Annotation Representations and the Construction of the DialogBank

Name: Kars Wijnhoven

ANR: 295543

Master's Thesis

Communication and Information Sciences

Specialisation: Human Aspects of Information Technology

Faculty of Humanities

Tilburg University, Tilburg

Supervisor: prof. dr. H.C. Bunt

Second Reader: dr. S. Wubben

August, 2016

Table of Contents

Abstract

The DialogBank is a new language resource consisting of dialogues with gold standard annotations according to the ISO 24617-2 standard. Two human-friendly tabular representation formats for ISO 24617-2 annotations, DiAML-TabSW and DiAML-MultiTab, have been introduced in the DialogBank as alternatives to the computer-friendly DiAML-XML format specified in the ISO standard. First, two problems in using the ISO standard that became apparent during the construction of the DialogBank are discussed. Solutions to these problems – which stem from theoretical and practical shortcomings of the ISO standard – are proposed, taking into account their effects on the standard's abstract syntax and representation formats. In addition, a conversion program is introduced that allows for meaning-preserving conversions between annotations in the three representation formats. The successful conversions form a prove of concept that the new formats are *ideal* (complete and unambiguous) and allow users of the DialogBank to choose the format in which they want to view the annotations. Finally, an experiment was conducted among Dutch university students (N = 47) comparing the three representation formats in the support they give in information retrieval tasks and in spotting annotation inaccuracies, and also in their subjective appreciation. Supportive evidence was found for the new representation formats as good alternatives to the DiAML-XML format. The DiAML-TabSW format was found to be an improvement over the DiAML-XML format in almost every respect.

Keywords: Annotation representations, ISO standard, Semantic annotation.

1. Introduction

In 2010 Apple acquired the app ‘SIRI’, a personal voice assistant, allowing users of Apple devices to verbally communicate with their device, for instance by giving it some instructions or by having a ‘casual’ conversation with SIRI (Lohr, 2012; www.apple.com/ios/siri). In recent years, many large companies have introduced such personal assistants, take for instance Microsoft’s ‘Cortana’, Samsung’s ‘S Voice’, and Google’s ‘Google Now’. These assistants are examples of dialogue systems: computational agents that engage in interactions with humans using human language in form of speech or text (Domain Adaptations for Dialog Agents/DADA, 2016). In the mid-nineties already, as part of the TRAINS project, researchers developed a system that could use English language to converse with and help humans in problem solving tasks, more specifically the system involved an interactive planning assistant that helped a user construct and monitor plans about a railroad system (Allen et al., 1995). Above mentioned dialogue systems and computer systems in general that are developed to interact with humans using natural language typically need example interactions to learn from (Allen et al., 1995; DADA, 2016; Petukhova et al., 2014).

This is one of the domains where language corpora and more specifically dialogue corpora come into play and prove their worth. These corpora contain large collections of dialogues (e.g. in textual, audio or video formats) between two or more dialogue participants. Note that corpora are also incredibly important and valuable research instruments, as much of the current linguistic research is based on such corpora. Several corpora, such as TRAINS (Allen et al., 1995), AMI (Carletta et al., 2005), AMIDA (Hain et al., 2007), HCRC Map Task (Anderson et al., 1991), ICSI MRDA (Shriberg, Dhillon, Bhagat, Ang, & Carvey., 2004), VerbMobil (Alexandersson et al., 1997), and SWBD-DAMSL (Jurafsky, Shriberg, & Biasca, 1997) have been used for analysing and modelling human behaviour and designing human-computer natural language based dialogue systems (Petukhova, Malchanau, & Bunt., 2014). In the TRAINS project, for instance, the system's behaviour and decision making was largely based on knowledge acquired from analysing a corpus that included human-human dialogues, i.e. dialogues between humans (Allen et al., 1995). More recently, researchers within the DBOX project have been developing interactive games based on natural language human-computer dialogues, having modelled the computer's behaviour based on the analysis of a collected corpus of human-human dialogues (Petukhova et al., 2014).

Analyses such as in the TRAINS and DBOX projects often require dialogues to be annotated according to an annotation scheme that specifies the concepts to be used and how to use them. In a general sense, annotation may be defined as the adding of notes to parts of a text or other types of data (Lee & Romary, 2010). For instance, one can attach data in the form of names, attributes and comments to a particular document, word or other type of data (e.g. "this document was created at 03/05/2010", "this word is a verb", or simply highlighting important sentences). There are various ways in which words, sentences, texts or dialogues can be annotated. The International Organisation for Standardisation (ISO) has recently established standards that cover various ways of annotating language data. Relevant to the current study, semantic annotation (ISO 24617) includes the marking up of units of spoken, written or multimodal language, such as texts, speech transcripts, pictures, or even recordings of multimodal or nonverbal communicative behaviour, with semantic information. In other words, semantic annotation enriches primary data with information about their meaning (Bunt, 2015).

Dialogues are no stranger to being subject of semantic annotation studies, schemes and other initiatives, and can be defined as a spoken conversational exchange (possibly including non-verbal behaviour) between two or more participants. When there is some sort of interactivity between two or more participants the annotation of semantic information involves the interpretation of how the speaker wants to influence the interactive situation (Bunt et al., 2012). In dialogues the meaning of communicative behaviour is often captured by annotating the dialogue according to a 'dialogue act annotation scheme'. Dialogue act annotation is defined as the activity of marking up stretches of dialogue with information about the dialogue acts which it contains and, as will be discussed in following sections, usually focuses on marking up the communicative functions of utterances (Bunt et al., 2012). For now, think of a dialogue act as a concept covering the intention of a sentence (Reithinger & Klesen, 1997). Dialogue acts play an important role in studies of dialogue, more specifically in the interpretation of the behaviour of dialogue participants and ultimately in the design of spoken dialogue systems (Bunt, 2011). Over time, corpora, projects and studies have defined a variety of annotation schemes for the annotation of dialogues (Bunt et al., 2010). This variety has made it difficult to interpret the annotations across different platforms and frameworks, negatively affecting their interoperability. Developing corpora and other language resources that are interoperable depends, among other things, on the application of common annotation and representation schemes (Bunt, Kipp, & Petukhova, 2012).

To facilitate the interoperability of corpora and other language resources, researchers from around the world in fields such as semantics and computational linguistics have been collaborating with the International Organisation for Standardisation and specifically its branch TC 37/SC 4 aiming to establish an international standard for semantic annotation. Working Group 2, which is part of the above branch and is concerned with semantic annotation, has created the Semantic Annotation Framework (SemAF), a framework that has incorporated various international standards that cover a variety of semantic phenomena, including e.g. ISO 24617-1 on *time and events* (ISO, 2012a), ISO 24617-4 on *semantic roles* (ISO, 2014a), ISO 24617-7 on *spatial information* (ISO, 2014b), and ISO 24617-8 on *semantic relations in discourse* (ISO, 2016). The current thesis covers the annotation standard ISO 24617-2 (ISO, 2012b), including the semantic phenomenon known as a *dialogue act*.

Following the rules of the 'Linguistic Annotation Framework' (LAF) – developed by the ISO branch TC37/SC4 to provide an architecture for representing linguistic annotations of language data that can serve the needs of all the annotation activities in the field of computational linguistics and offer full interoperability among annotation formats (Ide & Suderman, 2014; ISO 24612) – an important distinction that is relevant for the annotation standards in SemAF is that between 'annotation' and 'representation' (Ide & Romary, 2005). Annotation here refers to the linguistic information that is added to segments of data, independent of the format in which the information is presented. The format in which the annotation is presented (e.g. using XML-expressions), independent of its content, is referred to as representation (Bunt et al., 2012). Following this distinction, Bunt (2013) has developed a methodology ('CASCADES'; **C**onceptual analysis, **A**bstract syntax, **S**emantics and **C**oncrete syntax for **A**nnotation **DES**ign) for developing semantic annotation languages with, among other things, an 'abstract syntax' of annotations and a 'concrete syntax' of representations. These are two important components that underlie the annotation languages within SemAF. First, an abstract syntax defines a 'conceptual inventory' including a specification of particular collections of concepts, and possible relations between the concepts that an annotation may be composed of, also known as 'annotation structures'. For now think of such concepts as for instance a sender, an addressee, and the sender's intention of a particular communicative contribution to a dialogue. Secondly, a concrete syntax defines a representation format for representing such annotation structures. Most standards within SemAF specify as a concrete syntax an XML-based format with predefined XML-elements, attributes and values to store and represent dialogue information. As will be shown in following sections, DiAML ('Dialogue Act Markup Language'), the language of the ISO 24617-2 annotation scheme, also defines an abstract syntax with a conceptual inventory and a specification of its annotation structures, and as a concrete syntax an XML-based representation format called DiAML-XML.

Following the CASCADES methodology, annotation languages in SemAF specify in addition to an abstract and concrete syntax also a semantics, which is defined for an abstract syntax and is shared by all its representation formats. As Bunt (2013) notes, the interpretation of XML-expressions, which is the basis of most annotation representation formats, is not fixed and is left to the human- or machine-user. However, semantic annotations are meant to capture meaning in text and since annotation representations in XML do not have a well-defined meaning there is no reason to believe that XML captures meaning better than the text itself (Bunt, 2013). Therefore, Bunt and Romary (2004) have introduced the requirement of 'semantic adequacy' for semantic annotation, meaning that such languages require a well-defined semantics. The semantics specifies what annotation structures in the abstract syntax are intended to mean (Bunt, 2013). Defining the semantics for semantic annotation languages in this way is beneficial for improving the interoperability of semantic annotations, as any format that represents annotation structures shares these same semantics (Bunt, 2013).

As will be discussed in more detail in following sections, semantic annotation languages in SemAF, such as DiAML, are not tied to one compulsory, predefined representation format. On the contrary, as long as a concrete syntax is *complete* and *unambiguous* relative to the abstract syntax it may be used as a representation format to represent annotations (Bunt et al., 2012). Complete in the sense that the concrete syntax defines a representation for every structure defined by the abstract syntax. Unambiguous in the sense that every expression defined by the concrete syntax represents one and only one structure defined by the abstract syntax. A representation format that is both complete and unambiguous is called an *ideal* representation format (Bunt et al., 2012). As all ideal representation formats share the same underlying semantics, any ideal representation format can be converted through a meaning-preserving mapping (i.e. conversion) to any other ideal representation format (Bunt, 2013). A fictional and highly simplified example of an abstract syntax could include an annotation structure consisting of multiple 'participant structures' of the form <P= S, A, s>. These participant structures (P), which are created for each sentence (s), are defined in terms of a sender (S) and an addressee (A). Subsequently, a concrete syntax specifies a representation format which displays these structures defined by the abstract syntax with for instance XML-expressions as in (1) or in tabular format as in Table 1. In both examples the *ideal* representation formats cover the following dialogue fragment; John: “Hello Mary”, Mary: “Hello John”, John: “How are you?”, Mary: “I’m fine, how are you?”.

(1) <participants S=’John’ A=’Mary’ sentence=’Hello Mary’/>
<participants S=’Mary’ A=’John’ sentence=’Hello John’/>
<participants S=’John’ A=’Mary’ sentence=’How are you’/>
<participants S=’Mary’ A=’John’ sentence=’I’m fine how are you />

Table 1

Tabular Example of an Ideal Representation Format

Sentence	Sender	Addressee
Hello Mary	John	Mary
Hello John	Mary	John
How are you	John	Mary
I’m fine how are you	Mary	John

Recently, the DialogBank corpus has been established at Tilburg University (Bunt, Petukhova, Malchanau, Fang, & Wijnhoven., 2016; henceforth referred to as “DB16”). This new language resource (currently in beta phase, see https://dialogbank.uvt.nl) contains a collection of dialogues annotated according to the rules and guidelines of the above mentioned international standard ISO 24617-2 for dialogue act annotation. The main motivation for the establishment of the DialogBank is the perceived need for semantically annotated resources which are interoperable, i.e. resources whose annotations have been made according to international standards (DB16). Most of the dialogues in the DialogBank have been taken from existing corpora, often already annotated according to some other annotation scheme, such as the SWBD-DAMSL (Jurafsky et al., 1997) and DIT (Bunt, 1994; 1999) schemes. All dialogues present in the DialogBank have been updated according to the ISO 24617-2 annotation scheme, and to allow for comparisons between ISO 24617-2 and other schemes the dialogues in the DialogBank often also have their original annotations.

First, two problems that were encountered during the construction of the DialogBank are examined and discussed in the current thesis. These problems stem from particular theoretical and practical shortcomings of the ISO 24617-2 scheme and are related to the annotation of so-called feedback dependence relations and rhetorical relations. Finding ways to deal with them have been a major driving force behind this thesis. In Section 3.1 these problems are explained in detail. In Section 5.1 potential solutions are proposed to both of these problems, aiming to solve them for revised versions of the ISO 24617-2 scheme. The first research question is the following:

RQ1: What theoretical and practical shortcomings of the ISO 24617-2 standard have been discovered during the construction of the DialogBank, and how can these be solved?

ISO 24617-2 annotations are commonly represented using the DiAML annotation language, which specifies an XML-based reference format called ‘DiAML-XML’ for its representation (Bunt et al., 2012). The DialogBank has been established to provide users with annotations according to the ISO 24617-2 annotation scheme, therefore the plan initially was to add annotations in the scheme’s DiAML-XML format to the DialogBank. Therefore, the plan initially was to add annotations in this format to the DialogBank. First, a number of dialogues from the HCRC Map Task corpus (Andersson et al., 1991) were considered for updating according to ISO 24617-2 and, ultimately, implementation in the DialogBank. These dialogues were already annotated and represented in a particular XML-based representation format (not

DiAML-XML). Conveniently, this format allowed for their processing in an annotation tool called 'Anvil' (Kipp, 2001). The existing annotations, which were already for a large part annotated according to the ISO 24617-2 scheme, were updated manually with this Anvil tool into annotations with ISO 24617-2 compliant names and concepts. Finally, a special Anvil facility was used to automatically export the updated annotations into the DiAML-XML format. At the time of writing, the DialogBank contains six ISO-24617-2 compliant Map Task dialogues in DiAML-XML format.

However, annotations from other corpora were not available in the same XML-format as the Map Task dialogues. As will be shown in following sections, these annotations were often represented in some tabular format with rows and columns. Unfortunately, the Anvil annotation tool does not support processing of such formats, making the annotations unsuitable for use in Anvil and automatic exporting into the DiAML-XML format. Moreover, Anvil is especially useful for making new annotations from the start, and less useful for, relevant to the DialogBank, the updating of existing annotations. Therefore, some other way needed to be found to turn potential DialogBank annotations into fully ISO 24617-2 compliant annotations. During this process it became clear rather quickly that manually updating annotations into the DiAML-XML format would be difficult, inconvenient, and most of all extremely time-consuming. XML ('Extensible Markup Language') is a textual markup language and while it is designed to be human-readable (some XML expertise is needed however) and computer-readable, it is primarily a useful language for computers to express and exchange data. The DiAML-XML representation format serves its purpose as a format to store (semantic) dialogue information. However, a DiAML-XML file quickly tends to become extremely lengthy, mainly due to its extensive and repetitive use of XML elements and XML attributes. As a result, the format often leaves much to be desired in terms of readability for human users. So, with Anvil out of the picture, and taking into account the potential time needed to manually generate ISO

24617-2 compliant DiAML-XML annotations as well as doubts about the readability of the DiAML-XML representation format, new ways to update and represent potential ISO 24617-2 annotations for the DialogBank were necessary.

