speak, continuing to speak, or ceasing to speak should not be annotated as expressions of Turn Management functions.

A particularity of the Turn Management dimension is that the dimension-specific functions are divided into two subclasses, turn-initial and turn-final ones, that could be considered as separate dimensions. Usually only the first segment in a turn has a turn-initial function and only the last one a turn-final one. The non-final utterances in a turn have a turn-keeping function when the speaker signals (for example 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*").

Time Management

Time management functions are concerned with the sender buying some time. This standard 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. The first case may for example occur because the speaker is looking for the right words to express what he wants to convey, or because he hasn't quite made up his mind as to which information to convey, or because he needs a bit of time to look up something. The second case (*Pausing*) may for example occur 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 ("*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 like "*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*.

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*).

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 and slow tempo) to indicate that a dialogue can be closed, indicating positive feedback concerning the entire preceding dialogue. There do not seem to exist dialogue acts that have the sole function of closing a dialogue; conventionally, a dialogue is considered closed when the participants have exchanged farewell greetings. This standard 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 is performed which addresses this new topic. Only explicitly signaled topic (actual or intended) shifting should be annotated as such.

Contact Management

Contact Management acts serve the purpose of monitoring audio-visual contact and attention. Looking the speaker into the eye can be used to indicate that attention is paid to the speaker; in telephone conversations and textual chats, "*Hello?"* with a question-like intonation can be used in English to check contact/attention with someone on the other end of the line; other languages also have expressions for this purpose, like "*Pronto?"* in Italian, and "*Hello*" with a statement-like intonation can be used to make or confirm contact. These functions are semantically similar to dialogue-initial greetings, and in many languages the same or very similar expressions can be used for greeting as well as for contact management.

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.. An expression of thanks can for instance be used to signal that the speaker intends to terminate the dialogue, and can also be used for positive feedback.

A.3.3 Qualifiers

a. Certainty
Certainty qualifiers may be used when the sender of a dialogue act expresses uncertainty about the correctness of the information provided in an information-providing act, or about his commitment to perform an action that forms the semantic content of a commissive act. This is illustrated in (A9) 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 such expressions are not present, as in (A10), 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*.

(A9) 1. A: Do you know who'll be coming tonight?
2. B: I have a hunch that Mary won't come.
3. B: Peter, Alice, and Bert will probably come.
4. B: I heard that Tom and Anne might come.
5. B: I'm not sure if Bill will come.

(A10) 1. A: Do you know who'll be coming tonight?
2. B: Mary won't come.
3. B: Peter, Alice, and Bert will come.
4. B: I heard that Tom and Anne will come.
5. B: Bill will come.

Speakers may also signal being *very* certain, as exemplified in (A11). For such cases, the DiAML encoding with `certainty="certain"` is recommended,

(A11) 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 can not only be 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.

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

(A12) 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 Fig. A1 can be used.

b. Conditionality
Conditionality qualifiers may be used to indicate the sender’s view on the possibility (with respect to ability and power), the necessity, or his willingness to perform an action; the qualifiers *conditional* and *unconditional* can therefore be attached to action-discussion functions for this purpose. The following examples illustrate this.

(A13) 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.

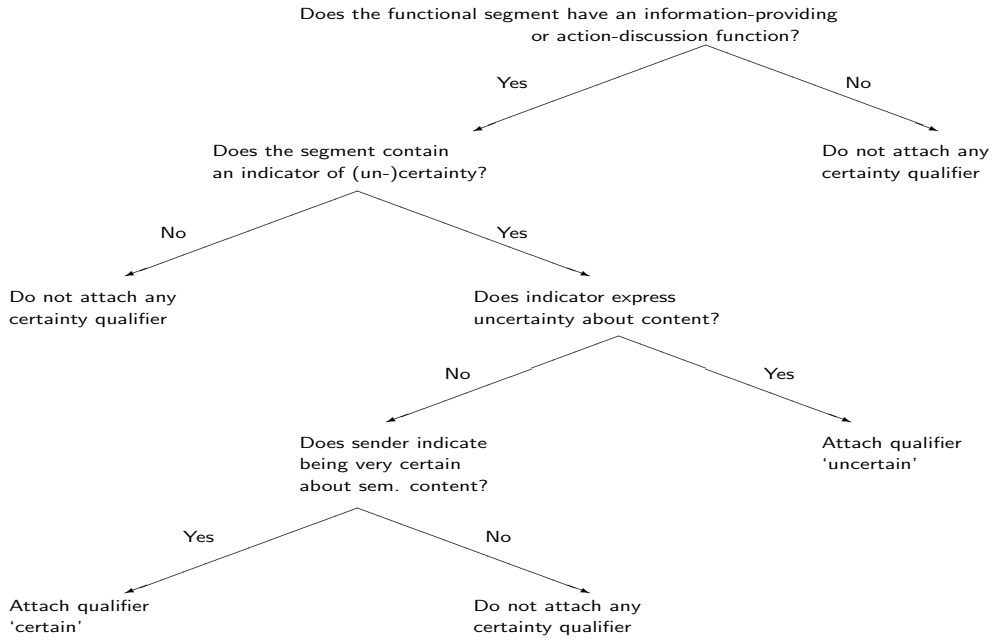

Figure A1. Decision tree for applying certainty qualifiers

In (A13a) we see the conditional acceptance of an offer; in (A13b) a conditional request, with a conditional acceptance; in (A13c) a conditional promise; in (A13d) a conditional acceptance of a request and a conditional rejection; and in (A13e) a conditional rejection of a suggestion. The absence of expressions indicating a condition corresponds to indicating unconditionality, hence the default case is *unconditional*, and does not need to be marked up. Annnotation of conditionality, like that of certainty, is thus a case of syntactic optionality (see Clause 5.2). Explicit expressions of 'unconditionality', are "*definitely" and "unconditionally"*.

Conditional dialogue acts can often be recognised by the use of conditional expressions such as *"if ..."*, "*unless*" and "*just"* (as in (A13d)). Like 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 of Fig. A2 can be used.

c. Sentiment
A particular sentiment associated with the performance of a dialogue act may be annotated if the sender indicates a positive/negative feeling or attitude concerning the semantic content or concerning the addressee, verbally or nonverbally, or both. A sentiment can be expressed verbally by the choice of adjectives or adverbs..Nonverbal expressions of sentiment exist in abundance and in great variety, including for instance smiling (positive), pressing lips together (negative), and sighing (negative).

Positive and negative sentiment both require explicit expression; there is no default attitude (a 'neutral' attitude would just mean not positive and not negative). The annotation of sentiment is a case of semantic optionality (see Clause 5.2).

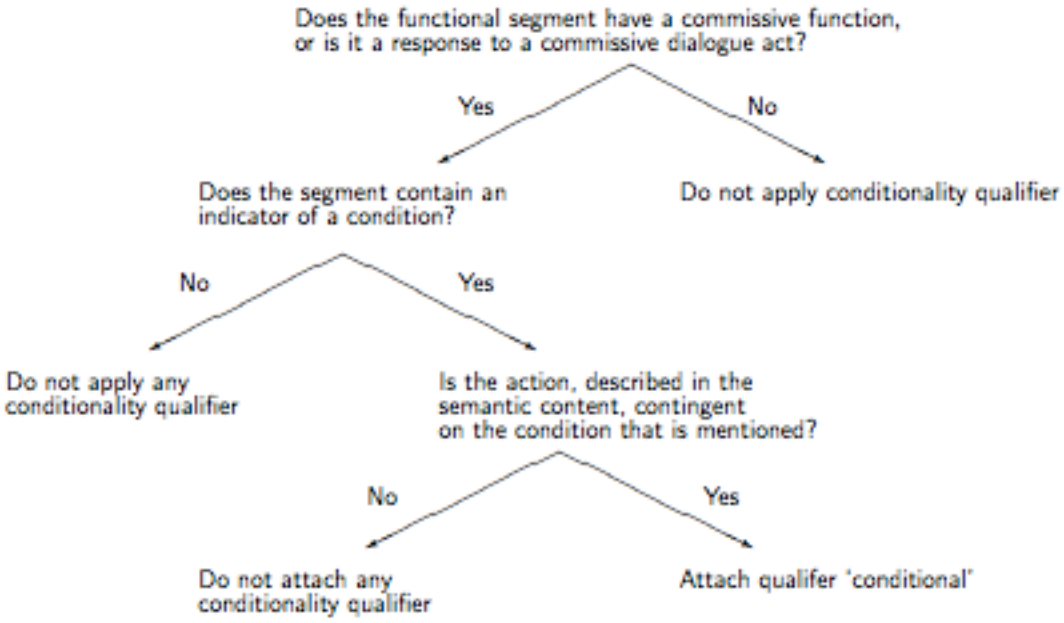

Figure A2 Decision tree for applying conditionality qualifiers

A.3.4 Functional dependences, feedback dependences, and rhetorical relations

Functional dependence
A dialogue act A1 is functionally dependent on a previous dialogue act A2 (its 'functional antecedent'), if its communicative function by its very nature responds to another dialogue act, contributed by another participant, and its semantic content is determined by the combination of the local semantic content of the

functional segment that expresses A1 and the semantic content of A2. This is the case for the following communicative functions defined in this standard:

(A14) - Answer, Confirm, Disconfirm;
- Agreement, Disagreement, Correction;
- Address Request, Accept Request, Decline Request;
- Address Suggestion, Accept Suggestion, Decline Suggestion;
- Address Offer, Accept Offer, Decline Offer;

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 (A15) or (b) responds to an implicit dialogue act.

(A15) 1. U: Can you tell me what time there are trains from Harwich to York?
2. S: What day would you like to travel?
3. U: Tomorrow morning.
4. S: On Tuesday morning there are trains at 6:45, 70:30,...(etc.)

In (A15), utterance 4 forms a functional segment with the function *Answer*, which responds to the question formed by the dialogue acts expressed by utterances 1 and 3 together. In such a case it is recommended to mark functional dependence relations to each of those dialogue acts which together make up the question; in this example, to both the dialogue acts expressed in utterances 1 and 3. (Note that, in a complete dialogue annotation, the dialogue act in 3 is marked as an answer act with as its functional antecedent the question in utterance 2, which itself asks for a clarification of the question in 1.)

(A16) <dialogueAct xml:id="da4" target="#m4" speaker="#s" addressee="#u" dimension="task" communicativeFunction="answer" functionalDependence=”#da1 #da3”/>

The case of responding to an implicit dialogue act is illustrated by example (B6) in Annex B, where the dialogue system operates on the assumption that the user has a question about train journeys, and queries the user for parameter values until it believes it knows the user's question, which it subsequently answers. This question is not explicit anywhere in the dialogue. In such a case it is recommended to follow the same strategy as for multi-act antecedents, and mark up functional dependences .

Feedback dependence
Feedback acts that have a dimension-specific communicative function (i.e. *Auto-Positive, Auto-Negative, Allo-Positive, Allo-Negative,* or *Feedback Elicitation* as well as the fine-grained functions introduced in Annex E) and OCM- and PCM acts with a dimension-specific function have a feedback dependence relation. This is either a relation with previous dialogue acts or with previous reference segments.

Encoding a feedback dependence relation means using the value of the feedbackDependence attribute to refer to the dialogue act(s) or reference segment(s) that the feedback is about. Most often this concerns 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 (A16).

