Using DiAML and ANVIL for multimodal dialogue annotation

Harry Bunt[1], Michael Kipp[2], and Volha Petukhova[3]

[1]Tilburg Center for Cognition and Communication, Tilburg University, The Netherlands,
[2]University of Applied Sciences, Augsburg, Germany,
[3]Human Speech and Language Technologies, Vicomtech-IK4, San Sebastián, Spain
harry.bunt@uvt.nl, michael.kipp@hs-augsburg.de, v.v.petukhova@gmmail.com

Abstract

This paper shows how interoperable annotations of multimodal dialogue, which apply the annotation scheme and the markup language (DiAML, Dialogue Act Markup Language) defined ISO standard 24617-2, can conveniently be obtained using the newly implemented facility in the ANVIL annotation tool to produce XML-based output directly in the DiAML format. ANVIL offers the use of multiple user-defined 'tiers' for annotating various kinds of information. This is shown to be convenient not only for multimodal information but also for dialogue act annotation according to ISO standard 24617-2 because of the latter's multidimensionality: functional dialogue segments are viewed as expressing one or more dialogue acts, and every dialogue act belongs to one of a number of dimensions of communication, defined in the standard, for each of which a different ANVIL tier can conveniently be used. Annotations made in the multi-tier interface can be exported in the ISO 24617-2 format, thus supporting the creation of interoperable annotated corpora of multimodal dialogue.

Keywords: multimodal dialogue, annotation standards, annotation tools

1. Introduction

The creation of interoperable language resources, such as annotated corpora, depends crucially on the application of common annotation and representation schemes on the one hand, and the availability of tools for using these schemes on the other hand. In the area of semantic annotation, ISO standard 24617-2[1] provides a comprehensive application-independent scheme for dialogue act annotation, which is applicable to spoken, typed, and multimodal dialogue, and includes the definition of the Dialogue Act Markup Language (DiAML).

The ANVIL[2] annotation tool (Kipp, 2001, 2008, 2012) was developed was for the annotation of digital video, offering a graphical user interface for creating annotation elements on temporal, hierarchical, user-defined tiers. ANVIL has proved to be a very useful tool for the annotation of multimodal and spoken dialogue (see e.g. Petukhova and Bunt, 2009a; 2009b), where its tiered representation form is convenient for annotating the communicative behaviour of a dialogue participant in each modality in a separate tier (e.g. using one tier for speech, one for gaze direction, one for head movements, and one for body posture). See the illustrative example in Figure 1.

ANVIL's tiered format has also proved convenient for *multidimensional* annotation, when stretches of communicative behaviour are marked up with multiple tags, especially when the various tags are provide functional information relating to a particular dimension of interaction, such as feedback, turn taking, or time management (see Petukhova, 2011; Petukhova and Bunt, 2012, and see Section 2).

An attractive features of ANVIL is its customizability, allowing user-defined tiers and the import of tag sets. Annotations made with ANVIL can be exported in various formats, including an XML format. As a service to the community, ANVIL has recently been extended with the possibility to export annotations in the DiAML representation format, thus supporting the creation of ISO-compatible, interoperable dialogue act annotations. In this paper we describe the application of this new version of ANVIL to support the creation of multidimensional annotations according to ISO 24617-2 and DIT^{++} (release 5, see below).

2. Multidimensionality in ISO 24617-2, DIT++, and ANVIL

The development of the ISO 24617-2 annotation scheme took as its starting point the DIT^{++} scheme (Bunt, 2007), which combined the original DIT scheme (Bunt, 1994) with concepts from DAMSL (Allen and Core, 1997) and various other schemes into a comprehensive domain-independent annotation scheme. Parallel to the development of ISO 24617-2, a new release of the DIT^{++} scheme was also defined,[3] which includes the ISO 24617-2 scheme and extends it with concepts for annotating contact management activities and fine-grained forms of feedback.

The ISO 24617-2 and DIT^{++} schemes share a number of basic design features, which are relevant for the discussion in this paper.

1. Communicative behaviour in dialogue is viewed as *multifunctional*, i.e. as having multiple communicative functions (Bunt, 2011). This view leads to 'multidimensional' annotation, i.e., to the annotation of stretches of dialogue behaviour with multiple functional tags. The ISO standard has adopted the DIT^{++} approach of interpreting this phenomenon in terms of

[1]ISO 24617-2 Semantic Annotation Framework, Part 2: Dialogue Acts, was accepted as an international standard in 2011.

[2]www.anvil-software.de

[3]DIT^{++} Release 5, see http://dit.uvt.nl.