As mentioned, DiAML allows for new representation formats to be designed, as long as they are ideal representation formats. Most of the potential DialogBank annotations, for instance from the TRAINS corpus and Switchboard corpus, were already represented in some tabular representation format according to an annotation scheme. Updating the annotations to ISO 24617-2 compliant annotations in their respective tabular formats appeared to be significantly less difficult and time-consuming than manually updating them into the DiAML-XML format. The structure of the tabular formats was adjusted in such a way that fully compliant ISO 24617-2 annotations could now be represented in largely similar tabular formats. More precisely, a representation format specifically for annotations of Switchboard dialogues, and a representation format for other dialogues were created. These formats were later called DiAML-TabSW and DiAML-MultiTab, respectively. Two formats were created (not just one) because the tabular structure of the Switchboard dialogue annotations differed considerably from annotated dialogues of other corpora, such as from the TRAINS corpus and the Dutch DIAMOND corpus (Geertzen et al., 2004).

Because of the design and introduction of these new formats, annotations in the DialogBank are now represented in one or more of the DiAML-XML, DiAML-TabSW, and DiAML-MultiTab annotation representation formats. As will be shown in following sections the three formats differ in the way they display information. However, they essentially contain the same information, as they use the same concepts and relations that are defined in the underlying abstract syntax of the ISO 24617-2 scheme. Secondly, in this thesis it is examined if the two new representation formats, similar to DiAML-XML, indeed are ideal representation formats. If so, the three formats should be interconvertible, meaning that an annotation represented in one of the formats can be converted into one of the other representation formats without losing any information. Therefore, the second research question is the following:

RQ2: To what extent are the new representation formats, DiAML-MultiTab and DiAML-TabSW, *ideal* representation formats?

This will be examined in two ways. First, it will be shown, theoretically, that DiAML structures can be represented in the DiAML-MultiTab and DiAML-TabSW formats, and vice versa. Secondly, by creating a program that allows for the automatic conversion between the DiAML-XML, DiAML-MultiTab, and DiAML-TabSW formats - via the underlying abstract syntax - the aim is to prove, in practice, that the new formats are indeed ideal representation formats for ISO 24617-2 annotations. Also, successful conversions would support the CASCADES methodology introduced by Bunt (2013) for the design of semantic annotation languages with an abstract syntax, a semantics, and concrete syntax(es).

As mentioned, manually producing annotations in the DiAML-XML format is a relatively difficult and time-consuming activity. The new tabular formats provide a way to make ISO 24617-2 compliant annotations without requiring this representation format. As a result, the new formats eased and accelerated the process of making ISO 24617-2 compliant annotations for the DialogBank. Moreover, the new formats appear to improve the ability to satisfy retrieval needs such as finding the sender of a particular dialogue segment or a segments' precise composition in terms of primary data (e.g. words). Furthermore, due to the length and nature of the DiAML-XML format, annotation inaccuracies such as missing or incorrect relations were at times overlooked. As the new formats are, arguably, clearer than DiAML-XML it appeared easier to spot such annotation inaccuracies and subsequently correct them. Also, the tabular structure of the new formats appears to decrease the time needed to navigate through annotations, and subsequently spot inaccuracies and satisfy particular needs faster. However, note that the above is not supported by scientific evidence yet. It is unclear if the new formats are viable alternatives to the DiAML-XML format, and how the new formats are advantageous for users of annotations represented in these formats compared to the DiAML-XML format. Therefore, the third and final research question is the following:

RQ3: How are the new representation formats an improvement over the DiAML-XML representation format for users of these formats?

In the current thesis an answer to this research question is provided by comparing the three formats in an experimental setting. While in Section 4.3 the method for the experiment is discussed, the results of this experiment are presented in Section 5.3. Five hypotheses that are related to this experiment are defined:

H1: More information retrieval needs are satisfied in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format.

H2: More annotation inaccuracies are uncovered in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format.

H3: Information retrieval needs are satisfied quicker in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format.

H4: Annotation inaccuracies are uncovered quicker in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format.

H5: Users are more positive towards the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format.

2. Theoretical Background

2.1 ISO 24617-2

2.1.1 A Dialogue Act Annotation Scheme.

The following sections cover the ISO 24617-2 dialogue act annotation scheme, discussing, among other things, the notions of 'communicative function' and 'dimension, and the scheme's core concepts based on the scheme's metamodel. The ISO 24617-2 scheme is a relatively new dialogue act annotation scheme and is acknowledged by ISO as the international standard for dialogue act annotation. The scheme builds on previously designed dialogue act annotation schemes such as DAMSL (Allen & Core, 1997), MRDA (Shriberg, Dhilon, Bhagat, Ang, & Carvey, 2004), Verbmobil (Alexandersson, Reithinger, & Maier, 1997), DIT, and DIT++ (Bunt, 2009) (DB16).

Bunt et al. (2012) identified three specific reasons that underlie the establishment of the ISO 24617-2 scheme as an international standard for dialogue act annotation. First, ISO 24617-2 has been developed taking into account the need for an application-independent dialogue act annotation scheme that is both empirically and theoretically well founded. Secondly, they identified a perceived need for a dialogue act annotation scheme that can adequately deal with typed, spoken and multimodal dialogue. Finally, they identified a perceived need for a dialogue act annotation scheme that can be effectively used both by human annotators and automatic annotation methods. The main motivation for the design of annotation standards is to promote and facilitate the interoperability of annotated corpora (DB16). This interoperability of annotations largely depends on the availability of interchangeable representation formats (Bunt, Kipp, & Petukhova, 2012).

But what exactly are dialogue acts? The term dialogue act is often used – rather loosely – in the sense of 'speech act used in dialogue' (Bunt et al., 2012). However, in the ISO standard a more specific and semantic view on dialogue acts is taken. According to Petukhova, Prévot and Bunt (2011), participants in dialogue produce utterances in order to provoke change in their addressees. They view dialogue utterances as actions with intended state-changing effects on 'information-states'. An information-state is defined as the totality of a dialogue participant's beliefs, assumptions, expectations, goals, preferences, hopes and other attitudes that may influence the participant's interpretation and generation of communicative behaviour (ISO 24617). This view is known as the 'information-state update' or 'context-change' approach to the analysis of dialogue (Traum & Larsson, 2003; Bunt, 2000). Take the following example where person A utters the sentence "Do you know what time it is?" (Bunt, 2009). Person B can perceive or interpret this utterance in more than one way. Taking into account the information-state view, when B perceives it as a question about the time, then B's information state now includes, among other things, the information that A does not know, but would like to know, the current time. If, however, B perceives it as an accusation of being late, then B's information state now includes, among other things, the information that A does know the current time. These information-state changing actions are referred to as dialogue acts. They are a common concept in many dialogue annotation schemes, such as DAMSL, DIT++, and ISO 24617-2. Dialogue acts have two main components: (1) a communicative function and (2) a semantic content (or 'dimension'). Communicative functions correspond to the *type of action* that is performed, such as an offer, feedback act, or rejection, and specify how an addressee is intended to update his information state. Dialogue act annotation often involves the assignment of such communicative functions to stretches of communicative behaviour in dialogues (Bunt et al., 2012). Taking into account the two above examples, A wants to know the time and A accuses B of being late, different communicative functions are in play that specify how, when B understands the dialogue act, B should update his or her information state. The communicative functions are 'question' and 'accusation' respectively, provided that B has interpreted A's intention correctly.

A more complete description of a particular functional stretch of communicative behaviour also provides information about the *type of semantic content* (Bunt et al., 2012). The DAMSL annotation scheme specifies three dimensions (Task, Task Management, and Communication) indicating whether the semantic content of the dialogue act is concerned with performing the underlying task of the dialogue, with discussing how to perform this task, or with the communication (Bunt et al., 2012). The ISO 24617-2 standard defines more fine-grained dimensions by dividing the DAMSL communication dimension into more subclasses, such as a class with information about the processing of something that was said before ('Auto-feedback' and 'Allo-feedback' dimensions), information about the allocation of turns ('Turn Management' dimension), and information about the use of time in a dialogue ('Time Management' dimension). The communicative functions and dimensions that are included in the ISO 24617-2 scheme are further explained and exemplified in Section 2.1.

2.1.2 Dialogue Segmentation.

To assign meaning in terms of dialogue acts to stretches of communicative behaviour, there needs to be some way of identifying meaningful stretches in a dialogue (Bunt, 2009). The process of identifying such stretches is called dialogue segmentation and is generally part of a larger three-step-plan for dialogue act annotation (Petukhova, Prévot & Bunt, 2011). The annotation process usually starts with the collection of the primary data. As mentioned, this data can be in various forms such as audio recordings containing speech, video recordings containing speech and non-verbal behaviour (including e.g. head movements and in-breaths), texts and speech transcripts. Next, it is common practice to divide the primary data into smaller pieces of data called dialogue segments or units. This phase is also known as the segmentation phase. Finally, the third phase involves the marking up of those data segments with semantic information, the actual annotation phase. Over time, annotation schemes have used various notions of dialogue segments. Petukhova et al. (2011) mention, among other things, the use of ‘turn units’ as dialogue segments, defining turn units as stretches of speech produced by one speaker, bounded by periods of silence by that speaker. However, they argue that turns can be quite lengthy and complicated and for most purposes are not flexible enough to consider as semantic units (Petukhova et al., 2011). As introduced in the DIT++ annotation scheme (Bunt, 2007; Bunt, 2009), and incorporated in the ISO 24617-2 scheme, semantic information in terms of dialogue acts can be more accurately assigned to so-called ‘functional segments’. These functional segments are defined as minimal stretches of (verbal and/or nonverbal) behaviour which have one or more communicative functions (Bunt et al., 2012). They allow for more flexibility and differ from regular turns as they may overlap, may be discontinuous, and may include parts of more than one turn (Petukhova et al., 2009; Bunt et al, 2012). The condition of being minimal ensures that a functional segment does not include parts that do contribute to the expression of a communicative function that identifies the segment (ISO 24617). Take the following example (Bunt, 2009):

(2) 1: C: What time is the first train to the airport on Sunday?
2: I: The first train to the airport on Sunday is at… let me see… 5.32.

While C’s turn in (2) contains just one communicative function (question) and does not cause any segmentation problems, I’s turn is not suitable as dialogue segment. According to the ISO 24617-2 scheme, the latter turn unit consists of three functional segments. First, the discontinuous dialogue segment (“The first train to the airport on Sunday is at 5.32”), which carries an answer function. Secondly, the embedded dialogue segment (“The first train to the airport on Sunday”), which carries a positive feedback function, indicating that I has understood C’s question. Finally, the dialogue segment (“let me see”), which carries a stalling (for time) function. The first functional segment illustrates the possible discontinuity of functional segments, while the first and second segments together illustrate overlapping functional segments. In (3) it is illustrated that a dialogue act and its corresponding functional segment may include parts of multiple turns (ISO 24617). B's answer to A's question consists of a list of items that B communicates one by one (ISO 24617).

(3) A: Could you tell me what departure times there are for flights to Frankfurt on Saturday?
B: Certainly. There's a flight in the morning leaving at 08:15.
A: yes,
B: and a KLM at 08:50,
A: yes,
B: and a flight at 10:30,
A: yes,

2.1.3 Multifunctionality and multidimensionality.

One of the main concepts of the ISO 24617-2 scheme is the communicative function of a dialogue act, which has also been incorporated in older dialogue act annotation schemes such as DIT++ and DAMSL. The communicative function specifies what the speaker is trying to achieve (Bunt, 2009), or as mentioned before, how the behaviour changes a person's information state when that person understands the behaviour. Communicative functions as specified by ISO 24617-2 include, among others, 'inform' (e.g. "The train to Tilburg leaves at 14.00"), 'instruct' (e.g. "Turn left at the crossroads"), 'propositional question' (e.g. "Does the match start at three o'clock?"), and 'Auto-feedback' (e.g. "Uh-huh", or nonverbally: nodding).

Another important concept of the scheme is the 'dimension' of a dialogue act. A dimension refers to the class of dialogue acts that are concerned with some aspect of communication, and corresponds to a type of semantic content (Bunt et al., 2012). Communication in general includes various distinct dimensions. Taking part in a dialogue involves more than just pursuing a particular goal, task or activity. A dialogue act could, besides advancing the activity that motivates the dialogue ('task' dimension), include: the allocation of the speaker role ('turn management' dimension), dealing with some social obligation such as greeting and apologizing ('social obligations management' dimension), or managing the use of time ('time management' dimension). Table 2 displays all dimensions that are included in the ISO 24617-2 scheme.

Table 2
ISO 24617-2 Dimensions and their Definitions.

Dimension	Definition
Task	Category of dialogue acts whose performance contributes to pursuing the task or activity that motivates the dialogue.
Auto-feedback	Category of dialogue acts where the sender discusses or reports on his processing of previous dialogue contributions.
Allo-feedback	Category of dialogue acts where the sender discusses the addressee's processing of previous dialogue contributions.
Turn management	Category of dialogue acts whose performance is meant to regulate the allocation of the speaker role.
Time management	Category of dialogue acts which concern the allocation of time to the participant occupying the speaker role.
Discourse structuring	Category of dialogue acts which explicitly structure the interaction.
Own communication management	Category of dialogue acts where the speaker edits his own speech within the current turn.
Partner communication management	Category of dialogue acts which are performed by a dialogue participant who does not have the speaker role, and edits the speech of the participant who does occupy that role.
Social obligations management	Category of dialogue acts performed for taking care of social obligations such as greeting, thanking, and apologizing.

Notes. Source: (ISO 24617), see also http://semantic-annotation.uvt.nl/DiAML-Data-Categories.pdf.

Communicative functions can be divided into two main groups, known as *dimension-specific* functions and *general-purpose* functions. Dimension-specific functions have the characteristic that they can only occur within one particular dimension. Examples of dimension-specific functions are 'stalling' (e.g. "ehm"), which can only occur within the 'time management' dimension, and 'turn accept' and 'turn take' which are specific for the 'turn management' dimension. General-purpose functions on the other hand may occur in more than one dimension. Take for example the function 'inform' which can occur, for instance, in the 'task' dimension (e.g. "It's now noon when you arrive in Avon", where the dialogue is about transporting some goods) and the 'discourse structuring' dimension ("I have something else to say", where the speaker notifies the addressee that he is about to speak about 'something else' and thus structures the dialogue). Another example is the communicative function 'check question', which when used with a task-related dimension (i.e. advancing the task, goal or activity of the dialogue) can be in the 'task' dimension, or when checking correct understanding of a previous utterance can be in the 'allo-feedback' dimension.

In total the ISO 24617-2 taxonomy consists of 9 dimensions, 41 dimension-specific functions, and 26 general-purpose functions (Petukhova, Malchanau & Bunt, 2014). Table 3 includes some examples of dimension-specific functions and general-purpose functions. For a complete overview of the scheme's communicative functions, dimensions and other concepts see e.g. http://semantic-annotation.uvt.nl/DiAML-Data-Categories.pdf.