Rhetorical relations
Many of the relations that may occur between units in discourse such as *cause, elaboration,* or *exemplification*, and which in the linguistic literature are often called ‘rhetorical relations' or ‘discourse relations', may also occur between dialogue acts. This standard does not specify any particular set of such relations, and therefore does not provide guidelines for their encoding. However, it is recommended to use the plug-in described in Annex E, which is based on the ISO 24617-8 standard for the annotation of rhetorical relations, and use the definitions of the relations as specified in Annex E in order to decide on the use of such relations in DiAML annotations.

Annex B: Annotated examples (informative)

B.1 Overview

This annex illustrates the use of this standard. Section B.2 shows the annotation of some short exchanges, like question-answer pairs. Section B.3 contains annotations of a complete spoken human-computer dialogue and of an extended fragment of a multimodal human-human dialogue.

B.2 Short exchanges

Example (B1) 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 "m1" and "m2", respectively. In this case, m1 is everything said by participant P1, and m2 is everything said by P2. 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.

(B1) a. P1: Where I should check in for Munich? [functional segment m1]
P2: For Munich go to counters 31 to 40. [functional segment m2]

b.
```
<diaml xmlns:"http://www.iso.org/diaml/"/>
  <dialogueAct xml:id="da1" target="#m1" speaker="#p1" addressee="#p2"
       communicativeFunction="setQuestion" dimension="task"/>
  <dialogueAct xml:id="da2" target="#m2" speaker="#p2" addressee="#p1"
       communicativeFunction="answer" dimension="task"
       functionalDependence="#da1"/>
</diaml>
```

An answer to a question may be argued to always entail positive feedback, since a question can only be answered successfully if it has been understood. It is not necessary to annotate entailed communicative functions; 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.

Example (B2) 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 standard 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.2b)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.

(B2) a. P1: Do you know what time the next train to Utrecht leaves? [m1]
P2: The next train to Utrecht leaves I think at 8:32.
m2.1: The next train to Utrecht leaves
m2.2: The next train to Utrecht leaves I think at 8:32.

b. <diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#m1" sender="#p1" addressee="#p2"
communicativeFunction="request" dimension="task" conditionality="conditional"/>
<dialogueAct xml:id="da2" target="#m2.1" sender="#p2" addressee="#p1"
communicativeFunction="autoPositive" dimension="autoFeedback"
feedbackDependence="#da1"/>
<dialogueAct xml:id="da3" target="#m2.2" sender="#p2" addressee="#p1"
communicativeFunction="answer" dimension="task" functionalDependence="#da1"/>
</diaml>

Example (B3) is a three-turn fragment of a dialogue from the HCRC Map Task corpus (Carletta et al., 1996). It 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.

(B3) a. 1. P1: Move up [m1]
2. P2: Slightly northeast? [m2]
3. P1: Yeah [m3.1] | very slightly [m3.2]

b. <diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#m1" sender="#p1" addressee="#p2"
communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da2" target="#m2" sender="#p2" addressee="#p1"
communicativeFunction="checkQuestion" dimension="autoFeedback"
feedbackDependence="da1"/>
<dialogueAct xml:id="da3" target="#m3.1" sender="#p1" addressee="#p2"
communicativeFunction="confirm" dimension="alloFeedback"
functionalDependence="#da2"/>
<dialogueAct xml:id="da4" target="#m3.2" sender="#p1" addressee="#p2"
communicativeFunction="inform" dimension="alloFeedback"/>
<rhetoricalLink dact="#da4" rhetoRelatum="#da2" rhetoRel="elaboration"/>
</diaml>

Example (B4) 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.

(B4) a. P1: engine E3 is going to pick up the bananas, back to Avon, dro... [m1]
P2: to pick up the oranges [m2]
b. <diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#m1" sender="#p1" addressee="#p2"
communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da2" target="#m2" sender="#p2" addressee="#p1"
communicativeFunction="correctMisspeaking" feedbackDependence="#da1"/>
</diaml>

Example (B5), 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.

(B5) a. P1: Would you like to say something at this point? [m1]
P2: Certainly. [m2]

b.
```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#m1" sender="#p1" addressee="#p2"
    communicativeFunction="offer" dimension="turnManagement"/>
<dialogueAct xml:id="da2" target="#m2" sender="#p2" addressee="#p1"
    communicativeFunction="acceptOffer" dimension="turnManagement"
    functionalDependence="#da1"/>
</diaml>
```

B.3 Annotated dialogues

B.3.1 Human-computer dialogue

The dialogue that is annotated here comes from the OVIS corpus of dialogues over the telephone between a user and the experimental OVIS information system for train travel in the Netherlands.[8]

In (B6) the transcription of the dialogue is shown with its segmentation into functional segments. Segment boundaries are indicated by'|'; end-of-turn boundaries are omitted. In the case of overlapping functional segments, the notation [m] … [/m] is used to indicate begin and end.

As in example (B3) above, the annotation makes use of the rhetorical relation 'elaborate', which as such is not part of this standard, to show how rhetorical relations can be annotated in DiAML.

(B6) *User – System dialogue (with OVIS information system) - segmentation*

1. S: [m1.1:] good morning | [m1.2:] this is the automatic information system of Public Transport Travel Information | [m1.3:] this system can only give information about train travel. | [m1.4:] from which station to which station do you want to travel?
2. U: [m2:] from Harwich to York.
3. S: [m3.1:] when do you want to travel [m3.2:] from Harwich to York? [/m3.2] [/m3.1]
4. U: [m4.1:] uh... | [m4.2:] Monday evening
5. S: [m5:] so you want to leave between eighteen and twenty-four hours?
6. U: [m6.1:] [m6.2:] uh [/m6.1] yes [/m6.2] [m6.3:] Monday evening
7. S: [m7.1:] you said that you do not want to leave on Thursday April twenty-six between eighteen and twenty-four hours, | [m7.2:] can you please correct me?
8. U: [m8.1:] yes | I would like to leave on Monday... | [m8.2:] April twenty-six.
9. S: [m9.1:] what time do you want to travel [m9.2:] the day after tomorrow? [/m9.2] [/m9.1]
10. U: [m10.1:] at [m10.2:] uh... [/m10.2] around eight o'clock [/m10.1]
11. S: [m11:] so you want to leave at eight o'clock in the morning?
12. U: [m12.1:] oh yes | [m12.2:] sorry | [m12.2:] eight o'clock in the evening.
13. S: [m13:] so you want to leave at twenty o'clock?
14. U: [m14:] yes
15. S: [m15.1:] I have found the following connection. | [m15.2:] 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. |[m15.3:] would you like me to repeat the connection?
16. U: [m16.1:] oh no | [m16.2:] thank you
17. S: [m17:] would you like to know another connection?
18. U: [m18:] do you have another connection?
19. S: [m19.1:] I will have a look for you |.[m19.2:] no, I don't think that there is something earlier
20. U: [m20:] is there an earlier possibility?

[8] http://www.let.rug.nl/vannoord/Ovis/ The dialogue was originally in Dutch.

21. S: [m21.1:] I'm sorry | [m21.2:] there's no earlier possibility.
22. U: [m22:] thank you very much
23. U: m23.1:] Public Transportation Information wishes you a pleasant journey | [m23.2:] goodbye

(B7) *DiAML-XML annotation*

```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#m1.1" sender="#s" addressee="#u"
        communicativeFunction="initGreeting"/>
<dialogueAct xml:id="da2" target="#m1.2" sender="#s" addressee="#u"
        communicativeFunction="selfIntroduction"/>
<dialogueAct xml:id="da3" target="#m1.3" sender="#s" addressee="#u"
        communicativeFunction="inform" dimension="task"/>
<rhetoricalLink dact="#da3" rhetoRelatum="#da2" rhetoRel="elaborate"/>
<dialogueAct xml:id="da4" target="#m1.4" sender="#s" addressee="#u"
        communicativeFunction="setQuestion" dimension="task"/>
<dialogueAct xml:id="da5" target="#m2" sender="#u" addressee="#s"
        communicativeFunction="answer" dimension="task"
        functionalDependence"#da4"/>
<dialogueAct xml:id="da6" target="#m3.1" sender="#s" addressee="#u"
        communicativeFunction="setQuestion" dimension="task"/>
<dialogueAct xml:id="da7" target="#m3.2 sender="#s" addressee="#u"
        communicativeFunction="autoPositive" feedbackDependence="#da5"/>
<dialogueAct xml:id="da8" target="#m4.1" sender="#u" addressee="#s"
        communicativeFunction="stalling"/>
<dialogueAct xml:id="da9" target="#m4.2" sender="#u" addressee="#s"
        communicativeFunction="answer" dimension="task" feedbackDependence="#da7"/>
<dialogueAct xml:id="da10" target="#m5" sender="#s" addressee="#u"
        communicativeFunction="checkQuestion" dimension="autoFeedback”/>
<dialogueAct xml:id="da11" target="#m6.1" sender="#u" addressee="#s"
        communicativeFunction=”stalling”/>
<dialogueact xml:id=”da12” target=”#m6” sender=”#u# addressee=”#s”
        communicativeFunction=”answer” dimension=”alloFeedback”
        certainty=”uncertain” functionalDependence=”#da10”/>
<dialogueAct xml:id=”da13” target=”#m6.3” sender=”#u” addressee=”#s”
        communicativeFunction=”inform” dimension=”alloFeedback”/>
<rhetoricalLink dact=”#da13” rhetoRelatum=”#da12” rhetoRel=”elaborate”/>
<dialogueAct xml:id="da14" target="#m7.1" sender="#s" addressee="#u"
        communicativeFunction="inform" dimension="autoFeedback"/>
<dialogueAct xml:id="da15" target="#m7.2" sender="#s" addressee="#u"
        communicativeFunction="request" dimension="autoFeedback”/>
<dialogueAct xml:id="da16" target="#m8.1" sender="#u" addressee="#s"
        communicativeFunction="acceptRequest" dimension="alloFeedback"
        functionalDependence="#da15"/>
<dialogueAct xml:id="da17" target="#m8.2" sender="#u" addressee="#s"
        communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da18" target="#m8.3" sender="#u" addressee="#s"
        communicativeFunction="inform" dimension="task"
<rhetoricalLink dact="#da18" rhetoRelatum="#da17" rhetoRel="elaborate"/>
<dialogueAct xml:id="da19" target="#m9.1" sender="#s" addressee="#u"
        communicativeFunction="setQuestion"   dimension="task"/>
<dialogueAct xml:id="da20" target="#m9.2" sender="#s" addressee="#u"
        communicativeFunction="autoPositive" feedbackDependence="#fda17 da18"/>
<dialogueAct xml:id="da21" target="#m10.1" sender="#u" addressee="#s"
        communicativeFunction="answer" dimension="task"
        functionalDependence="#da19"/>
<dialogueAct xml:id="da22" target="#m10.2" sender="#u" addressee="#s"
        communicativeFunction="stalling"/>
```

```
<dialogueAct xml:id="da23" target="#m11" sender="#s" addressee="#u"
        communicativeFunction="checkQuestion" dimension="autoFeedback"
        feedbackDependence="#m10.2"/>
<dialogueAct xml:id="da24" target="#m12.1" sender="#u" addressee="#s"
        communicativeFunction="inform" dimension="autoFeedback”/>
<dialogueAct xml:id="da25" target="#m12.2" sender="#u" addressee="#s"
        communicativeFunction="apology"/>
<dialogueAct xml:id="da26" target="#m12.3" sender="#u" addressee="#s"
        communicativeFunction="disconfirm" dimension="alloFeedback"
        functionalDependence="#da23"/>
<dialogueAct xml:id="da27" target="#m12.3" sender="#u" addressee="#s"
        communicativeFunction="correction" dimension="task"
        functionalDependence="#da23"/>
 <dialogueAct xml:id="da28" target="#m13" sender="#s" addressee="#u"
        communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<dialogueAct xml:id="da29" target="#m14" sender="#u" addressee="#s"
        communicativeFunction="confirm" dimension="alloFeedback"
        functionalDependence="#da26"/>
<dialogueAct xml:id="da230" target="#m15.1" sender="#s" addressee="#u"
        communicativeFunction="interactionStructuring"/>
<dialogueAct xml:id="da31" target="#m15.2" sender="#s" addressee="#u"
        communicativeFunction="answer" dimension="task"
        functionalDependence="#da5 #da9 #da27"/>
<dialogueAct xml:id="da32" target="#m15.3" sender="#s" addressee="#u"
        communicativeFunction="offer" dimension="alloFeedback"/>
<dialogueAct xml:id="da33" target="#m16.1" sender="#u" addressee="#s"
        communicativeFunction="declineOffer" dimension="autoFeedback"
        functionalDependence="#da32"/>
<dialogueAct xml:id="da34" target="#m16.2" sender="#u" addressee="#s"
        communicativeFunction="thanking"/>
<dialogueAct xml:id="da35" target="#m17" sender="#s" addressee="#u"
        communicativeFunction="offer" dimension="task"/>
<dialogueAct xml:id="da36" target="#m18" sender="#u" addressee="#s"
        communicativeFunction="propositionalQuestion" dimension="task"/>
<dialogueAct xml:id="da37" target="#m19.1" sender="#s" addressee="#u"
        communicativeFunction="inform" dimension="discourseStructuring"/>
<dialogueAct xml:id="da38" target="#m19.2" sender="#s" addressee="#u"
        communicativeFunction="answer" dimension="task"
        qualifier="uncertain" functionalDependence="#da36"/>
<dialogueAct xml:id="da39" target="#m20" sender="#u" addressee="#s"
        communicativeFunction="propositionalQuestion" dimension="task"/>
<dialogueAct xml:id="da40" target="#m21.1" sender="#s" addressee="#u"
        communicativeFunction="apology"/>
<dialogueAct xml:id="da41" target="#m21.2" sender="#s" addressee="#u"
        communicativeFunction="answer" dimension="task" functionalDependence="#da39"/>
<dialogueAct xml:id="da42" target="#m22" sender="#u" addressee="#s"
        communicativeFunction="thanking"/>
<dialogueAct xml:id="da43" target="#m23.1" sender="#s" addressee="#u"
        communicativeFunction="interactionStructuring" dimension="discourseStructuring"/>
<dialogueAct xml:id="da44" target="#m23.2" sender="#s" addressee="#u"
        communicativeFunction="initGoodbye"/>
</diaml>
```