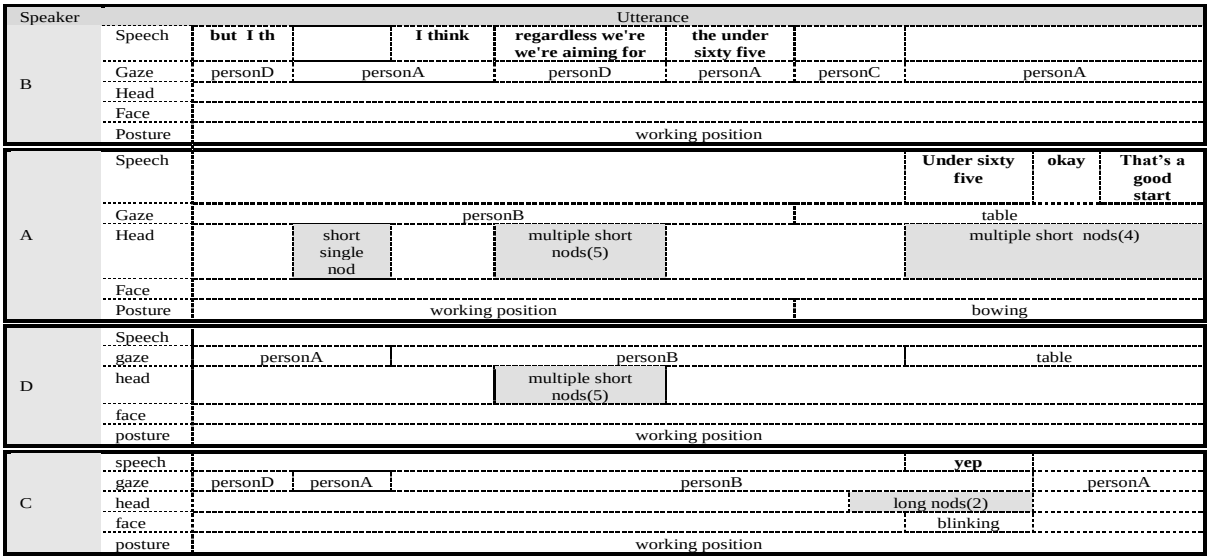

Figure 1: Example of coding multimodal dialogue behaviour.

'dimensions', defined as orthogonal aspects of communication that dialogue acts can be concerned with. On this view, a dialogue utterance can have a function in more than one dimension, but not more than one in any given dimension (*modulo* implied functions); this is because any two communicative functions that can be used in given dimension are either exclusive alternatives, or one implies the other.

2. Dialogue acts are viewed semantically as operators for updating the information states of dialogue participants. A dialogue act has two main properties: a semantic content, that describes the objects, properties, events,... that the dialogue act is about, and a communicative function, that specifies how the semantic content should be used to update an information state. Dimensions are categories of semantic content. ISO 24617-2 defines nine dimensions: addressing information about 1) a certain (*Task*); 2) the processing of utterances by the speaker (*Auto-feedback*) or 3) by the addressee (*Allo-feedback*); 4) the management of difficulties in the speaker's contributions (*Own-Communication Management*) or 5) that of the addressee (*Partner Communication Management*); 6) the speaker's need for time to continue the dialogue (*Time Management*); 7) the allocation of the speaker role (*Turn Management*); 8) the structuring of the dialogue (*Dialogue Structuring*); and 9) the management of social obligations (*Social Obligations Management*). The DIT^{++} scheme defines one more dimension, that of *Contact Management*.

3. For most annotation schemes, such as DAMSL (Allen and Core, 1997) or HCRC Map Tak (Carletta et al., 1996), dialogue act annotation comes down to assigning one or more communicative functions to stretches of dialogue behaviour, but for ISO-24617-2 and DIT^{++} the annotation involves both communicative functions and dimensions, and optionally certain relations in which dialogue acts participate (see 7). For example, the annotation reflects the difference between a task-related question and a feedback question, as shown in (1):

   (1) a. Do you know where the meeting is? [Set Question, Task]

   b. What did you say? [Set Question, Auto-Feedback]

4. ISO 24617-2 defines a hierarchically organized set of communicative functions, divided into *general-purpose functions* and *dimension-specific functions*. Functions of the latter type can be used only in one particular dimension; examples are *Turn Take* and *Turn Release* in the *Turn Management* dimension; *Stalling* in the *Time Management* dimension, and *Apology* and *Thanking* in the *Social Obligations Management* dimension. Examples of general-purpose functions include *Inform, Question, Answer, Offer, Request, Promise, Suggest, Instruct* and *Confirm*.
   Dialogue acts may be expressed in a way that indicates an emotion or attitude on the part of the speaker, as illustrated in (2). In (2a), B accepts an offer *happily*; in (2b), A accepts a request *conditionally*, and in (2c) H answers a question with *uncertainty*. For taking these phenomena into account, ISO 24617-2 defines so-called 'qualifiers', which can be attached to a communicative function. This allows marking up a stretch of dialogue as e.g. an *uncertain Answer*.

   (2) a. A: Would you like to have a cup of tea?
   B: Yes, please! [*with a big smile*]

   b. C: Can we just go back to that.
   A: Only if we can do it very quickly.

   c. P: Are you going to the lunch meeting?
   H: Probably not.

