ISO/TC37/SC 4 **N442 rev00**

Date: 2008-02-18

ISO/WD 24617-2

ISO/TC 37/SC 4/WG 2

Secretariat: KATS

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

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

Document type: International Standard
Document subtype: Not applicable
Document stage: (20) Working Draft
Document language: E

Reference Number
ISO/CD 24617-2(E)

da.tex 2008-02-18

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

Published in Switzerland

Contents

Page

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

Foreword

The International Organization for Standardization (ISO) 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 3.

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.

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

NOTE The ACL SIGSEM Working Group is headed by Harry Bunt, harry.bunt@uvt.nl, Tilburg University.

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 1 has been approved by a majority of the ISO/TC 37/SC 4 P-members for submission to the ISO Central Secretariat and registration as a DIS, draft international standard.

The main parts of ISO 24617-2 consist of:

a) Scope

b) Normative references

c) Terms and definitions

d) Motivation and requirements

e) Basic concepts and metamodel

f) Specification of ISO-DiAML, a formal annotation (specification) language for dialogue acts in natural language

g) Semantics of ISO-DiAML

The proposed international standard provides one normative annex providing core annotation guidelines and a number of informative annexes. Four annexes provide annotated examples, one being a set of completely annotated examples of English and the other consisting examples from various languages other than English. Two other annexes deal with the ISO-DiAML DTD and schema. Information is given on past and current activities on dialogue act annotation and also on tools and templates. In the final annex, editorial or authorship information is provided with a list of editors and contributors along with meeting records. This document is concluded with a bibliography.

Introduction

This standard proposal results from preliminary studies conducted in ad-hoc Thematic Domain Group 3 (TDG 3) of ISO committee TC 37/SC 4. The TDG 3 group was created in 2004 to conduct investigations into the feasibility and timeliness of standardization activities in the area of semantic annotation and representation. Five areas of semantic information were chosen to focus the investigations, and for each area a project leader was appointed:

a) temporal entities and relations - project leaders: Kiyong Lee and Harry Bunt

b) referential entities and relations - project leader: Laurent Romary

c) semantic roles and argument structures - project leader: Thierry Declerck

d) dialogue acts - project leader: Harry Bunt

e) discourse relations - project leader: Koiti Hasida

The investigations were reported and discussed in a series of TDG 3 meetings, listed in Appendix D. In order to obtain opinions and feedback from a larger community, a Working Group on the Representation of Multimodal Semantic Information was created within SIGSEM, the Special Interest Group on Computational Semantics of the Association for Computational Linguistics (ACL). The ISO TDG 3 and the SIGSEM WG have been holding four joint workshops – see Appendix D.

In order to boost the TC 37/SC 4 activities aiming at supporting the development of interoperable language resources, a project was set up within the eContent framework of the European Union. This project, called LIRICS (Linguistic Infrastructure for Interoperable Resources and Systems) ran from January 1, 2005 to July1, 2007. The project was organized in such a way that its deliverables would be input for ISO TC 37/SC 4 activities. The LIRICS project adopted the areas a) - d) mentioned above as its semantic focus, and produced feasibility studies as well as proposals for standard annotation concepts as inputs to TC 37/SC 4 working groups.

As an outcome of the ISO TDG 3 and LIRICS activities, the area of temporal information was chosen for the development of an international standard for semantic annotation. The standard under development in that project, “Semantic Annotation Fraework Part 1: Time and events” constitutes the first part of standard 24617, of which the present proposal forms Part 2.

It is envisaged that this standard will be organized into further parts, some of which will be concerned with other areas of semantic information (such as semantic roles, reference, spatial information, quantification,..) and one part that will be concerned with semantic annotation and representation from an integrated point of view, dealing with the combined annotation of semantic information from several areas.

1 Scope

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

2 Normative references

For this international standard there are two main normative references:

- ISO 8879: 1986 (SGML) as extended by TC2 (ISO/IEC JTC 1/SC 34 N029: 1998-12-06).
- ISO 19757-2, Document Schema Definition Language, part 2.

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

Besides these ISO documents, there are two documents from the LIRICS project (project deliverables) to be considered:

- "Documented compilation of semantic data categories". Deliverable D4.3 of European eContent project LIRICS, Linguistic Infrastructure for Interoperable Resources and Systems. http://let.uvt.nl/research/ti/iso-tdg3/D4-3.pdf
- "Multilingual test suites for semantically annotated data", Deliverable D4.4 of European eContent project LIRICS, Linguistic Infrastructure for Interoperable Resources and Systems. http://let.uvt.nl/research/ti/iso-tdg3/D4-4.pdf

3 Terms and definitions

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

3.1
turn
Stretch of communicative behaviour produced by one speaker, bounded by periods of inactivity of that speaker or by activity of another speaker.

NOTE After Allwood (2000).

3.2
dialogue act
Semantic unit in the description of dialogue behaviour, characterizing how the information state(s) of the agent(s) at whom the behaviour is directed are changed when he/they understands) the behaviour.

3.3
speaker
Property of a dialogue act, indicating the dialogue participant who produces the communicative behaviour that expresses the dialogue act.

3.4
addressee
Property of a dialogue act, indicating a dialogue participant at whom the communicative behaviour that expresses the dialogue act is directed.

3.5
overhearer
Participant in a dialogue who witnesses a dialogue act and whose information state may be affected by it, without being an addressee of the dialogue act.

3.6
semantic content of a dialogue act
Property of a dialogue act, specifying the entities that the act refers to and the states of affairs involving these entities that the dialogue act describes.

3.7
communicative function
Property of a dialogue act, specifying how the act's semantic content changes the addressee's information state when (s)he understands the dialogue act.

3.8
dialogue segmentation
Way of identifying stretches of communicative behaviour in dialogue that express one or more dialogue acts.

3.9
functional segment
Stretch of dialogue behaviour that expresses one or more dialogue acts.

NOTE A functional segment a stretch of behaviour that does not need to be continuous.

4 Purpose and justification

Dialogue acts are widely used in studies of dialogue phenomena, in dialogue annotation efforts, and in the design of dialogue systems. A dialogue act has a functional part, often called its communicative function, which expresses what the speaker is trying to achieve, and a referential/propositional part, which describes the information that is being addressed. Dialogue act annotation is the activity of marking up stretches of dialogue behaviour (linguistic, nonverbal, or multimodal) with communicative functions.

During the 1980s and 1990s a number of alternative dialogue act annotation schemas have been developed, such as those of the TRAINS project in the US, the Map Task studies in the UK, and the Verbmobil project in Germany. These schemas were all designed for a specific purpose and a specific application domain; they contained overlapping sets of communicative functions and made use of overlapping, often mutually inconsistent terminology. Later in the 1990s a group of dialogue researchers came together as the "Discourse Research Initiative", and in two workshops drafted the general-purpose schema for multidimensional dialogue act annotation called DAMSL: Dialogue Act Markup using Several Layers (Allen and Core, 1995; Core et al., 1998). This represented a important step forward in dialogue act annotation, with its focus on multidimensionality and domain-independence.

Due to the lack of a stable organizational structure behind the initiative, the design of DAMSL was left in an unfinished state with substantial gaps and deficiencies, for example in the areas of feedback, turn management, and social obligations management, and several researchers have constructed variations of the DAMSL schema for their own purposes, such as Switchboard-DAMSL (Jurafsky, 1997) and COCONUT (De Eugenio et al., 1998). Recent years have witnessed a rapidly growing interest in the annotation of dialogue corpora in terms of dialogue acts, and in the development of dialogue act annotation schemas. Existing schemas differ not only in their precise sets of tags, but more importantly with respect to (1) the underlying approach to dialogue modeling; (2) the definitions of the basic concepts; and (3) the level of granularity of the defined tag set. Generally, dialogue act annotation schemas can be divided into one- and multidimensional ones.

One-dimensional schemas allow coding dialogue utterances only with one tag, and their tag sets are mostly kept very simple (such as the LINLIN schema ((Ahrenberg et al., 1995) and the HCRC schema (Carletta et al., 1996). Because of their simplicity, they are thought to be reliable and to take less efforts to apply consistently by annotators. Several researchers note, however, that one-dimensional annotation schemas also have serious disadvantages (see e.g. e.g. Klein et al., 1998; Larsson, 1998; Popescu-Belis, 2005).