B.3.2 Human-human dialogue

The following excerpt from a dialogue in the HCRC Map Task corpus (Carletta et al., 1996), illustrates the occurrence of nonverbal and multimodal segments. There are several occurrences of heavy breathing (in or out) which have a communicative function; 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 (1) a 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, mentioned in Clause 7.3, 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 the present standard, the encoding in (B9) 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, like head gestures, the representation will be more complex and involve the representation of several features such as duration, direction, speed, and number of repetitions. See Annex D for TEI-compliant encoding of functional segments and the anchoring of dialogue acts in primary data.

(B8)
```
<functionalSegment xml:id="m1" textSegment="#ts1" vocalSegment="#vs1"
        lipSegment="#ls1"/>
    <textSegment xml:id="ts1" from="w11.16" to="w11.17"/>
    <vocalSegment xml:id="vs1" value="inbreath" from="1024135" to="1186249"/>
    <lipSegment xml:id="ls1" value="lipsmack" from="2135618" to="2382694"/>
```

(B9) *Fragment of multimodal human-human Map Task dialogue, segmentation*

1. P1: [m1.1:] okay, | [m1.2:] starting off, | [m1.3:] we are .. above .. a caravan park
2. P2: [m2:] mmhmm
3. P1: [m3.1:] we are going to go due south | [m3.2:] NONVOC_noise ... | [m3.3:] straight south ... | [m3.4:] and NONVOC_noise ... | [m3.5:] then we're going to [m3.6:] g-- turn [/m3.6] turn straight back round and head north... past an old mill ... on the right ... hand side
4. P2: [m4.1:] VOC_outbreath ... | [m4.2:] due south and then back up again
5. P1: [m5.1:] yeah | [m5.2:] south and then straight back up again | [m5.3:] with an old mill on the right | [m5.4:] and you're going to pass on the left-hand side of the mill
6. P2: [m6:] right okay
7. P1: [m7.1:] okay | [m7.2:] and then we're going to turn [m7.3:] ... VOC_inbreath [/m7.3] east [/m7.2]
8. P2: [m8:] mmhmm
9. P1: [m9.1:] not ... straight east ... slightly sort of northeast | [m9.2:] ... VOC_outbreath ...
10. P2: [m10.1:] s-- | [m10.2:] slightly northeast
11. P1: [m11.1:] slightly slightly yeah | [m11.2:] very slightly | [m11.3:] VOC_inbreath ... | [m11.4:] and we're going to continue straight along [m11.5:] ... GES_lipsmack VOC_inbreath ... um ... [/m11.5] quite a wee distance on that course and then we're going to turn north again" [/m11.4]
12. P2: [m12.1:] right | [m12.2:] mmhmm
13. P1: [m13.1:] and ... | [m13.2:] immediat-- | [m13.3:] well | [m13.4] a distance below that turning point there's a fenced meadow | [13.5:] ... VOC_inbreath ... | [m13.6:] but you should be avoiding that by quite a distance
14. P2: [m14:] okay
15. P1: [m15.1:] okay | [m15.2:] so we've turned | [m15.3:] and we're going up north again

(B.10) *DiAML-XML annotation of the dialogue in (B.9)*

```
<diaml xmlns:"http://www.iso.org/diaml/">
<dialogueAct xml:id="da1" target="#m1.1" sender="#p1" addressee="#p2"
      communicativeFunction="turnTake"/>
<dialogueAct xml:id="da2" target="#m1.1" sender="#p1" addressee="#p2"
      communicativeFunction="opening"/>
<dialogueAct xml:id="da3" target="#m1.2" sender="#p1" addressee="#p2"
      communicativeFunction="interactionStructuring"/>
<dialogueAct xml:id="da4" target="#m1.3" sender="#p1" addressee="#p2"
      communicativeFunction="inform" dimension="task"/>
```

```
<dialogueAct xml:id="da5" target="#m2" sender="#p2" addressee="#p1"
      communicativeFunction="autoPositive" feedbackDependence="#da4"/>
<dialogueAct xml:id="da6" target="#m3.1" sender="#p1" addressee="#p2"
      communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da7" target="#m3.2" sender="#p1" addressee="#p2"
      communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da8" target="#m3.2" sender="#p1" addressee="#p2"
      communicativeFunction="stalling"/>
<dialogueAct xml:id="da9" target="#m3.3" sender="#p1" addressee="#p2"
      communicativeFunction="inform" dimension=”task”/>
<drLink arg1=”#da9” arg2=”#da6” rel=”elaborate”>
      <argRole arg=”#da9” role=”specific”/>
      <argRole arg=”#da6” role=”broad”/>
</drLink>
<dialogueAct xml:id="da10" target="#m3.4" sender="#p1" addressee="#p2"
      communicativeFunction="stalling"/>
<dialogueAct xml:id="da11" target="#m3.4" sender="#p1" addressee="#p2"
      communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da13" target="#m3.5" sender="#p1" addressee="#p2"
      communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da12" target="#m3.6" sender="#p1" addressee="#p2"
      communicativeFunction="selfCorrection"/>
<dialogueAct xml:id="da14" target="#m4.1" sender="#p2" addressee="#p1"
      communicativeFunction="turnTake"/>
<dialogueAct xml:id="da15" target="#m4.2" sender="#p2" addressee="#p1"
      communicativeFunction="stalling"/>
<dialogueAct xml:id="da16" target="#m4.3" sender="#p2" addressee="#p1"
      communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<dialogueAct xml:id="da17" target="#m5.1" sender="#p1" addressee="#p2`'
      communicativeFunction="confirm" dimension="alloFeedback"/>
<dialogueAct xml:id="da18" target="#m5.2" sender="#p1" addressee="#p2"
      communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da19" target="#m5.3" sender="#p1" addressee="#p2"
      communicativeFunction="inform" dimension="task"/>
<drLink arg1="#da19" arg2="#da18" rel="elaborate">
      <argRole arg=”#da19” role=”specific”/>
      <argRole arg=”#da18” role=”broad”/>
</drLink>
<dialogueAct xml:id="da20" target="#m5.4" sender="#p1" addressee="#p2"
      communicativeFunction="instruct" dimension="task"/>
<drLink arg2="#da20" arg2="#da18" rel=”elaborate”>
      <argRole arg=”#da20” role=”specific”/>
      <argRole arg=”#18” role=”broad”/>
</drLink>
<dialogueAct xml:id="da21" target="#m6" sender="#p2" addressee="#p1"
      communicativeFunction="autoPositive" feedbackDependence="#da20"/>
<dialogueAct xml:id="da22" target="#m7.1" sender="#p1" addressee="#p2"
      communicativeFunction="autoPositive" feedbackDependence="#da21"/>
<dialogueAct xml:id="da23" target="#m7.1" sender="#p1" addressee="#p2"
      communicativeFunction="turnGrab"/>
<dialogueAct xml:id="da24" target="#m7.2" sender="#p1" addressee="#p2"
      communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da25" target="#m7.3" sender="#p1" addressee="#p2"
      communicativeFunction="stalling"/>
<dialogueAct xml:id="da26" target="#m7.3" sender="#p1" addressee="#p2"
      communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da27" target="#m8" sender="#p2" addressee="#p1"
      communicativeFunction="autoPositive" feedbackDependence="#da24"/>
<dialogueAct xml:id="da28" target="#m9.1" sender="#p1" addressee="#p2"
```

```
        communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da29" target="#m9.2" sender="#p1" addressee="#p2"
        communicativeFunction="stalling"/>
<dialogueAct xml:id="da30" target="#m9.2" sender="#p1" addressee="#p2"
        communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da31" target="#m10.1" sender="#p2" addressee="#p1"
        communicativeFunction="turnGrab"/>
<dialogueAct xml:id="da32" target="#m10.2" sender="#p2" addressee="#p1"
        communicativeFunction="checkQuestion" dimension="autoFeedback"/>
<dialogueAct xml:id="da33" target="#m11.1" sender="#p1" addressee="#p2"
        communicativeFunction="confirm" dimension="alloFeedback"
        functionalDependence="#da32"/>
<dialogueAct xml:id="da34" target="#m11.2" sender="#p1" addressee="#p2"
        communicativeFunction="inform" dimension="task"/>
<drLink arg1="#da34" arg2="da33" rel="elaborate">
        <argRole arg="”#da34” role=”specific”/>
        <argRole arg="”#da33” role=”broad”/>
</drLink>
<dialogueAct xml:id="da35" target="#m11.3" sender="#p1" addressee="#p2"
        communicativeFunction="stalling"/>
<dialogueAct xml:id="da36" target="#m11.3" sender="#p1" addressee="#p2"
        communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da37" target="#m11.4" sender="#p1" addressee="#p2"
        communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da38" target="#m11.5" sender="#p1" addressee="#p2"
        communicativeFunction="stalling"/>
<dialogueAct xml:id="da39" target="#m11.5" sender="#p1" addressee="#p2"
        communicativeFunction="turnKeep"/>
 <dialogueAct xml:id="da40"     target="#m12.1" sender="#p2" addressee="#p1"
        communicativeFunction="autoPositive" feedbackDependence="#da38"/>
<dialogueAct xml:id="da41"     target="#m12.2" sender="#p2" addressee="#p1"
        communicativeFunction="autoPositive” feedbackDependence="#da38"/>
<dialogueAct xml:id="da42" target="#m13.1" sender="#p1" addressee="#p2"
        communicativeFunction="turnTake"/>
<dialogueAct xml:id="da43" target="#m13.1" sender="#p1" addressee="#p2"
        communicativeFunction="stalling"/>
<dialogueAct xml:id="da44" target="#m13.2" sender="#p1" addressee="#p2"
        communicativeFunction="retraction"/>
<dialogueAct xml:id="da45" target="#m13.3" sender="#p1" addressee="#p2"
        communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da46" target="#m13.4" sender="#p1" addressee="#p2"
        communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da47" target="#m13.5" sender="#p1" addressee="#p2"
        communicativeFunction="stalling"/>
<dialogueAct xml:id="da48" target="#m13.5" sender="#p1" addressee="#p2"
        communicativeFunction="turnKeep"/>
<dialogueAct xml:id="da49" target="#m13.6" sender="#p1" addressee="#p2"
        communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da50" target="#m14" sender="#p2" addressee="#p1"
        communicativeFunction="autoPositive" feedbackDependence="#da49"/>
<dialogueAct xml:id="da51" target="#m15.1" sender="#p1" addressee="#p2"
        communicativeFunction="autoPositive" feedbackDependence="#da50"/>
<dialogueAct xml:id="da52" target="#m15.2" sender="#p1" addressee="#p2"
        communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da53" target="#m15.3" sender="#p1" addressee="#p2"
        communicativeFunction="instruct" dimension="task"/>
</diaml>
```