Table 3

Examples of Dimension-Specific and General-Purpose Functions

Dimension-specific Functions	
Function	Example
Auto-positive	"Okay"
Stalling	"Ehm"
Initial-greeting	"Good morning"
General-purpose Functions	
Set-question	"How far is it to the station?"
Check-question	"The meeting starts at ten, right?"
Inform	"The 6.34 to Breda leaves from platform 2."
Confirm	"Indeed"
Correction	"To Montreal, not to Ottawa."
Suggest	"Let's wait for the speaker to finish."
Request	"Please drive very carefully."

Notes. Source: (ISO 24617). For all communicative functions defined in ISO 24617-2 and their definitions, see also http://semantic-annotation.uvt.nl/DiAML-Data-Categories.pdf.

Utterances in dialogues are very often multifunctional, which means that they have more than one communicative function (Bunt, 2011). Therefore, dialogue analysis and annotation frameworks are often ‘multidimensional’, as they allow the assigning of more than one communicative function (in as many dimensions) to functional segments (Bunt, 2011). Note, however, that each functional segment may only contain one communicative function per dimension. The dialogue fragment[1] in (4) illustrates the concepts of multifunctionality and multidimensionality.

(4) 1. S: Which way do we wanna go?
2. U: Uh to Dansville to Corning to Elmira.

In the first utterance S asks U a question. More specifically, taking into account the ISO 24617-2 annotation scheme, the first utterance may be annotated as having the communicative function ‘set question’, and since the subject of the dialogue includes the navigation of transport the ‘task’ dimension is most suitable here. In the second utterance U then takes the turn (‘turn take’ function/’turn management’ dimension), stalls by the use of ‘uh’ (‘stalling’/’time management’) and finally answers the question (‘answer’/’task’).

Whereas schemes such as DAMSL (and MRDA) do not have an explicit definition of dimension and use ‘dimension’ as an umbrella-concept of clusters of communicative functions (i.e. groups of communicative functions that are alike), the ISO 24617-2 scheme, following the DIT scheme, defines dimensions as ‘types of semantic content’ (Bunt et al., 2012). In the DIT scheme and subsequently in the ISO 24617-2 scheme, this has resulted in the distinction between the nine aforementioned dimensions (see Table 2) and allows for a communicative function to be related to some dimension (note that this applies to general-purpose functions,

[1] Fragment from dialogue TRAINS 2 (TRAINS corpus) available in the DialogBank

and only to a certain extent to dimension-specific functions as they are specific to a particular dimension).

2.1.4 Core Concepts.

Figure 1 shows the metamodel for dialogue act annotation according to the ISO 24617-2 annotation scheme, with the core concepts and relations between these concepts that may occur in its annotations (ISO 24617). In comparison to most of its predecessors, the ISO 24617-2 (and DIT++) scheme allows for semantically more complete annotations (DB16). This is, among other things, because of the introduction of qualifiers, functional dependence relations, feedback dependence relations, and rhetorical relations. The dialogue fragment[2] in (5) illustrates most of the concepts that are identified in the metamodel.

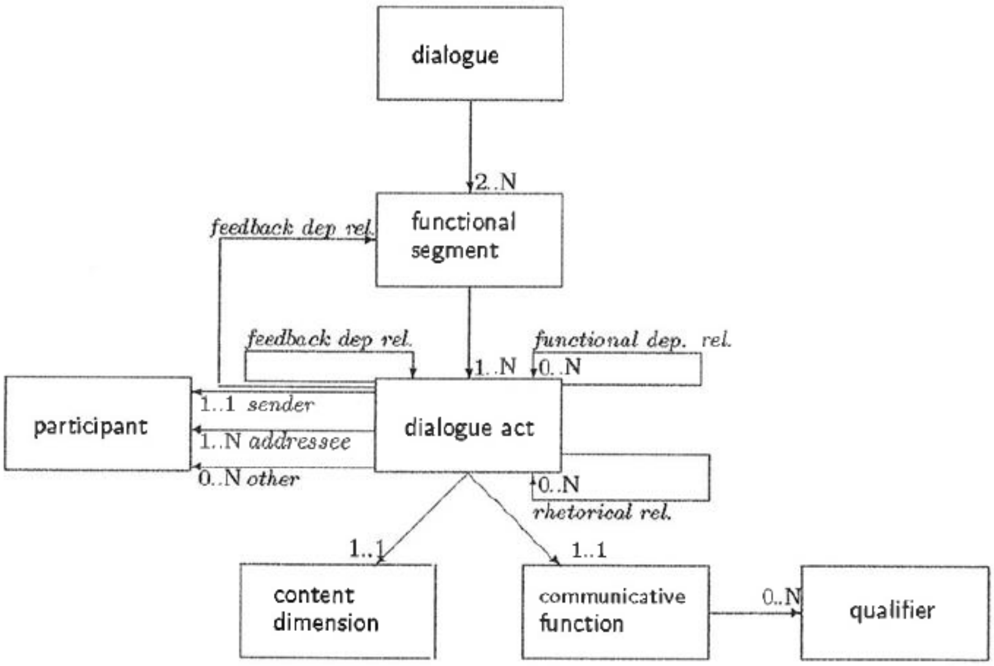

Figure 1. ISO 24617-2 metamodel for Dialogue Act Annotation.

(5) fs1. G: go south and you'll pass some cliffs on your right
fs2. F: uhm…
fs3. G: and some adobe huts on your left
fs4. F: oh okay

[2] Fragment from HCRC Map Task dialogue

First, the metamodel shows that a dialogue must be composed of at least two functional segments ("2 … N"). The dialogue fragment in (5) contains four functional segments, each corresponding to at least one dialogue act with a communicative function and a dimension; fs1: 'inform'/'task', fs2: 'turn take'/'turn management' and 'stalling'/'time management', fs3: 'inform'/'task', and finally fs4: 'auto positive'/'auto feedback'.

Then, the model specifies that each dialogue act has a sender, one or more addressees, and zero or more other participants (e.g. other people in the same room). In fragment (5) the sender and addressee are in turns G and F, while other participants are not present. Furthermore, zero or more '(function) qualifiers' may be assigned to a communicative function. Qualifiers are defined for expressing that a dialogue act is performed with 'certainty' (value: certain or uncertain), 'conditionality' (value: conditional or unconditional) or some 'sentiment' (value: an emotion or attitude) (Bunt et al., 2012). The following examples illustrate these qualifiers:

Certainty: "I'm not sure if I'll be there tomorrow." (value = uncertain)
Conditionality: "Only if you'll be there." (value = conditional)
Sentiment: "Woah, I never expected that!" (value = surprise)

The metamodel also includes so-called functional dependence relations. These describe relations between dialogue acts that are semantically dependent on one or more dialogue acts that occurred previously in a dialogue. This means that the semantic content of the dialogue act can only be determined by considering the semantic content of these earlier dialogue acts (Petukhova et al., 2011). This is for instance the case for indicating which question is answered or which offer is rejected, where the meaning of the answer and the rejection is 'co-determined' respectively by the question that is answered and the offer that is rejected. Functional dependence relations are annotated in ISO 24617-2 by providing a dialogue act and a (second) previous dialogue act if determination of the semantic content of the first act requires the semantic content of the second act.

In addition to functional dependence relations ISO 24617-2 defines feedback dependence relations. These relate a feedback act, i.e. a dialogue act which provides or elicits information about the processing of something that was said before, to the part of the dialogue that the feedback is about (Bunt, 2013). The difference between both dependence relations is that feedback acts can also be about the processing of something that was said earlier, and are not necessarily, like functional dependence relations, a response to the dialogue acts that were expressed (Bunt et al., 2010). Take the examples in (6) (Bunt et al., 2010):

(6) a. S: That's at two-thirty.
T: I see.
b. J: I would like to come on Thursday.
C: On Thursday?

In (6a) T expresses a (positive) feedback act indicating that (s)he heard and understood what S has said, i.e. the semantic content of the dialogue act. However, in (6b) C checks if his or her perception of what J has said is correct. It could be that J was hesitating between Tuesday and Thursday, or maybe J has difficulty pronouncing the 'th' (Bunt et al., 2010).

In (5) the feedback act corresponding to fs4 has a feedback dependence relation with the dialogue acts corresponding to fs1 and fs3, indicating F has understood G. Feedback dependence relations are annotated in ISO 24617-2 by providing the feedback act (dialogue act where the feedback is given) and the stretch of communicative behaviour that the feedback is about (the feedback antecedent), which according to ISO 24617-2 must be either a dialogue act or functional segment.

Finally, a dialogue act may have a rhetorical relation with one or more other dialogue acts. In (5) there is a rhetorical relation between the first dialogue act and the third dialogue act.

More specifically, the dialogue acts are related with an ‘expansion’ relation as the information (‘inform’) expressed with da3 expands on the information that is expressed with da1 (‘instruct’). The ISO 24617-2 scheme does not specify a fixed set of rhetorical relations, and can include rhetorical relations such as ‘cause’, ‘exemplification’ and ‘contrast’ (ISO 24617). In Section 3.1.1 and Section 3.1.2 feedback dependence relations and rhetorical relations are discussed in more detail, respectively, taking into account the aforementioned problems that came to light during the construction of the DialogBank.

2.2 The Dialogue Act Markup Language

2.2.1 Annotation Language.

The ‘Dialogue Act Markup Language ‘(DiAML) is the annotation language of ISO 24617-2. DiAML specifies the scheme’s concepts, possible ways of combining these concepts, and an XML-based reference format called DiAML-XML as representation of its annotations. As one of the annotation languages of SemAF, DiAML follows the rules of the Linguistic Annotation Framework (LAF, ISO 24612) and makes a distinction between ‘annotation; the linguistic information that is added to segments of data, and ‘representation’; the format in which the annotation is presented, independent of its content. According to LAF, annotations rather than representations are the proper level of standardization. Following the ISO principles of Semantic Annotation (ISO 24617-6), this is reflected in DiAML’s abstract syntax, which specifies basic concepts and ways of combining them, and the concrete syntax, which may include various representation formats as long as they are ideal formats.

2.2.2 Abstract Syntax vs Concrete Syntax.

The definition of DiAML follows the distinction between annotation and representation according to SemAF principles, requiring a class of abstract annotation structures in terms of set-theoretical pairs and tuples, and a class of XML-based representation structures (Bunt, 2011). The concrete syntax defines an XML representation of the abstract annotation structures.

The abstract syntax of DiAML is composed of two parts. First, a conceptual inventory which specifies the elements from which annotation structures are built up. This inventory includes six finite sets (ISO 24617), see (7).

(7) 1. *DP*: dialogue participants, possibly including besides the active participants also 'other participants' that do not actively participate.
2. *Dim*: dimensions;
3. *CF*: communicative functions;
4. *FS*: functional segments (provided by the segmentation of the primary data);
5. *QV*: qualifiers;
6. *RR*: rhetorical relations.

Secondly, the abstract syntax contains a specification of the possible ways these elements may be combined to form annotation structures in terms of n-tuples of concepts. An annotation structure consists of a set of *entity structures*, which contain semantic information about a functional segment, and *link structures*, which describe semantic relations between functional segments. These structures are defined as follows.

An *entity structure* is a tuple, $\varepsilon = <m, \alpha>$, consisting of a component 'α' that characterises a dialogue act structure and a functional segment 'm' that the dialogue act is anchored to. A dialogue act structure is a 6-tuple or a 7-tuple, composed of maximally seven elements; $\alpha = <S, A, H, d, f, Q, (\Delta)>$, where S refers to the sender of the dialogue act, A to the addressee(s) of the dialogue act, H to other participants (may be empty), d to the dimension of the dialogue act, f to the communicative function, Q to the qualifiers (may be empty), and Δ to the other dialogue acts or functional segments that the dialogue act may depend on (functional or feedback dependence). A dialogue act that does not depend on any other dialogue act(s) or functional segment(s) does not have the latter component.

Note that rhetorical relations are not part of the meaning of a particular dialogue act, in contrast to feedback and functional dependence relations. Concerning functional dependence relations, the meaning of an answer, for instance, cannot be determined without taking into account the question that is being answered. This is also the case for the acceptance or rejection of offers, suggestions and requests and the acceptance of apologies and thankings (DB16). The same goes for feedback dependence relations, where the meaning of a feedback act depends on earlier contributions to the dialogue. Rhetorical relations, by contrast, are not part of the meaning of a dialogue act but add semantic information about how a dialogue act is related to one or more other dialogue acts. The abstract syntax therefore specifies besides entity structures also *link structures* that specify these rhetorical relations. A link structure is a triple, $\mathrm{L} = <\varepsilon, E, \rho>$, composed of an entity structure (ε), a set of one or more entity structures (E) that the entity structure in question has a rhetorical relation with, and a rhetorical relation (ρ) relating the dialogue acts in ε and E.

The concrete syntax defines a way to represent the annotation structures, as specified by the abstract syntax, in the XML-based format known as DiAML-XML (more on this format in the following section). As mentioned in the introduction, for a concrete syntax to be called ideal, two conditions have to be met; the format has to be *complete* and *unambiguous* relative to the abstract syntax. Completeness refers to the concrete syntax being able to define a representation for every structure defined by the abstract syntax. Unambiguity refers to the concrete syntax representing one and only one structure defined by the abstract syntax. The representations defined by a concrete syntax that is both complete and unambiguous can be converted through a meaning-preserving mapping to any other ideal representation format, as they are semantically equivalent (Bunt, 2011; DB16). Therefore, representation formats other than DiAML-XML may be designed for the given DiAML abstract syntax, as long as they meet the above conditions.

The model in Figure 2 shows the relations between an abstract syntax, its semantics, and multiple ideal concrete syntaxes (ISO 24617-6). The functions F_1, F_i, F_j, and F_n designate the encoding of abstract annotation structures according to the concrete syntaxes 1, i, j, and n. The functions F_1^{-1}, F_i^{-1}, F_j^{-1}, and F_n^{-1} designate the inverse decodings. Due to the existence of these encodings and decodings, the abstract syntax can operate as an interlingua between ideal representation formats. The meaning-preserving conversions prove the interoperability of these formats. The arrows C_{ij} and C_{ji} in figure 444 between 'Ideal Concrete Syntax i' and 'Ideal Concrete Syntax j' indicate their one-on-one interoperability.

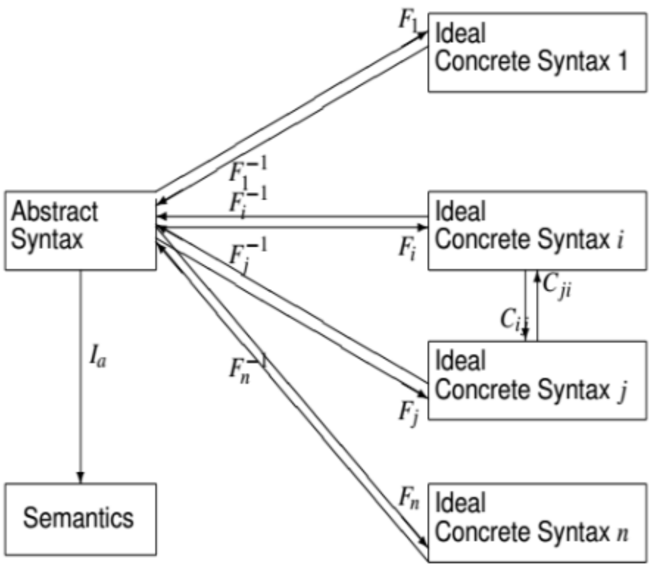

Figure 2. Conversions between multiple concrete syntaxes via the abstract syntax and semantics.

