Plug-ins for content annotation of dialogue acts

Harry Bunt
Department of Cognitive Science and Artificial Intelligence
Tilburg University, Tilburg, The Netherlands
harry.bunt@uvt.nl

Abstract

Experience in using the concepts of the ISO 24617-2 standard for dialogue act annotation and the DIT++ dialogue act taxonomy has made it clear that a severe limitation for their on-line use in dialogue systems is the lack of semantic content. This paper explores a possibility to remedy this by introducing the concept of an 'annotation plug-in'. It is argued that this can help to make these schemes more directly useful for on-line use as well as to increase both the coverage and the precision of the annotations they support.

1 Introduction

Dialogue act taxonomies, such as those of DIT++ and the ISO 24617-2 annotation standard, have two prime use cases: annotation of dialogue corpora and on-line use in dialogue systems. The DIT++ taxonomy was originally developed as on the one hand an instrument in the analysis of dialogue structure and dialogue mechanisms[1], and on the other hand a basis for the design of dialogue management modules in interactive systems[2]. The theory underlying both schemes, Dynamic Interpretation Theory, defines a dialogue act much in the spirit of speech act theory as having as its most important components a communicative function and a semantic content (besides dependence relations to other dialogue acts and possibly certain qualifiers). As the DIT++ taxonomy evolved into the ISO 24617-2 standard for dialogue act annotation (also referred to as 'ISO-DiAML'), a strong focus came to lie on its use for annotating the functional aspects of dialogue utterance meanings, and thus on communicative functions and inter-dependences of dialogue acts at the expense of attention to their semantic content, since other ISO standards were developing for aspects of semantic content. This limits the on-line use of ISO-DiAML in dialogue systems, where the semantic content of utterances is equally important as their communicative function.

The dialogue acts defined in ISO-DiAML have in recent years been used in dialogue management systems, getting around its limitations by invoking a semantic parser or entity extraction engine, as provided in NLU tools like LUIS or RASA for chatbots,[3] to add semantic content in a way that is suitable for the task domain. This paper explores a more general approach in the form of 'three-level annotation plug-ins', annotation mini-schemes that can be added on to a host annotation scheme and provide semantic content information in various degrees of detail.

This paper is organized as follows. Section 2 very briefly provides some background and motivation for the rest of the paper. Section 3 summarises the definition of the DiAML markup language for dialogue act annotation, defined as part of the ISO-DiAML standard, including its tripartite syntactic and semantic

[1] See e.g. Geertzen and Bunt (2006), Keizer and Morante (2006), Bunt et al. (2007), Geertzen (2009), Petukhova and Bunt (2009a, b), Wlodarczak et al. (2010), Bunt (2011), Fang et al. (2011), Petukhova et al (2011), Bunt (2012), Petukhova et al. (2015), Fang et al. (2018), and related studies such as Besser and Alexandersson (2007), Piwek (1998), Traum and Hinkelman (1992), Traum (1994), Poesio and Traum (1997).

[2] E.g. in the TENDUM system, see Bunt et al. (1984); in the DENK system (Bunt et al. 1998), and in the PARADIME system, see Keizer et al. (2011)

[3] LUIS: https://docs.microsoft.com/en-us/azure/cognitive-services/luis/what-is-luis; RASA: https://www.rasa.com/docs/nlu/

definition. Section 4 introduces the device of three-level annotation plug-in, first for adding very simple, ad hoc annotations of semantic content, and second for adding more principled annotations based on ISO standards. Section 5 considers the use of plug-ins for other purposes, first, for allowing more accurate annotations of discourse relations in dialogue, and second for supporting the import of task-specific types of dialogue act. Section 6 concludes by indicating the perspectives for extending the idea of annotation plug-ins.

2 Background

ISO standards are examined every five years for the need of correction, extension, or other updates. Since ISO-DiAML was formally established late 2012, it has come up for consideration in late 2017.

Experiences in using the ISO-DiAML scheme in annotation projects and dialogue system development have led to the awareness of certain limitations and desirable improvements, discussed in Bunt et al. (2017) and Bunt et al. (2018). These limitations concern mainly (1) the lack of semantic content information; (2) the absence of domain-specific dialogue act types; (3) the somewhat sloppy annotation of rhetorical relations, and (4) the sometimes inaccurate annotation of certain dependence relations.

Of these limitations, the latter one will be addressed in a revision of the standard. The third limitation can only be overcome by somehow adding semantic content information to dialogue act annotations, and the first two limitations are inherent to the scope of ISO-DiAML, and can only be tackled by radically extending its scope, which is not really an option. In order to nonetheless deal with these limitations, we introduce in this paper the notion of a 'three-level annotation scheme plug-in', or 'plug-in' for short, which allows semantically richer and customised annotations, and enhances the on-line usability in dialogue systems.

3 DiAML

3.1 Annotation scheme architecture

The ISO-DiAML annotation scheme has been designed according to the ISO principles of semantic annotation (ISO standard 24617-6, 'SemAF Principles', see also Bunt (2015) and Pustejovsky et al. (2017). This means that the scheme has a three-part definition consisting of (1) an abstract syntax that specifies the possible *annotation structures* at a conceptual level as set-theoretical constructs, such as pairs and triples of concepts; (2) a semantics that specifies the meaning of the annotation structures defined by the abstract syntax; (3) a concrete syntax, that specifies a representation format for annotation structures (for example using XML).