5. The unit of dialogue that may have one or more communicative functions is taken to be a 'functional segment', rather than e.g. a turn. A functional segment is defined as a *minimal stretch of behaviour which has one or more communicative functions, minimal in the*

sense that it does not include parts which do not contribute to the expression of its communicative function in the dimension under consideration. For example, in the dialogue fragment (5), B's utterance contains the discontinuous functional segment *"The next train to Tilburg leaves at 9.32"* which has a communicative function in the *Task* dimension; the part *"let me see... um..."* does not contribute to this function and therefore does not belong to this segment.

6. Closely related to the previous point, a functional segment is defined *relative to a given dimension*. In the example just discussed, the discontinuous stretch *The next train to Tilburg leaves in just over two hours* is a functional segment in the *Task* dimension, while the stretch *let me see...* is not a functional segment in that dimension, but is a functional segment in the *Time Management* dimension. This leads to *multidimensional segmentation*, which is discussed in the next section.

7. A dialogue act can occasionally be understood on its own, but much of the time a full understanding requires taking into account how the dialogue act is related to other units in the dialogue, in particular to preceding dialogue acts. ISO 24617-2 distinguishes three types of relations within a dialogue:

   - A dialogue act can be 'functionally dependent' on a previous dialogue act, such as an answer being dependent on a question. This happens for those types of dialogue act which are inherently 'responsive' in nature, such as *Answer, Confirm, Accept Offer, Decline Offer, Accept Apology, Turn Accept, Return Greeting,...*, whose meaning depends on which dialogue act they respond to.
   - The meaning of a feedback act, which by definition provides or elicits information about the processing of a previous utterance, can only be established by taking into account which utterance(s) the feedback is about. This semantic relation is called *feedback dependence*.
   - Dialogue acts may also be pragmatically related by rhetorical relations, as illustrated by the following examples:

     (3) B: I keep losing them.
     A: That's because they don't have a fixed location.

     (4) D: I also want to discuss the target audience.
     D: I think that may influence many of our decisions.

     In (3), the semantic content of A's statement is *causally* related to that of B's contribution. In (4), D's second contribution is related to the first through a *motivation* relation.

     For representing this kind of relations between dialogue acts, ISO 24617-2 includes the possibility to annotate rhetorical relations, although the standard does not define any particular set of such relations, given the lack of a general agreement on the choice of such a set. For any particular annotation task, a set of rhetorical relations that is appropriate for that task may be specified and used as values for the attribute that the standard defines for that purpose.

3. Multidimensional Segmentation

Multidimensional segmentation means that a dialogue is not cut up into a sequence of units, but that in every dimension those segments are identified which have a communicative function in that dimension. As an example, consider the segmentation of B's turn in the following dialogue fragment.

(5)
1. A: Do you know what time the next train to Tilburg leaves?
2. B: The next train to Tilburg leaves ... let me see ... um,.. at 9.32.

Upon multidimensional segmentation of B's utterance the functional segments are identified, shown in Table 1. Note

Dimension	*Functional segment*
Auto-Feedback	the next train to Tilburg leaves
Task	the next train to Tilburg leaves at 9.32
Turn Man.	let me see...
Time Man.	let me see...
Time Man.	... um,..
Turn Man,	... um,..

Table 1: Functional segments in dialogue fragment (5).

that functional segments may be discontinuous and may overlap (e.g. a segment carrying a feedback function may overlap with a segment that carries a task-related function); they may also contain parts from more than one turn, and may have parts contributed by different speakers.

Multidimensional segmentation is useful for identifying the relevant units in spoken dialogue, due its flexibility and its strictly functional definition, rather than units defined in terms of linguistic or behavioral properties. For the same reasons it also forms a useful approach for identifying relevant segments of nonverbal behaviour in multimodal dialogue. Communicative functions may have an expression in multiple modalities, e.g. in speech, in facial expression, in nonverbal vocal sounds (chuckling, sighing, whistling, heavy breathing..), and in head gestures. This makes the notion of a functional segment in multimodal dialogue a complex object, with components in various modalities.

In (6) a short fragment is shown of a dialogue from the HCRC Map Task corpus (Carletta et al., 1996), in which three modalities are used: speech, nonverbal vocal sounds (heavy breathing) and lip gestures (lip smacks). Figure 3 shows an XML encoding of a part of this fragment, where we see encodings of (a) stretches of speech, represented in a TEI-compliant fashion by their tokenisation; (b) lip gestures, with an indication of the dialogue participant who produced the movement and with timing information; and (c) vocal (nonverbal) contributions, likewise with information about who produced the behaviour and timing information.