Annex C Formal definition of DiAML (normative)

C.1 Overview

The Dialogue Act Markup Language DiAML has been designed in accordance with the ISO Linguistic Annotation Framework (LAF, ISO 24612:2009), 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. 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 proposed originally by Bunt (2010) and developed in more detail in ISO 24617-6 (Principles od semantic annotation), according to which the syntax specification of an annotation language 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 a metamodel, like the one shown in Figure 1. This standard specifies a reference representation format for DiAML annotation structures using 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 and form annotation structures.

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

- *DP:* dialogue participants;
- *Dim:* dimensions;
- *CF:* communicative functions, with a specified subset of responsive communicative functions;
- *M:* markables, divided into *FS:* functional segments and *RS*: reference segments;
- *QV:* a set of finite sets $Q_1,.. Q_k$, of qualifiers.
- *RR:* rhetorical relations; *AR*: argument roles; and a function ρ from *RR* to *AR* .

b. 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.

b1. 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 the functional segment it is anchored to, and how it relates to other acts in the dialogue. Formally, an entity structure is a pair $\langle m,\alpha\rangle$ consisting of a functional segment (markable) *m*, and a 'dialogue act structure' α.

A *dialogue act structure* contains the characterizes a single dialogue act, which is a 7-tuple $\langle S, A, H, d, f, q, E\rangle$ specifying the sender, the addressee(s), possibly other participants, a dimension, a communicative function, qualifiers (optional), and functional and feedback dependences (if any).

b2. Link structures. A link structure is either a triple $\langle \varepsilon, E, R\rangle$ consisting of an entity structure ε, a set *E* of one or more entity structures, and a rhetorical relation *R*, which relates the dialogue act in ε to those in *E*, or a nested triple $\langle\langle \varepsilon,\rho1\rangle, \langle E, \rho2\rangle, R\rangle$, where $\rho1$ and $\rho2$ are the argument roles of the rhetorical relation *R*.

C.3 Concrete syntax

The concrete syntax is defined in accordance with the methodology for defining semantic annotation languages described in ISO. 24617-6 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.[9]

The DiAML-XML concrete syntax specification consists of a vocabulary, specifying names of XML tags, attributes, and values for the various ingredients of the conceptual inventory, and a specification of the possible ways of combining these elements in XML representation structures. 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*.

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 one or more dialogue participants (if it refers for example to two participants, identified as p1 and p2, then the attribute has the value "#p1 #p2");
- the optional attribute @otherParticipant, whose values refer to one or more of dialogue participants;
- the obligatory attribute @communicativeFunction, whose value names one of the communicative functions defined in this standard;
- 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 it is a dimension-specific function. Its value names one of the dimensions defined in this standard;
- the attribute @functionalDependence, which is obligatory for dialogue acts with a responsive communicative function. It does not apply in other cases; its values refer to one or more dialogue acts.
- the attribute @feedbackDependence, which is obligatory for dialogue acts with a dimension-specific function in the dimensions Auto-Feedback, Allo-Feedback, Own Communication Management, and Partner Communication Management, and which is not applicable in other cases. Is values refer to one or more dialogue acts or reference segments
- the optional attributes @certainty, @conditionality, and @sentiment, whose values specify one of the communicative function qualifiers defined in this standard.

A *link structure representation* is either an element called <rhetoricalLink>, which has the obligatory attributes @dact, @rhetoRelatum, and @rhetoRel, or an element caelled < drLink >, which has the obligatory attributes @arg1, @arg2, and @rel, and embedded <argRole> elements. The annotated dialogues in Annex B contain several illustrations of the use of both types of link structure representations.

C.4 Semantics

C.4.1 Semantics of dialogue acts and communicative functions

DiAML has a formal semantics defined in terms of information-state updates. The details of such a semantics depend on the implementation of information states. The specification of the DiAML semantics makes no further assumptions than that an information state has various components representing different kinds of information, an assumption that is commonly made in proposals for information states[10], and that an information state has a part (called the 'pending context') for buffering update information that needs to be checked for consistency

[9] See Bunt (2010) for formal definitions and proofs.

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

before it can be 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 or that of other participants (like bystanders, or the members of an audience). To simplify matters, only the update of a single addressee's information state is considered in this specification.

The most important structure defined by the DiAML abstract syntax, the dialogue act structure, is a *functional* characterization of a dialogue act. It does not correspond to a full-blown dialogue act, since it does not include a 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 $\langle m,\alpha\rangle$, containing a dialogue act structure α, to the semantic content of the dialogue act For a dialogue act which has no functional or feedback dependences this is shown in (C2), where I_{DA} stands for the interpretation function for full-blown dialogue acts, and I_a as before for the interpretation function for DiAML annotation structures. In the absence of any dependences, the semantic content of the dialogue act is the semantic interpretation $\kappa_1(s)$ of the functional segment that expresses the dialogue act.. (The semantics of dependence relations is considered below). The result is an information state update operation that represents the meaning of the dialogue act.

(C2) $I_{DA}(\langle s,\alpha\rangle) = I_a(\alpha)(\kappa_1(s))$

The interpretation $I_a(\alpha)$ of a dialogue act structure α is computed as follows for those structures which do not contain qualifiers (the semantics of qualifiers is considered below), where case a) applies to dialogue acts with a general-purpose communicative function and case b) to acts with a dimension-specific function:

(C3) a) $I_a(\langle S,A,f,d\rangle) = I_a(f)(I_a(S), I_a(A), I_a(d))$

b) $I_a(\langle S,A,f_d\rangle) = I_a(f_d)(I_a(S), I_a(A))$

A dialogue act structure is thus interpreted as the interpretation of its communicative function, applied to (the interpretations of) its sender, its addressee, and, if present, its dimension. As will become clear below, the result is a function that can be applied to a semantic content.

C.4.3 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 precedence relations between their functional segments, where the updates of two coinciding entity (or link) structures are unifiied (∪), 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 precedes β and is unified with β if the two coincide.

(C4) $I_a(\{e_1,.., e_n, L_1,.., L_k\}) = I_a(e_1)$;⁄∪ ..;⁄∪ $I_a(e_n)$;⁄∪.. ;⁄∪ $I_a(L_1)$;⁄∪ .. ;⁄∪ $I_a(L_k)$

The semantic relatedness between dialogue acts, as e.g. visualised in Fig. 2, is brought out in their interpretation as information state updates. Compare, for example, a *Confirm* act and an *Answer*. According to their definitions an *Answer* with semantic content *p* tells an addressee (A) three things: (1) that the sender (S) wants to make the information *p* available to A; (2) that S believes that A wanted to know whether *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 that *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 a single update item. 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 update functions and a fourth one, which expresses the difference between *Confirm* and *Answer*.

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

(C5) $U_{10}(X,Y,D_i,p)$: add to component D_i of *Y*'s pending context the information that participant *X* wants to know whether *p*;

$U_{11}(X,Y,D_i,p)$: 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:*

(C6) I_a(Propositional Question) = $\lambda X.\ \lambda Y.\ \lambda D_i.\ \lambda z.\ U_{10}(X,Y,D_i,z) \cup U_{11}(X,Y,D_i,z)$

The function specified in the right-hand side can be inserted in the interpretation of a dialogue act annotation structure, as defined in (C3). Applied to two participants *Sys* and *Usr* and a task-related question, the result is the update function in (C7), in which Sys_{TaskC} denotes the component of the system's pending context where task-related information is buffered:

(C7) $I_a(Prop.\ Question)(Usr, Sys, TaskC) = \lambda p.\ U_{10}(Usr, Sys, Sys_{TaskC}, p) \cup U_{11}(Usr, Sys, Sys_{TaskC}, p)$

Applied to a value for p, this specifies 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)=9:15, then if *Sys* understands *Usr* correctly, the component Sys'_{TaskC} of the system's pending context (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 Qualifiers and dependence relations

C.4.3.1 Qualifiers

Qualifiers make the information state updates of communicative functions more specific.They come in two varieties, ‘restrictive' and ‘additive' ones (see Bunt, 2011). 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 enrich a communicative function with additional information, for instance adding that a request is accepted reluctantly. The ‘certainty' and ‘conditionality' qualifiers of this standard are restrictive; ’sentiment' qualifiers are additive.