2.2.3 DiAML-XML.

DiAML-XML is the representation format specified in the ISO 24617-2 standard. Dialogues annotated in this representation format can be found at https://git.io/vKmTz[3], https://git.io/vKmTp[4] or the DialogBank website. Example (9) contains a DiAML-XML annotation of the fragment in (8) from dialogue "TRAINS 2" from the TRAINS corpus. Example (10) contains the segmentation of this fragment in functional segments.

[3] HCRC Map Task dialogue "q1ec5"
[4] DBOX dialogue "gl-Diana"

(8) S: hello, can I help you?

U: Yes, I have a problem. I need to transport two tankers of OJ [= orange juice] to Avon and three boxcars of bananas to Elmira, the bananas must arrive in Elmira by nine p.m.

(9)

```
<diaml xmlns="http://www.iso.org/diaml">
   <dialogueAct xml:id="da1" sender="#s" addressee="#u" dimension="discourseStructuring" target="#fs1"
      communicativeFunction="opening" />
   <dialogueAct xml:id="da2" sender="#s" addressee="#u" dimension="socialObligationsManagement" target="#fs1"
      communicativeFunction="initialGreeting" />
   <dialogueAct xml:id="da3" sender="#u" addressee="#s" dimension="discourseStructuring" target="#fs2"
      communicativeFunction="offer" />
   <dialogueAct xml:id="da4" sender="#u" addressee="#s" dimension="discourseStructuring" target="#fs3"
      communicativeFunction="acceptOffer" functionalDependence="da3" />
   <dialogueAct xml:id="da5" sender="#u" addressee="#s" dimension="task" target="#fs4"
      communicativeFunction="inform" />
   <dialogueAct xml:id="da6" sender="#u" addressee="#s" dimension="task" target="#fs5"
      communicativeFunction="inform" />
   <rhetoricalLink dact="#da6" rhetoAntecedent="#da5" rhetoRel="expansion" />
   <dialogueAct xml:id="da7" sender="#s" addressee="#u" dimension="autoFeedback" target="#fs6"
      communicativeFunction="autoPositive" feedbackDependence="da6"/>
</diaml>
```

(10) fs1. S: hello

fs2. S: can I help you

fs3. U: yes

fs4. U: I have a problem

fs5. U: I need to transport two tankers of OJ to Avon and three boxcars of bananas to Elmira

fs5. U: the bananas must arrive in Elmira by nine p.m.

The DiAML-XML concrete syntax reflects the definition of entity structures and link structures in the abstract syntax by defining entity structure representations and link structure representations. As illustrated in (9), these structures are represented by the XML attributes and values inside the <dialogueAct> and <rhetoricalLink> elements, respectively. In (11) all possible attributes and values of the <dialogueAct> and <rhetoricalLink> elements are specified. Note that, following an XML convention, the values in the DiAML-XML format prefixed by the symbol '#' are assumed to be identified elsewhere in the representation or in some other layer of data.

(11) The XML element <dialogueAct> has the following attributes:

- *xml:id*, whose value specifies a unique identifier of an entity structure;
- *target*, whose value refers to a functional segment;
- *sender*, whose values refers to a dialogue participant;
- *addressee* and *otherParticipant* whose values refer to a set of dialogue participants (the latter is often omitted if the set is empty);
- *dimension* and *communicativeFunction*, whose values are one of the possible dimensions and communicative functions defined in the ISO 24617-2 annotation scheme, respectively;
- *functionalDependence* and *feedbackDependence*, whose values refer to one or more dialogue acts or functional segments;
- *certainty* (values: certain, uncertain), *conditionality* (values: conditional, unconditional), and *sentiment* (values: not specified by ISO 24617-2).

The XML element <rhetoricalLink> has the following attributes:

- *dact*, dialogue act under consideration (where the rhetorical relation becomes apparent);
- *rhetoAntecedent* (or *rhetoRelatum*), whose value refers to one or more related dialogue acts;
- *rhetoRel*, whose value represents a rhetorical relation.

For the first utterance of participant S in dialogue fragment (8), "Hello can I help you?", the DiAML-XML representation in (9) specifies two functional segments as values of the 'target' attribute, namely fs1 and fs2. The first and second 'dialogueAct' elements, da1 and da2, are both anchored to the first functional segment, respectively annotating fs1 with the communicative function 'opening' in the 'discourse structuring' dimension, and the communicative function 'initial greeting' in the 'social obligations management' dimension. This example illustrates the possible multidimensionality of functional segments, as more than one dialogue act is assigned to a functional segment. Both dialogue acts have the same 'sender' (S) and 'addressee' (U). Furthermore, the DiAML-XML representation shows that the second dialogue act (anchored to the functional segment: "can I help you") and third dialogue act (anchored to the functional segment "yes") are related with a functional dependence relation as

U accepts the offer that S made. Moreover, the example shows that two dialogue acts in U's utterance, namely da6 (functional segment: "the bananas must arrive in Elmira by nine p.m.") and da5 (functional segment: "I have a problem … bananas to Elmire."), are related with a rhetorical relation; the information expressed with da6 is an 'expansion' on the information expressed with da5.

2.2.4 Three-Level Architecture.

DiAML assumes a three-level-architecture including (1) the primary data (e.g. a textual transcription or speech recording), (2) the marking of functional segments from the primary data, and (3) the actual annotation associated with the functional segments (ISO 24617-6; Petukhova, et al., 2011). DiAML annotation constitutes the third level and follows the stand-off annotation approach, where the annotations are stored separately from the primary data. DiAML annotation uses functional segment identifiers that are specified at the second level and which in turn refer to primary data elements at the first level. The following describes the three-level architecture in DiAML-XML in more detail.

A DiAML-XML file generally consists of two parts; (1) a coding and (2) an annotation (Petukhova & Bunt, 2012). The coding specifies level 1 of the annotation, i.e. the primary data in terms of words, timestamps, and/or non-verbal behaviour, and level 2, in terms of the specification of functional segments from the primary data. Example (12) [5] shows a level-1 representation of primary data in DiAML-XML according to the TEI ('Text Encoding Initiative') guidelines (TEI P5, 2007). More specifically, it displays the specification of words and non-verbal behaviour, plus corresponding times of occurrence in the dialogue (the example only contains the first seven elements of primary data). For instance, according to the annotator the uttering of the word 'okay' ('wp11', meaning the first word of the first participant) started at time '0.2635' ('TWSp11') and ended at time '0.583' ('TWEp11'). Note that such a

[5] Fragment from HCRC Map Task dialogue 'q1ec5', available in the DialogBank.

representation of primary data may also be composed of only words, only timestamps, and/or only nonverbal behaviour (or any combination).

(12)

```
<div>
   <timeline xml:id="TL01" unit="s">
      <when xml:id="TW0" absolute="00:00:00"/>
      <when xml:id="TWSp10" interval="0" since="#TW0"/>
      <when xml:id="TWEp10" interval="0.2635" since="#TW0"/>
      <when xml:id="TWSp11" interval="0.2635" since="#TW0"/>
      <when xml:id="TWEp11" interval="0.583" since="#TW0"/>
      <when xml:id="TWSp12" interval="0.583" since="#TW0"/>
      <when xml:id="TWEp12" interval="0.6456" since="#TW0"/>
      <when xml:id="TWSp13" interval="0.6456" since="#TW0"/>
      <when xml:id="TWEp13" interval="1.1644" since="#TW0"/>
      <when xml:id="TWSp14" interval="1.1644" since="#TW0"/>
      <when xml:id="TWEp14" interval="1.4394" since="#TW0"/>
      <when xml:id="TWSp15" interval="1.4394" since="#TW0"/>
      <when xml:id="TWEp15" interval="1.581" since="#TW0"/>
      <when xml:id="TWSp16" interval="1.581" since="#TW0"/>
      <when xml:id="TWEp16" interval="1.6568" since="#TW0"/>
      <when xml:id="TWSp17" interval="1.6568" since="#TW0"/>
      <when xml:id="TWEp17" interval="2.4177" since="#TW0"/>
      <when xml:id="TWSp18" interval="2.4177" since="#TW0"/>
      <when xml:id="TWEp18" interval="2.5149" since="#TW0"/>
      <when xml:id="TWSp19" interval="2.5149" since="#TW0"/>
      <when xml:id="TWEp19" interval="3.078" since="#TW0"/>
      <when xml:id="TWSp110" interval="3.0781" since="#TW0"/>
      <when xml:id="TWEp110" interval="3.1242" since="#TW0"/>
      <when xml:id="TWSp111" interval="3.1242" since="#TW0"/>
      <when xml:id="TWEp111" interval="3.4503" since="#TW0"/>
      <when xml:id="TWSp112" interval="3.4503" since="#TW0"/>
      <when xml:id="TWEp112" interval="3.5524" since="#TW0"/>
      <when xml:id="TWSp113" interval="3.5524" since="#TW0"/>
      <when xml:id="TWEp113" interval="4.0185" since="#TW0"/>
      <when xml:id="TWSp114" interval="4.0185" since="#TW0"/>
      <when xml:id="TWEp114" interval="4.4795" since="#TW0"/>
   </timeline>
</div>

<div>
   <u who="p1">
      <w xml:id="wp10" start="#TWSp10" end="#TWEp10">_</w>
      <w xml:id="wp11" start="#TWSp11" end="#TWEp11">okay</w>
      <w xml:id="wp12" start="#TWSp12" end="#TWEp12">_</w>
      <w xml:id="wp13" start="#TWSp13" end="#TWEp13">inbreath</w>
      <w xml:id="wp14" start="#TWSp14" end="#TWEp14">_</w>
      <w xml:id="wp15" start="#TWSp15" end="#TWEp15">now</w>
      <w xml:id="wp16" start="#TWSp16" end="#TWEp16">you're</w>
      <w xml:id="wp17" start="#TWSp17" end="#TWEp17">starting</w>
      <w xml:id="wp18" start="#TWSp18" end="#TWEp18">_</w>
      <w xml:id="wp19" start="#TWSp19" end="#TWEp19">ehm</w>
      <w xml:id="wp110" start="#TWSp110" end="#TWEp110">_</w>
      <w xml:id="wp111" start="#TWSp111" end="#TWEp111">above</w>
      <w xml:id="wp112" start="#TWSp112" end="#TWEp112">the</w>
      <w xml:id="wp113" start="#TWSp113" end="#TWEp113">diamond</w>
      <w xml:id="wp114" start="#TWSp114" end="#TWEp114">mine</w>
   </u>
</div>
```

The identification and specification of functional segments from the primary data, as in example (13), corresponds to ‘level 2’. The example shows that the functional segments ‘fsp1TUMCV0’, ‘fsp1TIMCV0’, and ‘fsp1TSKCV0’ are composed of the primary data elements that are specified as values of the ‘from’ attributes. As mentioned before, these particular values are specified at the first level, see (12).

(13)

```
<div>
   <spanGrp xml:id="vesp1TUMCV0" type="functionalVerbalSegment">
      <span xml:id="tsp1CV0l1" from="#wp11"/>
   </spanGrp>
   <fs xml:id="fsp1TUMCV0" type="functionalSegment">
      <f name="verbalComponent" fVal="#vesp1TUMCV0"/>
   </fs>
   <spanGrp xml:id="vesp1TIMCV0" type="functionalVerbalSegment">
      <span xml:id="tsp1CV0l1" from="#wp19"/>
   </spanGrp>
   <fs xml:id="fsp1TIMCV0" type="functionalSegment">
      <f name="verbalComponent" fVal="#vesp1TIMCV0"/>
   </fs>
   <spanGrp xml:id="vesp1TSKCV0" type="functionalVerbalSegment">
      <span xml:id="tsp1CV0l1" from="#wp15"/>
      <span xml:id="tsp1CV0l2" from="#wp16"/>
      <span xml:id="tsp1CV0l3" from="#wp17"/>
      <span xml:id="tsp1CV0l4" from="#wp111"/>
      <span xml:id="tsp1CV0l5" from="#wp112"/>
      <span xml:id="tsp1CV0l6" from="#wp113"/>
      <span xml:id="tsp1CV0l7" from="#wp114"/>
   </spanGrp>
   <fs xml:id="fsp1TSKCV0" type="functionalSegment">
      <f name="verbalComponent" fVal="#vesp1TSKCV0"/>
   </fs>
</div>
```

The coding of this dialogue fragment is formed by (12) and (13), while (14) shows the DiAML-XML annotation of this fragment at level 3. The annotation uses functional segment identifiers that point to functional segments at the second level. These identifiers are specified as values of the ‘target’ attribute. In (14) the third ‘dialogueAct’ element, for instance, includes the functional segment ‘#fsp1TSKCV0’ (here the symbol # means the functional segment is specified at the second level). As shown in (13) this second level functional segment consists of the primary data elements ‘#wp15’, ‘#wp16’, ‘#wp17’, ‘#wp111’, ‘#wp112’, ‘#wp113’, and ‘#wp114. Together they form the following functional segment: “now you’re starting above the diamond mine” (see (12)). The annotation in (14) shows, among other things, that the dialogue act that is associated with this functional segment has the communicative function ‘instruct’ in the ‘task’ dimension. Furthermore, a turn take function in the turn management dimension has been assigned to the primary data element ‘wp11’ (“okay”) and a stalling function in the time management dimension has been assigned to the primary data element ‘wp19’ (“ehm”). The new ISO 24617-2 annotation representation formats DiAML-MultiTab and DiAML-TabSW have the three-level architecture implemented as well.

(14)

```
<diaml xmlns="http://www.iso.org/diaml">
   <dialogueAct xml:id="dap1TUM0" sender="#p1" addressee="#p2" dimension="turnManagement"
      communicativeFunction="turnTake" target="#fsp1TUMCV0"/>
   <dialogueAct xml:id="dap1TIM0" sender="#p1" addressee="#p2" dimension="timeManagement"
      target="#fsp1TIMCV0" communicativeFunction="stalling"/>
   <dialogueAct xml:id="dap1TSK0" sender="#p1" addressee="#p2" dimension="task"
      target="#fsp1TSKCV0" communicativeFunction="instruct"/>
</diaml>
```

3. The DialogBank

The current section is divided into three parts. Section 3.1 is related to the first research question, describing two problems in using ISO 24617-2 that were encountered during the construction of the DialogBank. The next two sections are related to the second and third research question and are focused on the new representation formats and their interoperability.

3.1 Problems Encountered in Using ISO 24617-2

3.1.1 Feedback Dependence Relations.

During the construction of the DialogBank and more specifically during the updating and re-annotation of existing annotations, two theoretical problems were encountered that both have to do with the annotation of relations between units in a dialogue. The first problem concerns feedback dependence relations. As briefly discussed in Section 2.1.4, these are relations between a feedback act and the stretch of communicative behaviour whose processing the act provides or elicits information about. These stretches are often called ‘feedback antecedents’ and can be either dialogue acts or functional segments.

The DIT++ annotation scheme distinguishes between feedback related to understanding and other aspects of processing dialogue utterances – using feedback acts such as Attention Positive Feedback, Perception Positive Feedback, Interpretation Positive Feedback, Evaluation

Positive Feedback, Execution Positive Feedback, and similar acts for 'negative feedback' and 'feedback elicitation' – resulting in a fine-grained collection of communicative feedback functions, concerned with different levels of processing, such as hearing, understanding, and accepting something (Bunt, 2012). The ISO 24617-2 scheme, for which the DIT++ scheme has been a major source of inspiration, has a more coarse-grained collection of feedback acts. This was mainly done to simplify the annotation process for annotators, as the various types of feedback specified by DIT++ may be hard to differentiate. Therefore, the ISO 24617-2 scheme only distinguishes between positive and negative auto- and allo feedback. See (15), where B's utterance is about B's own processing/understanding (auto-feedback), and A's second utterance is about B's processing/understanding (allo-feedback).