Defining the semantics at the level of the abstract syntax puts the focus of an annotation standard at the conceptual level, as required by the ISO Linguistic Annotation Framework (ISO 24612, see also Ide and Romary, 2004) . rather than at the level of representation formats. Annotators (human or automatic) deal with concrete representations only, but they can rely on the existence of an underlying abstract syntax and its semantics.

3.2 Abstract syntax

The abstract syntax specifies a store of basic concepts, called the 'conceptual inventory'. The DiAML conceptual inventory consists of:

- a set of nine dimensions (Task, Auto-Feedback, Allo-Feedback, Turn Management, ...);
- a set of communicative functions;
- a set of qualifiers, partitioned into those for certainty, for conditionality, and for sentiment;
- a set of dialogue participants;
- a collection of primary data, segmented into markables.

Given a conceptual inventory, the abstract syntax specifies certain set-theoretical structures like pairs, triples, and more complex nested structures made up from the elements of the inventory. Two types of structure are distinguished: *entity structures* and *link structures*. An entity structure contains semantic information about a segment of primary data and is formally a pair ⟨m, s⟩ consisting of a markable, which refers to a segment of primary data, and certain semantic information. A link structure contains information about the way two or more segments of primary data are semantically related.

A dialogue act is characterised by eight components: (1) a sender S, (2) one or more addressees A, (3) zero or more other participants H, (4) a dimension D, (5) a communicative function F, (6) zero or more dependence relations to a set E of other dialogue acts, (7) zero or more qualifiers Q, and (8) a semantic content c, where the components H, T, and Q are not necessarily present. In DiAML a full-blown entity structure is a septet $\langle S, A, H, D, F, E, Q\rangle$ rather than an octet, since the semantic content is not annotated. For example, the entity structure for a task-related question, expressed through the markable m_1, and addressed by participant p_1 to participant p_2, is the nested quadruple in (1)

(1) $\langle m_1, \langle p_1, \{p_2\}, Task, Question\rangle\rangle$

A link structure in DiAML is a triple $\langle e, E, R\rangle$ consisiting of an entity structure e, a set of entity structures E, and a rhetorical relation R. For example, the rhetorical relation represented in (14) corresponds in the abstract syntax to the link structure $\langle da_2, da_3, Cause\rangle$

A full-blown annotation structure for a dialogue in DiAML is a set $\{e_1, ..., e_n, L_1, ..., L_k\}$ of entity structures structures $(e_1, ..., e_n)$ and link structures $(L_1, ..., L_k)$. For example, the annotation structure for a dialogue fragment consisting of a single question-answer pair, is the following nested structure, in which the entity structure for the question re-appears inside the one for the answer, indicating that the answer is an answer to that question (a ‘functional dependence’ relation):

(2) $\{\langle m_1, \langle p_1, \{p_2\}, Task, Question\rangle\rangle,$
$\langle m_2, \langle p_2, \{p_1\}, Task, Answer, \{\langle m_1, \langle p_1, \{p_2\}, Task, Question\rangle\rangle\}\rangle\rangle\}$

3.3 Concrete syntax

The conceptual structures defined by the DiAML abstract syntax can be represented in a variety of ways; the ISO 24617-2 standard specifies a pivot XML-based representation format (also referred to as ‘DiAML-XML’).

For the representation of entity structures an XML element <dialogueAct> is defined, with an attribute @xml:id whose value is a unique identifier; an attribute @target, whose value anchors the annotation in the source data, having a markable (or a sequence of markables) as its value; and furthermore the following attributes, corresponding to its conceptual components: @sender, @addressees, @other participants (optional), @dimension, @communicative function, @dependences (optional), and @qualifiers (optional). For example, the entity structure shown in (1) is represented as:

(3) <dialogueAct xml:id=”da1” target=”#m1” sender=”#p1” addressee=”#p2”
dimension=”task” communicativeFunction=”question”/>

For the representation of link structures the XML element <rhetoricalLink> is defined, with the attributes @dact, @rhetorelatum, and @relType. Their use is illustrated in example (14).

3.4 Semantics

The DiAML semantics consists of the specification of a recursive interpretation function I_{DA} that defines a functor which, applied to a semantic content, forms an information state update operation. The DiAML semantics is compositional in the sense that the interpretation of an annotation structure is obtained by combining the interpretations of its component entity structures and link structures; see Bunt ((2014) for details.

Formally, according to SemAF Principles (ISO 24617-6) an annotation scheme is defined by a triple formed by specifications of an abstract syntax (AS), a concrete syntax (CS), and a semantics (SEM), each of which is further structured:

(4) $A = \langle AS, CS, \text{SEM}\rangle = \langle\langle CI, AC\rangle, \langle VC, CC, eF\rangle, \langle M, I\rangle\rangle$

The abstract syntax specification consists of the conceptual inventory CI and the specification of conceptual structures AC; together, these define the class of well-formed annotation structures.