Mutidimensional approaches to dialogue act annotation have been recognised by many researchers as empirically better motivated, and allowing the modeling of theoretically important distinctions. Natural dialogue involves several activities beyond those strictly related to performing the task or purpose for which the dialogue is instrumental (such as obtaining certain information, instructing another participant, negotiating an agreement, etc.). In natural conversation, among other things, dialogue participants constantly 'evaluate whether and how they can (and/or wish to) continue, perceive, understand and react to each other's intentions' (Allwood, 1997). They share information about the processing of each other's messages, elicit feedback, manage the use of time and taking turns, of contact and attention, etc. Communication is therefore a complex, multi-faceted activity, and dialogue utterances are often multifunctional. While the multifunctionality of dialogue utterances has been widely recognised, computationally oriented approaches to dialogue generally see multifunctionality as a problem, both for the development of annotation schemas and for their application when designing dialogue systems (Traum, 2000b). Information that may be obtained through a multifunctional analysis is therefore often sacrificed for simplicity in computational modelling.

Preliminary studies in the ISO TC 37/SC 4 ad-hoc Thematic Domain Group on Semantic Content (TDG 3) have indicated that the area of dialogue act annotation is sufficiently mature for a new and sustained effort (supported by the stable organizational structure of ISO) to design a comprehensive general framework for dialogue act annotation. In the LIRICS project, a set of data categories for communicative functions has been defined following ISO standard 12620, which has been documented in LIRICS deliverable D4.3 and endorsed by TDG 3 in 2007. The LIRICS data categories have been designed to fit into a multidimensional annotation schema. They have been tested for their usability and coverage in the manual annotation of dialogues in English, Dutch and Italian (documented in LIRICS deliverable D4.4). This set of data categories forms one of the starting points of the present proposal.

To support the use of the proposed annotation standard, this project will provide annotation guidelines and examples. An important issue in the annotation of dialogue is its segmentation into functionally meaningful parts. The multidimensional approach behind the LIRICS data categories provides a basis for segmenting dialogues in multiple ways and allowing markables to be discontinuous as well as overlapping. The project will provide guidelines for how to effectively perform such segmentation.

While it seems feasible to develop standard annotation concepts for a set of core dialogue acts, given the current state of the art, researchers and application designers should also be supported in adding their own concepts for specific domains or purposes. The project will therefore provide general principles and guidelines for extending its core concepts.

5 Basic concepts and metamodel

The idea of interpreting dialogue behaviour in terms of communicative acts such as statements, questions, promises, requests, and so on, goes back to speech act theory (Austin, 1962; Searle, 1969), which has been an important source of inspiration for modern dialogue act theory. Where speech act theory is primarily a direction within the philosophy of language, however, dialogue act theory is a data-driven approach to the computational modeling of language use in dialogue. As a way to describe meaning in interactive language use, dialogue acts, are semantic concepts that can be defined by the way they affect the information state of an addressee when (s)he understands the behaviour. For instance, when an addressee understands the utterance *Do you know what time it is?* as an indirect question about the time, then the addressee's information state is updated to contain (among other things) the information that the speaker would like to know what time it is. If, by contrast, an addressee understands the utterance as a reproach, then his information state is updated to include the information that the speaker knows very well what time it is and wants to bring that to the addressee's attention. Distinctions such as that between a question and a reproach refer to the communicative function of a dialogue act, and the entities that are referred to, such as the time at the moment of speaking, to its semantic content. Fundamental notions associated with the dialogue act concept are therefore:

a) speaker

b) addressee

c) communicative function

d) semantic content

These concepts form the backbone of the proposed metamodel for dialogue act annotation. A dialogue act has a sender, also referred to as "speaker", and one or more addressees; in addition there may be various types of side-participants who witness a dialogue without participating in it, but whose information states may be affected. The behaviour of the (real) participants may also be influenced by the presence of side-participants, if they are aware of their presence, as in a television interview or a talk show. Clark (1996) makes a distinction between 'overhearers', 'side-participants' and 'bystanders; we will use 'overhearer' as a cover term here, allowing further refinement when necessary.)

The assignment of meaning to stretches of communicative behaviour in dialogue presupposes a way to identify meaningful stretches of dialogue behaviour. The identification of such stretches is called the segmentation of the dialogue. Many studies in the assignment of dialogue acts to communicative behaviour have assumed dialogue to be segmented at the level of 'utterances' or 'turns'. A turn can be defined as a stretch of communicative behaviour produced by one speaker, bounded by periods of inactivity of that speaker or by activity of another speaker. An advantage of segmentation at turn level is that turn boundaries can be detected relatively easily; however, a turn often contains smaller meaningful parts, which makes turns too coarse-grained for many purposes. (The observation that dialogue behaviour is often multifunctional, in the sense of having more than one communicative function (Allwood, 2000) is partly exlained by the fact that turns may contain several functionally relevant substretches of behaviour. Multifunctionality does not go away when smaller stretches are considered, however; see Bunt, 2007a.)

Utterances, on the other hand, are linguistically defined stretches of (linguistic) behaviour that have one or more communicative functions. Levinson (1983) writes: *'An utterance is the issuance of a sentence, a sentence-analogue, or sentence-fragment, in an actual context.'* They are usually smaller than turns. While utterances have the advantage of being more fine-grained, the detection of utterance boundaries is a highly nontrivial task. Syntactic and prosodic features are often used as indicators of utterance endings (e.g. Shriberg et al., 1998; Stolcke et al., 2000; Nöth et al., 2002), but are in general not reliable. In the case of nonverbal or multimodal communication, the notion of an utterance as a linguistically defined unit is even less clear.

Apart from the difficulties of automatic utterance boundary detection, from a semantic point of view the stretches of behaviour that are relevant as markables for dialogue act annotation may be discontinuous, may overlap, and may even contain parts from different turns. The following example illustrates the potential discontinuity of dialogue act markables:

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

In this example the utterance *The next train is at 7.48*, which answers the preceding (indirect) question, is interrupted by *at... let me see...* which expresses that the speaker cannot answer immediately, but needs a little time (a 'Stalling' act). As a result, the communicative behaviour expressing the answer is discontinuous.

The following example illustrates that dialogue acts may overlap:

(2) A: What time is *the first train to the airport on Sunday*?
B: *The first train to the airport on Sunday* is at 06:25.

In this example, B's response as a whole is an answer to the preceding question, but the part *The first train to the airport on Sunday* is in addition a positive feedback act, displaying B's understanding of A's question.

These complications can be handlled by taking a *multidimensional* view on communication. On such a view, participating in a dialogue involves performing actions in several separate 'dimensions' of the interaction, such as pursuing the activity that motivates the dialogue, giving and eliciting feedback on the communicative process, taking turns, managing time, and taking care of social aspects such as thanking and apologizing when appropriate. Moreover, participants act in these different dimensions not purely serially, but often simultaneously, which explains the fundamentally multifunctional nature of dialogue behaviour. It therefore makes sense to segment a dialogue in multiple ways, identifying functionally meaningful stretches in every relevant dimension. The unit of dialogue behaviour that then emerges is the natural candidate for being a markable, i.e. the unit that is annotated with communicative function information. We call this unit a *functional segment* or, if no confusion is likely to arise, an *utterance* (as in the metamodel in Figure 1).

Figure 1 shows a representation of the fundamental upper-level concepts that support dialogue act annotation. Dialogue acts are tied to functional segments, which are usually part of a turn but sometimes parts of more than one turn. Turns are themselves parts of a dialogue. A defining feature of a turn is that it is performed by only one sender, regardless of whether the addressee interjects with nonverbal feedback or backchannels (these are seen as a separate though concurrent turn). In the case of multiparty dialogue, there may be multiple addressees, and not all the utterances in a turn necessarily have the same addressee(s). In addition, there may be one or more overhearers.

The metamodel in Figure 1 captures the considerations that have been discussed in this section, and additionally includes the potential for showing functional dependencies between dialogue acts. This would encompass such phenomena as indicating to which question a response is intended to be an answer, or to which previous utterance a feedback act applies. Note also, and most importantly, that an utterance may correspond to multiple dialogue acts, due to the multidimensional nature of communication and the multifunctionality of natural language utterances.