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

(C8) a. $I_a(\langle f,q_r\rangle) = I_a(f)(I_a(q_r))$

b. $I_a(\langle f, q_a\rangle) = \lambda S.\ \lambda z.\ [I_a(f)(S,z) \cup I_a(q_a)(S,z)]$

This semantics is used to obtain the semantic interpretation of a dialogue act structure with qualifiers:

(C9) $I_a(\langle S,A,d,f,q\rangle) = I_a(\langle f,q>)(I_a(S), I_a(A), I_a(d))$

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

(C10) a. S: The KL 476 departs I think at 19:15.

b. V_a(Inform,*uncertain*) = $[\lambda s.\ \lambda A.\ \lambda B.\ \lambda C_i.\ \lambda p.\ I_a$(Inform)$(A,B,C_i,p,s)](I_a$(*uncertain*))
$= \lambda A.\ \lambda B.\ \lambda C_i.\ \lambda p.\ U_1(A,B,C_i,p,\ weak) \cup U_2(A,B,C_i,p,weak)$

The update schemes U_2 and U_2 are defined as follows:

(C11) $U_1(X,Y,D_i,p,s)$: add to component D_i of Y's pending context the information that participant X wants participant Y to believe that p with belief strength s.
$U_2(X,Y,D_i,p,s)$: add to component D_i of Y's pending context by extending the information that participant X believes that p with belief strength s.

C.4.3.2 Dependence relations

For a dialogue act without dependence relations the semantic content is determined by the functional segment that expresses the dialogue act - see (C2). For a dialogue act that has a dependence relation to a set E of one or more entity structures, the semantic content is determined by the local semantic content of its functional segment together with that in the segments in E. Two combination functions are defined for combining the local semantic information and the information from the elements that the act depends on, one for functional dependences (κ_2) and one for feedback dependences (κ_3). This is expressed in (C12).

(C12) a. functional dependence: $I_{DA}(\langle m,\langle S,A,H,d,f,q,,E\rangle\rangle) = I_a(\langle S,A,H,d,f,q\rangle)(\kappa_2(\kappa_1(s), \{\kappa_1(s_i) \mid e_i \in E\}, f_i, f_a))$

b. feedback dependence: $I_{DA}(\langle m,\langle S,A,H,d,f,q,,E\rangle\rangle) = I_a(\langle S,A,H,d,f,q\rangle)(\kappa_3(s, \{e_i \mid e_i \in E\}, f_a))$

Note that the combinator κ_2 for functional dependences takes as its arguments not only the semantic interpretation of all the functional segments involved, but also the communicative function f_a of the responsive acts and the functions of its antecedents. For example, for an answer act responding to a propositional question, κ_2 delivers as value the semantic content of the question if the answer has the local content TRUE, as in the case of the answer "*Yes*", and it delivers the negation of that if the answer has local content FALSE.

C.4.3.3 Rhetorical relations

Link structures, annotating a rhetorical relation in a dialogue, can only be interpreted if a plug-in for DiAML is defined that specifies a set of relations (and argument roles). Their semantics then defined as 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. For more details on the DiAML semantics see Bunt (2014).

Annex D: DiAML technical schema (normative)

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.

D.1 Overview

This representation relies on a three-level architecture:

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

b) the marking of functional segments from the primary source;

c) the actual dialogue act annotation associated with functional segments.

This annex provides a specification for this third level, 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 standard, but would for example include such relations as `elaborate`, `justify`, `exemplify`..

Functional segments are identified by means of the `functionalSegment` element, which groups together the components of multimodal communicative behaviour that form 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 spoken contribution, following 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 represent a discontinuous stretch of speech. The `@target` attribute 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 example shows how the three levels, mentioned above, may be instantiated in the case of a tokenised 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.

(C.1) 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.

