

Guidelines for using ISO standard 24617-2

Harry Bunt
Tilburg University
harry.bunt@uvt.nl

Overview

Section 1 of these guidelines considers some general issues in dialogue act annotation. Section 2 explains the segmentation of a dialogue into functional segments. Section 3 provides guidelines for the use of DiAML and the annotation schema defined in ISO 24617-2.

1 General issues in DA annotation

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 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.

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 other physical activities including head-, hand-, arm-, and shoulder gestures, forms of touching (stroking, caressing, hugging, shaking hands, patting on the shoulder, etc.), and body posture changes. All these verbal and nonverbal activities may have a communicative meaning which can be made explicit in terms of dialogue acts. ISO 24617-2 has a general emphasis on its use for creating interoperable language resources, but it has been successfully applied also to a range of nonverbal and multimodal behaviours. (See e.g. Petukhova and Bunt, 2009e on the analysis of nodding as feedback signals.)

1.2 Dialogue settings and participants

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

1). This participant, the ‘speaker’, speaks either to the single other participant in the case of a two-person dialogue, or to one or more participants in the case of multi-party dialogue. These participants are the addressees of the dialogue acts performed by the speaker.

There are certain formalized interactive situations where the role of addressee does not coincide with the person(s) that the speaker is in fact addressing. For example, in debates in the British House of Commons the person who occupies the speaker role is formally addressing the Speaker of the House, but his words are in fact aimed at a particular representative or cabinet member, or at a 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. Communicative functions are defined in ISO 24617-2 as the way in which the speaker intends to affect the information state(s) of the addressee(s), hence in such situations the annotation of the speaker’s utterances should be determined by considering whose information states the speaker is principally trying to influence.

1.3 Annotation purposes and unusual annotation situations

ISO 24617-2 is intended for use by human annotators and by automatic annotation systems. It has been tested for being useful for both these purposes.

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

1.4 Explicit and implicit, implied and indirect functions

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

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

A’s utterance is an Offer; B’s response is an Accept Offer by virtue of its linguistic form and the fact that it occurs immediately after an Offer. Since an offer can only be accepted

when it has been understood, B's response by implication also has a positive auto-feedback function. The Accept Offer in this example has a functional dependence relation to the preceding Offer.

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:

1. A communicative function F_2 is *logically entailed* by another one F_1 because F_1 is a special case of F_2 . This happens in hierarchies of communicative functions like the general-purpose functions of these standard, where for instance a Confirm is a special case of an Answer, and a Correction is a special case of a Disagreement, which in turn is a special case of an Inform. Another type of entailment exists between a dialogue act that responds to a previous dialogue act, and feedback about the processing of that previous utterance. For example, an Accept Offer entails positive auto-feedback about the processing of the utterance which expressed the Offer.
2. A communicative function F_1 may have another function F_2 as a pragmatic *conversational implicature*, i.e. in most situations where a functional segment has the function F_1 it also has the function F_2 , assuming that the dialogue participants behave cooperatively. For example, a thanking act like *Thank you* will normally be understood as also a signal of positive feedback.

Should implicit communicative functions be annotated? Annotating logically entailed functions would be redundant, since by their very nature such functions can be inferred from explicit functions. For conversationally implicated functions the situation is different, since these functions do not necessarily follow from an explicit function. It is therefore recommended to annotate implicated functions. An annotator running into the situation where a functional segment has an explicitly expressed communicative function and an implied function, should decide whether the implied function is a logical consequence or a matter of what is plausible in the given context. In the first case the implied function should not be annotated; in the second case it should. For more details about types of implicit functions and strategies for how to deal with them see Bunt (2011).

Standard speech act theory mostly regards indirect speech acts as just another *form* of the same communicative act as the direct form. By contrast, ISO 24617-2 incorporates the view that indirect forms signal subtly different packages of beliefs and intentions than direct ones. For example, the direct request *Tell me what time it is please* carries the assumption that the addressee knows what time it is, whereas the indirect request *Do you know what time it is?*, or *Can you tell me what time it is?*, does not carry that assumption (it does at least not *express* that assumption; in fact it questions it), and is best interpreted as *Please tell me what time it is, if you know/can*.

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

‘conditionality’ having the value ‘conditional’. This is represented in DiAML as follows, where the notation `target="#fs1"` is used, with `#` preceding an identifier, following the joint ISO-TEI¹ standard for attaching information to source documents. More generally, this kind of notation is used for referring to entities that have been defined in another annotation, for instance at another layer of annotation, or defined in the metadata of a given document (like the sender and the addressee in this example).

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

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.