(15)
1. A: Now move up.
2. B: Slightly northeast?
3. A: Slightly yeah.

Both auto- and allo-feedback can be positive, indicating successful processing, and negative, indicating unsuccessful processing (Bunt, 2012). As will be explained in the following sections, this simplified collection of feedback functions raises the question what to do with feedback antecedents.

Positive feedback, such as "yeah", "okay", or head nodding, is often a response on the level of dialogue acts indicating for instance understanding, agreement or acceptance of what was said. Negative feedback can also be a response at the level of dialogue acts, for instance "Really?", or "Good question". In these cases ISO 24617-2 proposes the use of dialogue acts as feedback antecedents. Much of the time, however, negative feedback has to do with not understanding something (e.g. "huh", "what?", "sorry?"). In such cases the feedback is often not a response to a dialogue act, but rather a response to something that was said. Take the following example; “P1: Amsterdam to Barcelona is two hours. P2: “three hours?”, where P2 may have misheard the word “two”. The ISO 24617-2 standard, as a derivative of DIT++, says that in such cases the feedback antecedent is a functional segment, i.e. a functional stretch of primary data. This is incorrect, however, as the feedback antecedent may be ‘just’ a collection of primary data – for instance the primary data element “two” in (15) – which is not a functional segment and does not correspond to a dialogue act.

In DiAML-XML feedback dependence relations are annotated with the ‘feedbackDependence’ attribute in the ‘dialogueAct’ element where the feedback is given. The value of the feedbackDependence attribute is the dialogue act(s) (or functional segment) that the feedback is about. The first two utterances of the fragment in (15) for instance are annotated in DiAML-XML format as in (16). The value of the feedbackDependence attribute here refers to the dialogue act corresponding to the first utterance, i.e. “Slightly northeast?” the ‘check-question’ function in the ‘auto-feedback’ dimension has a feedback dependence relation with “now move up”.

(16)

```
<diaml xmlns: “http://www.iso.org/diaml/”/>
<dialogueAct xml:id=”e1” target=”#fs1” sender=”#a” addressee=”#b”
        communicativeFunction=”instruct” dimension=”task”/>
<dialogueAct xml:id=”e2” target=”#fs2” sender=”#b” addressee=”#a”
        communicativeFunction=”checkQuestion” dimension=”autoFeedback”
        feedbackDependence=”#e1”/>
<diaml/>
```

Now, take the following example:

(17) G: keep going down south past a forge on your right

F: past a what?

Here, the feedback act consists of the words ‘past a what’, and the feedback antecedent is the word ‘forge’. It is safe to say that F has understood the communicative meaning of the dialogue act (the ‘instruct’ function) expressed by G’s utterance and the utterance parts ‘keep going down south’ and ‘on your right’. In fact, the repetition of ‘past a’ may even be taken as positive feedback. However, F may have misheard what G said or may not know what a ‘forge’ is. Therefore, annotating the whole functional segment or dialogue act as feedback antecedent would be incorrect, as the scope of the feedback is not the entire (preceding) functional segment or dialogue act, but rather a smaller stretch of primary data. Specifying the word ‘forge’ as a functional segment and annotating this segment as the feedback antecedent would also be incorrect, as this word is not assigned a communicative function and is therefore not ‘functional’.

The above problem had come to light during the re-annotation of HCRC Map Task dialogues according to ISO 24617-2, and a way of dealing with the problem was used in these existing ISO 24617-2 Map Task annotations (Petukhova, PhD): whenever a feedback antecedent consisted of a non-functional stretch of primary data, new functional segments were created that consisted of these particular primary data elements. Such a functional segment was then used as the value of the feedbackDependence XML-attribute. However, as no dialogue acts were assigned to these segments, these newly created ‘functional segments’ were incorrectly annotated as being functional segments.

A similar problem may arise in the case of dialogue acts in the ‘own communication management’ and ‘partner communication management’ dimensions, which are in a way also negative feedback acts since some dialogue segment is corrected. Dialogue acts in these dimensions do not have a feedback dependence according to ISO 24617-2 (only feedback acts do). However, like feedback acts, they depend on something that was said before. Therefore, these dependences have also been represented using the ‘feedbackDependence’ attribute. This is not compliant with ISO 24617-2, as the scheme says that in these cases functional dependences occur rather than feedback dependences. Note, however, that following the ISO 24617-2 standard in this regard would also lead to incorrect annotations. See for instance the 'own- and partner communication management' example in (18) where, similar to some feedback acts, the feedback antecedent is neither a dialogue act nor a functional segment.

(18) 1. U: that's where the orange juice factory is
2. U: … orange factory.

In (18) speaker U corrects himself in the second utterance ("orange" as correction of "orange juice"). The annotation of this dialogue fragment could look as in (19), while a fully ISO 24617-2 compliant representation would have a functionalDependence rather than a feedbackDependence. Note that both are incorrect, because there is no dependence with a dialogue act or with a functional segment.

(19)
```
<diaml xmlns: "http://www.iso.org/diaml/"/>
<dialogueAct xml:id="da1" target="#fs1" sender="#u" addressee="#s"
        communicativeFunction="inform" dimension="task"/>
<dialogueAct xml:id="da2" target="#fs2" sender="#u" addressee="#s"
        communicativeFunction="selfCorrection"
        dimension="ownCommunicationManagement" feedbackDependence="#da1"/>
<diaml/>
```

The feedback antecedent that is annotated here is the preceding dialogue act ('inform'), while in fact the self-correction applies only to the words "orange juice".

In conclusion, a particular annotation problem emerges when the antecedent of a feedback dependence relation is a stretch of primary data that does not correspond to a dialogue act or a functional segment. A solution for this problem is proposed in Section 5.1.1. In Section

4.1.1 the methodology that has been used to come to this potential solution is discussed. For now, feedback antecedents of feedback dependence relations in DialogBank annotations will be dialogue acts or functional segments, following ISO 24617-2.

3.1.2 Rhetorical Relations.

The second theoretical problem that was encountered during the construction of the DialogBank concerns the annotation of rhetorical relations as specified by DiAML (see also Section 2.1.4). These relations, also called 'discourse relations', indicate how events, states, facts, and propositions in a discourse (or dialogue) are related to each other. They are important for the understanding of a discourse, beyond the meaning conveyed by individual clauses and sentences (Prasad & Bunt, 2015). The international ISO 24617-8 standard, also known as ISO DR-Core, which is currently under construction and is expected to be established in 2016, is concerned with these relations (see Bunt, Prasad, & Joshi, 2012; Prasad & Bunt, 2015; Bunt & Prasad, 2016). The ISO DR-Core standard has specified its own annotation language, DRelML ("**D**iscourse **Rel**ations **M**arkup **L**anguage"), that like DiAML defines an abstract syntax, a concrete syntax, and a semantics. DiAML has some facilities for the annotation of these relations, in the form of 'link structures' in the abstract syntax, and <rhetoricalLink> elements in the DiAML-XML concrete syntax. As the construction of the DialogBank and the ISO DR-Core standard progressed, it became clear that the DiAML annotation of rhetorical relations was inadequate.

The ISO DR-Core standard defines a set of rhetorical relations, such as 'Cause', 'Condition', 'Elaboration', and 'Exemplification'. See Bunt and Prasad (2016) for the proposed set of relations. Rhetorical relations can occur explicitly, i.e. by means of discourse connectives (e.g. words as 'because' or phrases as 'a major reason'), as in example (20a), or implicitly as in example (20b). Both examples – taken from the Penn Discourse Treebank (PDTB), see

Miltsakaki, Pradas, Joshi & Webber (2004); Prasad et al., (2008) – in (20) illustrate a 'Cause' relation.

(20) a. Mr. Taft, who is also president of Taft Broadcasting Co., said he bought the shares because he keeps a utility account at the brokerage firm of Salomon Brothers Inc., which had recommended the stock as a good buy.
b. Some have raised their cash positions to record levels. (implicit = (because)) High cash positions help buffer a fund when the market falls.

An important distinction in the literature on rhetorical relations, and incorporated in ISO DR-Core, is the so-called 'semantic-pragmatic' distinction. Example (21a) illustrates a causal relation between the semantic contents of the respective dialogue acts, where B gives a reason for A's problem. Example (21b), on the other hand, illustrates a pragmatic cause relation where J's 'expensive camera' is not the reason why the gates at the airport might be unsafe, but rather the motivation for asking the question.

(21) a. A. I can never find my remote control.
B. That's because they don't have a fixed place.
b. J. Is it safe to put my camera through here? Because it's a very expensive camera.

As seen in previous sections, the ISO 24617-2 scheme and its annotation language DiAML do take into account rhetorical relations, though in a less elaborate way than DRelML. A DiAML annotation of rhetorical relations contains the two related dialogue acts and the relation. According to ISO DR-Core, the two dialogue acts that have a rhetorical relation are the two 'arguments' of the relation. The DiAML annotation of example (20b) is shown in (22), where the first dialogue act is anchored to the functional segment "Some have … record levels" and the second dialogue act to "High cash … market falls". The utterances corresponding to da1/fs1 and da2/fs2 are the first and second argument of the Cause relation, respectively.

(22)
```
<diaml xmlns: “http://www.iso.org/diaml”>
<dialogueAct xml:id=”da1” target=”#fs1” sender=”#p1” addressee=”#p2”
        communicativeFunction=”inform” dimension=”task”/>
<dialogueAct xml:id=”da2” target=”#fs2” sender=”#p1” addressee=”#p2”
        communicativeFunction=”inform” dimension=”task”/>
<rhetoricalLink dact=“da2” rhetoAntecedent=”da1” rhetoRel=”cause”/>
<diaml/>
```

According to ISO DR-Core, the two arguments of a rhetorical relation always play a particular ‘role’. In (20b) for instance the first part of the relation ‘causes’ the second part. Therefore, the first and second argument are annotated in DRelML as the ‘reason’ and the ‘result’ of the relation, respectively. Similarly, in an ‘elaboration’ relation one argument describes the same situation in more detail than the other argument (‘specific’ vs ‘broad’). These are examples of asymmetrical rhetorical relations. Examples of symmetrical relations are a ‘contrast’ relation where one or more differences between the arguments of the relation are highlighted (‘arg1’ vs ‘arg2’) or a ‘similarity’ relation where one or more similarities between the arguments are highlighted (‘arg1’ vs ‘arg2’). DiAML only specifies the rhetorical relation (e.g. ‘Cause’), and disregards the specific roles that the arguments of the relation play. Another limitation of DiAML is that it does not distinguish between semantic and pragmatic variants of a rhetorical relation.

In other words, DiAML annotation of rhetorical relations is inadequate, as information that is valuable to the interpretation of these relations is not annotated. In the current thesis, two solutions are proposed that aim to enrich and improve the annotation of rhetorical relations. In Section 4.1.2 the methods that have been used for both solutions are discussed. In Section 5.1.2 and Section 5.1.3 the solutions are explained in detail.

3.2 Alternative Representation Formats

3.2.1 Tabular Representation Formats.

Most of the dialogues present in the DialogBank had already been annotated according to some annotation scheme and presented in some representation format. These representations were often in tabular form, in which the rows relate to a dialogue's segmentation of primary data and the columns to the actual annotation (DB16).

Table 4 includes an annotation according to the DIT annotation scheme in a tabular form, produced with the DitAT tool (Geertzen, 2007). Table 5 shows an annotation according to the SWBD-DAMSL annotation scheme in another tabular form.[6] These tabular representation formats do not resemble the DiAML-XML format in many ways. However, the information represented in each of the formats is not that different. For instance, the numbers in the first column in Table 4, which are identifiers for functional segments, resemble some of the numbers in the first column in Table 5 (the four numbers after the dialogue name specification 'sw02-0224'). In the DiAML-XML format, see (22), these functional segment identifiers are represented by the 'target' attribute and its value. Also, the second column in Table 4, the letter after the functional segment id in Table 5, and the value of the 'sender' attribute in (22) all specify the sender of a dialogue segment. The third column in Table 4 and Table 5 represent the transcripts of the functional segments, while the rest of the columns in Table 4 and the second column in Table 5 represent the dialogue act annotation. A DiAML-XML representation of the third row in Table 4 would look as in (22).

(22)

```
<diaml xmlns: “http://www.iso.org/diaml/”/>
    <dialogueAct xml:id=”da4” target=”#fs3” sender=”#U” addressee=”#S”
            dimension=”turnManagement” communicativeFunction=”turnTake”/>
    <dialogueAct xml:id=”da5” target=”#fs3” sender=”#U” addressee=”#S”
            dimension=”timeManagement” communicativeFunction=”stalling”/>
    <diaml/>
```

---

[6] Fragment from dialogue sw02-0224, Switchboard corpus.

Table 4

DIT Annotation in Tabular Representation Format

Id	Sp	transcript	Task	AutoF	AlloF	TuM	TiM	DS	CM	OCM	PCM	SOM
1	S	hello							Contact indication			Initial greeting
2		can I help you						Offer				
3	U	Uhm,				Turn Take	Stalling					
4	U	yes hello, maybe		Eval. Pos.				Accept offer				
5		I'd like to take a tanker ..	inform					Topic intro				

Notes. Sp = speaker, autoF = auto feedback, alloF = allo feedback, TuM = turn management, TiM = time management, DS = discourse structuring, CM = contact management, OCM = own communication management, PCM = partner communication management, SOM = social obligations management, Eval. Pos. = evaluation positive.

Table 5

SWBD-DAMSL Annotation in Tabular Representation Format

id	function	transcript
sw02-0224-0001-A001-01	o	A.1 utt1: Okay. /
sw02-0224-0002-B002-01	o	B.2 utt1: All right. /
sw02-0224-0003-B002-02	qy	B.2 utt2: {F Uh, } do you have any friends that have children? /
sw02-0224-0004-A003-01	na	A.3 utt1: I do have friends that have children, /
sw02-0224-0005-A003-02	ny	A.3 utt2: yes. /

Notes. 'o' stands for 'other', 'qy' for 'yes/no question', 'na' for 'affirmative non-yes answers' and 'ny' for 'yes answers' (Jurafsky, Shriberg, & Biasca, 1997).

While the above tabular formats succeed in expressing some of the information presented in DiAML-XML, these formats lack 'information-richness'. For instance, the semantic information that is assigned to dialogue segments only includes communicative functions (Table 5), and communicative functions and dimensions (Table 4). Moreover, both tabular formats do not allow the annotation of qualifiers and possible feedback dependence relations, functional dependence relations and rhetorical relations. Furthermore, taking into account the text transcriptions in the third columns, both formats only allow for neighbouring, non-overlapping dialogue segments (DB16).