The concrete syntax specification CS contains a vocabulary VC, the specification CC of a class of syntactic structures, such as XML elements, and an encoding function (including a mapping from AC to CC). The components VC and CC together define a class of well-formed representations, and eF assigns such a representation to every well-formed annotation structure.

The semantic specification SEM is in general a pair $\langle M, I\rangle$, consisting of a model and an interpretation function. For the ISO-DiAML host annotation scheme, the semantics uses a context model (or 'information state') for M and an interpretation function defined in terms of context updates.

4 Content plug-ins

The semantic content of a dialogue act, expressed by a markable (functional segment) m_i, can of course be supplied by a semantic parser-interpreter that delivers the semantic content of m_i. This is what Keizer et al (2011) and Malchanau (2019) do for an application domain where the semantic content has a simple structure, that can be described by a list of attribute-value pairs. Here we explore an alternative possibility.

The various parts of the ISO Semantic Annotation Framework (SemAF, ISO 24617) other than ISO-DiAML are all concerned with the annotation of aspects of sentence meaning, and these annotations have a compositional semantics. A methodologically elegant option that presents itself would be to combine ISO-DiAML with some of these other annotations and use their semantics to obtain the semantic content of dialogue acts. This can be accomplished by defining 'plug-ins' for the DiAML annotation language, as described in the rest of this section.

4.1 A domain-specific content plug-in

To introduce the idea of a three-level annotation plug-in, we first consider the case of a simple domain-specific plug-in that could for example be useful in a journey planning domain where a task can be described by a few attribute-value pairs, specifying departure place, destination, travel date, etc. For example, the utterance *"I would like to leave around ten in the morning"* from a client PA1 could be semantically annotated as in (5b):

(5) a. PA1: "I would like to leave around ten in the morning" (= markable m1)

b. <avContent xml:id="av1" target="#m1" attribute="departureTime" value="10:00"/>

According to the ISO principles of semantic annotation, as laid down in ISO 24617-2 (see also Bunt, 2015), a semantic annotation should have an underlying abstract syntax and a semantics. Underlying the representation used in (5b) would be a conceptual inventory that lists the attributes and their possible values, and an entity structure in the form of a nested pair $\langle m, \langle A_i, v_{ij}\rangle\rangle$ made up of a markable, an attribute and a value (taken from the conceptual inventory). The semantics of such an entity structure could e.g. be a feature structure of the form $[A'_i : v'_{ij}]$ which, according to the ISO standard 24612 for feature structures can be viewed as the function $\lambda x.A'_i(x) = v'_{ij}$.

The syntax and semantics of such AV-entity structures define a very simple annotation language L_{AV}, the semantics of which is a defined by:

(6) $I_{AV}(\langle A_i, v_{ij}\rangle) = [I_{AV}(A_i) : I_{AV}(v_{ij})] = [A'_i : v'_{ij}]$

Attribute-value pairs $\langle A_i, v_{ij}\rangle$, their XML encoding as in (5b), and the specification of their semantics as in (6) in fact define a mini-annotation-scheme for semantic content annotation that can be combined with DiAML annotation – such a mini-scheme is what we call an annotation *plug-in*. A plug-in is thus formally characterised as a triple $PL_a = \langle AS_a, CS_a, \textsc{Sem}_a\rangle$. The formal specification of the three-level attribute-value content plug-in PL_{AV} for DiAML is as follows:

- AS_{AV}: the conceptual inventory lists attributes and their possible values; entity structures for semantic content of the form ⟨markable, ⟨attribute, value⟩⟩.
- CS_{AV}: vocabulary items for attributes and values; encoding of entity structures as in (5b).
- $\textsc{Sem}_{AV}$: I_{AV} as in (6).

To integrate plug-in annotations of semantic content with DiAML-annotations of the functional aspects of dialogue acts, there are two options.

- extend the entity structures describing a dialogue act from septets to octets, including their semantic content, and add a corresponding attribute @semContent to the XML encoding of in a <dialogueAct> element. This leads to representations of the following form:

  (7) <dialogueAct xml:id="da1"target="#m1" speaker="#s" addressee="#a"
      dimension="task" communicativeFunction="inform" semContent="#av1"/>
  <avContent xml:id="av1" target="#m1" attribute="departureTime" value="10:00"/>

- introduce a new link structure that relates a dialogue act to its semantic content, and add a corresponding <contentLink> element in the XML encoding. This leads to representations in the following form:

  (8) <dialogueAct xml:id="da1"target="#m1" speaker="#s" addressee="#a"
      dimension="task" communicativeFunction="inform"/>
  <avContent xml:id="av1" target="#m1" attribute="departureTime" value="10:00"/>
  <contentLink dAct="#da1" content="#av1"/>

According to the underlying theory, a dialogue act is formally characterized as an octet, including a semantic content (see Section 3.2 above), so the first option may seem most plausible. However, it requires the introduction in DiAML of an attribute like @semContent of which the possible values cannot be specified, since that depends on the plug-in that is used. The second option has several interesting advantages:

1. it makes it very clear that the dialogue annotation of a dialogue act does not require the specification of a semantic content: the use of the plug-in is optional.[4]

2. the use of an explicit link between the functional aspects of a dialogue act and its semantic content allows the specification of additional information attached to the link, such as (un-)certainty scores and alternatives, supporting the management of semantic ambiguity.