1. Do as an addressee would do.

When assigning annotation tags to a dialogue utterance (a ‘functional segment’, more precisely), put yourself in the position of the participant(s) to whom the utterance was addressed, and imagine that you try to understand what the speaker is trying to achieve. Why does he say what he says? What are the purposes of the utterance? What assumptions does the speaker express about the addressee? Answering such questions should guide you in deciding which annotation tags to assign, regardless of how exactly the speaker has expressed himself. Use all the available information that you would have if you were the actual addressee, and like the addressee, try to understand the speaker’s communicative behaviour. (As mentioned in Section 1.3 depending on the purpose of the annotation, it may also be an option for you to look ahead in the dialogue.)

2. Think functionally, not formally.

The linguistic form of an utterance often provides vital clues for choosing an annotation tag, but such clues can also be misleading; in choosing your 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, Set Questions are questions where the speaker wants to know which elements of a certain domain have a certain property. In English, such questions often contain a word beginning with “wh”, such as *which* as in *Which books did you read on your holidays?* or *where* in *Where do your parents live?* In other languages this is not the case. 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, such as *Is The Hague the capital of the Netherlands?* or *Do you like peanut butter?* But not all sentences of this form express a Propositional

¹Text Encoding Initiative, see <http://www.tei-c.org/>)

Question; for example, *Do you know what time it is?* is most often used as an indirect way of requesting to tell the time. Similarly, *Would you like some coffee?* is most likely an offer, rather than a question, and *Shall we go?* a suggestion.

3. Be specific

Among the communicative functions that you can choose from, there are differences 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.

2 Segmentation

According to ISO 24617-2, dialogue acts correspond to *functional segments*, defined as *????*. In this definition, a functional segment is characterised as a *minimal* stretch of communicative behaviour that has a communicative function; the requirement of being 'minimal' has been added in order to ensure that communicative functions are assigned as accurately as possible to those stretches of behaviour which express these functions. Consider the following example (from a Map Task dialogue):

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

This stretch of behaviour could be marked up as expressing an **Instruct** act and an **Inform** act which explains the term "moon lander". However, in order to do that accurately it is best to segment this stretch into two functional segments: *fs1* = "and then go direction that moon lander" and *fs2* = "that thing on those legs", and to be accurate in assigning the **Instruct** function to segment *fs1* only and the **Inform** function to *fs2* only, rather than assigning both of them to the entire utterance. More fine-grained segmentation also allows us to indicate the fact that the **Inform** in *fs2* is an explanation of something in the **Instruct** in *fs1* as in the following example, where a 'rhetoricalLink' connects the dialogue act da2 with the dialogue act da1 through an 'explanation' relation:

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

There are cases where the identification of the minimal stretch of behaviour that corresponds to a functional segment is not obvious, in particular when a longer stretch could

be said to express a particular function, but where it consists of smaller parts which could also be said to express that same function. Example (5) illustrates this (from a Map Task dialogue):

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

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

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

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

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

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

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

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

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

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

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

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

If the pre-segmentation does not distinguish the segment formed by (8.1) and (8.3) as a single functional segment, but treats them as two separate segments, then it is best to give each of these parts the same tag (Answer, in this example), and code them all as having a functional dependency relation with the same question. In this way it is clear that they are all part of an answer to the same question.

3 Representing annotations in DiAML

According to the abstract syntax of DiAML, as specified in ISO 24617-2, a DiAML annotation structure formally consists of a functional segment and a set of annotations, which contain information about sender, addressee(s), communicative functions, function qualifiers, dimensions, and functional and feedback dependence relations, and rhetorical relations. In order to be compliant with the ISO Linguistic Annotation Framework (LAF ????), the XML-based representation of these structures assumes a three-level architecture, consisting of:

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

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

According to the metamodel in Figure 1, a dialogue act has a sender, at least one addressee, possibly other participants, a semantic content category, a communicative function (which may have one or more qualifiers), and possibly functional and feedback dependence relations. This is reflected in the concrete XML-based DiAML representation of dialogue act annotation in the fact that a `dialogueAct` element has obligatory attributes `sender`, `addressee`, `communicative function`, and `dimension`, and optional attributes whose values represent qualifiers, while the optional functional relations, feedback relations, and rhetorical relations correspond to relational XML elements which may be added.

For a given functional segment in a dialogue, the sender and addressee roles are usually easy to assign. For assigning communicative functions, see sections 3.1 and 3.2. For assigning

dimensions, the decision to be made is which kind of information or actions is addressed. Is it (1) concerning the underlying task/activity; or (2) concerning the speaker's processing of previous utterances; or (3) concerning the addressee's processing of previous utterances; or (4) concerning the allocation of the speaker role; or (5) concerning the time needed to continue the dialogue; or (6) concerning the editing of what the speaker is saying; or (7) concerning the editing of what the addressee is currently saying; or (8) concerning the structure of the dialogue; or (9) concerning social obligations?