P2's utterance 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 Clause 11.2

The transcription of spoken or multimodal dialogue is not part of this standard, 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>
            <fsspanGrp 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" dimension="task"
        communicativeFunction="setQuestion" conditionality="conditional"/>
    <dialogueAct xml:id="da2" target="#fs2" sender="#p2" addressee="#p1" dimension="task"
        communicativeFunction="answer" functionalDependence="#da1"/>
</diaml>
```

Annex E: Tripartite plug-ins for semantic content and other enrichments
(informative)

E.1 Overview

This annex describes the use of tripartite annotation plug-ins and plug-in interfaces, introduced in Clause 10.2. Section E.2 describes three alternative plug-ins for enriching dialogue act annotations with information about their semantic content. Sections E.3 - E.5 describe other plug-ins, which can be used to enrich the ISO 24617-2 annotation scheme or to make it more accurate in various ways:

- adding the possibility of marking up rhetorical relations in dialogue by importing concepts from the ISO standard 24617-8 for discourse relation annotation;
- adding communicative functions for more fine-grained annotation of feedback and of units of casual talk;
- adding application-specific communicative functions;
- adding the possibility of marking up emotional aspects of the performance of a dialogue act.

E.2 Annotation plug-ins for semantic content

E.2 Plug-ins for semantic content

For the use cases UC3 and UC4 mentioned in Clause 4.2, it is important to have intormation about the semantic content of dialogue acts. The degree of detail in which semantic content should be represented depends on the application domain. For some domains a simple representation as a list of attribute-value pairs may be adequate; for others a representation in terms of events with their participants, time and place may be more appropriate; for more advanced applications it may be necessary to take general aspects of natural language utterance meaning into account, including quantification and modification phenomena.

In any case, the use of a semantic content plug-in PL_c for the host annotation scheme L_a requires a plug-in interface ${}_aY_c$, which can be defined as shown in (E1): the abstract syntax ${}_aAS_c$ introduces the content link structure as a pair consisting of a dialogue act entity structure (‘a’) and a content entity structure (‘c’); the concrete syntax specifies its XML encoding using a <contentLink> element, and the semantics specifies its meaning as the application of the interpretation $I_a(a)$ of the dialogue act structure ‘a’, defined by the semantics of the host annotation scheme, to the argument $I_c(c)$, defined by the plug-in semantics. This semantics reflects the dialogue act theory underlying the ISO 24617-2 annotation scheme, according to which the semantics of a full-blown dialogue act is an update operation on information states, obtained by applying the semantics of the functional characterization of the dialogue act to its semantic content (see (C2) in Annex C; the semantic content is now computed as the interpretation of the content annotation).

(E1) ${}_aY_c = \langle {}_aAS_c, {}_aCS_c,, {}_aSm_c\rangle$, with:

${}_aAS_c, = \langle {}_aCI_c, {}_aAC_c\rangle = \langle \emptyset, \{\langle a,c\rangle \text{ content link structure}\}\rangle$

${}_aCS_c, = \langle {}_aV_c, {}_aCC_c, {}_aF_c\rangle \rangle = \langle\langle \emptyset, \{\text{<contentLink> element}\}, {}_aF_c(\langle a, c\rangle) =$

<contentLink dialAct=${}_aF_c(a)$ semContent= ${}_aF_c(c)$/>$\rangle$

${}_aSm_c, = \langle {}_aM_c, {}_aI_c\rangle = \langle\langle \emptyset, {}_aI_c(\langle a,c\rangle) = I_a(a)(I_c(c))\rangle$

The ISO 24617-2:2012 host annotation system $L_a = \langle AS_a, CS_a, Sem_a\rangle = \langle\langle CI_a, AC_a\rangle, \langle V_a, CC_a, F_a\rangle, \langle M_a, I_a\rangle\rangle$, a content plug-in $PL_c = \langle\langle CI_c, AC_c\rangle, \langle V_c, CC_c, F_c\rangle, \langle M_c, I_c\rangle\rangle$, and an interface ${}_aY_c = \langle\langle \emptyset, \{\langle a,c\rangle\}\rangle, \langle\langle \emptyset, \{\text{<contentLink>}\}, {}_aF_c\rangle, \langle \emptyset, {}_aI_c\rangle\rangle$ together define an extended annotation scheme L_{ac} formed by the unions of their components:

(E2) $L_{ac} = \langle\langle CI_a \cup CI_c, AC_a \cup AC_c\rangle, \langle V_a \cup V_c, CC_a \cup CC_c, F_a \cup F_c\rangle, \langle M_a \cup M_c, I_a \cup I_c \cup \{I_{ac}(\langle a,c\rangle) = I_a(a)(I_c(c))\}\rangle\rangle$

The union of these components forms a useful annotation scheme only if two important properties of the host annotation scheme are preserved: the orthogonality of the set of dimensions and the taxonomical structure of the set of communicative functions.

E.2.1 Attribute-Value plug-in

A simple domain-specific plug-in for semantic content described as (lists of) attribute-value pairs could for example be useful in a travel domain where a journey can be described by a few attribute-value pairs, specifying departure place, destination, travel date, etc. In such a context, when a client P1 says *"I'd like to leave around ten in the morning",* this could be annotated as in (E3b):

(E3) a. P1: *I'd like to leave around ten in the morning* (= markable m1)

b. <avContent xml:id="c1" target="#m1" attribute="departureTime" value="10:00"/>

Underlying the representation in (H3b) is a conceptual inventory that lists the attributes and their possible values, and the definition of entity structures consisting of attribute-value pairs $\langle A_i, v_{ij}\rangle$. The semantics of such an entity structure can be defined as a feature structure $[A_i': v_{ij}']$ which, according to ISO standard 24612 for feature structures, expresses the property $\lambda x.\ A_i'(x) = v_{ij}'$. The variable 'x' in the lambda abstraction can in this domain be thought of as ranging over journeys. The syntax and semantics of such AV-entity structures define a very simple annotation language L_{AV}, the semantics of which is a defined in (E4).

(E4) $I_{AV}(\langle A_i, v_{ij}\rangle) = [I_{AV}\}(A_i,): I_{AV}\}(v_{ij})] = [A_i,': v_{ij}']$

To link an AV-content annotation to dialogue act annotations, the XML element <contentLink>, defined in the interface (E1), can be used to obtain representations of the form (E5).

(E5)
```
<dialogueAct xml:id="da1"target="#m1" speaker="#s" addressee="#a"
        dimension="task"    communicativeFunction="inform"/>
<avContent xml:id="c1" target="#m1" attribute="departureTime" value="10:00/>"
 <contentLink dialAct="#da1" content="#c1"/>
```

The formal specification of the attribute-value content plug-in PL_{AV} for DiAML annotations is as follows:

- **Abstract syntax:** $AS_{AV} = \langle CI_{AV}, AC_{AV}\rangle$:
  - the conceptual inventory CI_{AV} lists attributes and their possible values;
  - AC_{AV}: content entity structures are triples of the form $\langle m, \langle A_i, v_{ij}\rangle\rangle$,[11] with m a markable;
- **Concrete syntax**: $CS_{AV} = \langle V_{AV}, CC_{AV}, F_{AV}\rangle$:
  - the vocabulary VC_{AV} lists names of XML attributes and values;
  - CC_{AV}: specification of XML element <avContent> - see (E3b).
  - encoding function F_{AV}: mapping from CI_{AV} to V_{AV}; encoding of AC_{AV} – entity structures: $F_{AV}(\langle m,\langle A_i,v_{ij}\rangle\rangle$ = <avContent xml:id="c1" target="#m" attribute= "$F_{AV}(A_i)$" value= "$F_{AV}(v_{ij})$"/>.
- **Semantics:** SEM_{AV} uses interpretation function I_{AV} as defined in (E4).

Note that the interface ${}_aY_{AV}$ for connecting an AV plug-in with DiAML, as defined by the schema in (E1), introduces in the abstract syntax 'content link structures', which are pairs $\langle a, c\rangle$ consisting of a dialogue act entity structure and a content entity structure. The semantic component of the interface combines the interpretation functions of the host annotation scheme and the plug-in according to (E6), which says that the interpretation of the dialogue act annotation is applied (as a function) to the argument formed by the interpretation of the content annotation.

(E6) ${}_aI_{AV}(\langle a, c\rangle) = I_a(a)(I_{AV}(c))$

This combination of the two interpretation functions is possible only if the interpretation function I_a of the host language is applicable to the output of the plug-in interpretation function. The interpretation function I_a makes use of elementary context update operators which are defined in a representation-neutral way, just stipulating that the given semantic content should be added to that part of the addressee's information state which

[11] This can of course be extended to finite lists of such triples.

contains information about the task that still has to be verified for consistency with other available information (the addressee's `pending semantic context'). When using this approach in a dialogue system, the elementary update operators must be instantiated for the representation formalism of the contents of the system's information state. The semantic content of dialogue acts has to be represented in a form that fits in with that formalism. For content expressed in the form of feature structures, as is the case for I_{AV}, this no problem. Existing DiAML implementations in dialogue systems, such as Keizer et al. (2011), Malchanau et al. (2019), and Malchanau (2019) use typed feature structures for information state representation, making the implementation of (E5) a straightforward matter.

E.2.2 Plug-in for events and semantic roles

The following more general content plug-in is based on ISO standard 24617-4 for the annotation of semantic roles. The annotation scheme of this standard, a.k.a. `SemAF-SR', marks up semantic information related to the question "Who did what to whom?" assigning semantic roles to the participants in an event. For example, the sentence *"The soprano sang an aria"* is analysed as mentioning a singing event and annotated as in (E9b), where "sing.01" refers to a verb sense in VerbNet:

(E9) a. "The soprano sang an aria"
Markables: m1="The soprano", m2="sang", m3="an aria"

b. <entity xml:id="x1" target="#m1" pred="soprano"/>
<event xml:id="e1" target="#m2" eventFrame="sing.01" type ="accomplishment"/>
<srLink event="#e1" participant="#x1" semRole="agent"/>
<entity xml:id="x2" target="#m1" pred="aria"/>
<srLink event="#e1" participant="#x1" semRole="theme"/>

SemAF-SR interprets such annotations as expressing the existence (or denied existence, in case of a clause with negative polarity) of certain states or events and participants in certain roles. For the example in (E9) the semantics can be expressed by the following DRS:

(E10) [e1, x1, x2 | sing01(e1), soprano(x1), aria(x2), agent(e1,x1), theme(e1,x2)]

The plug-in consists in this case of the abstract and concrete syntax of the SemAF-SR markup language and the semantic interpretation function which produces DRSs like those in (E10). The abstract syntax has a conceptual inventory that lists semantic roles and verb senses by reference to VerbNet, defines entity structures for eventualities and their participants, and defines link structures for relating participants to eventualities in a certain role. The concrete syntax defines XML encodings of the annotation structures defined by the abstract syntax, as illustrated in (E9b).

When defining a content plug-in for information about semantic roles, the question arises whether *all* the information encoded in SemAF-SR annotations should be taken along in the plug-in. This issue regards in particular the reference to event frames for VerbNet verb senses. While this seems appropriate for the purposes of SemAF-SR, it would bring a level of precision to the interpretation of verbs and deverbal nouns which is not pursued for other content words; it may therefore be more appropriate to make this optional in a plug-in, allowing users to choose whether they want to use a conceptual inventory with that level of granularity or a less fine-grained one. The annotation of time and events also needs to be considered: ISO-TimeML (ISO 24617-1) uses a classification of event types that differs from that of SemAF-SR, and includes other detailed information about events that is not considered in SemAF-SR (like tense and aspect). Again, it is not obvious how much of that information would seem appropriate to take along in a plug-in for DiAML.

The simplest content plug-in for semantic roles is one that takes a minimalist approach to event classifications, and uses a simple form like <event xml:id="e2" target="#m3" pred="sing"/> rather than the more fine-grained representations of SemAF-SR or ISO-TimeML. This plug-in (`PL_{SR}') is characterized by the following schema:

- **Abstract syntax**: the conceptual inventory lists the semantic roles defined in the ISO 24617-4 standard and a set of verb senses, distinguishing only senses which differ in their semantic roles; two kinds of entity structures are distinguished, for eventualities and for their participants, and just one kind of link structure, for indicating a semantic role.

- **Concrete syntax**: specifies names for the elements of the conceptual inventory; XML elements for encoding the entity and link structures.

- **Semantics**: translation of entity and link structures and their combination to DRSs.

Formally, this plug-in is defined as follows:

(E11) $PL_{SR} = \langle\langle CI_{SR}, AC_{SR}\rangle, \langle V_{SR}, CC_{SR}, F_{SR}\rangle, \langle M_{SR}, I_{SR}\rangle\rangle$, with

CI_{SR} = SR ∪ EP ∪ DP made up of a set SR set of semantic roles defined in ISO 24617-4 (see Table 1), a set EP of event predicates defined in VerbNet (such as 'sing01'), and a set DP of domain predicates defined in a domain ontology (like 'soprano' and 'aria')

AC_{SR} = {entity structures $\langle m, p\rangle$; link structures $\langle e_{EV}, e_P, R\rangle$}[12]

V_{SR} = names of CI_{SR} elements

CC_{SR} = {<event>, <entity>, <srLink>}

F_{SR} assigns a value in V_{SR} to each element of CI_{SR}[13];

$F_{SR}(\langle m, p\rangle)$ = <entity xml:id="x" target="#m1" pred=" $F_{SR}(p)$""/> if $p \in DP$;

$F_{SR}(\langle m,p\rangle)$ = <event xml:id="x" target="#m1" pred=" $F_{SR}(p)$"/> if $p \in EP$;

$F_{SR}(\langle e_{EV}, e_P, R\rangle)$ = <srLink event="$F_{SR}(e_{EV})$" participant="$F_{SR}(e_p)$" semRole="$F_{SR}(R)$"/>

$I_{SR}(p) = p'$; $I_{SR}(t) = t'$; $I_{SR}(R) = R'$; $I_{SR}(\langle m,p\rangle) = [x \mid p'(x)]$;

$I_{SR}(\langle e_{EV}, e_P, R\rangle) = [\ e\ x \mid R'(e,x))\ ] \cup I_{SR}(e_{EV}) \cup I_{SR}(e_P)$

The interface for using this plug-in in combination with the ISO 24617-2 annotation scheme is the same as the one defined in (E1).

Table E1 Semantic roles defined in ISO 24617-4

	Role	Definition
1.	Agent	Participant in an event who intentionally or consciously initiates an event, and who exists independently of the event.
2.	Beneficiary	Participant in an eventuality that is advantaged or disadvantaged by the eventuality, and that exists independently of the event.
3.	Cause	Participant in an event that initiates the event, but does not act with any intentionality or consciousness; the participant exists inde-pendently of the event.
4.	Goal	Participant in an event that is the (non-locative, non-temporal) end point of an action; the participant exists independently of the event.
5.	Instrument	Participant in an event that is manipulated by an agent, and with which an intentional act is performed; it exists independently of the event.
6.	Partner	Participant in an event that is intentionally or consciously involved in carrying out the event. Participant is not the principal agent of the event, and exists independently of the event.
7.	Patient	Participant in an event that undergoes a change of state, location or condition, is causally involved or directly affected by other participants, and exists independently of the event.
8.	Pivot	Participant in a state that is characterised as being in a certain position or condition throughout that state, and has a major or central role or effect in that state. A pivot is more central to the state than a participant in a theme role, and exists independently of the state.
9.	Purpose	Set of facts or circumstances that an agent wishes or intends to accomplish by performing some intentional action.
10.	Reason	Set of facts or circumstances explaining why a state exists or an event occurs.