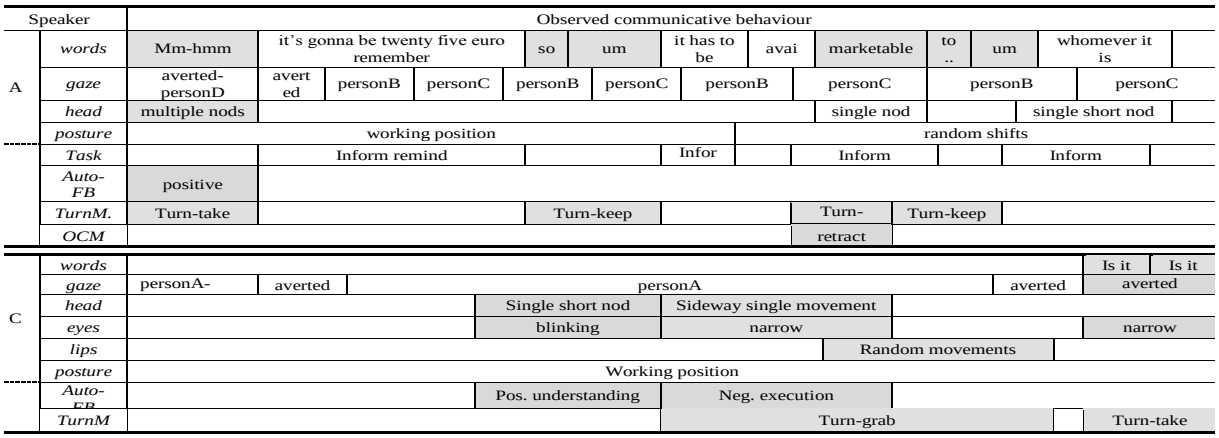

Figure 2: Example of coding and annotating multimodal dialogue behaviour.

(6)
1. P1: we're going to continue straight along ... GES_lipsmack VOC_inbreath .. um .. quite a wee distance .. um .. GES_lipsmack VOC_inbreath on that course and then we're going to turn north again
2. P2: right mm-hmm
3. P1: and...

Multidimensional segmentation applied to the first utterance in (6) yields four functional segments:

1. the purely verbal (discontinuous) functional segment *"we're going to continue straight along (...) quite a wee distance (...) on that course"*;
2. the multimodal functional segment *GES_lipsmack VOC_inbreath ... "um" ...* ;
3. the multimodal functional segment ... *"um".. GES_lipsmack VOC_inbreath*;
4. the verbal functional segment *"and then we're going to turn north again"*.

Figure 3 shows the encoding of the communicative behaviour first three of these segments, adding XML encodings of segments of 'lip behaviour' and nonverbal vocal behaviour to an encoding of verbal behaviour that follows the guidelines of the Text Encoding Initiative (TEI P5, 2007). Note that the encoding of the nonverbal behaviour includes a specification of who performs the behaviour, using the @WHO attribute, and of the start and end times of the behaviour, using the @START and @END attributes. Since it is of obvious importance to see the temporal relations between the verbal and nonverbal components of multimodal behaviour, we have added the attributes @WHO, @START and @END to sequences of verbal tokens as well (see the `verbContrib` elements in Figure 4). The information represented by these attributes is obviously available in ANVIL; the use of particular tiers in the interface, as shown in Figure 1, associates each verbal element with a speaker; moreover, ANVIL keeps a timeline (not shown in Figure 1), and the annotator's choices of the start and end of a stretch of verbal or nonverbal behaviour registers precise values for the @START and @END attributes.

Figure 8 shows an XML representation of these functional segments and their verbal and nonverbal components. In multimodal dialogue, as mentioned above, a functional segment is in general a complex object, with components formed by segments of communicative behaviour in multiple modalities. This is illustrated in Figure 4 by the functional segments `fs11` and `fs12`, both of which have (a) a verbal component, (b) a vocal component, and (c) a lip gesture component. Note also the use of the `spanGrp` element to identify the discontinuous utterance *"we're going to continue straight along (...) quite a wee distance (...) on that course"*, which forms the sole component of the purely verbal functional segment `fs10`.

4. Annotation using DiAML

The Dialogue Act Markup Language DiAML has a 3-part definition (see Bunt et al, 2010), consisting of:

(a) an *abstract syntax* that defines the class of well-defined annotation structures in set-theoretical terms;

(b) a *formal semantics* of this class of structures[4]

(c) a *concrete syntax* defining a reference representation format in XML.

In this paper we only consider the representations defined by the concrete syntax, and whenever we speak of "annotations in DiAML", we mean annotations expressed in the XML-based reference representation format defined by the DiAML concrete syntax (Ide and Bunt (2010) have shown that the representations defined by a concrete syntax of the type specified for DiAML can be converted in a straightforward way into an alternative general representation format known as the GrAF format (Ide and Suderman, 2007) which makes use of annotation graphs.) The functional segments identified in the dialogue fragment (6) according to Table 1, are represented in XML in Figure 4; the DiAML annotation of this fragment is shown in Figure 5. Note that the functional segment `fs11` is multifunctional, having both a function in the *Time Management* dimension (viz. *Stalling*) and a function in the *Turn Management* dimension (viz. *Turn Keep*).