3.1 Encoding general-purpose functions

Information transfer functions

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

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

Action discussion functions

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

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

3.2 Encoding dimension-specific functions

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

Auto- and Allo-Feedback

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

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

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

In (9.2) B literally repeats part of A's question, thereby displaying what he perceived what A said. In (9.3), by contrast, B paraphrases parts of A's question, and this can be taken to indicate not only what B heard but also how B interpreted what A said (which in this example may be particularly relevant for the interpretation of '*next Saturday*').

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

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

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

Turn Management

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

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

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

Similarly, does A, by asking a question, express that he wants B to occupy the speaker role next? The answer to this question is clearly No, since A can continue for a while occupying the speaker role after asking the question, as in the following example:

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

Note that in example (11) participant A continued in the speaker role after asking a question simply by continuing to speak. This raises a rather troubling question: does continuing to speak indicate the speaker's goal to keep the turn? In that case, one should assign a turn-keeping function to nearly everything that a speaker says. A recommendation for how to go about assigning turn-management functions is to only assign such a function to those stretches of communicative behaviour which have the sole (or the main) purpose to obtain, to keep, or to get rid of the speaker role. Just starting to speak, continuing to speak, or ceasing to speak should not be annotated as expressions of Turn management functions.

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

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

Time Management

Time management functions are concerned with the sender buying some time. ISO 24617-2 distinguishes two cases:

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

In both cases there may be several reasons why the sender wants to buy some time. In the first case this is most probably because he is looking for the right words to express what he wants to convey or that he is gathering (or calculating) the information that he wants to convey. In the second case this may be because he is aware that collecting/computing

the information requires more time than is reasonable to take while continuing the dialogue, or is too complicated to allow him to continue to participate in the dialogue, or it may be that something more urgent came up, or that a disturbing sound was prohibiting him from continuing to interact.

Stalling acts often take the form of filled pauses (*ehm, 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 current status* should not be marked as Pausing acts, but as requests in the Time Management dimension, using the general-purpose function **Request**.

Own and Partner Communication Management

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

Partner Communication Management (PCM) acts similarly edit the addressee's speech, who at that moment occupies the speaker role. Two important cases are the correction of the addressee/current speaker (Correct Misspeaking), 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). In both cases the sender of the PCM act barges in and grabs the turn, or takes the turn which has become available because the addressee/current speaker is hesitating.

Discourse Structuring

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

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

Social Obligations Management (SOM)

The kind of social obligations that should be annotated depends on the kind of dialogue. Welcome and farewell greetings that play a role in starting and ending a dialogue are domain-independent, however, as are apologies and their acceptances, acts for introducing oneself, and thanking acts and their acceptances. All of these types of acts have conventional forms ('formulas') in every language. They tend to come in pairs: an initial greeting puts pres-

sure on the addressee to send a response greeting; introducing oneself puts pressure on the addressee to also introduce himself; an apology puts pressure on the addressee to accept the apology; a thanking puts pressure on the addressee to downplay what he is thanked for (like in *It was nothing; It was my pleasure*); and a farewell greeting puts pressure on the addressee to produce a response farewell greeting.

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

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

3.3 Encoding communicative function qualifiers

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

a. Certainty The sender of a dialogue act can express certainty or uncertainty about the correctness of the information provided in an information-providing act, or about his commitment to perform an action in a commissive act. This is illustrated in (12) for information-providing acts, where the expressions "*I have a hunch that*", "*probably*", "*might*", and "*I'm not sure if*" are indicators of the speaker's uncertainty. When these expressions are omitted, as in (13), the resulting sentences no longer contain any suggestion that the speaker is uncertain about the correctness of what he says. This indicates that the default value, corresponding to the unmarked case, is *certain*.

- (12) 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.

- (13) 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 (14). For such cases, the DiAML encoding with `certainty="certain"` is recommended,

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

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

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

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

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

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

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

- (16) a. A: Would you like to have some coffee?
B: Thanks, only if you have it ready.
- b. A: Can you to the presentation, if you're ready?
B: I can do that if you like.
- c. A; I'll send you an email if you give me your address.
- d. A: Can we just go over that again?
B: Just very quickly. I have to hurry you on here.
C: I don't think we have time for that, unless you make it very short.
- e. A: I can make the buttons larger.
B: No, only if we want basic things to be visible.