3. the annotation of dialogue acts in DiAML remains the same, therefore any formal operations and any software defined for DiAML annotations are still applicable.

We therefore choose the second option. The introduction of a content link structure shows that the combination of a host annotation scheme with a plug-in requires in general an *interface* that connects the two. The content link structure is neither part of DiAML nor of the plug-in, but is part of the interface between the two. An interface ${}^{p}Y_h$ between a host scheme h and a plug-in p can be formally specified in a similar way as the host scheme itself (see (4)) and the plug-in, with an abstract and concrete syntax

[4] See Bunt (2018) for a discussion of different forms of optionality in semantic annotations.

and a semantics, except that the interface does not require the introduction of new basic concepts or vocabulary items, but only of one or more link structures for relating host and plug annotations.

For the present plug-in, the abstract syntax of the interface ${}^{AV}Y_{DA}$ defines the content link structure as a pair $\langle \epsilon_a,\, \epsilon_c \rangle$ consisting of a dialogue act entity structure and a content entity structure; no relation between the two needs to be specified since no other relation between the two kinds of structure are envisaged. The concrete syntax of the interface defines the <contentLink> element and the encoding of pairs $\langle \epsilon_a,\, \epsilon_c \rangle$ using this element. As noted above (Section 3.4), the DiAML semantics specifies an interpretation function I_{DA} which defines a functor that, applied to a semantic content, forms an information state update operation. The semantic component of the interface specifies the interpretation of the new content link structure in a way that expresses exactly that:

(9) $I_{{}^{AV}Y_{DA}}(\langle \epsilon_a, \epsilon_c \rangle) = I_{DA}(\epsilon_a)(I_{AV}(\epsilon_c))$

Using DiAML with the PL_{AV} plug-in comes down to applying an extended annotation scheme of which the abstract syntax is formed by the DiAML abstract syntax, the PL_{AV} abstract syntax, and the content link stsructure of the interface; the concrete syntax likewise is that of DiAML extended with PL_{AV}-representations and the <contentLink> element; the semantics is that of DiAML extended with $I_{AV}(\epsilon_{AV})$ for content entity structures and with (9) for content link structures.

A caveat: the semantic part of the interface, as formed by (9), assumes that the interpretation function I_{DA} of the host language is applicable to the output of the plug-in interpretation function I_{AV}. The DiAML semantics makes use of elementary context update operators which are defined in a representation-neutral way, just stipulating for example that a given semantic content should be added to that part of the addressee's information state which contains information about the task that still has to be verified for consistency with other available information (the addressee's 'pending semantic context'). To apply this approach in a dialogue system, the elementary update operators must be instantiated for the representation formalism of the system's information state. The semantic content of dialogue acts has to be represented in a form that fits in with that formalism, and if necessary has to be converted to that. For content expressed in the form of feature structures, as is the case for I_{AV}, this will not be an obstacle. Existing DiAML implementations in dialogue systems, such as Keizer et al. (2011), Malchanau et al. (2017), and Malchanau (2019) use typed feature structures for information representation, making the implementation of (9) a straightforward matter.

It may be noted that the annotation of a semantic content formed by an attribute-value pair, by means of an XML-element <avContent>, can be viewed as just a conveniently compact abbreviated notation of an ISO/TEI-conformant XML-representation using the TEI vocabulary. For example, the representation used in (5b) is equivalent to the XML-representation in (10), which makes use of TEI-defined vocabulary items and of feature structures as specified in ISO standard 24612. The abbreviating representation used in (5), (7), and (8) could be automatically expanded into this representation.

```
(10) <u xml:id=”m1”>I would like to leave around ten in the morning</u>
       <annotationBlock type=”semanticContent”>
         <spanGrp type=”markable”>
           <span target=”#m1” ana=”#fs1”/ >
         </spanGrp>
       <fs type=”avContent” xml:id=”fs1”>
         <f name=”departureTime”>
           <string>10:00</string>
         </f>
       </fs>
       </annotationBlock>
```

4.2 Semantic roles

We next consider a more general content-plug, based on an ISO standard.

Semantic annotation standard ISO 24617-4 for semantic role labelling, a.k.a. ‘SemAF-SR’, marks up semantic information related to the question *Who did what to whom?*, assigning semantic roles to the participants in an event. For instance, the example sentence *“I would like to leave around ten in the morning”* would be analysed as mentioning two eventualities, a like-state and a travel-event, and would be annotated as follows, where “like.01” and “leave.01” correspond to verb senses in VerbNet:

(11) a. PA1: ”I would like to leave around ten in the morning”
Markables: m1=“I”, m2=“like”, m3=“leave”, m4=“ten in the morning”

b. <eventuality xml:id=”e1” target=”#m2” eventFrame=”like.01” eventualityType=”state”/>
<entity xml:id=”x1” target=”#m1” pred=”#pa1”/>
<srLink event=”#e1” participant=”#x1” semRole=”experiencer”/>
<eventuality xml:id=”e2” target=”#m3” eventFrame=”leave.01” eventualityType=”activity”/>
<srLink event=”#e1” participant=”#e2” semRole=”theme”/>
<srLink event=”#e2” participant=”#x1” semRole=”agent”/>
<entity xml:id=”x2” target=”#m4” pred=”10:00”/>
<srLink event=”#e2” participant=”#x2” semRole=”time”/>