For designing annotations of dialogue act information, it is useful to consider the various aspects of a dialogue act. ISO 24617-2 defines a dialogue act as:

[4]See Bunt (2011).

```
<xml version="1.0" encoding="UTF-8">
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <profileDescr>
<teiHeader> (...)
    <particDescr xml:id="p1"> <p>the first participant</p> </particDescr>
    <particDescr xml:id="p2"> <p>the second participant</p </particDescr></profileDescr>
</teiHeader>
<timeline unit="ms">
  <when xml:id="t1" absolute="122725"/>
  <when xml:id="t2" absolute="298377"/>
  ...
  <when xml:id="t14" absolute="1943268"/></timeline>
<head>Verbal dialogue contributions, segmented into tokens (TEI-compliant)</head>
<u><w xml:id="w99">we're</w>
  <w xml:id="w100">going</w>
  <w xml:id="w101">to</w>
  <w xml:id="w102">continue</w>
  <w xml:id="w103">straight</w>
  <w xml:id="w104">along</w></u>
<u><w xml:id="w105">um</w></u>
  <u><w xml:id="w106">quite</w>
  <w xml:id="w107">a</w>
  <w xml:id="w108">wee</w>
  <w xml:id="w109">distance</w></u>
<u><w xml:id="w110">um</w></u>
<u><w xml:id="w111">on</w>
  <w xml:id="w112">that</w>
  <w xml:id="w113">course</w></u>
<head>Nonverbal dialogue behaviour, segmented and time-stamped</head>
<kinesic type="lipMove" subtype="lipsmack" xml:id="lmv1" who="#p1"
  start="#t3" end="#t4"/>
<vocal xml:id="voc1" type="inbreath" who="#p1" start="#t5" end="#t6"/>
<kinesic type="lipMove" subtype="lipsmack" xml:id="lmv2" who="#p1"
  start="#t9" end="#t10"/>
<vocal xml:id="voc2" who="#p1" type="inbreath" start="t11" end="#t12"/>
```

Figure 3: Encoding of tokenized multimodal dialogue fragment, using TEI P5 (s;ightly simplified).

(7) *communicative activity of a participant in dialogue, interpreted as having a certain communicative function and semantic content.*

A note, added to the definition, remarks that "A dialogue act may additionally have certain functional dependence relations, rhetorical relations, and feedback dependence relations". The dialogue participant who produces a dialogue act is called the 'sender' (or 'speaker', when the dialogue act is in spoken form); being a form of 'communicative activity', there must also be one or more addressees that the sender is directing his action to.

We have also seen that communicative functions may be 'qualified' for the sender's emotion/attitude, conditionality and certainty. DiAML therefore defines concepts for annotating the following properties of a dialogue act:

(8)
1. sender
2. addressee
3. communicative function
4. dimension (category of semantic content)
5. communicative function qualifier
6. functional dependence relations
7. feedback dependence relations
8. rhetorical relations

In DiAML annotations, central stage is played by an XML element called `dialogueAct` of which the following obligatory attributes correspond to four of these properties: `@sender`, `@addressee`, `@communicativeFunction`, and `@dimension`. Function qualifiers are annotated only if the sender explicitly expresses a sentiment, certainty, or condition, and correspond to the optional attributes `@sentiment`, `@conditionality`, and `@certainty`. Functional dependences are inherent to responsive communicative functions but are undefined for non-responsive communicative functions (such as *Inform, Question, Offer, Promise, Apology, Turn Release, Stalling* and many others); they can be represented by the value of the optional attribute `@functionalDependence`. Similarly, feedback dependences are inherent to feedback acts but are not defined for dialogue acts with other communicative functions; they can be represented by the value of the optional attribute `@feedbackDependence` (see Fig. 6).

Unlike functional and feedback dependences, rhetorical relations are not an aspect of the meaning of a dialogue act, so they are not represented by an attribute in a `dialogueAct` element, but as links that connect dialogue acts, using the XML element `rhetoricalLink`, whose

```
<head>Identification of functional segments</head>
 <spanGrp xml:id="ves10" type="verbalSegment">
   <span xml:id="ts10" type="textStretch" from="w99" to="w105"/>
   <span xml:id="ts11" type="textStretch" from="106" to="w110"/>
   <span xml:id="ts12" type="textStretch" from="111" to="w114"/></spanGrp>
 <fs type="verbalContrib" xml:id="vec1" vSpan="#ves10" who="#p1" start="#t1" end="#t2"/>
 <fs type="functionalSegment" xml:id="fs10" ana="#da1">
   <f name="verbalComponent" fVal="#vec1"/></fs>
 <span xml:id="ts13" type="textStretch" from="w105" to="w106"/>
 <fs type="verbalContrib" xml:id="vec2" vSpan="#ts13" who="#p1" start="#t7" end="#t8"/>
 <fs type="functionalSegment" xml:id="fs11" ana="#da2" "#da3">
   <f name="verbalComponent" fVal="#vec2"/>
   <f name="vocalComponent" fVal="#voc1"/>
   <f name="lipComponent" fVal="lmv1"/></fs>
 <span xml:id="ts14" type="textStretch" from="w110" to="w111"/>
 <fs type="verbalContrib" xml:id="vec3" vSpan="#ts14" who="#p1" start="#t11"
   end="#t12"/>
 <fs type="functionalSegment" xml:id="fs12" ana="#da4" "#da5">
   <f name="verbalComponent" fVal="#vec3"/>
   <f name="vocalComponent" fVal="#voc2"/>
   <f name="lipComponent" fVal="#lmv2"/></fs>
```