6 Defining systems of communicative functions

6.1 Deep and surface approaches to dialogue act definition

A distinction is sometimes made between 'surface speech acts' and 'deep speech acts'. What is meant by this distinction is that the interpretation of an utterance as a communicative act can be made either in a way that stays close to surface forms, or in a way that digs into the speaker's underlying intentions and beliefs. A 'surface speech act' would as such be determined by the lexical, prosodic and syntactic properties of an utterance, and would correspond to what the speaker is 'literally saying', rather than what he is really trying to achieve (see e.g. Appelt, 1982; Allen & Perrault, 1979; Traum, 1999). Many approaches to dialogue act annotation do not take a clear position on this issue, defining some communicative functions in terms of beliefs and intentions and others in terms of surface form.

We propose to take a strictly 'deep', semantic approach. But we also insist that every communicative function that is distinguished must be empirically justified in the sense that there are ways in which a speaker can indicate that his behaviour should be understood as having that particular function through the surface form of his behaviour. Successful communication depends on both participants understanding the communicative functions of each others' utterances, which they deduce from the utterance surface forms plus general background knowledge and their models of the dialogue context, including assumptions about each other's beliefs and goals.

The deep/surface distinction is particularly relevant in connection with the differences between human and automatic annotation. Human annotators are better at understanding and annotating dialogue utterances deeply, because they have more knowledge of intentional behaviour and richer context models. Since a general dialogue annotation schema should support human annotation, it should contain concepts with a depth and granularity that matches human understanding of the functions of dialogue utterances. In order to support automatic annotation, on the other hand, the schema should also contain concepts that are suitable for a more shallow form of automatic annotation that does not necessarily require deep semantic knowledge. These two requirements can be met by defining hierarchies of communicative functions, where functions deeper down in a hierarchy correspond to more specific intentions or assumptions on the part of the speaker than functions higher in the hierarchy.

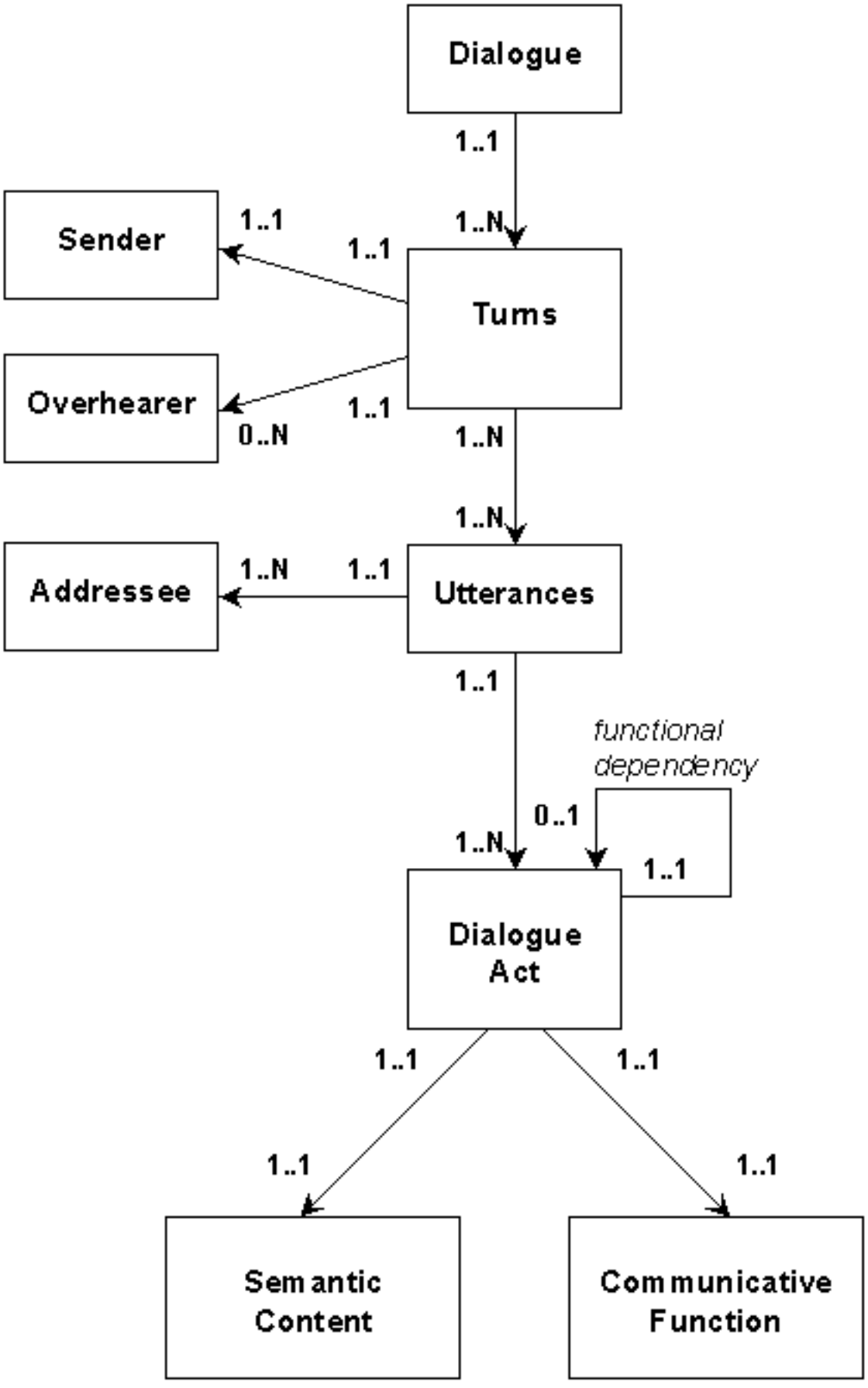

Figure 1. Dialogue act metamodel

6.2 Semantic content and communicative function definition

The very idea of distinguishing the communicative function and the semantic content as components of a dialogue act, inherited from the distinction between 'illocutionary force' and 'propositional content' in speech act theory, is that they describe separable and independent components of utterance meaning. As a consequence, the semantic content of a dialogue act cannot be a factor determining its communicative function. Although this may seem rather obvious, many dialogue act annotation schemas violate this principle and employ such functions as "accept_date" and "suggest_exclude_location", mixing semantic content into the communicative function (as in the 1995 version of the Verbmobil dialogue act taxonomy, see Jekat et al., 1995; this was remedied in a later stage of the Verbmobil project - see Alexandersson et al., 2002).

6.3 Indirect dialogue acts

Indirect dialogue acts have been largely ignored in dialogue act annotation systems. The DIT^{++} taxonomy of communicative functions (Bunt 2005) views indirect dialogue acts as having a communicative function which is slightly different from, though closely related to, that of the corresponding direct dialogue act, because their performance has slightly different effects on information states of addressees. For example, the difference between a direct SetQuestion such as *Where is Lee's office?* and an indirect one like *Do you know where Lee's office is?* is that in the indirect version the speaker does not express an expectation that the addressee knows the answer to the question, whereas the direct version does carry this assumption (see Bunt, 2000a for further discussion). The beliefs that the speaker subscribes to when he uses an indirect question are thus not exactly the same as when he uses a direct question. This provides a basis for their functional distinction in a dialogue act annotation schema.

6.4 Levels of feedback

DAMSL and other annotation schemas (with the exception of the DIT^{++} schema) do not have communicative functions for articulate description of communicative feedback. Inspired by the work of Allwood et al. (1994), the LIRICS annotation schema distinguishes the following levels of processing at which a dialogue participant may provide feedback about his processing of previous utterances:

a) attention: whether sufficiently much attention is paid to what the dialogue partner says in order to perceive (to hear, in particular) what is being said;

b) perception: whether the previous contribution from the dialogue partner was recognized well or encountered some difficulties;

c) interpretation: whether the meaning of the previous contribution from the dialogue partner was understood well or encountered some difficulties;

d) evaluation: whether the message expressed in the previous contribution from the dialogue partner was evaluated as fitting into the current dialogue context without difficulty or not (e.g. whether a statement by the dialogue partner was believed);

e) execution: whether the previous contribution of the dialogue partner led to the successful performance of certain actions, intended by the dialogue partner (e.g. whether a question can be answered, or a request can be fulfilled).

These levels are logically related in that successful processing at a lower level is a prerequisite for successful processing at higher levels. For instance, a question that was not understood cannot be answered. Due to these logical relations, feedback acts at one level imply feedback functions at other levels. For example, a negative feedback act at the level of perception (*"What did you say?"*) implies negative feedback at higher levels (interpretation, evaluation, and execution).