The new representation formats, DiAML-MultiTab and DiAML-TabSW, which are inspired by the tabular formats in Table 4 and Table 5 respectively, do not have these problems. By allowing for more information to be placed inside a cell, extra information such as qualifiers and relations can be annotated. In addition, the second problem can be overcome by following the stand-off requirement specified by LAF where the annotations are stored separately from the primary data, allowing non-neighbouring primary data elements to form a functional segment as well as allowing primary data elements to be part of more than one functional segment. For more on the stand-off requirement see also Section 2.2.4 and Sections 3.2.2 and 3.2.3 in the DiaML-MultiTab and DiAML-TabSW formats, respectively.

3.2.2 DiAML-MultiTab.

DiAML-MultiTab is one of the two new representation formats for ISO 24617-2 annotations introduced in (DB16). Appendix A includes a dialogue fragment from dialogue TRAINS 2 in this format. To see a full dialogue annotated in DiAML-MultiTab format see https://git.io/vKmU5[7] or the DialogBank website. Table 6 shows the column order and all column names that are present in the DiAML-MultiTab format.

[7] TRAINS dialogue "TRAINS 1", from TRAINS corpus.

Table 6

All Columns in the DiAML-MultiTab Format

Column order	Column name	Column order	Column name
1	Markables	9	alloFeedback
2	Sender	10	turnManagement
3	Addressee	11	timeManagement
4	Other Ps	12	ownCommunicationManagement
5	FS text	13	partnerCommunicationManagement
6	Turn transcript	14	discourseStructuring
7	Task	15	socialObligationsManagement
8	autoFeedback	16	Comments

Table 7 shows an ISO-24617-2 compatible annotation in simplified DiAML-MultiTab format covering the fragment in (23).[8] Due to space limitations nine columns have been omitted from Table 7, namely: ‘Addressee’, ‘Turn transcription’, ‘Other Ps, ‘alloFeedback’, ‘ownCommunicationManagement’,‘partnerCommunicationManagement’, ‘discourseStructuring’, ‘socialObligationsManagement’, and ‘Comments’.

(23) fs1. A: Jimmy,

fs2. A: so how do you get most of your news?

fs3. B: Well

fs4. B: I kind of, I

fs5. B: uh

fs6. B: I watch the national new every day […]

[8] Fragment from dialogue ‘sw-01-0105’, Switchboard corpus.

Table 7

Fragment from Switchboard Dialogue 'sw-01-0105' in Simplified DiAML-MultiTab format

Markables	Sender	FS text	Task	TiM	TuM	OCM
sw0105-fs1	A	Jimmy			da1:turnAssign	
sw0105-fs2	A	So how do you get most of your news	da2:setQuestion			
sw0105-fs3	B	Well		da3:stalling	da4:turnTake	
sw0105-fs4	B	I kind of I				da5:self-Correction
sw0105-fs5	B	Uh		da6:stalling		
sw0105-fs6	B	I watch the national news every day.	da7:answer (Fu:da2)			

Notes. TiM stands for time-management, TuM stands for turn-management, and OCM stands for own communication management.

The structure of the DiAML-MultiTab format largely resembles that of the tabular format in Table 4. The latter format includes for each dimension a separate column, with the communicative function specified in a cell in that column. The DiAML-MultiTab format similarly gives all nine dimensions specified in ISO 24617-2 its own column. However, different from Table 4, a cell in these columns can potentially contain, in addition to a communicative function, also a feedback or functional dependence relation, a rhetorical relation, and qualifiers. Table 8 shows how all of these elements are represented in a cell in the DiAML-MultiTab format. The order of the contents in a cell in DiAML-MultiTab format is as follows: "Dimension:function [qualifiers] (dependences) {rhetorical relations}".

Table 8

ISO 24617-2 Concepts in Cells in DiAML-MultiTab and DiAML-TabSW Formats

Concept	Cell representation
Communicative function	da5:instruct
Feedback dependence	da6:autoPositive (Fe: da5)
Functional dependence	da9:confirm (Fu: da8)
Rhetorical relation	da10: inform {Cause:Result da8 da9}[1]
Qualifiers:	
Certainty	da7: inform [uncertain] da7: answer [certain] (Fu: da6)
Conditionality	da7: inform [conditional]
Sentiment	da7: autoPositive [happiness]

Notes. [1]See Section 5.1.2; potential solution to Problem II. 'Fe' stands for **fe**edback dependence relation, 'Fu' stands for **fu**nctional dependence relation.

The first column in the DiAML-MultiTab format, 'Markables', contains identifiers of the functional segments in a given dialogue. These functional segments and their specification in terms of stretches of primary data are stored in a separate file (as a list of sequences of word tokens and/or timestamps). Note that, taking into account the three-level architecture, this particular file corresponds to the second level. This implementation of the second level overcomes the problem in of the representation used in Table 4, where overlapping or non-neighbouring dialogue segments are not allowed, since primary data elements from earlier (or later) in the dialogue may now be added to any functional segment. The primary data elements that correspond to the first level are specified in their own separate file. This means that the annotation of a dialogue in DiAML-MultiTab format consists of three files: (1) a file containing all primary data elements, (2) a file containing all functional segments that have been identified, and (3) the annotation file in the DiAML-MultiTab format in which information is added to the functional segments.

Similar to Table 4, the second column in the DiAML-MultiTab format specifies the sender of a particular functional segment, while the third and fourth column specify the addressee and possible other participants, respectively. The fifth column in DiAML-MultiTab format contains the text corresponding to the functional segments (‘FS text’), resembling the third column in the DIT example. In addition, the sixth column in the DiAML-MultiTab format, ‘Turn transcript’, specifies the contributions (as stretches of speech or nonverbal behaviour) of the dialogue participant for each turn. This particular column was added for readability purposes, as it allows users to see where functional segments occur in a dialogue (DB16). The next nine columns specify the dialogue acts per ISO 24617-2 dimension and functional segment. Finally, a column was added (‘Comments’) allowing annotators to add any kind of comment about (elements in) a particular row.

3.2.3 DiAML-TabSW.

DiAML-TabSW is the second representation format that is introduced in (DB16). Appendix B includes a fragment from a TRAINS dialogue annotated in the DiAML-TabSW format. For a full dialogue annotated in this format format see https://git.io/vKmk0[9] or the DialogBank website. Currently, the DialogBank contains three dialogues that are originally from the Switchboard corpus and have been re-segmented and re-annotated according to the ISO 24617-2 annotation scheme. The DiAML-TabSW format has been created based on existing SWBD-DAMSL annotation representations. Table 9 shows all columns that are present in the DiAML-TabSW representation format.

[9] Switchboard dialogue ‘sw00-0004’

Table 9

Columns in the DiAML-TabSW Representation Format

Column order	Column name	Column number	Column name	Column order	Column name
1	Markables	4	Sender	7	FS text
2	Da-ID	5	Addressee	8	Turn transcript
3	Dialogue Acts	6	Other Ps	9	Comments

Table 10 shows an ISO-24617-2 compatible annotation in simplified DiAML-TabSW format for the same dialogue fragment as in Table 7. Due to the limited space the 'Addressee' and 'Comments' columns have been omitted from the annotation in Table 10. See Table 5 for the original representation of a Switchboard dialogue fragment before re-segmentation and re-annotation.

Table 10

Fragment from Switchboard Dialogue 'sw01-0105' in Simplified DiAML-TabSW format

Markables	Da-ID	Dialogue acts	Sender	FS text	Turn transcription
sw0105-fs1	da1	TuM:turnAssign	A	Jimmy	Jimmy, so how do you get most of your news?
sw0105-fs2	da2	Ta:setQuestion	A	so how do you get most of your news	
sw0105-fs3	da3; da4	TiM:stalling; TuM:turnTake	B	Well	Well, I kind of, uh, I watch the, uh, national news every day.
sw0105-fs4	da5	OCM:selfCorrection	B	I kind of I	
sw0105-fs5	da6	TiM:stalling	B	uh	
sw0105-fs6	da7	Ta:answer (Fu:da2)	B	I watch the national news every day	

The content in a cell in the DiAML-TabSW format resembles that of a cell in the DiAML-MultiTab format (see Table 8). The only differences are that (1) in the DiAML-TabSW format the 'dialogue act identifier' is stored in a separate column ("Da-ID") and (2) the dimensions are preceding the communicative function in a cell instead of each having their own column (see Table 10). The first column specifies the markables as in the DiAML-MultiTab format. The second column stores the dialogue act identifiers that specify the dialogue act(s) expressed in a functional segment. In the SWBD-DAMSL example in Table 5 the third column contains the sender, the dialogue units in which the dialogue has been segmented – called 'slash-units' in the SWBD-DAMSL scheme – and transcripts of these slash-units. In the DiAML-TabSW format the sender is specified in the fourth column, while the slash-unit identifiers (slash-units often are re-segmented into ISO 24617-2 compliant functional segments) are represented in the first column as functional segment identifiers ('Markables'). The transcripts of the functional segments, similar to DiAML-MultiTab, are represented in the fifth column containing the text of a functional segment ('FS text'). Similar to DiAML-MultiTab, the DiAML-TabSW format contains a column specifying the text transcripts of the turns in a dialogue ('Turn transcript'). The representation of a dialogue annotation in DiAML-TabSW format again consists of three files corresponding to the three-level architecture as explained in Section 3.2.2.

3.3 Interoperability of Representation Formats

3.3.1 Encoding and Decoding Functions for New Formats.

In the current and following sections theoretical mappings between the ISO 24617-2 abstract syntax and the new representation formats are described. These mappings transform a collection of entity structures and link structures to a DiAML-MultiTab and DiAML-TabSW representation, and from a DiAML-MultiTab and DiAML-TabSW representation to a collection of entity structures and link structures.

Tabular representations can be seen as a matrix (i.e. a representation in rows and column) in which each row is an n-tuple of elements (DB16). Elements here refer to concepts specified by the abstract syntax, such as the communicative function, dimension and possible relations inside a row, and the n refers to the number of such elements in a row. Functional and feedback dependence relations give rise to nested structures in the abstract syntax (DB16). Take for instance an answer by participant A to a question by participant B about the task domain. Schematically, such an entity structure would look as follows: <m_1, <A, B, task, answer, <m_2, <B, A, task, question>>>>

The sets of ‘other participants’ and ‘qualifiers’ are omitted here as they are empty. The first markable (m_1) refers to the functional segment that the answer by participant A to B’s question is anchored to. This dialogue act has a nested relation with the dialogue act that forms B’s question. In the new tabular formats such a representation of annotation structures is facilitated by the introduction of dialogue act identifiers. See Table 8 for examples of such identifiers (e.g. ‘Fe: da7’).

In the following the mappings from the annotation structures as defined by the abstract syntax to the DiAML-MultiTab and DiAML-TabSW formats are described, $F_{MultiTab}$ and F_{TabSW} respectively. These mappings follow similar steps as specified in (DB16). The definition of the encoding functions $F_{MultiTab}$ and F_{TabSW} confirms and demonstrates the completeness of the two representation formats, while their unambiguity can be demonstrated by defining the inverse of the functions (DB16). The four functions together demonstrate the interoperability of the representation formats.

Note that the columns ‘FS text’ and ‘Turn transcription’ in the DiAML-MultiTab and DiAML-TabSW formats have been included solely to increase readability (DB16). These columns are not part of the third level DiAML annotation and are irrelevant for these mappings. The same is true for the ‘Comments’ column.

3.3.2 Mapping to DiAML-MultiTab.

The function $F_{MultiTab}$ includes the mapping of DiAML annotation structures to DiAML-MultiTab representations. The $F_{MultiTab}$ encoding includes five main steps (DB16). The starting point is an annotation structure, which is a pair of entity structures and link structures: $<\{\varepsilon_1, ..., \varepsilon_n\}, \{L_1, ..., L_k\}>$.

Step 1: First, each entity structure is assigned an identifier. Remember that an entity structure is composed of a markable and a dialogue act structure (see Section 2.2.2). Secondly, the entity structures are sorted according to their markables (i.e. fs1, fs2, …). Entity structures may contain the same markable, see for instance Table 7 where the third functional segment 'uhm' corresponds with two dialogue acts (da3 and da4). In such a case, sort the entity structures according to their dimension, where Task = 1, Auto-feedback = 2, Allo-feedback = 3, Turn Management = 4, Time Management = 5, Own Communication Management = 6, Partner Communication Management = 7, Discourse Structuring = 8, and Social Obligations Management = 9. Based on the ordering of entity structures that follows assign an index to each entity structure. This results in a set (E) of indexed entity structures: $E = \{<\varepsilon_i, 1>, ..., <\varepsilon_j, n>\}$

Step 2: This step is concerned with all elements inside an entity structure, extracting these elements from the indexed entity structures and transforming them for cells in DiAML-MultiTab format: $T_e<<m,<S, A, H, d, f, Q, \Delta>>, \acute{\imath}> = <<m, <S, A, H, <d,<\acute{\imath}, f, Q, \Delta>>>>>$. The right-hand side already resembles a representation of such structures, where each entity structure is described in its own row. Table 11 shows a representation of such structures in DiAML-MultiTab format. The elements identifying the index, communicative function, possible qualifiers, and possible dependence relations of each entity structure are specified in one cell in the column corresponding to its dimension.

Table 11

Entity Structure Elements Represented in DiAML-MultiTab Format

Elements	m	S	A	H	D (ί, f, Q, Δ)
Columns	Markables	Sender	Addressee	Other Ps	Task
Row	fs4	S	M		da5: confirm [certain] (Fu: da4)

Note. Column ‘Other Ps’ may be empty, as above.

Step 3: The third step concerns the restructuring of elements in link structures (rhetorical relations between dialogue acts) in entity-like form. A link structure L looks as follows: $L = <\varepsilon_1,\{ \varepsilon_2, ..., \varepsilon_k\}, p>$, where the first element, ε_1, refers to an entity structure as specified in step 1: $\varepsilon_1 = <m_1,< S_1, A_1, H_1, d_1, f_1, Q_1, \Delta_1>>$, the second element, $\{\varepsilon_2, ..., \varepsilon_k\}$, refers to one or more entity structures that are (rhetorically) related to the first element, and the third element, p, refers to the rhetorical relation between dialogue acts (e.g. cause, elaboration, contrast). The transformation in the third step includes copying the structures that are built in step 2 and extending them with rhetorical link information: $T_L(L) = <m_1,< S_1, A_1, H_1, d_1,< i_1, f_1, Q_1, \Delta_1<p, p_1,\{ i_2, ..., i_k \}>>>>>$. Note that the link structure information is included as ‘part’ of the relation element (Δ). The second to last element (p_1) refers to the argument role of the related dialogue acts (e.g. Cause:Reason or Cause:Result), while the last element, $\{i_2, ..., i_k\}$, refers to the indices of the structures from step 1. The representation of the resulting structure T_L in the new formats resembles the representation in Table 11, adding a rhetorical relation: e.g. “da5: inform {Cause:Reason da4}” (see also Table 8).

Step 4: This step includes the merging of the structures that are built in step 2, without rhetorical link information, and step 3, with rhetorical link information. The difference between the structures is the additional rhetorical link information in the latter structure. Only if the structures are otherwise identical the merge operation will succeed.

Step 5: In this step combinations are formed between all the structures that are constructed so far with the same markable. The operation is defined as follows: $<m, <S,<d_1,$

$\alpha>>> \oplus <m,<S,< d_2, \beta>>> = <m,<S,\{< d_1, \alpha>, <d_2, \beta>\}>>$. After the first four steps there are still structures that have the same markable. The markable 'ehm' for instance may correspond with the communicative functions 'stalling' in the 'time management' dimension and the communicative function 'turn take' in the 'turn management' dimension. In this step these structures, whose markable and sender must be identical, are combined.