In (16a) we see the conditional acceptance of an offer; in (16b) a conditional request, with a conditional acceptance; in (16c) a conditional promise; in (16d) a two conditional acceptances of a request; and in (16e) a conditional rejection of a suggestion. Similar to the case of certainty qualifiers, omission of the expressions indicating a condition leads to expressions that signal unconditional dialogue acts, hence the default value is *unconditional*, and does not need to be marked up. Explicit expressions of 'unconditionality' are hard to find, other than the adverb *unconditional*, which is hardly ever used in natural dialogue.

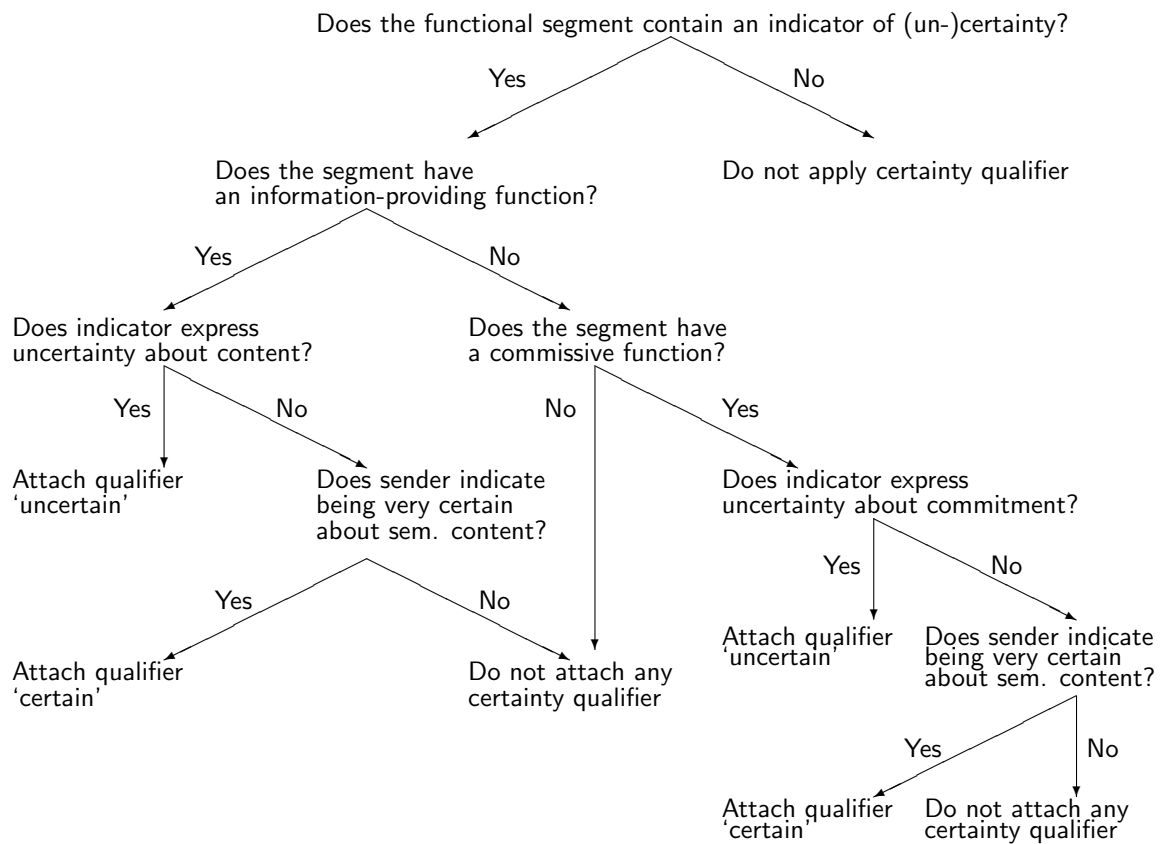


Figure 1: Decision tree for applying certainty qualifiers

Conditional dialogue acts can often be recognised by the use of conditional expressions such as *if ...* or *unless*, and *just* (as in (16d), first case) but just like in the case of certainty, these expressions can also be part of the semantic content rather than qualifiers of the communicative function. For deciding whether to add a conditionality qualifier to the annotation of a communicative function, the decision tree can be used which is displayed in Figure 4.

c. Sentiment A particular sentiment associated with the performance of a dialogue act may be annotated if the sender indicates an emotion or an attitude concerning the semantic content or the addressee, verbally or nonverbally, or both. Example (??) shows some verbal expressions of sentiment. Nonverbal expressions of sentiment exist in abundance and in