Within the area of feedback, a distinction has been made between *auto-* and *allo-feedback* (Bunt, 1995). Auto-feedback is the case where a speaker provides information about his processing of previous utterances by other participants, and is often called simply 'feedback'; allo-feedback is the case where the speaker provides or elicits feedback to/from his addressee(s) about his/their processing of previous utterances in the dialogue.

6.5 Turn management

Turn management is an area for which DAMSL and other proposed annotation schemas, with the exception of the DIT^{++} schema, do not provide communicative function concepts, in spite of the fact that turn management occurs virtually all the time in natural dialogue, especially in multi-party dialogue. Following the detailed studies of turn-taking in spontaneous conversation that have been performed (see e.g. Levinson, 1983), the LIRICS proposal distinguishes three functions for turn-initial turn management activity, and three for turn-final:

- Turn-initial functions:
  - TurnTake: Speaker takes the turn, which is available
  - TurnAccept: Speaker takes the turn upon request
  - TurnGrab: Speaker interrupts other speaker to take the turn
- Turn-final functions:
  - TurnAssign: Speaker requests other participant to take the turn
  - TurnRelease: Speaker makes the turn available
  - TurnKeep: Speaker signals that he wants to continue to have the turn

6.6 Other aspects of interaction management

The LIRICS set of data categories for dialogue acts furthermore includes communicative functions in 5 other dimensions relating to the of management of the interaction, namely Time Management, Contact Management, Discourse Structure Management, Own Communication Management and Partner Communication Management.

Table 1 shows some examples of dialogue acts in each of these dimensions.

6.7 Social obligations management

Social Obligations Management (SOM) acts have a 'social' function such as greeting, introducing oneself, thanking, and apologizing. Some of these acts occur only at the beginning or at the end of a dialogue; in particular, welcome greetings (*Hi!*) are the most common way to open a dialogue and valedictions (*Bye!*) are commonly used to close a dialogue. Self-introductions also occur in the beginning of a dialogue; thanking tends to occur only towards the end of a dialogue. Apologies can occur anywhere in a dialogue, in particular when a participant has misunderstood something or is unable to answer a question or fulfill a request.

SOM acts often come in pairs: greetings elicit response greetings; self-introductions elicit self-introductions by the other participant(s); thanking and apologizing elicit downplayers (*That's all right; No problem*).

7 Core dimensions and dialogue acts

The various annotation schemas for dialogue acts that have been proposed share a number of dialogue act types that are of obvious importance in any type of dialogue. These 'core dialogue acts' include various types of questions, answers, informs, requests, and acknowledgements. Traum and Hinkelman (1992) have used the term 'core dialogue acts' to refer to the types of acts that are most familiar from traditional speech act theory. These are often related to the use of performative verbs (such as *promise, invite, christen, fire*) and include the commissive and directive act types (offer, request, propose,...), the 'reportative' speech acts used for stating facts (assert, conclude), and the 'expressive' ones for expressing psychological states (apologize, thank, congratulate). We will use the term 'core dialogue acts' in a slightly different sense, covering the most important types of dialogue acts that are commonly in dialogues. These include the commissive, directive, and reportative acts from speech act theory and some of the expressive ones, but excluding the 'declarative' ones (christen, baptize, declare war, pronounce husband and wife, open meeting, pass judgement,..) and

including some others which have not been considered in speech act theory, such as acts for turn taking and time management.

In the LIRICS project, 58 communicative functions have been defined which are considered to be important for general, domain-independent dialogue act annotation efforts, and are taken as a starting point in the present project. . The question exactly which communicative functions should be considered as 'core' concepts and as forming a part of the proposed dialogue act annotation standard, will be an important issue in this project.

7.1 Dimensions

Multidimensional annotation often comes down to allowing language data to be marked up with multiple tags, and the notion of a dimension comes down to a certain set of mutually exclusive tags. This is for instance the case in the DAMSL schema, which has been presented explicitly as a multidimensional annotation schema, but where the notion of a dimension does not have an independent definition. This is not satisfactorily, as the following examples shows.

(3) 1. A: And what possibilities do you have on Tursday?
2. B: Did you say Thursday?

Utterance 2 expresses a problem in B's understanding of the previous utterance, and as such has a function in the Understanding dimension, where DAMSL has two possible values: Signal-understanding and Signal-non-understanding. Both of these are too crude, since speaker B has an understanding problem only with a particular word in the previous utterance the problem being a lack of certainty regarding the correct understanding of that word. A more accurate description than 'Signal-non-understanding' would be: B wants to make sure that he correctly understood the designated element in the previous utterance. But note that *B wants to make sure that...* is the essence of the communicative function known as Check, a function in the Info-request dimension (see Core et al., 1998). So an accurate characterization of the utterance in the Understanding dimension amounts to characterizing it in DAMSL's Info-request dimension! Clearly, the Info-request dimension is not orthogonal to the Understanding dimension. Similarly, an accurate characterization of *I did not quite hear whether you said 'Thursday'"* in the Understanding dimension would result in a characterization in the Statement dimension.

The point of these examples is that questions, assertions, checks and answers do not always relate to the task or activity that motivates the dialogue, but may also be about understanding, in which case they lead to a dialogue act in the Understanding dimension. In fact, these very general types of dialogue act can be about *any* aspect of the communication. For example, *"Are you there?"* is a question about the presence or attention of the dialogue partner, and *"I think we're done"* asserts the opinion that the dialogue can soon be ended since the motivating task has been completed. Questions, assertions, and answers are therefore functions that do not constitute separate dimensions, as is assumed in DAMS. Moreover, they do not belong to *any* particular dimension.

A similar analysis applies to requests, offers, instructions, suggestions, and other directive and commissive acts; like questions, statements and answers, dialogue acts with these functions can be about any aspect of the dialogue, as is illustrated by the following examples:

(4) a. Please repeat that.
b. Would you like me to repeat that?
c. Wait a minute please.

For this reason the DAMSL categories 'Committing-speaker-future-action' and 'Influencing-addressee-future-action', likewise, do not qualify as proper dimensions, and the functions in these classes do not belong to any specific dimension. This raises the question of how they fit into a multidimensional annotation schema. Following Bunt & Girard (2006), this proposal follows the solution that has been taken in the DIT^{++} and LIRICS schemas, and treat these functions, called *general-purpose functions*, as dimension-independent, and forming a part of the annotation schema separate from the dimension-specific communicative functions. When a dialogue act is constructed by combining a general-purpose function and a semantic content, then the type of semantic content determines the dimension of the resulting dialogue act.

In order to design a dialogue act annotation schema that is truly multidimensional, we start from the conceptual view that a participant in a dialogue has a number of things to manage besides the performance of the underlying task or activity that motivates the dialogue. An aspect of participating in dialogue that a dialogue agent has to manage, qualifies as a dimension of dialogue act analysis if the following two conditions are satisfied:

1. A dialogue participant can address this aspect of participating in a dialogue through linguistic and/or non-verbal behaviour that has a communicative function specific for this purpose;
2. This aspect of participating in a dialogue can be addressed independently of other aspects, i.e., dialogue utterances can have a communicative function in one dimension, independent of its functions in other dimensions.

The first of these criteria means that we are considering an aspect of communication that can be distinguished according to empirically observable behaviour in dialogue. The second condition requires dimensions to be orthogonal. (Any dependencies (implications or constraints) between functions in different dimensions are regarded as empirical facts about communication, rather than as properties of the system of dimensions.)

Consider, for instance, the category of time management. Utterances that address time management include those where the speaker wants to gain a little time in order to determine how to continue the dialogue; this function is called Stalling. Speakers indicate this by slowing down in their speech and/or by using fillers, as in *ehm, well, let me see,...* The existence of the act of Stalling means that the category of time management functions satisfies the first criterion. Moreover, the use of devices to indicate the Stalling function can be applied to virtually any utterance, which can have have any other function in any other dimension. Time management therefore satisfies the second criterion as well, and hence qualifies as a proper dimension.