Figure 4: Encoding of functional segments

attributes refer to the related dialogue acts.

Figure 5 illustrates the use of `dialogueAct` elements with their obligatory attributes for the three functional segments which occur in the first turn of dialogue fragment (6), as defined in Figure 4. The `@target` attribute, which can have any pointer reference as a value, is used to identify the functional segment where a dialogue act is expressed. In this representation, produced with ANVIL, five dialogue acts are represented for the three functional segments, since both the segments `fs11` and `fs12` express two dialogue acts, one in the *Turn Management* dimension and one in the *Time Management* dimension.

Figure 6 illustrates the use of the non-obligatory `dialogueAct` attributes and of `rhetoricalLink` elements in DiAML annotations, generated with ANVIL.

5. Coding DiAML in ANVIL

Before presenting the details of how DiAML structures are realized in the ANVIL tool, we describe the workflow from a practical user perspective. We assume that there is a corpus of video recordings of conversations to be analyzed with regard to communicative behavior.

5.1. Workflow

In ANVIL, the layout and functionality of tiers (also called tracks) is defined in a separate XML file, the so-called *specification file*. For DiAML, we provide a specific specification file[5] that can be used with minimal adaptation. A screenshot (just the annotation board) of an annotation session in progress can be seen in Fig. 7. Each video in the corpus is annotated manually, resulting in ANVIL data files (.anvil). These files can now be exported to DiAML format in ANVIL. In summary, the workflow is: (a) obtain and adjust our DiAML-ANVIL specification file, (b) annotate videos and store annotations in .anvil files, and (c) export each .anvil file to DiAML format.

[5]You can download this file on www.anvil-software.de: click on DiAML in the main menu.

5.2. ANVIL's encoding facilities

As a multi-tier coding tool, ANVIL provides various mechanisms to encode information (Kipp, 2001, 2012, and Kipp, *to appear*). To understand the nature of these mechanisms is crucial when crafting a mapping from ANVIL structures to formalisms like DiAML, especially when the formalism contains multidimensionality and dependencies across dimensions.

Rich annotation elements: Single elements contain not only a simple textual string (as is the case with actually all other coding tools) but a set of attribute-value pairs. This allows us to put various dimensions of data into a single element, avoiding visual clutter on the annotation board (in other tools, every attribute has to be encoded in another separate tier). For DiAML, we utilize 14 attributes in each element. In addition, these attributes have a *type* (text / number / controlled vocabulary / boolean etc.) which is reflected in the user interface (text box / number slider / drop-down menu etc.). This restricts user input, thus reducing errors.

Track temporal relationships: Two tracks may have a systematic temporal containment relationship. For instance, one may want to group elements of a track "words" such that they belong to a unique element of a track "sentence". ANVIL supports such relationships which again allow the user interface to offer more efficient coding and avoidance of errors because an explicit time-alignment of dependent elements is performed automatically. In DiAML (Fig. 7 all tracks of a participant depend on the participant's *utterance* track. In Fig. 7, the "checkQuestion" element (marked blue) in the *AutoFeedback* track consists of the two words "slightly northeast" in the *utterance* track (of participant B).

Cross-tier logical pointers: While track relationships encode a very specific temporal containment/equivalence relation, one may need to encode arbitrary logical relations

```
<diaml xmlns="http://www.iso.org/diaml">
<dialogueAct xml:id="da1" target="#fs10" sender="#p1" addressee="#p2"
  communicativeFunction="instruct" dimension="task"/>
<dialogueAct xml:id="da2" target="#fs11" sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da3" target="#fs11" sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/>
<dialogueAct xml:id="da4" target="#fs12" sender="#p1" addressee="#p2"
  communicativeFunction="stalling" dimension="timeManagement"/>
<dialogueAct xml:id="da5" target="#fs12" sender="#p1" addressee="#p2"
  communicativeFunction="turnKeep" dimension="turnManagement"/></diaml>
```