SemAF-SR interprets such annotations as expressing the existence (or denied existence, in case of a negated clause) of certain states or events and participants in certain roles. For the example in (11) the semantics can be expressed by the following DRS (where pa1 is a constant referring to the speaker of the utterance in (11a)):

(12) [e1 e2 x1 x2 | like01(e1), leave01(e2), x1=pa1, x2=10:00, experiencer(e1,x1), theme(e1,e2), agent(e2,x1), time(e2,x2)]

A content plug-in for DIAML consists in this case of the abstract and concrete syntax of the SemAF-SR markup language and the semantic interpretation function (not spelled out in the ISO standard) which produces DRSs like those in (12). The abstract syntax has a conceptual inventory that lists semantic roles and verb senses by reference to VerbNet, defines entity structures for eventualities and their participants, and link structures for relating participants to eventualities in a certain role. The concrete syntax defines XML encodings of the annotation structures defined by the abstract syntax, as illustrated in (11).

When defining a content plug-in for information about semantic roles, the question arises whether *all* the information encoded in SemAF-SR annotations should be taken along in the plug-in. This issue concerns especially the reference to event frames for VerbNet verb senses. While this seems appropriate for the purposes of SemAF-SR, it would bring a level of precision to the interpretation of verbs and deverbal nouns which is not pursued for other content words; it may therefore be more appropriate to make this optional in a plug-in, allowing users to choose whether they want to plug in a conceptual inventory with that level of granularity or a less fine-grained one. ISO-TimeML (ISO 24617-1), the ISO standard for annotating time and events, should also be considered here. It uses a classification of event types that differs from that of SemAF-SR, and it includes other detailed information about events that is not considered in SemAF-SR. Again, it is not obvious how much of that information would seem appropriate to take along in a plug-in for DiAML. A closely related issue concerns the analysis of the semantic roles of temporal objects. SemAF-SR distinguishes three temporal roles: Duration, Initial-time, and Final-time, whereas ISO-TimeML has a much larger and more fine-grained set of temporal relations.

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

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

Concrete syntax: specifies names for the elements of the conceptual inventory, and defines XML elements for encoding the entity and link structures.

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

Content annotation structures according to this plug-in can be linked to dialogue acts using content link structures in the same way as for the AV-plug-in. This allows the example utterance (17a) to be annotated as follows:

(13) a. PA1: ”I would like to leave around ten in the morning”
Markables: m1=“I”, m2=“like”, m3=“leave”, m4=“ten in the morning”

b.
```
<dialogueAct xml:id=”da1” target=”#m1” speaker=”#pa1” addressee=”#a”
   dimension=”task” communicativeFunction=”inform”/>
<eventuality xml:id=”e1” target=”#m2” pred=”like”/>
<entity xml:id=”x1” target=”#m1” pred=”#pa1”/>
<srLink event=”#e1” participant=”#x1” semRole=”experiencer”/>
<eventuality xml:id=”e2” target=”#m3” eventFrame=”leave”/>
<srLink event=”#e1” participant=”#e2” semRole=”theme”/>
<srLink event=”#e2” participant=”#x1” semRole=”agent”/>
<entity xml:id=”x2” target=”#m4” pred=”#10:00”/>
<srLink event=”#e2” participant=”#x2” semRole=”time”/>
<contentLink dAct=”#da1” content=”#e1”/>
```

The interface for this plug-in can be the same as for the AV plug-in, with the semantic part like (9) except that the interpretation function I_{SR} is used instead of I_{AV}.

5 Other plug-ins

5.1 More fine-grained rhetorical relations

ISO 24617-2 supports the marking up of rhetorical relations between dialogue acts, but does not specify any particular set of relations to be used; it only specifies *how* a rhetorical relation may be marked up as relating two dialogue acts. ISO standard 24617-8 for annotating semantic relations in discourse, also known as DR-Core, was established in 2016 and defines a core set of rhetorical relations. Comparing the ISO-DiAML and DR-Core annotation schemes, two limitations have been noted of the way rhetorical relations can be marked up in DiAML. First, many rhetorical relations have two arguments that play different roles, for example, a Cause relation has one argument that plays the role of a reason and another that plays the role of a result. ISO-DiAML has a provision for indicating the existence of a causal relation between two dialogue acts, but not for indicating their roles, as illustrated in (14):

(14) A: Have you seen Pete today?
B: Pete didn’t come in. He has the flu.

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

By contrast, DR-Core annotations make argument roles explicit, as illustrated in the DR-Core annotation of B's two utterances in (15):

(15) A: Have you seen Pete today?
B: Pete didn't come in. He sent me a message saying that he has the flu.
<drArg xml:id="a1" target="#m1" type="dialogAct"/>
<dRel xml:id="r1" rel="cause"/>
<drArg xml:id="a2" target="#m2" type="event"/>
<drLink rel="#r1" arg1="#a1" arg1Role="result" arg2="#a2" arg2Role="reason"/>

Second, many rhetorical relations may occur either between between the semantic contents of two dialogue acts, or between the semantic content of one dialogue act and the performance of another. This phenomenon is known in the literature as the 'semantic-pragmatic' distinction, and is illustrated by the difference between (14) and (15). B's two utterances in (14) are causally related in the sense that the semantic content of the second dialogue act forms the reason why the content of the first dialogue act is true. In (15), by contrast, there is a 'pragmatic' causal relation, in the sense that the second utterance expresses why B performs the dialogue act of informing A that Pete is not in. This distinction is represented in DR-Core by indicating the types of the arguments, where "dialogAct" is one of the possible types, and the possible types of the semantic content of a dialogue act are the other. This distinction cannot be expressed in DiAML.