A similar analysis can be applied to the other candidate dimensions mentioned above. Table 1 gives some examples of communicative functions within each of these dimensions, with typical utterance forms in English. Note that in natural dialogue many of these functions are often indicated not or not only linguistically, but also through nonverbal means such as facial expressions, head movements, direction of gaze, and hand gestures. Three other dimensions that correspond to conceptually distinct aspects of communication and that satisfy the two criteria mentioned above are: (1) own communication management, which is the category of functions a speaker may use to indicate his editing of his own speech; (2) partner communication management, which occurs when a speaker assists or corrects the dialogue partner in producing a dialogue contribution (e.g. by completing an utterance which the dialogue partner is struggling to complete); and (3) discourse structuring, which is the explicit opening or closing of a (sub-)dialogue, or announcing a topic shift.

The present proposal starts from the 10 dimensions proposed by the LIRICS project, shown in Table 1. The question which of these dimensions should be considered as 'core dimensions' will be addressed in this project.

7.2 Dimension-specific and general-purpose functions

A key distinction that emerges from the examples in (4) is that some communicative functions are specific to a certain dimension, while others can be used to construct a dialogue act in any dimension. The two types of function are called *dimension-specific* and *general-purpose communicative function*, respectively. The turn management functions listed in section 6.5 are clear examples of dimension-specific functions, while the various types of questions, informs, answers, requests, and offers are all general-purpose functions that can be used in every dimension. For example, a request like *Jim, will you say something about this please?* is a turn management act constructed with the general-purpose function Request, and a statement like *I am extremely grateful for your help* is a thanking act in the SOM dimension, with the communicative function Inform. Table 1 lists examples of dimension-specific communicative functions in the dimensions distinguished in the LIRICS project.

A consequence of the fact that general-purpose functions can be used to build a dialogue act in any dimension, depending on the type of semantic content, is that an adequate dialogue act annotation of a functional segment should have two components: the communicative function and the dimension that is addressed, as in the following examples:

Dimension	Dimension-specific comm, functions	Typical expressions
Task/Activity	OpenMeeting; CloseMeeting	domain-specific
	Appoint; Hire	fixed expressions
Auto-Feedback	PerceptionNegative	*Huh?*
	EvaluationPositive	*True.*
	OverallPositive	*OK.*
Allo-Feedback	InterpretationNegative	*THIS Thursday.*
	EvaluationElicitation	*OK?*
Turn Management	TurnKeeping	final intonational rise
	TurnGrabbing	hold gesture with hand
	TurnGiving	*Yes.*
Time Management	Stalling	slowing down speech; fillers
	Pausing	*Just a minute*
Contact Management	ContactChecking	*Hello?*
Own Comm. Man.	SelfCorrection	*I mean...*
Partner Comm. Man.	PartnerCompletion	completion of partner's utterance
Discourse Structure Management	TopicShiftAnnouncement	*Something else.*
Social Obligations Management	Apology	*I'm sorry.*
	Greeting	*Hello.*
		Good morning
	Thanking	*Thanks.*

Table 1: Examples of dimension-specific communicative functions and their expression for each dimension distinguished in the LIRICS project.

(5) a. Please repeat that.
<Auto-Feedback, Request>

b. I am very grateful for your help
<Social Obligations Management, Inform>

c. You got that?
<Allo-Feedback, CheckQuestion>

d. Jim, you go on.
<Turn Management,Instruct>

If a functional segment has a dimension-specific function, then obviously the specification of that function is sufficient, and specifying the dimension would be redundant. In general, however, the annotation of segments requires both components. This is reflected in the definition of the dialogue act annotation language presented below.

8 DiAML: Dialogue Act Markup Language

In the Linguistic Annotation Framework (Ide et al., 2003), which has been accepted as a proposed ISO standard, a distinction is made between *annotation* and *representation*. The term 'annotation' is used to refer to the process of adding information to segments of language data, or to refer to that information itself. This notion is independent of the format in which this information is represented. The term 'representation' is used to refer to the format in which an annotation is rendered, for instance in XML, independent of its content. According to the Linguistic Annotation Framework, annotations are the proper level of standardization, not representations. For marking up dialogue act information at the level of annotations, this project therefore defines a markup language for dialogue acts at the annotation level. This language is called DiAML: Dialogue Act Markup Language.

The distinction between annotations and representations is reflected in the definition of DiAML given below, where an *abstract syntax* is defined independent of a *concrete syntax*. The abstract syntax specifies the elements making up the information in an annotation and how these elements may be combined; these combinations are defined as set-theoretical structures. There are infinitely many ways in which these structures

can be represented concretely. In line with other ISO TC 37/SC 4 proposals, an XML-based concrete syntax is proposed for representing DiAML annotations. Any other representation that is a faithful rendering of the abstract syntax of DiAML can readily be converted into this XML representation and vice versa. DiAML has a formal semantics associated with its *abstract* syntax, which explains why all concrete representations of DiAML annotations are semantically equivalent.

The Linguistic Annotation Framework recommends the use of *stand-off annotation*, i.e. the construction of annotations in documents independent from the one containing the primary language data. Stand-off annotations refer to specific locations in the primary data by addressing byte offsets, linguistic elements such as words, or times associated with recorded data, to which the annotation applies. Compared to in-line annotation, stand-off annotation in general has the advantages of respecting the integrity of the primary data and of allowing multiple annotations to be layered over a given primary document. For dialogue act annotation, in-line annotation would moreover be fundamentally inadequate since functional segments can be discontinuous; moreover, the multiple segmentation that is used in multidimensional annotation would require alternative copies to be annotated for each dimension. Since semantic annotation typically occur at a relatively high level in a layered annotation structure, they do not necessarily refer directly to segments in the primary data, but may also refer to structures in other annotation layers.

8.1 DiAML abstract syntax

The abstract syntax of DiAML defines certain set-theoretical structures ("DiAML-structures") which contain all and exactly those elements that constitute the annotation of functional segments in dialogue with communicative functions according to a multidimensional annotation schema.

Definition. The abstract syntax of DiAML consists of a *conceptual vocabulary* and the syntactic definitions of *DiAML-segments, DiAML-tags*, and *DiAML-structures*.

a) Conceptual vocabulary:

- a finite set $P = \{P_1, P_2, ..., P_k\}$of elements called 'dialogue participants';
- a finite set $D = \{D_1, D_2, ..., D_N\}$ of elements called 'dimensions';
- a finite set of sets $DSF = \{DSF_1, DSF_2, ..., DSF_N\}$, where each element DSF_i is a finite set $DSF_i = \{F_{i1}, F_{i2}, ..., F_{in_k}\}$ of elements called 'dimension-specific communicative functions';
- a finite set $GPF = \{F_{01}, F_{02}, ..., F_{0n}\}$ of elements called 'general-purpose communicative functions';
- an ordered set Σ of elements called 'segment indicators' with ordering $\leq$.

b) Syntactic definitions:

- A DiAML-segment is a finite sequence of pairs $< \sigma_i, \sigma_j >$ of segment indicators, with $\sigma_i \leq \sigma_j$.
- A DiAML-tag is an n-tuple of pairs $< d, f >$ or triples $< d, f, t >$,where d is a dimension (element of D); $f \in DSF \cup GPF$ (i.e., f is a dimension-specific or a general-purpose communicative function); where t is a DiAML-tag; where $1 \leq n \leq N$, and where every dimension d occurs at most once.
- A DiAML-structure is a 4-tuple $< \alpha, \beta, \sigma, \mu >$, where α and β are dialogue participants; σ is a DiAML-segment and μ is a DiAML-tag.