Figure 5: DiAML annotations as output by ANVIL for the functional segments defined in Figure 4.

```
<diaml xmlns="http://www.iso.org/diaml">
<dialogueAct xml:id="da6" target="#fs6" sender="#p1" addressee="#p2"
  communicativeFunction="inform" dimension="task" />
<rhetoricalLink dact="#da6" rhetoAntecedent="#da5" rhetoRel="elaborate" />
<dialogueAct xml:id="da18" target="#fs18" sender="#p1" addressee="#p2"
  communicativeFunction="positiveAutoFeedback" dimension="autoFeedback"
  feedbackDependence="#da50"/>
<dialogueAct xmlns="" xml:id="da20" target="#fs20" sender="#p1" addressee="#p2"
  communicativeFunction="confirm" dimension="alloFeedback"
  functionalDependence="#da53" /></diaml>
```

Figure 6: DiAML annotations as output by ANVIL for DiAML annotations with functional and feedback dependences, communicative function qualifiers, and rhetorical relations.

between elements. To our knowledge ANVIL is the only coding tool that allows any annotation element to point to any other annotation element, independent of the tiers the elements reside in. This is realized with special attribute types where the coder can insert a number of links to other elements. In DiAML, links are used for dependence relations, i.e. for the fact that a feedback refers to a previous utterance of the interlocutor. In Fig. 7 the "checkQuestion" feedback (participant B) refers to the participant A's words (marked orange) in the topmost *utterance* track.

Comparison with other tools: There are various other multi-tiered coding tools, most notably ELAN (Wittenburg et al., 2006) and Exmaralda (Schmidt, 2004). ELAN is the tool that is most similar to ANVIL. It also allows user-defined coding schemes, offers various tier relationships and controlled vocabularies. It is widely used in linguistic communities. However, it lacks rich elements so that every attribute needs a separate tier and it does not allow logical links between elements. Exmaralda is a specialized tool for conversation analysis and is therefore text-based, i.e. it lacks the temporal precision that many multimodality researchers need.

5.3. Application to DiAML export

In ANVIL, dialogue acts are encoded in 10 tiers per speaker corresponding to the 10 DIT^{++} dimensions (see above): Task, Auto-feedback, Allo-feedback, Own-Communication Management, Partner Communication Management, Time Management, Turn Management, Dialogue Structuring, Social Obligations Management, and Contact Management.

In the current ANVIL-DiAML specification, there is one additional track to encode the utterances. This first track, called *utterance*, contains all words and vocal signals uttered by the speaker (see Fig.7). It is a *primary* track because all its elements are anchored in time. The 10 dimension tracks are *secondary* since they depend on the utterance track in the sense that their elements are made up of a 'span' of contiguous utterance elements (the so-called *span* track relationship). However, it may be the case that a feedback element does not refer to all the words contained in this span. Therefore, a logical link attribute lets the user specify which words/tokens exactly should be contained by this element (attribute is called "correlate verbal").

Dialogue act properties are encoded as *attributes* for each element in the dimension tracks. As mentioned above, this mechanism of rich elements containing 14 attributes avoids visual information overload on the coding board. The properties encoded as attributes are: addressee, communicative function, dimension (category of semantic content), communicative function qualifier (sentiment, conditionality, certainty), functional dependence relations, feedback dependence relations, rhetorical relations.

6. Conclusions

This paper shows how interoperable annotations of dialogue corpora, using ISO standard 24617-2 (or DTI^{++}) scheme and markup language DiAML, can conveniently be obtained using the ANVIL facility for producing XML-based output directly in the DiAML format. It is hoped that this will promote the creation of interoperable annotated corpora of spoken and multimodal dialogue.

7. References

Allen, J. and Core, M. (1997) Draft of DAMSL: Dialog Act Markup in Several Layers.

Bunt, H. (2009a) The DIT^{++} taxonomy for functional dialogue markup. In *Proceedings AAMAS 2009 Workshop*