11.	Result	Participant in an event that comes into existence through the event; it indicates a terminal point for the event: when that is reached, the event does not continue.

[12] 'e_{EV}' stands for 'event entity structure', e_P' for 'participant entity structure'.

[13] The value assigned to an argument a will be designated by a'.

12.	Setting	Set of (non-locative and non-temporal) facts or circumstances of the occurrence of an event or a state.
13.	Source	Non-locative, non-temporal starting point of an event. The source exists independently of the event.
14.	Theme	Participant in a state or an event that (i) in the case of an event, is essential to the event taking place, but does not have control over the way the event occurs and is not structurally changed by the event, and (ii) in the case of a state, is characterised as being in a certain position or condition throughout the state, and is essential to the state being in effect but not as central to the state as a participant in a pivot role. The theme of a state or event exists independently of the state or event.
15.	Manner	The way or style of performing an action or the degree/strength of a cognitive or emotional state.
16.	Medium	The physical setting, device or channel that allows an event to take place.
17.	Means	Procedure for performing an action in terms of component steps, or a methodology by which an intentional act is performed by an agent. A means does not necessarily exist independently of the event.
18.	Location	Place where an event occurs, or a state is true, or a thing exists.
19.	Initial Location	Participant in an event that indicates the location where an event begins or a state becomes true; initial-location exists independently of the event.
20.	Final Location	Location where an event ends or a state becomes false; final-location exists independently of the event.
21.	Path	Intermediate location or trajectory between two locations, or in a designated space, where an event occurs.
22.	Distance	Length or extent of space that plays a role in an eventuality.
23.	Time	Participant that indicates an instant or a time interval during which a state exists or an event takes place.
24.	Duration	Length or extent of time during which an event occurs or a state is true.
25.	Initial Time	Indication of the point in time when an event begins or a state becomes true.
26.	Final Time	Indication of a point in time when an event ends or a state ceases to be true.
27.	Amount	Quantity of something other than time or space, or number of objects of a certain kind, which plays a role in an event or a state.
28.	Attribute	Property that an event or state associates with one of the other participants. '(We [agent e1]) will (paint e1) (the front door [theme e1]) (dark green [attribute e1])'

E.2.3 Plug-in for events, participants, and quantification

A plug-in for the semantic content of dialogue acts is more general and more powerful as it takes more aspects into account of the meanings of phrases, clauses, sentences, and other natural language structures that may express semantic content. On top of the identification of events with their time and place and participants with their respective roles, the interpretation of quantifier and modifier structures forms the most important source of semantic information. The ISO standard 24617-12 under development can be the basis of a powerful plug-in for this type of information. See Bunt et al. (2018) and Bunt (2019) for the design of an annotation scheme or quantification and modification, and Bunt (2018) for a preliminary version of an annotation schemes.

E.3 Plug-ins for rhetorical relations

This second edition of ISO 24617-2 supports the marking up of rhetorical relations in dialogue in a more fine-grained way than the first edition, but does not specify any particular set of relations to be used. A plug-in for discourse relations does not require the introduction of any entity structures or link structures, since these have been defined in DiAML, and for the same reason no specification of an interface is required, but just the specification of a set of rhetorical relations and their argument roles. Note that such a plug-in still has the tripartite structure, but in a very simple form: the specification of a rhetorical relation appears in three places, for example: '*Cause'* occurs as the causal relational concept in the conceptual inventory of the abstract syntax; the string "cause" occurs as the value of an XML attribute in the vocabulary of the concrete syntax, and 'Cause' occurs as a binary predicate constant in the semantics.

The tripartite plug-in specified here takes the ISO 24617-8:2015 (DR-core) annotation scheme for rhetorical relations as the point of departure. The DR-core set consists of 20 relations which include 'functional dependence' and 'feedback dependence', which in this standard are treated differently. To the remaining 18 relations, the relation 'Evaluation' has been added, which has been felt to be missing. Table E2 lists the resulting 19 relations with their definitions, which describe their semantics in an informal way. Many other relations that are distinguished in other annotation schemes can be seen as special cases of DR-core relations, for example

'Explanation' as a case of 'Cause', 'Juxtaposition' as a case of 'Contrast', and 'Specification' as a case of 'Elaboration'.

Table E2. Rhetorical relations, based on ISO 24617-8.

	Relation	*Definition*
1.	Cause	The second argument provides a reason why the first argument occurs or holds true.
2.	Condition	The first argument is an unrealized situation which, when realized, would lead to the situation that forms the second argument.
3.	Negative Condition	The first argument is an unrealized situation which, when not realized, would lead to the situation that forms the second argument.
4.	Purpose	The second argument is the goal or purpose of the situation that forms the first argument.
5.	Manner	The second argument describes how the first argument comes about or occurs.
6.	Concession	The second argument cancels or denies an expected causal relation between the first argument and the negation of the second.
7.	Contrast	One or more differences between the two arguments are highlighted with respect to what each predicates as a whole or about some entities they mention.
8.	Exception	The second argument indicates one or more circumstances in which the situation that forms the first argument does not hold.
9.	Similarity	One or more similarities between the two arguments are highlighted with respect to what each predicates as a whole or about some entities they mention.
10.	Substitution	The two arguments are alternatives, the situation of the second argument being the favored or chosen alternative.
11.	Conjunction	The two arguments bear the same relation to some other situation evoked in the discourse. Their conjunction indicates that they both hold with respect to that situation.
12.	Disjunction	The two arguments bear the same relation to some other situation evoked in the discourse. Their disjunction indicates that they are non-exclusive alternatives with respect to that situation.
13.	Exemplification	The second argument is a situation that is an element of the set of situations described by the first argument. Arg1 describes a set of situations.
14.	Elaboration	The two arguments are the same situation, but the second argument is specified in more detail.
15.	Restatement	The two arguments are the same situation, but viewed from different perspectives.
16.	Synchrony	The two arguments form two temporally overlapping situations. All forms of overlap are included.
17.	Asynchrony	The first argument temporally precedes the second.
18.	Expansion	The two arguments are distinct situations that involve some shared entities; the second argument expands a narrative of which the first argument forms part of a certain narrative and Arg1 is a part, or expanding on the setting relevant for interpreting Arg1.
19.	Evaluation	The second argument provides an opinion on the social, esthetic, economic, or other qualities of the first argument.

This leads to the following plug-in PL_{DR} for discourse relations in dialogue:

Abstract syntax:

- CI_{DR} = the relations defined in DR-core, i.e. the set {*Cause, Condition, Negative Condition,… Expansion, Evaluation*}, corresponding to the contents of the left column in Table E2, and their argument roles, as specified in Table E3 below.

Concrete syntax:

- XML names for the relations in the conceptual inventory and for their argument roles. For convenience, font variations of the same strings are used: "cause", "condition", "negativeCondition",… and "reason", "result", etc. (Not all relations have distinct argument roles. For example, the relations *Contrast* and *Similarity* have two arguments that play semantically identical roles. In such cases, the

arguments are named “arg1” and “arg2”, and by convention the label “arg1” is given to the argument that occurs first in the discourse.)

Semantics:

- The meanings of the rhetorical relations and their argument roles are listed in Table E2 in an informal way. See also Annex C section C.4.3.3 for their use in the semantic interpretation of rhetorical link structures.

Table E3. Rhetorical relations and argument roles in the PL_{DR} concrete syntax.

Rhetorical relation	*First argument*	*Second argument*
cause	reason	result
condition	antecedent	consequent
negativeCondition	negatedAntecedent	consequent
purpose	enablement	goal
manner	means	achievement
concession	expectationRaiser	expectationDenier
exception	regular	exclusion
substitution	disfavoredAlternative	favoredAlternative
exemplification	set	instance
elaboration	broad	specific
asynchrony	before	after
expansion	narrative	expander
evaluation	situation	judgement
contrast, conjunction, disjunction, restatement, similarity, synchrony	arg1	arg2

Note that this plug-in is especially powerful in combination with a content-plug; In that case the distinction between ‘semantic’ and ‘pragmatic’ variants of discourse relations cannot be expressed, but the argument roles of discourse relations can be, making the annotations more precise than those of ISO 24617-2:2012.

Combination of this plug-in L_{DR} with the second edition of ISO 24617-2 and with a plug-in for semantic content does not require a plug-in interface, since the link structure in the abstract syntax of L_{DR} and its <drLink> encoding, as defined in DiAML in this edition already expect both dialogue acts and their semantic contents as possible arguments.

E.4 Plug-ins for additi0nal communicative functions

E.4.1 Overview

According to the general structure of a plug-in, $PL_a = \langle A_a, CS_a, SEM_a \rangle$, with $AS_a = \langle CI_a, AC_a \rangle$; $CS_a = \langle CV_a, CC_a, F_a \rangle$, and $Sm_a = \langle M, I_a \rangle$, a plug-in PL_{CF} for adding certain communicative functions would have a very simple specification since no new entity structures or link structures are needed, but only the following components:

- **Abstract syntax:** conceptual inventory CI_{CF} listing the new functions;
- **Concrete syntax:** CV_{CF}: corresponding XML function names, and mapping from CI_{CF} to CV_{CF};
- **Semantics:** the context-update semantics $I_a(f_j)$ for every $f_j \in CI_{CF}$.

Since no new entity structures or link structures are introduced, there is no need for a plug-in interface in this case.

E.4.2 Plug-in for application-specific communicative functions

The ISO 24617-2 standard was designed to be domain-independent, and for this reason does not include communicative functions that would be specific for a certain application domain. All its communicative functions are either general-purpose or belong to one of the dialogue control dimensions. The general-purpose functions of the scheme form a powerful battery of functions for use in any application, but some

applications may benefit from the availability of additional, domain-specific communicative functions. This is another area where plug-ins can be useful.

One important question that arises when designing a plug-in for domain-specific types of dialogue acts is how these functions relate to those of the host annotation scheme. It was noted in Clause 10 that any plug-in should respect the structural properties of the host annotation scheme, in particular the orthogonality of the dimensions and the semantic connectedness of the communicative functions. For example, in a negotiation domain one finds bids, counter-bids, accepts and rejects of (counter-)bids, and so on. Such acts can be viewed as special cases of the general-purpose functions Offer and AddressOffer, and they would thus fit well within the taxonomy of the standard.

E4.3 Plug-in for additional SOM functions

Most annotation schemes, including ISO 24617-2:2018, have been designed for annotating dialogue where the participants had a clear purpose, such as finding a train connection, making an appointment, designing a remote control, or finding a route on a map. Natural everyday conversations such as a chat with a neighbour or with a colleague at the coffee machine often do not have such a clear task as their motivation, but are aimed at a social purpose, such as establishing a pleasant atmosphere or maintaining a good relationship. Task-related dialogues often have an initial phases in which the participants are exchanging small talk before getting to a specific task, and such initial phases have often been omitted in dialogue corpora, where the initial small talk is viewed as occurring 'before' the 'actual' dialogue. An exception is the ADELE corpus of casual conversations (Gilmartin et al., 2018) in the form of textual chat dialogues. The dialogues in this corpus often have rather elaborate initial phases with greetings and discussions of each other's health, and sometimes also an extended leavetaking phase with various kinds of greetings and well-wishing. In order to annotate the communicative functions in such phases in a satisfactory way, the DIT++ annotation scheme (Release 5.2) includes several dimension-specific functions in the Social Obligations Management dimension.[14] Table E4 lists these functions and gives some examples. These functions can be used to specify a tripartite plug-in PL_{SOM} as follows.

- **Abstract syntax:** conceptual inventory CI_{SOM} listing the new functions; as in the case of PL_{DR}, font variants of the same name are used in the abstract and the concrete syntax - see the leftmost column in Table E4;
- **Concrete syntax:** CV_{SOM}: corresponding XML function names;
- **Semantics:** the context-update semantics $I_a(f_j)$ for every $f_j \in CI_{SOM}$. Table E4 specifies the semantics of these communicative functions in a semi-formal way, like the data categories in Annex F.

Table E4. Additional communicative functions for Social Obligations Management (from DIT++).

Communicative Function	*Definition*	*Example*
Follow-on Greeting	The sender, S, wants the addressee, A, to know that S has established the presence and the identity of the addressee, A.	"Hi Anne."
Politeness Question	The sender, S, wants to know the state of well-being of the addressee, A, or of someone close to A.	"How do you do?" "How is your mother?"
Return Politeness Question	The sender, S, wants to know the state of well-being of the addressee, A, or of someone close to A; S responds to a Politeness Question by A.	"And how about you?"
Opening Politeness Statement	The sender, S, wants the addressee, A, to know that S is pleased to meet A.	"Nice to meet you"
Closing Politeness Statement	The sender, S, wants the addressee, A, to know that S is pleased to have met A. S intends to soon close the dialogue.	"It was nice talking to you."
Farewell Wish	The sender, S, wishes the addressee, A, to be in a positive state of well-being, and	"Have a good time."

[14] See https://dit.uvt.nl.?

	intends to close the dialogue.	

E.4.4 Plug-in for additional feedback functions

For using this standard in applications that instantiate use case UC4, i.e. for the generation of dialogue acts in a dialogue system, it may be useful to have more fine-grained feedback functions available then the rather coarse positive and negative functions defined in ISO 24617-2:2012. More fine-grained functions are available in the DIT++ annotation scheme,[15] where a distinction is made between five levels of processing at which feedback may be provided or elicited: (1) attention; (2) perception; (3) interpretation (understanding); (4) evaluation; and (5) execution. These functions define a plug-in PL_{FB} as follows.