Together these five steps result in a set of structures of the following form: $<m, S, A, H, \{<d_1, \alpha_1>, ..., <d_k, \alpha_k>\}>$. Here, α_1 is maximally a quintuple (i.e. five elements): $< \iota, f, Q, \Delta, R_{rh}>$. These elements corresponds to the contents of a cell in one of the nine dimension columns in the DiAML-MultiTab representation format. Table 11 and Table 8 include examples of the representation of these elements in a cell.

3.3.3 Mapping to DiAML-TabSW.

The function F_{TabSW} includes the mapping of annotation structures to a DiAML-TabSW representation and also consists of five steps. Again, the starting point is an annotation structure composed of entity structures and link structures: $<\{\varepsilon_1, ..., \varepsilon_n\},\{L_1, ..., L_k\}>$.

Step 1: Identical to step 1 of $F_{MultiTab}$. The result of this step is a set of entity structures with indices: $E = \{<\varepsilon_i, 1>, ..., <\varepsilon_j, n>\}$

Step 2: This step extracts and transforms contents for cells from the output of step 1: $\text{Te}(<<m,<S, A, H, d, f, Q, \Delta>>, \iota >) = <m,< \iota, <S, A, H, <d, f, Q, \Delta>>>>$. Note the position of the identifier here compared to step 2 in $F_{MultiTab}$. This identifier is given its own column in the DiAML-TabSW format. Again, the right-hand side already largely resembles a representation in the DiAML-TabSW format. Table 12 shows a representation of such structures. The elements identifying the dimension, communicative function, possible qualifiers, and possible dependence relations of each entity structure are specified in one cell. As mentioned, the identifier is specified in a separate column.

Table 12

Entity Structure Elements Represented in DiAML-TabSW Format

Elements	*m*	*ί*	*S*	*A*	*H*	*(d, f, Q, Δ)*
Columns	Markables	Da-ID	Sender	Addressee	Other Ps	Dialogue acts
Row	fs4	da5	S	M		Ta:answer [certain] (Fu: da2)

Note. Column 'Other P's' may be empty, as above.

Step 3: The third step includes the addition of link structure information. $T_L(L)$ = <m_1,<ί,<S, A, H, <d, f, Q, Δ<p, p_1,{ $ί_2$, ..., $ί_k$}>>>>>.

Step 4: The merging of structures with and without rhetorical link information.

Step 5: In this step structures with the same markable are combined: <m_2,< $ί_1$<d_1, f_1, Q_1, $Δ_1$, R_1>>> ⊕ < m_1,< $ί_2$<d_2, f_2, Q_2, $Δ_2$, R_2 >>> = <m_1,< $ί_1$, $ί_2$>,< d_1, f_1, Q_1, $Δ_1$, R_1> , < d_2, f_2, Q_2, $Δ_2$, R_2>, *S, A, H*>. Note here that if one markable is assigned more than one dialogue act their identifiers are represented in one cell, as well as their (semantic) content. Take for instance Table 13, where m_1 = fs1, $ί_1$ = da6, $ί_2$ = da7, d_1 = turn management, f_1 = turn take, d_2 = time management, f_2 = stalling, S = P1, A = P2, and H = empty.

Table 13

DiAML-TabSW representation of Markable with Two Dialogue Acts

Elements	*m*	*ί*	*S*	*A*	*H*	(*d, f, Q, Δ, …*)
Columns	Markables	Da-ID	Sender	Addressee	Other Ps	Dialogue acts
Row	fs1	da6; da7	P1	P2		TuM:turnTake; TiM:stalling

Note. Column 'Other P's' may be empty, as above.

3.3.4 Mapping New Formats to Annotation Structures

The functions F_{TabSW}^{-1} and $F_{MultiTab}^{-1}$ decode any DiAML-TabSW and DiAML-MultiTab representation, respectively, constructing the annotation structure that it represents.

In a DiAML-MultiTab representation a row can be viewed as corresponding to the following 13-tuples: $<<m, S, A, H, \{<d_i, \alpha_i>, ..., <d_{9k}, \alpha_{9k}>\}>$. Here d_i denotes one of nine dimensions, and α_i denotes maximally a quintuple: $<\acute{\iota}, f, Q, \Delta, R_{rh}>$, corresponding to the contents of a cell in one of the nine dimension columns. Multiple dialogue acts – in as many dimensions – may thus be anchored to one functional segment. The first step includes separating the structures that have the same markable m. The first step in constructing the annotation structure represented by a given DiAML-MultiTab representation is to take these 13-tuples and in each of them separate the structures that the same functional segment (markable), sender, addressee, and other participants, such as: $< m_1, <S, A, H, <d_1, \alpha_1>>>$ and $< m_1, <S, A, H, <d_2, \alpha_2>>>$. Each of these structures in fact corresponds to one dialogue act, possibly including rhetorical link information. Next, the rhetorical link information is extracted from those structures that contain such information and is used to construct link structures: $<R, da_1, \{ da_2, ..., da\acute{\iota}_k\}>$. Finally, the structures $< m_1, <S, A, H, <d_i, \alpha_i>>>$ with $\alpha_i = <\acute{\iota}, f, Q, \Delta>$ are 'flattened' and reduced to $< m_1, <S, A, H, d_i, f, Q, \Delta>>$, corresponding to entity structures in the abstract syntax.

Similarly, in a DiAML-TabSW representation a row can be viewed as the following n-tuple, where n can be any number between 1 and 9: $<m_1, < \acute{\iota}_1, ..., \acute{\iota}_n>, < d_1, f_1, Q_1, \Delta_1, R_1>, ..., < d_n, f_n, Q_n, \Delta_n, R_n>, S, A, H>$. Note here the n dialogue act structures, and rhetorical relations, anchored to the same markable. The first construction step is to separate the structures that have the same markable, resulting in structures of the form $<m_k, < \acute{\iota}_k <d_k, f_k, Q_k, \Delta_k, R_k>, S, A, H >>$. The second step, as before, extracts the rhetorical link information from those structures that contain such information and use that to construct link structures: $<R, da_1, \{ da_2, ..., da\acute{\iota}_k\}>$. Finally, the various elements that should ultimately be in the entity structures representations are extracted and placed in the order that is used in the entity structures as defined by the abstract syntax.

4. Method

The method section is divided into three parts, each related to one of the three research questions. First, the methods for dealing with the two theoretical problems encountered in using ISO 24617-2 are briefly discussed. Secondly, various aspects of the conversion program are discussed. Finally, in the third section the experiment is discussed including the participants, materials, experimental design, and procedure.

4.1 Dealing with ISO 24617-2 Problems

During the construction of the DialogBank existing annotations have been re-annotated according to the ISO 24617-2 scheme. The existing annotations were often, to some extent, already according to the scheme (see for instance Table 4). Corrections and enrichments in these representation were made, using the two new ISO 24617-2 compliant formats DiAML-MultiTab and DiAML-TabSW. During this process two main limitations of the ISO 24617-2 scheme became apparent, see Section 3.1. In the two following sections the methods used for dealing with these limitations are described. For both, the CASCADES methodology is used, taking into account the ISO 24617-2 abstract syntax, semantics, and concrete syntax(es). The CASCADES model provides methodological support by means of procedures for how to go from one level of decision-making to the next (Bunt, 2013). It can be used for, among other things, the design of new annotation languages and the detection and repair of deficiencies in existing annotation languages. DiAML was developed systematically using this CASCADES approach (see steps in Figure 3), starting with the design of a metamodel (conceptual analysis, see Figure 1) and an abstract syntax. In addition to the design of new semantic annotation languages, feedback cycles in the CASCADES model – including steps back – are also useful when the starting point is an existing representation format, as is the case for the two encountered problems.

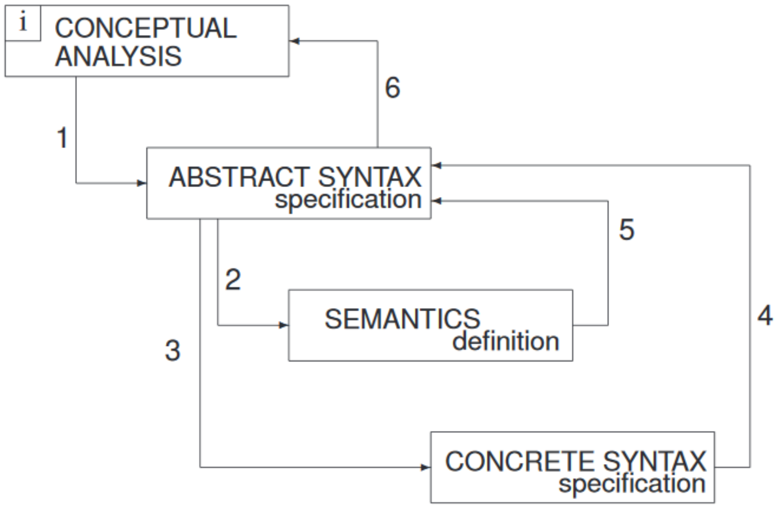

Figure 3. The CASCADES model (Bunt, 2013).

4.1.1 Dealing with Problem I.

In this section the method used for dealing with the feedback antecedent problem (see Section 3.1.1) is described. There are feedback dependence relations where the feedback antecedent is a functional segment or a dialogue act, mainly for feedback on higher levels (e.g. agreement). However, there are also feedback dependence relations where the feedback antecedent is a stretch of primary data, mainly for feedback on lower levels (e.g. mishearing). Currently, ISO 24617-2 does not allow correct annotation of these latter cases.

In the current thesis a solution to the problem is proposed which involves step 4 and step 3 of the CASCADES model, where changes to the concrete syntax require adaptations in the underlying abstract syntax. The proposed solution, explained in detail in Section 5.1.1, requires adjustments to the ISO 24617-2 representation formats and the abstract syntax. The semantics of the scheme are not affected by the solution.

4.1.2 Dealing with Problem II.

In this section the method used for dealing with the inadequate annotation of rhetorical relations is described. Two solutions to this problem are proposed in Section 5.1.2 and Section 5.1.3, respectively. The first proposed solution is rather straightforward and involves only a small change to the DiAML representation formats, while requiring no adjustments to the abstract syntax or semantics. The second proposed solution is more complex as it involves the integration of the annotation languages DiAML (ISO 24617-2) and DRelML (ISO DR-Core). It affects the concrete syntax, the abstract syntax, and the semantics of ISO 24617-2. Therefore, this solution not only requires the cycle of the CASCADES model formed by steps 4 and 3, but also the 'inner cycle': step 2 and step 5, to maintain semantic adequacy. As mentioned before, the elements of an annotation are seen as structures of which the meaning is determined by their effect on the information state(s) of the addressee(s). This applies to dialogue acts, but also to rhetorical relations. One of the few assumptions on information states is that all dialogue participants maintain a so-called 'dialogue history', which is an ordered registration of the dialogue acts in a dialogue. A rhetorical relation establishes a semantic link between two elements in the dialogue history, i.e. between two dialogue acts. These links are added to the dialogue history. As mentioned, the first proposed solution does not affect the abstract syntax and semantics. It allows for more explicit determination and storage of rhetorical relations between dialogue acts in the dialogue history.

However, this solution disregards, among other things, the inability of ISO 24617-2 to distinguish between pragmatic and semantic versions of rhetorical relations. This is because ISO 24617-2 only takes into account rhetorical relations between dialogue acts, and whether these relations are between dialogue acts as a whole or between the semantic contents of the dialogue acts cannot be expressed in ISO 24617-2 annotations. To be able to annotate this distinction it should be possible to say something about the semantic contents of dialogue acts. So far, the only thing that can be said about the semantic contents of dialogue acts is the dimension. To improve on this an integration of DiAML and DRelML is proposed in the current thesis. As will be explained in more detail, DRelML has variables that allow the expression of situations – with events, states, facts, and propositions as subclasses – that ideally need to be linked as semantic content to dialogue acts. For this to become a reality, the concrete syntax, abstract syntax and semantics of both DiAML and DRelML will need to be combined. The proposed integration of the DiAML and DRelML concrete syntaxes and abstract syntaxes is discussed and exemplified in Section 5.1.3.

4.2 Conversion Program

4.2.1 Objectives.

The conversion program that has been created allows for conversions between the DiAML-MultiTab, DiAML-TabSW, and DiAML-XML representation formats. As mentioned, an important objective of the conversion program was to demonstrate and prove that the DiAML-MultiTab and DiAML-TabSW representation formats are ideal representation formats - like DiAML-XML – by enabling meaning-preserving conversions of any DiAML representation format to any other DiAML representation format via the abstract syntax. Moreover, as the construction of the DialogBank progressed, another important objective became to provide annotations in not just the DiAML-XML representation format, but also in the DiAML-MultiTab and DiAML-TabSW representation formats, giving users of the DialogBank the option to choose their desired representation format(s). As a result this allows, for instance, for comparisons between representation. Both of these goals have been achieved, with multiple dialogues from the Switchboard and TRAINS corpora, as well as the DIAMOND, OVIS, and Map Task NL corpora already available in the DialogBank in all three DiAML representation formats after successful conversions using the conversion program.

A more long-term goal is to implement an online facility in the DialogBank that allows users to directly convert an annotation representation in one of the DiAML formats into a representation in one of the other DiAML formats. This would potentially allow users to write annotations in, for instance, the DiAML-MultiTab format and convert it to DiAML-XML – for instance to make the annotation(s) usable for particular computer applications – directly in the DialogBank. For this to become a reality various technicalities with respect to the implementation of such an online facility in the DialogBank website will have to be taken into account. Currently, however, users can still get their annotation(s) converted by contacting the DialogBank or running the program's script themselves.

4.2.2 Python.

The script of the conversion program has been written in the Python programming language (version 3.5) and is comprised of multiple Python 'functions' that are called depending on the conversion that is chosen. Currently, the program can be executed in any Python interpreter. In total, six conversions can be distinguished. Since the DiAML abstract syntax acts as interlingua between the three representation formats – to be able to prove their ideal nature – these conversions are: DiAML-MultiTab to the abstract syntax, DiAML-TabSW to the abstract syntax, DiAML-XML to the abstract syntax, the abstract syntax to DiAML-MultiTab, the abstract syntax to DiAML-TabSW, and the abstract syntax to DiAML-XML. For more on the principle of using the abstract syntax as interlingua see Section 2.2.2 and in particular Figure 2.

The conversion program requires input from the user and produces output for the user. When the input annotation is in DiAML-MultiTab or DiAML-TabSW format the program requires as input the respective annotation in tabular format and the level-1 and level-2 files specifying the primary data and segmentation of this primary data, respectively. When the input annotation is in DiAML-XML format the program only requires the annotation in DiAML-XML format as input. Based on the conversion the program then encodes the input data into an annotation structure according to the DiAML abstract syntax consisting of entity structures and link structures. Subsequently, the program decodes these structures – depending on the conversion - into an annotation representation in one of the other formats. In the Results section the conversion program is discussed in more detail, including, among other things, a description of the program's script.

4.3 Experiment

The following sections describe various aspects of the experiment that has been conducted to compare the representations formats, including among other things, the participants, materials, design, and procedure.