This can be remedied in the presence of a plug-in for semantic content annotation, in which case the necessary entity structures are already available, by adding a plug-in which provides a link structure, corresponding to the 'drLink' structure of DR-Core. for annotating the occurrence of a rhetorical relation. A simple plug-in PL_{DR} for rhetorical relations can then be defined as follows:

Abstract syntax:

- Conceptual inventory: the discourse relations defined in DR-Core;
- Link structures: those of DR-Core for rhetorical relations between entity structures as defined in DiAML or in the content plug-in.

Concrete syntax: XML names for the relations in the conceptual inventory and for their argument roles (like '@reason' and '@result'); encodings of DR-Core link structures.

Semantics: The DR-Core interpretation of discourse relations as binary predicates.

For the semantic interpretation of a rhetorical link between a dialogue act and the semantic content of another one, or between the semantic contents of two dialogue acts, the following semantic interface is needed:

(16) $I_{SRY_{DA}}(\langle\epsilon_1, \epsilon_2, \rho\rangle) = I_{DR}(\rho)(\epsilon'_1, \epsilon'_2)$, where ϵ'_i is the representation of ϵ_i in the DiAML context model if ϵ is a DiAML (dialogue act) entity structure; and if ϵ_i is a content entity structure then ϵ'_i is the representation in the context model of the semantic content plug-in.[5]

It may be noted that DR-Core is limited to annotating strictly semantic discourse relations with only a small set of 18 core relations. This is too limited for many applications, as has for example been noted when adding rhetorical relations in the DialogBank (Bunt et al., 2018). A more powerful plug-in could have a two-part conceptual inventory where one part corresponds to the DR-Core set of relations and another part to additional relations, needed for a given annotation task. Moreover, if one wants to annotate additional aspects of discourse relations such as their argument order in the discourse, as in the CCR theory of discourse (Sanders et al., 1992, 2018), then this could be taken care of by introducing relation qualifiers in the plug-in, in a similar way as dialogue act qualifiers are used in DiAML.

[5] The DIAML semantics assumes its context model to include a Dialogue History, a chronological representation of the dialogue acts that constitute a dialogue and of the relations between them (see Bunt, 2014). Clause (16) inserts such relations in the context model, taking semantic content into account.

5.2 Application-specific dialogue act types

The DIT++ dialogue act taxonomy and the ISO-DiAML annotation scheme were both designed to be domain- and task-independent, i.e. to be applicable in virtually every task domain. This is part of their strength, but it is also a limitation, since tasks other than the exchange of information may involve task-specific dialogue act types. For example, the chair person of a meeting may perform meeting-specific dialogue acts such as opening and closing the meeting, and suspending and resuming the meeting; in an interview in a human resource management context, acts such as appointing, promoting, hiring and sacking may occur. For applying ISO-DiAML or DIT++ to meetings and HRM interviews, such communicative functions should be added to the inventories of these annotation schemes.

For negotiation dialogues, Petukhova et al. defined 15 dialogue act types, such as Elicit-offer-value (*“How do you feel about...?”*), Offer-value (*“I could live with just a ban in public transportation”*), Counter-offer-value (*“I go for twenty five then if you’re so bad”*), Bargain-down (*“Okay, I can go for somewhat less restrictive’”*), Deal (*“That’s a deal!*), and Exit-deal (*“We have to re-discuss this.”*). For each of these a context-update semantics is defined[6] in similar terms as for the ISO-DiAML dialogue act types, making their addition conceptually relatively easy. A plug-in would just list the 15 negotiation-specific functions as the conceptual inventory; no additions to the DiAML entity structures or link structures are needed, and hence no additions to the concrete XML representations other than the names of the additional 15 functions. Semantic accommodation is very simple since the semantic component of the plug-in consists of the context-update specifications of the 15 additional functions, which do not interfere with the DiAML semantics and can simply be added.

5.3 Emotions

ISO 24617-2 has no provisions for expressing the emotional aspect that a dialogue act may have. The W3C recommendation EmotionML[7] was designed in part with the aim to serve as a plug-in for other annotation schemes. It characterizes emotions as complex entities, including an emotion category such as *irritated, excited*, or *amused*, an intensity (‘valence’), and , and various alternative ways of characterizing emotions, notably in terms of ‘action tendencies’, ‘appraisals’, and multiple ‘dimensions’. Since EmotionML is defined only at the level of concrete syntax, it cannot directly be used as a plug-in for semantic annotation, however, the concrete syntax could be used as the starting point for applying the CASCADES development method (see Pustejovsky et al., 2017) to build a full-fledged plug-in.