- **Abstract syntax:** conceptual inventory CI_{FB} listing the new functions; as in the case of PL_{SOM}, font variants of the same name are used in the abstract and the concrete syntax - see Table E5 for the 10 fine-grained auto-feedback functions; similar tables define the allo-feedback functions (10 feedback-providing and 10 feedback eliciting functions);
- **Concrete syntax:** CV_{FB}: corresponding XML function names;
- **Semantics:** the context-update semantics $I_a(f_j)$ for every $f_j \in CI_{FB}$. Table E5 specifies the semantics of these communicative functions in a semi-formal way, like the data categories in Annex F.

Table E5. Fine-grained communicative functions for auto-feedback (from DIT++).

	Communicative Function	*Definition (in brief)*
1.	Attention Positive Auto-Feedback	The speaker has noticed that something was said/done.
2.	Perception Positive Auto-Feedback	The speaker has registered (heard/seen/felt,..) what was said/done.
3.	Interpretation Positive Auto-Feedback	The speaker has interpreted what was said/done.
4.	Evaluation Positive Auto-Feedback	The speaker has evaluated what was said/done.
5.	Execution Positive Auto-Feedback	The speaker has acted on what was said/done.
6.	Attention Negative Auto-Feedback	The speaker has failed to notice that something was said/done.
7.	Perception Negative Auto-Feedback	The speaker has not been able to register (hear/see/feel,..) what was said/done.
8.	Interpretation Negative Auto-Feedback	The speaker has not been able to interpret what was said/done.
9.	Evaluation Negative Auto-Feedback	The speaker has not been able to evaluate what was said/done.
10.	Execution Negative Auto-Feedback	The speaker has not been able to act on what was said/done.

The fine-grained auto- and allo-feedback functions form three hierarchical systems, due to the logical relations between the five levels of processing: acting successfully on what was said presupposes successful evaluation, understanding, and perception of what was said, which all presuppose that the speaker noticed that something was said in the first place. Positive feedback at the highest level of processing ('execution') is thus more specific than evaluation feedback, which in turn is more specific than interpretation feedback, and so on. For negative feedback functions the inverse order applies, since failing to notice that something was said entails being unable to determine what was said, and so on.

E.5 Plug-ins for emotion annotation

The sender of a dialogue act may expresses a certain emotion associated with the performance of the dialogue act, such as amusement, irritation, or disappointment. ISO 24617-2 has no provisions for annotating such aspect. Qualifiers, as used for sentiment and certainty, could also be considered for dealing with a speaker's affective/emotional state, but this would assume that such a state can be characterized in a one-

[15] See https://dit.uvt.nl.

dimensional way, through a single predicate. Such an approach may be useful for some use cases, but is in general too simple.

The W3C recommendation EmotionML is a flexible scheme, designed with the aim of being combined with other annotation schemes. It characterizes emotions as complex entities, including 'emotion categories' such as "anger", "happiness", or "surprise", an intensity value (called 'valence'), and a confidence value, as well as various alternative other ways of describing emotions, notably in terms of 'action tendencies', 'appraisals', and multiple 'dimensions'.[16] An emotion in EmotionML may have components of various categories; for instance, in the snippet (E41), taken from the document https://www.w3.org/TR/emotionml/, an emotion is annotated as being a form of anger with elements of sadness and fear.

(E12)
```
<emotion category-set="http://www.w3.org/TR/emotion-voc/xml#big6">
    <category name="sadness" value="0.3"/>
    <category name="anger" value="0.8"/>
    <category name="fear" value="0.3"/>
</emotion>
```

Observing that there is no general agreement in the community, EmotionML does not provide a single repository of emotion descriptors, but gives users a choice to select a suitable emotion vocabulary in their annotations. In order to promote interoperability, EmotionML provides a number of emotion vocabularies that can be used for this purpose. The guiding principle for selecting emotion vocabularies has been to list vocabularies that are either commonly used in technological contexts, or represent current emotion models from the scientific literature. One of the best known repositories is Ekman's 'big six', (Ekman, 1972), a set of basic emotions with universal facial expressions; emotions that are recognized and produced in all human cultures. Example (E41) shows how this repository or one of the others listed by EmotionML[17] is referenced in an annotation.

EmotionML is defined only at the level of concrete syntax, so it cannot directly be used as a plug-in for ISO 24617-2 dialogue annotation. However, an abstract syntax for EmotionML can be developed using the CASCADES methodology in reverse engineering mode (Bunt, 2016), and a semantics can be added for those parts of EmotionML markups that are truly semantic in nature, in contrast with e.g. confidence values.

An emotion has an experiencer and an object that the emotion is about. The emotional aspect associated with a dialogue act is a relation between the speaker, as the experiencer of the emotion, and (the semantic content of) the dialogue act as the object of the emotion. For example, in (E13) the experiencer of the emotion associated with the acceptance of the preceding offer is participant P2 and the object is the semantic content of this offer and its acceptance, viz. P2 having a cup of coffee.

(E13) a. P1: Would you like to have a cup of coffee? (= markable m1)
P2: That would be wonderful! (= markable m2)

b.
```
<dialogueAct xml:id="da1" target=#m1" speaker=#p1" addressee="#p2"
    dimension="social" communicativeFunction="offer"/>
<contentLink dialAct="#da1" content="#e1"/>
<dialogueAct xml:id="da2" target="#m2" speaker="#p2" addressee="#p1"
    dimension="social" communicativeFunction="acceptOffer"
    functionalDependence="#da1"/>
<event xml:id="e1" target="#m2" pred="have-coffee"/>
<srLink event="#e1" participant="#p2" semRole="agent"/>
<contentLink dialAct="#da2" content="#e1"/>
<emotion xml:id="em1" target=`"#m2" category="happiness" value="0.8"/>
<emoLink holder="#p2" object="#e1" emotion="#em1"/>
```

[16] See Emotion Markup Language (EmotionML) 1.0, W3C Recommendation 22 May 2014. Available at http://www.w3.org/TR/2014/REC-emotionml-20140522/. See also Burkhardt, F., Pelachaud, C., Schuller, Bj., and Zovato, E. (2017) EmotionML. In D. Dahl (ed.) Multimodal Interaction with W3C Standards. Springer, Cham, pp. 65-80.

[17] The EMotionML repositories are listed in the W3C document 'Vocabularies for EmotionML', https://www.w3.org/TR/emotion-voc/.

A simple plug-in PL_e for adding emotion annotation to dialogue acts, based on EmotionML, can be defined as follows.

Abstract syntax:

- CI_e = set of emotion categories; intensity values (any floating point number in the interval [0,1];
- Entity structures: pairs ⟨m, ⟨c, v⟩⟩ consisting of an emotion category and an intensity value.

Concrete syntax:

- XML names for the emotion categories in the conceptual inventory; numerical values representing emotion intensities;
- encodings of entity structures using <emotion> elements (as defined in EmtionML, but simplified).

Semantics:

- Pairs ⟨c, v⟩ are interpreted as attribute-value pairs where c denotes a two-place function, applicable to the experiencer and the object of an emotion, so $I_e(\langle c, v\rangle)$ is defined as the two-place predicate $\lambda x.\ \lambda y.\ I_e(c)(x,y) = I_e(v)$.

For linking emotion specifications to dialogue act annotations, a plug-in interface is needed that defines the <emoLink> element used in (E13b) with its underlying abstract syntax and semantics. In the abstract syntax, an emotion link structure is a triple ⟨p, s, e⟩ formed by a dialogue participant 'p' who is the sender of a dialogue act, the semantic content 's' of this dialogue act, and an emotion 'e'. These components correspond in the concrete syntax to the values of the attributes @holder, @object, and @emotion in an <emoLink> element, as illustrated in (E13). The semantics of the emotion link structures is defined by (E14), where ${}_{a+c}I_e$ designates the semantic interpretation function of the interface of the emotion plug-in with the annotation scheme formed by the ISO 24617-2 host scheme with a semantic content plug-in L_c.

(E14) $\quad {}_{a+c}I_e(\langle p, s, e\rangle) = I_e(e)(I_a(p), I_c(s))$

Annex F: Data categories for core concepts (normative)

F.1 Overview

This annex contains data categories for the core concepts of this standard, namely *dimensions, communicative functions,* and *qualifiers*.

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

F.2 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 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 apologizing.
-- Source	DIT

/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

F.3.1 General-purpose functions

F.3.1.1 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, not does it include rhetorical questions, which from a semantic point of view are not questions at all but rather the expression of an opnioin.
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 standard prefers the term ‘propositional question' because the term ‘YN-Question' carries the suggestion that this kind of question can only be answered by “yes” or “no”, which is 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

/testQuestion/	
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	Test questions have the linguistic same form as ordinary questions; their occurrence in a particular setting with participants in specific roles (such as tutor and student) makes a question recognisable as a test question.

F.3.1.2 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 standard 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”

F.3.1.3 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	“I will look that up for you”

/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	“Shall I begin?”; “Would you like to have some coffee?”

/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 wlll 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.”

F.3.1.4 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	“Yes please"; French: "Je vous en prie”

/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".

F.3.2 Dimension-specific functions

F.3.2.1 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 DITand SLSA schemes.
Example	English: "I beg you 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?"

F.3.2.2 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/	
Conceptuak 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

F.3.2.3 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…”)

/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 orde 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"

F.3.2.4 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 activites 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

/topicShift/	
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 going to continue the dialogue on a different topic.
-- Source	DIT
Example	“Something else.”

F.3.2.5 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: Repaird (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

F.3.2.6 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.”.

/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).
Example	S: "Bye bye, see you later"

/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."

/compliment/	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, aware that S likes A's appearance, certain of A's qualities, or something that A has achieved.
-- Source	Commonplace
Explanation	Follow-on greetings usually come in response to opening greetings and/or self-introductions.
Example	"Well done!", "You look great".

/congratulation /	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, aware that the speaker takes pleasure in the successful achievement of something by A or in A's good fortune.
-- Source	After Merriam-Webster.
Example	"Congratulations!".

/sympathyExpression /	
Definition	Communicative function of a dialogue act performed by the sender, S, in order to make the addressee, A, aware that the speaker is sorry for something that happened to the addressee.
-- Source	
Example	"I'm sorry to hear that?".

F.3.2.7 Contact management functions

/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”

F.4 Qualifiers

F.4.1 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"

/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.”

F.4.2 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)

F.4.3 Sentiment

/sentiment/	
Definition	Class of predicates which can be associated with a communicative function to express an attitude of the sender towards the semantic content of the dialogue act.
-- Source	Petukhova and Bunt (2009a)
-- Note	In the absence of a widely agreed set of sentiment values, this standard does not define any data categories for sentiment values.

/positive/	
Definition	Predicate which expresses that the sender of a dialogue act with that function feels good about the semantic content of the dialogue act.
-- Source	
Example	

/negative/	
Definition	Predicate which expresses that the sender of a dialogue act with that function does not feel good about the semantic content of the dialogue act.
-- Source	
Example	

Annex G: Dimensions in dialogue act annotation schemes (informative)

As part of the development of the present standard a detailed study was conducted in order to provide theoretical and empirical arguments for identifying dimensions and communicative functions (Petukhova and Bunt, 2009a)[18]. The study included a survey of the literature on dialogue analysis and of the use of functions and dimensions in 18 annotation schemes. Moreover, a number of statistical and machine-learning tests were carried out in order to identify dependencies among potential dimensions. The independence of dimensions was investigated by investigating co-occurrences of communicative functions across dimensions, by calculating the semantic relatedness between dimensions, and by determining for a range of candidate dimensions the frequencies of occurrence of functional segments addressing only those dimensions. The tables G.1 and G.2 are from this study. Table G.1 shows the relative frequencies of functional segments in ten dimensions 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 ('orthogonal').

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

It has been observed that Time Management acts co-occur frequently with Turn Management acts, since speakers often need a bit of time to formulate their contribution when they take (or have and want to keep) the turn. This consideration applies only to *Stallings* under certain context conditions, however; *Pausing*, by contrast, does not imply that the speaker wants to keep the turn. It should be also noticed that *Stallings* do not always imply that the speaker wants to keep the turn; extensive amounts of protraction accompanied by certain non-verbal behaviour may instead 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 are used for turn-keeping purposes; even in monologues, where speakers do not need to keep the turn, time delays occur frequently. Time and Turn Management are therefore better kept apart rather than considered as one dimension.

Stallings are often produced unintentionally, and it has been suggested that 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 an emotional or cognitive state, which is obviously relevant 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

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

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 unintentional performance does not provide a convincing argument against viewing *Stallings* as dialogue acts.

For each of the ten dimensions occurring in Table G.1 the question was studied whether functional segments occur which express a dialogue act addressing that 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