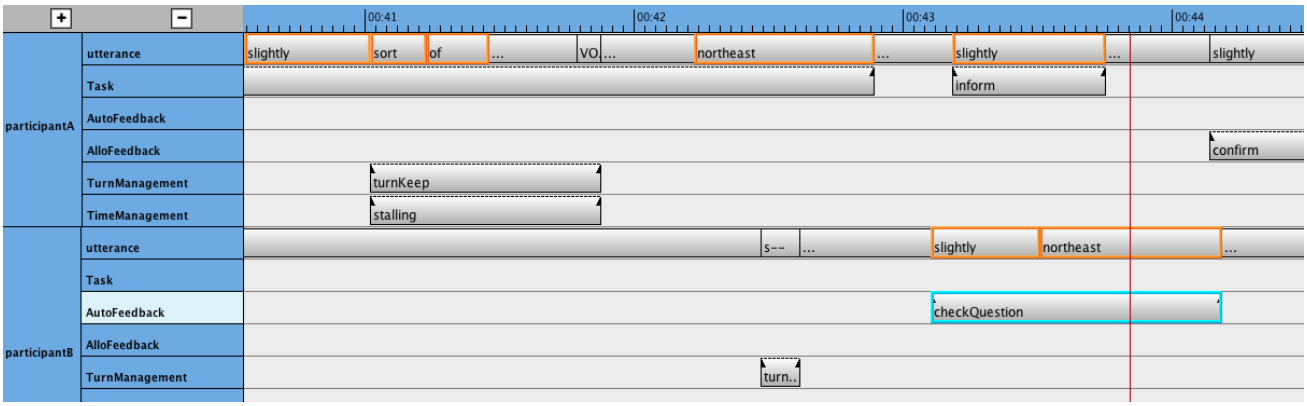

Figure 7: In ANVIL, cross-tier links are used to encode verbal correlates and dependencies. The selected annotation is marked blue, the orange frames indicate linked up elements. Note that some tracks are hidden for clarity.

'Towards a Standard Markup Language for Embodied Dialogue Acts', Budapest, pp. 13-24.

Bunt, H. (2010). Multifunctionality in Dialogue. *Computers, Speech, and Language*, 22, 224-245.

Bunt, H. (2011). The semantics of dialogue acts. In *Proceedings of the 9th International Conference on Computational Semantics IWCS 2011*, Oxford, pp. 1-14.

Bunt, H., J. Alexandersson, J. Carletta, J.-W. Chae, A. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and D. Traum (2010). Towards an ISO standard for dialogue act annotation. In *Proc. 7th Intern. Conference on Language Resources and Evaluation (LREC 2010)*, Malta, pp. 2548–2558.

Bunt, H., J. Alexandersson, J.-W. Chae, A. Fang, K. Hasida, V. Petukhova, A. Popescu-Belis, and D. Traum (2012). ISO 24617-2: A semantically-based standard for dialogue annotation. In *Proc. 8th Intern. Conf. on Language Resources and Evaluation (LREC 2012)*, Istanbul.

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

Geertzen, J., V. Petukhova and H. Bunt (2007) A multidimensional approach to utterance segmentation and dialogue act classification. In *Proceedings 8th SIGdial Workshop on Discourse and Dialogue*, pp. 140-149.

Ide, N. and L. Romary (2003). International Standard for a Linguistic Annotation Framework. *Natural Language Engineering* 10: 211–225.

Ide, N. and Suderman, K. (2007) GrAF: A graph-based format for linguistic annotations. In *Proceedings of the First Linguistic Annotation Workshop*, Prague, pp. 1–8.

Ide, N. and H. Bunt 2010 Anatomy of annotation schemes: Mappings to GrAF. In *Proceedings of the 4th Linguistic Annotation Workshop LAW IV*, pp. 115-124.

ISO 24617-2 (2011) Language resource management – Semantic annotation framework (SemAF), Part 2: Dialogue acts. ISO, Geneva, January 2011.

Kipp, M. (2001) ANVIL - a Generic Annotation Tool for Multimodal Dialogue. In *Proceedings of Eurospeech 2001*, Aalborg, pp. 1367–1370.

Kipp, M. (2008) Spatiotemporal coding in ANVIL. In *Proc. 7th Intern. Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, pp. 2242-2245.

Kipp, M. (2012) Multimedia Annotation, Querying and Analysis in ANVIL. In M. Maybury (ed.) *Multimedia Information Extraction*, IEEE Computer Society Press.

Kipp, M. (to appear) ANVIL: A Universal Video Research Tool. In: J. Durand, U. Gut, G. Kristofferson (eds.) *Handbook of Corpus Phonology*, Oxford U. Press.

Petukhova, V. (2011) Multidimensional Dialogue Modelling. PhD Thesis, Tilburg University.

Petukhova, V. and H. Bunt (2009a) Grounding by nodding. In *Proceedings 2009*, Poznán.

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

Petukhova, V. and Bunt, H. (2010b). Towards an integrated scheme for semantic annotation of multimodal dialogue data. In *Proccedings of the 7th international conference on language resources and evaluation, LREC 2010*, Malta, pp. 2556- 2563.

Petukhova, V. and H. Bunt (2011) Incremental dialogue act understanding. In *Proc. IWCS*, Oxford, pp. 235-244.

Petukhova, V. and H. Bunt (2012) The coding and annotation of multimodal dialogue acts. In *Proceedings 8th International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul.

Popescu-Belis, A. (2005) Dialogue Acts: One or More Dimensions? ISSCO Working Paper 62, ISSCO, Geneva.

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

Schmidt, T. (2004) Transcribing and Annotating Spoken Language with Exmaralda. In *Proceedings LREC-Workshop on XML based Richly Annotated Corpora*, Lisbon.

TEI (2007) Guidelines of Electronic Text Encoding and Interchange, Edition P5. Text Encoding Initiative, Charlottesville, Virginia.

Wittenburg, P., H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes (2006) ELAN: A Professional Framework for Multimodality Research. In *Proc. LREC 2006*, 1556-1559.