An EmotionML-based plug-in could use an <emoLink> element as the interace in the concrete syntax for relating the sender of a dialogue act to an EmotionML annotation representation, with corresponding abstract link structures and semantics in terms of updates of the senders emotional state. The following example illustrates the possibilities.

(17) a. PA1: Would you like to have a cup of coffee?
PA2: That would be wonderful!

b.
```
<dialogueAct xml:id=“da1” target=“#m1” speaker=“#pa1” addressee=“#pa2”
  dimension=“social” communicativeFunction=“offer”/>
<dialogueAct xml:id=“da2” target=‘#m2” speaker=‘#pa2” addressee=‘#pa1”
  dimension=“social” communicativeFunction=‘acceptOffer” funcDep=“#da1”/>
<event xml:id=“e1” target=“#m2” pred=“have-coffee”/>
<srLink event=“#e1” participant=“#pa2” semRole=‘agent”/>
<contentLink dAct=“#da1” content=“#e1”/>
<emotion xml:id=“em1” target=“#m2” category=“happy” valence=“0.8”/>
<emoLink dialogAct=“#da2” emotion=“#em1”/>
```

[6] See EU project Metalogue Deliverable 4.1, Annex 11.3.

[7] Available at http://www.w3.org/TR/2014/REC-emotionml-20140522/.

6 Conclusions and perspectives

In this paper we have introduced the concept of a plug-in for annotation schemes, and shown how this device opens possibilities for overcoming certain limitations of the ISO-DiAML and DIT++ annotation schemes. As a matter of principle, the ISO-DiAML standard does not deal with the semantic content of dialogue acts and includes only domain-independent communicative functions. Both these restrictions limit the possibility of using the standard on-line in dialogue systems without defining domain-related extensions. Plug-ins for semantic content and for domain-specific functions may form a well-defined and flexible way to overcome these limitations. Such plug-ins have been implemented in DIT++ release 5.2 (see `https://dit.uvt.nl`).

A plug-in for more precise annotation of rhetorical relations in dialogue was described, based on the ISO DR-Core standard, which presupposes a plug-in for semantic content annotation. The support of multiple plug-ins can clearly make the dialogue annotation scheme not only more powerful but also more accurate. Plug-ins derived from ISO-TimeML and ISO-Space (ISO 24617-7) could add temporal and locational information to the events introduced by the semantic roles plug-in.

Two crucial aspects of semantic content annotation that are still missing concern coreference and quantified predicate arguments. For each of these, the development of an ISO annotation standard has recently started (ISO WD 24617-9; ISO WD 24617-12), which opens the perspective of future additional plug-ins to further enhance the ISO-DiAML scheme for dialogue act annotation, analysis and interpretation, making it more useful both for annotation and for on-line use in intelligent interactive systems.

References

Besser, J. and J. Alexandersson (2007). A comprehensive disfluency model for multi-party interaction. In *Proceedings 8th SIGDIAL Workshop on Discourse and Dialogue*, Antwerp, pp. 182 – 189.

Bunt, H. (2011). Multifunctionality in dialogue. *Computer, Speech and Language 25*, 222–245.

Bunt, H. (2012). The semantics of feedback. In *Proceedings of SeineDial, 16th Workshop on the Semantics and Pragmatics of Dialogue*, Paris.

Bunt, H. (2014). A context-change semantics for dialogue acts. In H. Bunt, J. Bos, and S. Pulman (Eds.), *Computing Meaning, Vol. 4*, pp. 177–201. Berlin: Springer.

Bunt, H. (2015). On the principles of semantic annotation. In *Proceedings 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11)*, London, pp. 1–13.

Bunt, H., R. Ahn, R. Beun, T. Borghuis, and C. van Overveld (1998). Multimodal cooperation with the DENK system. In H. Bunt, R.-J. Beun, and T. Borghuis (Eds.), *Multimodal Human-Computer Communication*, pp. 39–67. Berlin: Springer.

Bunt, H., R. Beun, F. Dols, J. van der Linden, and G. Schwartzenberg (1984). The theoretical foundations of the TENDUM system. *IPO Annual Progress Report 19*, 105–113.

Bunt, H., E. Gilmartin, C. Pelachaud, V. Petukhova, L. Prévot, and M. Theune (2018). Downward compatible revision of the dialogue act annotation. In *Proceedings 14th Joint ISO - ACL Workshop on Interoperable Semantic Annotation (ISA-14)*, Santa Fé, New Mexico, pp. 21–34.

Bunt, H., R. Morante, and S. Keizer (2007). An empirically based computational model of grounding in dialogue. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, pp. 283–290.

Bunt, H., V. Petukhova, and A. Fang (2017). Revisiting the ISO Standard for Dialogue Act Annotation. In *Proceedings 13th Joint ISO - ACL Workshop on Interoperable Semantic Annotation (ISA-13)*, Montpellier, pp. 37–50.

Bunt, H., V. Petukhova, A. Malchanau, and K. Wijnhoven (2018). The DialogBank: Dialogues with interoperable annotations. *Language Resources and Evaluation 2018*. url: https://doi.org/10.1007/s10579-018-9436-9.