8.2 DiAML: concrete XML-based syntax

A concrete syntax consists of he specification of names for the various ingredients in the conceptual vocabular, and a specification of how to represent DiAML-structures.

a) Concrete vocabulary:

- XML attributes to designate the representations of the components of DiAML annotation structures (dimensions, segments, communicative functions, speaker and addressee)
- names to identify dialogue participants (e.g. S, A);
- dimension names (as attribute values) : `activity, autoFeedback, alloFeedback,...`

- communicative function names (as attribute values): `inform, propositionalQuestion, setQuestion, checkQuestion, setAnswer, posAutoFeedbackInterpretation, negAutoFeedbackPerception, turnGrab, turnTake, turnKeep, turnAssign, initGreeting, reactGreeting, Apology, acceptApology,...`
- a set of segment indicator identifiers, identifying begin and end points of stretches in the language data (sequences of which forming potentially discontinuous functional segments).

b) Representation of DiAML-structures:
The following XML structure exemplifies the representation of DiAML annotation structures. In this example a functional segment of behaviour of participant S, directed to participant A, is annotated with three communnicative functions (in three dimensions). The functional segment consists of two parts, identified as `st1` and `st2`, these parts are continuous stretches of behaviour defined by their begin- and end points. The segment as a whole does not need to be continuous: there may be something in between the two constituent stretches. In the dimension `d2` the communicative function `f2` has a functional dependency on some other functional segment `s3`, i.e., it responds in some way to that segment.

```
<diaml diamlID="da1" speaker="S" addressee="A" functionalSegment="s1" communicativeFuncti
  &lt;segment sgid="s1" initStretch= "st1" finalStretch="st2" >
   <stretch stid="st1" start="st1a" end="st1b" nextStretch="st2" >
   <stretch stid="st2" start="st2a" end="st2b" >
  &lt;/segment&gt;
  <comfuns id="fs1" d1fFunction="df1" d2Function="df2" d3Function="df3" >
    <dimfun dim = "d1" function = "f1">
    <dimfun dim = "d2" function = "f2" functionalDependency="s3">
    <dimfun dim = "d3" function = "f3">
   </comfuns>
</diaml>
```

8.3 DiAML semantics

Annotating natural language documents with tags that do not have a clear semantics would be a fundamentally dubious enterprise. A fundamental requirement of semantic annotation is therefore that semantic markups should have a clear semantics (see Bunt & Romary, 2002; Bunt, 2007b). The DiAML language has a formal model-theoretic semantics associated with its abstract syntax as follows.

Definition. A model for DiAML is a pair $M = < D, F >$, where

- D is the model structure, which is itself a pair $D = < IS, P >$, where

  IS is a set of information states (viz. those of the dialogue participants)

  P is a set of functions ('update operations') from IS to IS.

- F the interpretation function, assigning to each dialogue participant (S and A) an information state from IS and to every element in $DSF \cup GPF$ a value in P. In other words, every communicative function is interpreted as an update operation on information states. An update operation is conceived here as a function that, given a semantic content, specifies how a given information state is to be updated with that content. For any particular specification of the sets of general-purpose and dimension-specific communicative functions, corresponding update operations are specified in the model structure.

The model structure, that is extremely simple, may be made more articulate by introducing structured information states, with dimensions as particular types of information that are addressed in certain update operations.)

9 Principles for schema extensions

The annotation schema defined in this project, with its core dimensions and communicative functions, cannot be expected to be ideal for every type of dialogue analysis and for every conceivable kind of dialogue. The general principles underlying the design of the schema and the annotation language should however also be useful for defining extensions or modifications.

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

- dialogue behaviour is viewed as multifunctional, i.e. each stretch of communicatively meaningful behaviour may have more than one communicative function;
- communicative functions are defined semantically in terms of how they use a semantic content to change the addressee's (and the speaker's) information state when the addressee understands (and is believed to understand) the dialogue act having that function and that content.
  It follows from this principle that there cannot be communicative functions such as Repetition, Hesitation, or Response, because such labels are descriptive of superficial characteristic of the behaviour, not of its communicative function.
  Candidate new communicative functions, proposed for being added to the annotation schema, should satisfy these conditions. They should moreover constitute the consistent set of functions with the ones already present in the annotation schema.
- The semantic content of a dialogue act cannot contribute to the definition of its communicative functions.
  It follows from this principle that there cannot be communicative functions such as *Propose Date* or *Accept Location*.
  Candidate new communicative functions, proposed for being added to the annotation schema, should satisfy these conditions.
- Indirect dialogue acts are just like ordinary dialogue acts. Each indirect dialogue act is closely related to, but semantically slightly different from, its direct counterpart.
- aspect of participating in dialogue that a dialogue agent has to manage, qualifies as a dimension of dialogue act analysis if the following two conditions are satisfied:

  1. A dialogue participant can address this aspect of participating in a dialogue through linguistic and/or nonverbal behaviour that has a communicative function specific for this purpose;
  2. This aspect of participating in a dialogue can be addressed independently of other aspects, i.e., dialogue utterances can have a communicative function in one dimension, independent of its functions in other dimensions.

  The first of these criteria means that we are considering an aspect of communication that can be distinguished according to empirically observable behaviour in dialogue. The second condition requires dimensions to be orthogonal. Both conditions should be satisfied by any potential dimension that is proposed to be added to the annotation schema.

Annex A
(informative)
LIRICS data categories for communicative functions

To be copied from LIRICS deliverable D4.3 and reformatted.

Annex B
(informative)
Annotation guidelines

noteThis section has been copied from he DIT^{++} annotation guidelines – see `http://dit.uvt.nl`, and should in a later stage be adapted to the details of the final annotation schema defined in this project.

Dialogue act annotation is about indicating the kind of intention that the speaker had; what kind of thing was he trying to achieve? When participating in a dialogue, this is what agents are trying to establish. The first and most important two guidelines follow from this.

1. First and most important guideline: **"Do as the Addressee would do!"**
When assigning annotation tags to a dialogue utterance, put yourself in the position of the participant at whom the utterance was addressed, and imagine that you try to understand what the speaker is trying to do. Why does (s)he say what (s)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 information that you could have if you were the actual addressee, and like the addressee, try to interpret the speaker's communicative behaviour as best as you can.

2. Second and equally important guideline: **"Think functionally, not formally!"**
The linguistic form of an utterance often provides vital clues for choosing an annotation, but such clues may also be misleading; in making your choice of annotation tags you should of course use the linguistic 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, SetQuestions are questions where the speaker wants to know which elements of a certain domain have a certain property. In English, such questions often contain a word beginning with "wh", such as *which* as in *Which books did you read on your holidays?* or *where* in *Where do your parents live?*. But in other languages this is not the case; moreover, even in English not all sentences of this form express a SetQuestion: *Why don't you go ahead* is for instance typically a Suggestion rather than a question.

Similarly, PropositionalQuestions are questions where the speaker wants to know whether a certain statement is true or false. Such sentences typically have the form of an interrogative statement, such as *Is The Hague the capital of the Netherlands?* or *Do you like peanut butter?* But not all sentences of this form express a PropositionalQuestion; for example, *Do you know what time it is?* functions most often as in IndirectSetQuestion (*What time is it?*), and *Would you like some coffee?* is an Offer; *Shall we go?* is a Suggestion.

3. Another important general guideline is: **"Be specific!"**
Among the communicative functions that you can choose from, there are differences in specificity, corresponding with their relative positions in hierarchical subsystems. For instance, a CheckQuestion is more specific than a PropositionalQuestion, in that it additionally carries the expectation that the answer will be positive. Similarly, a Confirm is more specific than a PropositionalAnswer, in that it carries the additional speaker that the addressee expects the answer to be positive.

In general, try all the time to be as specific as you can. But if you're in serious doubt about specific functions that you could choose between, then simply use a less specific function tag that subsumes the more specific tags.

4. On indirect speech acts: **"Code indirect speech acts just like direct ones."**
Standard speech act theory regards indirect speech acts, such as indirect questions, as just an indirect *form* of the same illocutionary acts. By contrast, the DIT++ taxonomy incorporates the idea that indirect dialogue acts signal subtly different packages of beliefs and intentions than direct ones. For example, the direct question *What time is it?* carries the assumption that the addressee knows what time it is, whereas the indirect question *Do you know what time it is?* does not carry that assumption (it does at least not *express* that assumption; in fact it

questions it).

5. On implicit functions: **"Do not code implicit communicative functions, that can be deduced from functions that you have already assigned."**
Implicit communicative functions occur in particular for positive feedback. For example, someone answering a question may be assumed to (believe to) have understood the question. So any time you annotate an utterance as an answer (of some sort), you might consider annotating it also as providing positive feedback on the interpretation of the question that is answered. Don't! It would be redundant.

Notice also that the definition of a positive (auto-) feedback act concerning interpretation stipulates that the speaker wants the addressee to know that he (speaker) has understood the question. A speaker who answers a question does not so much want to tell the addressee that his question was understood – that's just a side-effect of giving an answer, that no speaker can avoid.
Similarly for reacting to an offer, a request, a suggestion, etc.

6. Guidelines for the annotation of feedback functions.
Negative feedback, where the speaker wants to indicate that there was a problem in processing a dialogue utterance, is always explicit and as such mostly easy to annotate.