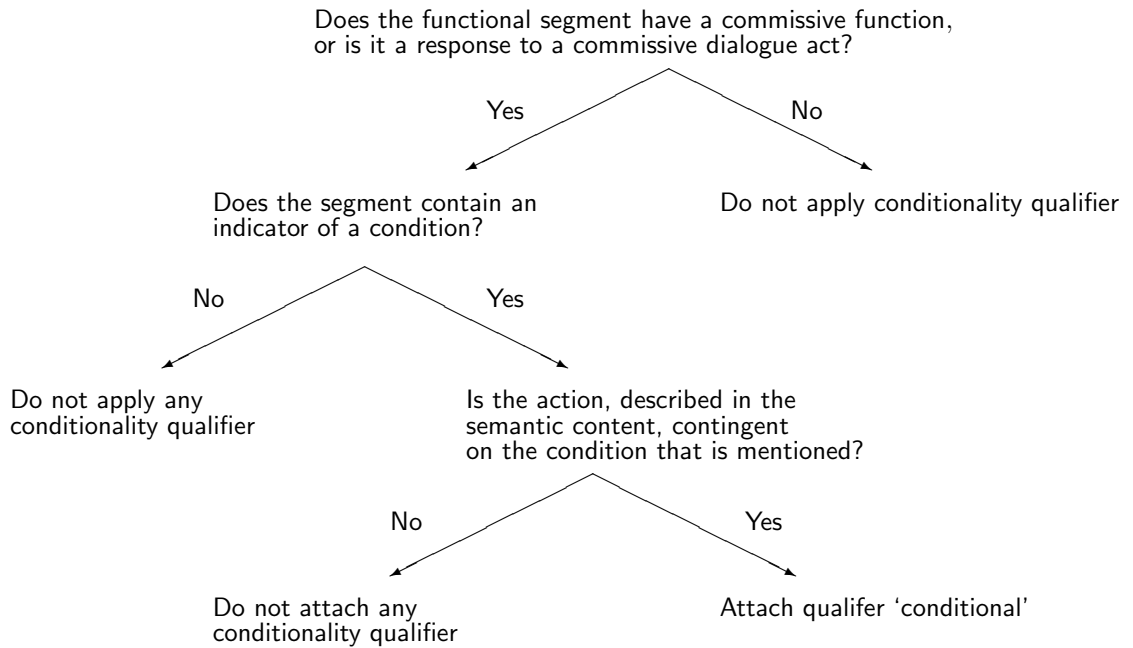


Figure 2: Decision tree for applying conditionality qualifiers

great variety, including for instance smiling (happiness), eyebrow raising (surprise), pressing lips together (angst), and sighing (sadness). Specific guidelines for sentiment annotation cannot be given here, since the class of sentiment qualifiers is not specified in ISO 24617-2.

3.4 Encoding functional dependences, feedback dependences, and rhetorical relations

Functional dependence

A dialogue act A1 is functionally dependent on a previous dialogue act A2 (its ‘functional antecedent’), if its communicative function by its very nature responds to another dialogue act. This is the case for the following core communicative functions defined in ISO 24617-2:

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

Encoding a functional dependence relation means identifying the functional antecedent and linking the two dialogue acts by means of a **functionalDependence** element. The identification of a functional antecedent is not straightforward if (a) the current dialogue act does not respond to a single dialogue act but to a combination of dialogue acts, as in (18), or (b) responds to an implicit dialogue act.

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

In (18), utterance 4 forms a functional segment with function Answer, which responds to the question formed by the dialogue acts expressed by utterances 1 and 3 together. In such a case it is recommended to mark functional dependence relations to both these dialogue acts.

Feedback dependence

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

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

Rhetorical relations

Many of the relations which may occur between units in discourse such as Justification, Explanation, Cause-Effect, or Summarization, and which in the linguistic literature are often called ‘rhetorical relations’ or ‘discourse relations’, may also occur between dialogue acts. ISO 24617-2 does not specify any particular set of such relations, and therefore does not provide detailed guidelines for their encoding. So-called ‘discourse markers; like *also*, *but*, *because*, *for example*, *in short*, *so* signal such relations as Elaboration, Contrast, Motivation, Cause, Exemplification, Summarization and Conclusion, and they are often multifunctional; for example, a protracted turn-initial *An..d,..* may be a functional segment with the functions **Turn Take**, **Stalling**, **Auto-Positive**, and also be the first part of a longer functional segment expressing a dialogue act which has an Elaboration relation to a previous dialogue act.²

²(See Petukhova and Bunt, 2009a on multifunctionality of discourse markers, and Hovy and Maier, 1995; and Mann and Thompson, 1988 more generally on discourse relations and rhetorical relations.