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

Fang, A., H. Bunt, J. Cao, and X. Liu (2011). Relating the semantics of dialogue acts to linguistic properties: A machine learning perspective through lexical cues. In *Proceedings of the 5th International Conference on Semantic Computing (IEEE-ICSC 2011*, Stanford University, Palo Alto, California, USA.

Fang, A., Y. Liu, J. Cao, and H. Bunt (2018). Chinese Multimodal Resources for Discourse Act Analysis. In C.-R. Huang, Z. Jing-Schmidt, and B. Mesterernst (Eds.), *The Routledge Handbook of Chinese Applied Linguistics*. Oxford, UK: Routledge.

Geertzen, J. (2009). *The automatic recognition and prediction of dialogue acts. Ph.D. Thesis*. Tilburg University.

Geertzen, J. and H. Bunt (2006). Measuring annotator agreement in a complex, hierarchical dialogue act schema. In *Proceedings SIGDIAL 2006*, Sydney.

Ide, N. and L. Romary (2004). International standard for a linguistic annotation framework. *Natural Language Engineering 10*, 221–225.

ISO (2012a). *ISO 24617-1: 2012, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 1: Time and events*. Geneva: International Organisation for Standardisation ISO.

ISO (2012b). *ISO 24617-2:2012, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 2: Dialogue acts*. Geneva: International Organisation for Standardisation ISO.

ISO (2014a). *ISO 24617-4: 2014, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 4: Semantic roles*. Geneva: International Organisation for Standardisation ISO.

ISO (2014b). *ISO 24617-7: 2014, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 7: Spatial information*. Geneva: International Organisation for Standardisation ISO.

ISO (2015). *ISO 24617-6:2015, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 6: Principles of semantic annotation*. Geneva: International Organisation for Standardisation ISO.

ISO (2016). *ISO 24617-8:2016, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 8: Semantic relations in discourse, Core annotation scheme (DR-Core)*. Geneva: International Organisation for Standardisation ISO.

ISO (2019a). *ISO 24617-12:2019, Language Resource Management: Semantic Annotation Framework (SemAF) - Part 12: Quantification*. Geneva.: International Standard. International Organisation for Standardisation ISO.

ISO (2019b). *ISO 24617-9:2019, Language Resource Management: Semantic Annotation Framework (SemAF) - Part 9: Reference Annotation Framework*. Geneva.: International Standard. International Organisation for Standardisation ISO.

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

Keizer, S. and R. Morante (2006). Dialogue acts as context operators constraining the generation of dialogical discourse. In C. Sidner, J. Harpur, and P. Kühnlein (Eds.), *Proceedings of the Workshop on Constraints in Discourse*, Maynooth, Ireland, pp. 117–124.

Lee, K., L. Burnard, L. Romary, S. Bauman, H. Bunt, E. de la Clergerie, T. Erjavec, T. Declerck, L. Clement, A. Roussanaly, and C. Roux (2004). Towards an International Standard for the Representation of Feature Structures. In *Proceedings 4th International Conference on Language Resources and Evaluation (LREC 2004)*, Paris. ELRA.

Malchanau, A. (2019). *Cognitive Architecture for Multimodal Multidimensional Dialogue Management*. Saarbrücken: PhD Thesis, University of Saarland.

Malchanau, A., V. Petukhova, and H. Bunt (2017). Virtual Debate Coach design: Assessing multimodal argumentation performance. In *Proceedings International Conference on Multimodal Interaction (ICMI 2017)*, Glasgow.

Petukhova, V. and H. Bunt (2009a). Grounding by nodding. In *Proceedings GESPIN, Conference on Gestures and Speech in Interaction*, Poznán.

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

Petukhova, V., A. Malchanau, and H. Bunt (2015). Modelling argumentation in parliamentary debates. In *Proc. of CMNA XV, 15th Workshop on Computational Models of Natural Argument*, Bartinoro, Italy.

Petukhova, V., A. Malchanau, and H. Bunt (2017). Computing negotiation update semantics in multi-issue bargaining dialogues. In *Proceedings 21st Workshop on the Semantics and Prgatics of Dialogue (SEMDIAL 2017 - Saardial)*, Saarbrücken.

Petukhova, V., L. Prévot, and H. Bunt (2011). Multi-level discourse relations between dialogue units. In *Proceedings 6th Joint ACL-ISO workshop on Interoperable Semantic Annotation (ISA-6)*, Oxford, pp. 18–28.

Piwek, P. (1998). *The logic of conversation*. PhD thesis, Tilburg University.

Poesio, M. and D. Traum (1997). Conversational actions and discourse situations. *Computational Intelligence 13(3)*, 309 – 347.

Pustejovsky, J., H. Bunt, and A. Zaenen (2017). Designing annotation schemes: From theory to model. In N. Ide and J. Pustejovsky (Eds.), *Handbook of Linguistic Annotation*, pp. 21–72. Berlin: Springer.

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

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

Traum, D. (1994). *A computational theory of grounding in natural language conversation.* PhD Thesis, University of Rochester.

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

Wlodarczak, M., H. Bunt, and V. Petukhova (2010). Entailed feedback: evidence from a ranking experiment. In *Proceedings 14th Workshop on the Semantics and Pragmatics of Dialogue (PozDial)*, Poznan, Poland, pp. 159–162.