6.1 *Implicit and explicit positive feedback.*
Positive feedback is sometimes given explicitly, and very often implicitly.
Examples of explicit positive auto-feedback are the following utterances by B, where he repeats part of the question by A:
A: *What time does the KLM flight from Jakarta on Friday, October 13 arrive?*
B: *The KLM flight from Jakarta on Friday, October 13 has scheduled arrival time 08.50*
B: *The flight from Jakarta on Friday has scheduled arrival time 08.50*
B: *The KLM flight from Jakarta on October 13 has scheduled arrival time 08.50*
B: *The flight from on October 13 has scheduled arrival time 08.50*

In such cases, the utterance by B should be annotated as having, besides the general-purpose function SetAnswer in the Activity dimension, also a function in the Auto-Feedback dimension (see below).

By contrast, the short answer: *At 08.50* would carry only implicit feedback information, and should therefore, following Guideline 5, not be coded in the Auto-Feedback dimension.

6.2 *Levels of feedback.*
Several levels of feedback may be distinguished:
1. participant A *pays attention* to participant B's utterance.
2. A *perceives* B's utterance, i.e. A recognizes the words and nonverbal elements in B's contribution.
3. A *understands* B's utterance, i.e. A assigns an interpretation to B's utterance, including what A believes B is trying to achieve with this utterance (what are his goals and associated beliefs about the task/domain and about A).
4. A *evaluates* B's utterance, i.e. A decides whether the beliefs about B that characterize his understanding of B's utterance, can be added to A's model of the dialogue context, updating his context model without arriving at inconsistencies.
5. A *'executes'* B's utterance, i.e. A performs actions which are appropriate for achieving a goal that he had identified and added to his context model. (For instance, executing a request is to perform the requested action; executing an answer is to add the content of the answer to one's information; executing a question is to look for the information that was asked for.)

There are certain relations between these levels: in order to execute a dialogue act one must have evaluated it positively ("accepted" it); which is only possible if one (believes to) have understood the corresponding utterance; which presupposes that one perceived the utterance in the first place, which, finally, requires paying attention to what is said. So for instance positive auto-feedback about the acceptance of the addressee's previous utterance implies positive feedback at the "lower" levels of understanding, perception, and attention. *For positive feedback functions a higher-level function is more specific than the lower-level functions.* (Remember that a function is more specific if it implies other functions.)

For negative feedback the reverse holds: when a speaker signals the impossibility to perceive an utterance, he implies the impossibility to interpret, evaluate and execute it. So *negative feedback at a lower level implies negative feedback at higher levels.*

Since, following Guideline 3, you should always be as specific as possible, you should observe the following guideline for annotating feedback functions:

Guideline 6: **When assigning a feedback function, choose the most specific level of feedback in the case of positive feedback that you feel to be appropriate, and choose the least specific level in the case of negative feedback.**

While this guideline instructs you to be as specific as possible, sometimes you'll be in serious doubt. You may for instance find yourself in a situation where you have no clue whether a feedback signal (such as *OK*) should be interpreted at the level of interpretation or that of evaluation. In such a case you should use the less specific of the two, since the more specific level would mean that you "read" more into this utterance than you can justify.

In practice, it is often difficult to decide the level of feedback that should be chosen. One of the reasons for this is that the same verbal and nonverbal expressions may be used at most of the levels (with a tendency to signal feedback (positively or negatively) with more emphasis as higher levels of processing are involved). It may happen that you encounter a feedback signal and you have no clue at all at which level you should interpret that signal. In this situation the annotation schema allows you to use the labels 'Positive' and 'Negative', which leave the level of feedback unspecified.

7. Guidelines for the annotation of Interaction Management functions.
7.1 Turn Management.
General guideline: **"Code Turn Management functions only when these are not just implied."**
In a spoken dialogue, the participants take turns to speak. (Their nonverbal behaviour is not organised in turns; both participants use facial expressions and gestures more or less all the time.) A turn, that is a stretch of speech by one of the participants, in general consists of smaller parts that have a meaning as a dialogue act; these parts we call "utterances". Turn Management acts are the actions that participants perform in order to manage the allocation of the speaker role. These acts are subdivided into acts for taking the turn (utterance-initial acts) and those for keeping the turn or giving it away (utterance-final acts). Usually only the first utterance in a turn has an utterance-initial function and only the last an utterance-final one. The non-final utterances in a turn do not have an utterance-final function, except when the speaker signals (for example by using a rising utterance-final intonation) that the utterance is not going to be the last one in the turn, that he wants to continue. In that case the utterance has a Turn Keeping function. Except for the first one, the utterances in the turn do not have an utterance-initial function; the speaker does not have to perform a separate act in order to continue; all he has to do is to continue speaking.

When a speaker accepts a turn that the addressee has assigned to him through a Turn Assign act, the utterance should be annotated as having the utterance-initial function Turn Accept only when the speaker performs a separate act for the purpose of accepting the turn, so don't code this when the turn is accepted implicitly by simply starting to speak.

Similarly, an utterance should be annotated as having the utterance-initial function Turn Take only if the speaker performs a separate act to that effect. If he just goes ahead and makes a contribution to the dialogue, without first signalling his intention to do so, then the utterance should not be marked with an utterance-initial Turn Management function.

The verbal as well as nonverbal activities that a speaker performs to seize the turn should be marked as Turn Grabbing, but the utterance that follows *after* he has seized the turn should not be marked as having an utterance-initial Turn Management function.

7.2 Time Management.
When a speaker is buying time, using fillers such as *Well,..; Let's see,...* , then the utterance should be annotated as having the Stalling function in the Time Management dimension. There may be several reasons why a speaker wants to have more time; it may be that the speaker has trouble completing his current utterance, or that he is interrupted by some urgent event that requires his attention before he can continue the dialogue. but it may also be that he needs some time to find some information (for instance, for answering a question). So when you encounter a Stalling act, you may well pay attention to the reason why the speaker is stalling. (For instance, Stalling often goes hand in hand with turn acceptance or turn keeping.) However, don't speculate; only code additional functions for which you have evidence.

7.3 Topic Management.
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 both being a subtopic of a more general topic. Implicit topic management should not be encoded; it would be redundant. Topic Management functions should be annotated only if the speaker explicitly introduces or closes a topic, or signals his intention to do so.

7.4 Contact Management.
The management of contact in the sense of both partners being ready to send and receive messages to and from each other, is important especially in other than face-to-face situations, such as telephone conversations, video-conferencing, and internet chatting.
Note that in many languages expressions used for establishing contact can often be used for other purposes as well, for example for greeting (*Hello!*). When annotating a dialogue where this happens, the utterance in question should be marked as having both a Contact Management function and a Social Obligation Management function.

7.5 Own Communication Management.
Own Communication Management (OCM) functions should be coded whenever a speaker signals that he made a speech error and/or wants to edit what he is saying. Since this typically requires some extra effort and time, OCM acts often go hand in hand with acts whose function is to win time, such as hesitations (*Ehm...*), which have a Stalling function. See also 7.2.

7.6 Partner Communication Management.
Partner Communication Management (PCM) functions should be coded whenever a speaker signals that he believes the addressee made a speech error or has difficulty in completing an utterance, for instance being unable to recall a name or to find the right words to express something. The use of dimension-specific PCM functions for this purpose is typically only possible by interrupting the dialogue partner or in immediate response to a partner utterance.

7.7 Dialogue Structuring.
These functions should be coded only when the speaker explicitly signals something about his intention to open or close the dialogue, or to continue in a particular way.

Across the board, the following guideline applies to the encoding of Interaction Management functions:

Guideline 7: **"Code only explicit Interaction Management functions."**

8. Guidelines for annotating Social Obligation Management (SOM) functions.
Utterances that serve a 'social' purpose such as greetings, thanks, and apologies can often be used for other purposes as well. Greetings like *Hello!* , for example, can be used also for establishing contact (Contact Management function) and/or for opening a dialogue (a Dialogue Structuring function). Also, an expression of thanks can be used to signal that the speaker wants to soon end the dialogue (Dialogue Structuring function PRE-CLOSING), and can also be used for overall positive feedback. In such cases, the utterances should be coded with the appropriate functions in all these dimensions.

Guideline 8: **"When coding an utterance as having a SOM function, look out for additional functions in other dimensions."**

B.1 Note on segmentation

The segmentation of a dialogue into utterances may present several difficulties.

First, if you're working form a transcription of a spoken dialogue, the segmentation in the transcription is not necessarily perfect. You may run into cases where you would prefer the utterance to be segmented as a sequence of parts that each have a functional interpretation. In such a case it is best to assign the various tags that you would prefer to assign to the parts to the utterance as a whole. Or conversely, it may also happen that a turn has been segmented into certain parts, where you would want to annotate the longer utterance formed by these parts together. In such a case it is recommended to annotate all these parts with the same tags.

Second, an utterance may be self-interrupted by a part that has a different communicative function, as in the following example: *When***, I mean what time,** *does the train to* **ehm,...** *Viareggio leave?* Here the utterance as a whole is a SetQuestion; it includes a Self-Correction (*I mean what time*) and a Stalling utterance (*ehm*). In such cases, again, it is best to assign the tags for the intervening parts of the utterance to the utterance as a whole.

Third, it may happen that a dialogue act corresponds to (parts of) more than one turn, as in the following example, where the utterances in turns 1 and 3 together form a SetAnswer:

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.*

In such a case it is best to give each of these parts the same tag (WH-ANSWER, in this example).

Annex C
(informative)
Completely annotated examples

To be added: examples from LIRICS test suites as documented in LIRICS deliverable D4.4.

Annex D
(informative)
ISO-DiAML schema

To be provided.

Annex E
(informative)
Editors, contributors and meetings

E.1 Editors and contributors

This document has been produced by Harry Bunt making use of documents from the LIRICS project, and of presentations and discussions at ISO TC 37/SC 4 meetings and joint workshops with the ACL-SIGSEM Working Group on the Representation of Multimodal Semantic Information.

Co-authors of background LIRICS documents and publications include Amanda Schiffrin, Jeroen Geertzen, Volha Petukhova and Laurent Romary; LIRICS participants in discussions include Claudia Soria, Nicoletta Calzolari, Tomaso Casselli, Monica Monachini, Valeria Quocchi, Anna Joan Casademont, Koiti Hasida, Thierry Declerck and Gil Francopoulo. Through participation in joint TDG 3/SIGSEM WG meetings many other people have directly or indirectly contributed to this work, including Jens Allwood, Jae-Woo Chae, Chu-Ren Huang, Nancy Ide, Dafydd Gibbons, Jerry Hobbs, Simon Keizer, Staffan Larsson, Roser Morante, Andrei Popescu-Belis, James Pustejovsky, David Traum, and Thorsten Trippel.

E.2 Meetings

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

- January 10-11, 2005 in Tilburg, The Netherlands, in conjunction with the 6th International Conference on Computational Semantics (IWCS-6). Joint meeting with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information.
- September 22-23, 2005, at INRIA-LORIA in Nancy, France during the Nancy Inference Week. Joint meeting with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information.
- January 20-22, 2006 at Jeju island, Korea (as part of TC 37/SC 4 meeting).
- April 20-22, 2006, Marina del Rey, California, Institute for Information Sciences (ISI). Invitation-only joint workshop with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information.
- August 22-24, 2006, Beijing (at ISO TC 37 annual meeting).
- October 26-28, 2006, Brandeis University, Boston, USA. Meeting to start the work on temporal annotation in the form of ISO TC 37/SC 4 project "Semantic Annotation Framework (SemAF) Part 1, Time and Events".
- January 8-9, 2007, Tilburg, The Netherlands. Joint meeting with the ACL SIGSEM Working Group on the Representation of Multimodal Semantic Information in conjunction with the 7th International Conference on Computational Semantics (IWCS-7).
- May 7-9, 2007, Paris, France, in conjunction with a meeting of the LIRICS project.
- January 12-13, 2008, Hong Kong (in conjunction with ISO TC 37/SC 4 meeting and ICGL conference).

Bibliography

[1] Ahrenberg, L., N. Dahlback & A. Jonsson (1995) Codings schemas for Studies of Natural Language Dialogue. In: Working Notes from the AAAI Spring Symposium, Stanford.

[2] Allen, J. & M. Core (1997) DAMSL: Dialogue Act Markup in Several Layers (Draft 2.1). Technical Report, Multiparty Discourse Group, Discourse Resource Initiative, September/October 1997. http://www.cs.rochester.edu/research/cisd/resources/damsl/RevisedManual/

[3] Allen, J. and R. Perrault (1979) Plans, Inferences, and Indirect Speech Acts. In *Proceedings of the 17th Annual Meeting of the ACL*.

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

[5] Allwood J. (1997) A critical look at speech act theory. In: O. Dahl (ed.) *Logic, Pragmatics, and Grammar*.

[6] Allwood, J., J. Nivre & E. Ahlsen (1994) Semantics and Spoken Language Manual for Coding Interaction Management. Report from the HSFR project Semantik och talsprak.

[7] 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.* John Benjamins, Amsterdam, pp. 47–80.

[8] Appelt, D. (1982) Planning Natural Language Utterances to Satisfy Multiple Goals. SRI International AI Center Technical Note 259.

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

[10] Bunt, H. (2000a) Dialogue Pragmatics and Context Specification. In: Harry Bunt & William Black (eds.) *Abduction, Belief and Context in Dialogue*. Benjamins, Amsterdam: 81–150.

[11] Bunt, H. (2005). A Framework for Dialogue Act Specification. Joint Workshop of ACL SIGSEM Working Group on the Representation of Semantic Information and ISO TC 37/SC 4 Thematic Domain Group 3: Semantic Content. Tilburg, January 2005.
`http://sigsem.uvt.nl/wg/meeting4.fdas.pdf`

[12] Bunt, H. (2006). Dimensions in Dialogue Act Annotation. In Proceedings LREC 2006 Workshop.

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

[14] Bunt, H. (2000) Dialogue pragmatics and context specification. In H. Bunt & W. Black (eds.) Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics. Amsterdam: Benjamins, 81-150.

[15] Bunt, H. (2007a) Multifunctionality and multidimensional dialogue act annotation. In: E. Ahlsén et al., editors, *Communication - Action - Meaning*. Gothenburg University, pp.237-259.

[16] Bunt, H. (2007b) 7-9 he Semantics of Semantic Annotation. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation (PACLIC-21)*, Seoul, November 2007, 1-30, pp. 13-28.

[17] Bunt, H. & Y. Girard (2005) Designing an open, multidimensional dialogue act taxonomy. In C. Gardent & B. Gaiffe (eds.) DIALOR'05, Proceedings of the Ninth Workshop on the Semantics and Pragmatics of Dialogue, Nancy, June 2005, 37-44.

[18] Bunt, H. and L. Romary (2002). Towards multimodal content representation. In Proceedings LREC 2002 Workshop.

[19] Bunt, H. & L. Romary (2004) Standardization in Multimodal Content Representation: Some Methodological Issues. Proceedings LREC 2004, June, Lisbon, 2219-2222.

[20] Bunt, H. & A. Schiffrin (2006) Methodological aspects of semantic annotation. In Proceedings LREC 2006, Genova, May 2006.

[21] Bunt, H. & A. Schiffrin (2006) LIRICS deliverable D4.2, Preliminary set of semantic data categories.

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

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

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

[25] Core, M., M. Ishizaki, J. Moore, C. Nakatani, N. Reithinger, D. Traum & S. Tutiya (1998) The Report of The Third Workshop of the Discourse Resource Initiative, Chiba University, Chiba, Japan. http://cogsci.l.chiba-u.ac.jp/∼tutiya/ Publications/DRI98Report.pdf

[26] Core & Allen (1997) Coding Dialogs with the DAMSL Annotation schema. *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, MA.

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

[28] Geertzen, J. & H. Bunt (2006) Measuring annotator agreement in a complex, hierarchical dialogue act schema. In *Proc. SIGDIAL 2006*, Sydney.

[29] 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, September 2007, pp. 140–149.

[30] Jekat, S., A. Klein, E. Maier, I. Maleck, M. Mast & J.J. Quantz (1995) Dialogue acts in Verbmobil. Verbmobil Report 65, April 1995. DFKI, Saarbrücken.

[31] Jurafsky, D., E. Schriberg, and D. Biasca (1997) Switchboard SWBD-DAMSL Shall-Discourse-Function Annotation Coders Manual. `http://stripe.colorado.edu/∼jurafsky/manual.august.html`

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

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

[34] Petukhova, V. V. (2005) Multidimensional interaction of multimodal dialogue acts in meetings. MA thesis,Tilburg University.

[35] 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`

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

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

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

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

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

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

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