4.3.1 Participants.

47 students from Tilburg University (33 women and 14 men) ranging in age from 18 to 28 years old (mean age = 20.91) participated in the experiment. Mastering the Dutch language and English language was a prerequisite for participation. All participants signed up via the university's participant's pool ('Proefpersonenpool') – at the end of their study a certain amount of credits are required – and received one credit for their participation. Each participant was assigned to one of three conditions; DiAML-MultiTab, DiAML-TabSW, or DiAML-XML. The participants were sequentially assigned to the conditions; the first participants to condition 1, second participant to condition 2, third participant to condition 3, fourth participant to condition 1, and so forth. Consequently, the participants were distributed over the three conditions as follows; 16 participants in both the DiAML-MultiTab and DiAML-TabSW conditions, and 15 participants in the DiAML-XML condition.

4.3.2 Materials.

A dialogue from the Switchboard corpus – dialogue "sw01-0105" – was selected as material for the experiment (also available in the DialogBank). The length of the dialogue was reduced by omitting some sentences. Also, some changes were made to sentences that were in the final version of the dialogue. This resulted in a customized transcript of the dialogue, see Appendix C or https://git.io/vrcjv.

The customized dialogue was annotated according to the ISO 24617-2 scheme in the DiAML-XML, DiAML-MultiTab, and DiAML-TabSW representation formats. Since it was expected that participants had no prior experience with dialogue annotation (and ISO 24617-2) and an important goal was to avoid information overload and confusion, the annotation representations were adjusted by changing names of concepts, merging dimensions, and omitting some of the information in the formats. Ultimately, the information present in the three annotations was identical, see https://git.io/vrcj3 (DiAML-XML), https://git.io/vrcjW (DiAML-MultiTab), and https://git.io/vrcjB (DiAML-TabSW).

In addition to the three annotation representations, a 'level-1' and a 'level-2' document was created for the DiAML-MultiTab and DiAML-TabSW conditions. The level-1 document contained all verbal and nonverbal behaviour that occurred in the dialogue combined with the number of their occurrence, see https://git.io/vrVv2. The level-2 document contained all dialogue segments and the respective numbers of the words and nonverbal behaviour that constituted these segments, see https://git.io/vrVvV. Note that in a DiAML-XML annotation these levels are included in the DiAML-XML annotation file (see Section 2.2.4).

Furthermore, for each representation format a document was created that contained information on (dialogue) annotation in general and the annotation representation format. Again, an important goal was to avoid information overload and confusion. Therefore, the information was presented in a clear and concise way. For DiAML-XML see Appendix D, https://git.io/vrcjp (Dutch) or https://git.io/vrVfT (English), for DiAML-MultiTab see Appendix E, https://git.io/vrCev (Dutch) or https://git.io/vrVvp (English), and for DiAML-TabSW see Appendix F, https://git.io/vrCek (Dutch) or https://git.io/vrVfv (English). The participants received the Dutch versions of these documents.

Finally, three surveys were created (in Dutch) – one for each condition – with 'Qualtrics', a survey service developed particularly for research purposes. The surveys, however, essentially contained the same information and questions. The first difference was that in the DiAML-XML condition participants were asked whether they had any prior experience with XML (or HTML). The second and last difference was related to the naming of the annotation formats in questions about, among other things, their experience with working in their respective annotation format; "Tabular format" vs "XML format".

The survey (singular, as the three surveys were similar except for above differences) was structured as follows. The first part contained questions on various characteristics of the participants; gender, age, highest level of education, and whether they had any prior experience with annotations (and with XML/HTML in DiAML-XML condition). Then, the surveys included a small text indicating that two steps were supposed to be carried out before advancing to the next page; (1) reading the dialogue transcript and (2) reading the general information. Moreover, the text indicated that the participants were provided with a total of twenty questions (one-by-one; one question on each page) and that for some of the questions it was necessary to take into account all three levels.

There were nineteen multiple-choice questions (four possible answers) and one open question. A so-called 'timing question' was added after each of the 20 questions, allowing the capturing of, among other things, the time that each participant spent on each of the questions (time spent on each page before advancing to the next page). These 'timers' were not visible to participants. The first fifteen questions were specific search questions about the annotation, including all three levels. The first few questions were relatively easy to let the participants get some familiarity with the annotation document(s) (e.g. "How many dialogue segments can be found in the annotation?" and "Which dimension is annotated at dialogue act 14?"). More difficult questions included those that required multiple levels to be taken into account, for instance: "Which function belongs to the dialogue segment that consists of the following word numbers: w78,w80,w82,w83,w84?" and "Dialogue act 14 has a feedback dependence relation with a previous dialogue act. Which word numbers together make up the dialogue segment that belongs to this previous dialogue act?". The last five questions were about inaccuracies (errors) in the dialogue annotation which were deliberately included, for instance: "What is the error in the annotation of dialogue act 87?" and "The annotation of dialogue act 24 is incomplete. What is missing?". For each participant a 'total score' on a 0-100 scale based on all questions, (each question being worth five points: 10 correct questions resulted in a score of 50, 16 correct question resulted in a score of 80, and so forth), a 'search score' based on the answers to the search questions (Q1-Q15) on a scale of 0-15, and an 'inaccuracy score' based on answers to the inaccuracy questions (Q16-Q20) one a scale of 0-5 were calculated.

Finally, the survey contained 15 statements; 5 statements about one's experience while searching for the answers to the 20 questions ('the search statements'), 5 statements about the XML or tabular format specifically ('the format statements'), and 5 statements about the three levels ('the levels statements'). Participants were asked to indicate their level of agreement with the statements on a 7-point Likert scale (strongly disagree – strongly agree). Statements were taken from the "System Usability Scale" (Brooke, 1996), also known as the "SUS survey", and the "Post-Study Usability Questionnaire" (Lewis, 1995), also known as the "PSSUQ Survey". Some of these statements were slightly modified (positive to negative and vice versa) to fit in with statements that were specifically designed for the current research. See Appendix G or https://git.io/vrVTG for the questions and statements of the DiAML-XML survey in English, and https://git.io/vrVtW for the DiAML-XML survey in Dutch (the surveys for the other conditions can be derived from this with the earlier mentioned differences).

Reliability analyses were carried out and three subscales were found: the 'search subscale', the 'format subscale', and the 'levels subscale'. Prior to these analyses reverse-worded items were reverse-scored (i.e. responses to negatively worded items were reversed). The search subscale showed a Cronbach's alpha of .704 (N = 5) which could be increased to .722 by deleting the second statement ("… I sometimes got disoriented"). The format subscale showed a Cronbach's alpha of .781 (N = 5) which could be increased to .903 by deleting the first statement ("The … format was unclear"). The levels subscale showed a Cronbach's alpha of .791 (N = 5) which could be increased to .819 by deleting the third statement ("… which level to search").

The surveys and dialogue annotations (XML format and tabular formats) used in the experiments were presented to the participants digitally. The other documents were presented on paper.

4.3.3 Design and Procedure.

The experiment included a between subjects design; a participant participated in only one of the three conditions (see also Section 4.3.1). The three conditions and more specifically the three corresponding representation formats; DiAML-XML, DiAML-MultiTab, and DiAML-TabSW, were the independent variables in the experiment (more precisely, representation format as the independent variable with three levels). The survey was designed with three main objectives in mind. First, to examine the effect of the type of representation format on the ability to satisfy information retrieval needs, in terms of both number of search questions answered correctly and time spent on these questions. Secondly, to examine the effect of the type of representation format on the ability to spot and correct annotation inaccuracies, again in terms of both number of questions answered correctly and time spent on these questions. Finally, to examine the effect of type of annotation format on experience during the answering of the questions, the experience of working with the XML or tabular formats, and the experience of working with the three-level architecture.

The procedure was as follows. First, all participants were asked to read a consent form including, among other things, the goal, duration, compensation, and voluntary nature of the experiment, and if they were willing to participate to sign the form (see https://git.io/vrchJ (Dutch), https://git.io/vK3it (English) or Appendix H (English)). Then, the participants were provided with the documents belonging to their condition. In the DiAML-XML condition these included (1) the DiAML-XML annotation, (2) the general information on annotations and the

DiAML-XML format, and (3) a scratch paper. In the DiAML-MultiTab and DiAML-TabSW conditions these included (1) the DiAML-MultiTab or DiAML-TabSW annotation, (2) general information on annotations and their respective format, the (3) level-1 and (4) level-2 documents, and (5) a scratch paper. Participants were then instructed to complete the survey.

Participants were allowed to ask questions to the researcher. The researcher, however, decided per case if and to what extent the question was answered. This was mainly done to avoid any potential negative or undesired effects on the results of the experiment. In this respect it is also relevant to note that only two participants asked a question about any of the materials during the experiments. While reading the consent form both participants indicated that they were unfamiliar with the term ‘annotation’, which was then briefly explained to them. It took most participants 30 to 40 minutes to complete the experiment, from reading and signing the consent form to answering the last question of the questionnaire.

4.3.4 Pre-test.

A pre-test was conducted (N = 3) which primarily functioned as a check for the three surveys and other materials in the experiment. Some valuable results were yielded. First, an error was found in the second-level document as it contained 106 instead of 107 dialogue segments; missing the specification of one segment. This was corrected afterwards. Secondly, the general information documents were slightly adjusted. Initially, the columns in the tabular formats were described as column 1, column 2, columns 3, et cetera. One participant suggested to change the numbers to letters (i.e. column A) as this would reflect existing Excel column ‘titles’. This was adjusted to be more clear and avoid potential confusion. Finally, one participant suggested to emphasize the fact that it was allowed to use the general information document at any time during the experiment. Therefore, a sentence was added to ‘step 2’ in the survey (“Step 2: Read and study the ‘general information’. You can consult this information during the entire experiment."). Finally, the pre-test showed that it took the participants approximately 35 minutes to complete their experiment.

4.3.5 Data Analysis.

The research data was collected with 'Qualtrics' (see also Section 4.3.1). However, all data was exported to and analysed with 'SPSS' (Statistical Package for Social Sciences) version 23. For all participants the answers to the questions, the time spent on the questions, and the responses to the 15 statements have been analysed. In Section 5.3 the results of the statistical analyses that have been performed are presented.

5. Results

The result section is, similar to the method section, divided in three parts. In Section 5.1 the solutions to the feedback antecedent problem and the problems regarding the annotation of rhetorical relations are described. In Section 5.2 various aspects regarding the conversion program are explained. Finally, the statistical analyses of the data from the experiment are presented in Section 5.3.

5.1 Solutions to ISO 24617-2 Problems

5.1.1 Solution to Problem I.

The solution to the first problem, which arises when the feedback antecedent of a feedback dependence relation is a stretch of primary data instead of a dialogue act or functional segment, involves an extension of one of the sets in the conceptual inventory of the abstract syntax. The notion of a 'feedback segment' is introduced; a stretch of source text that a dialogue act has a feedback dependence with and that does not form a dialogue act or a functional segment (see also (DB16)). The conceptual inventory of DiAML contains six sets of elements from which annotation structures are built up (see Section 2.2.2). One of these is the *FS* set containing the functional segments that are identified in a given dialogue. A functional segment is a markable: a specification of a stretch of primary data that is relevant to the annotation process. Note that a markable is not the same as ‘a stretch of primary data’ as a markable is relevant for an annotation and often contains extra information. In (24) two markables from dialogue ‘sw-02-0224’ from the Switchboard corpus are shown. The first nine symbols represent the name of the particular dialogue. The four numbers that follow stand for the row number in the tabular representation format of the annotation. The sender and turn count (i.e. a count of twenty-six means that the turn has changed twenty-five times) are specified with the last four symbols. Another example of a markable could be a word token produced by a tokenizer that denotes a collection of primary data with a reference to a lexicon entry.

(24) “sw02-0224-0001-A001”
“sw02-0224-0037-B026”

In addition to functional segments, feedback segments are also examples of markables as they are relevant to the annotation of the scope of feedback acts. Moreover, they could be used as antecedents of dialogue acts in the own- and partner communication dimensions. Therefore, a new set called *markables* is introduced that replaces the *FS* set and consists of two subsets: (1) a set of functional segments and (2) a set of feedback segments. Note that both sets are mutually exclusive; a feedback segment is never also a functional segment. How does this affect dependence relations and their antecedents? The antecedent in a functional dependence relation is always a dialogue act. A feedback dependence relation, however, may have a dialogue act or a feedback segment as antecedent. The notion of a ‘functional segment’ remains unchanged; a segment with one or more communicative functions.

This extension of DiAML’s conceptual inventory has an effect on the composition of entity structures, $\varepsilon = <m, \alpha>$, and more directly on dialogue act structures, $\alpha = <S, A, H, d, f, Q, \Delta>$, which are currently (maximally) a 7-tuple. The last element (Δ) consists of one or more dialogue acts (functional dependence or feedback dependence) or functional segments (feedback dependence) that have a dependence relation with the dialogue act in question. In the new situation this last element defines, for functional dependences, one or more dialogue acts, or, for feedback dependences, one or more dialogue acts, functional segments, or feedback segments.

The addition of feedback segments also affects the DiAML concrete syntaxes to some extent. First, in the DiAML-XML format the feedback segments would be created at the second level similar to how functional segments are created (see Section 2.2.4, example (13)). The only two changes are the value of the 'type' attribute from 'functionalSegment' to 'feedbackSegment', and the value of the 'xml:id' attribute from 'fs#' to 'fbs#'. In the third level annotation the 'xml:id' values can be used as identifiers of feedback segments for feedback acts as well as for own- and partner communication management acts. Example (25) contains a simplified DiAML-XML second (a) and third (b) level specification of the following dialogue fragment: "That's where the orange juice factory is … orange factory" (see also (18)). The words are specified as primary data elements at the first level in a separate file (w1 = that's, w2 = where, w3 = the, w4 = orange, w5 = juice, w6 = factory, w7 = is, w8 = orange, w9 = factory). Note the above mentioned difference between the representations of the two functional segments and the feedback segment. Also, note that the XML element 'fs' stands for 'feature structure' (XML attribute-value pairs) and not for 'functional segment' or 'feedback segment' (ISO 24610-1).

The DiAML-MultiTab and DiAML-TabSW formats also require some changes. Again, feedback segments should be identified at level-2. This means that the level-2 file that specifies functional segments in terms of primary data elements should also specify feedbacks segments in terms of primary data elements. Subsequently, in the actual annotation one can refer to these feedback segments. Example (26) contains for the DiAML-MultiTab and DiAML-TabSW formats the second (a) and simplified third (b1 & b2, respectively) level specification of the above dialogue fragment.

(25) a.

```
<fs xml:id=“fs1” type=“functionalSegment”/>
<spanGrp xml:id=“sg1”>
    <span xml:id=“sp1” from=“#w1”/>
    <span xml:id=“sp2” from=“#w2”/>
    <span xml:id=“sp3” from=“#w3”/>
    <span xml:id=“sp4” from=“#w4”/>
    <span xml:id=“sp5” from=“#w5”/>
    <span xml:id=“sp6” from=“#w6”/>
    <span xml:id=“sp7” from=“#w7”/>
<spanGrp/>
<fs xml:id=“fs2” type=“functionalSegment”/>
<spanGrp xml:id=“sg2”>
    <span xml:id=“sp8” from=“#w8”/>
    <span xml:id=“sp9” from=“#w9”/>
<spanGrp/>
<fs xml:id=“fbs1” type=“feedbackSegment”/>
<spanGrp xml:id=“sg3”>
    <span xml:id=“sp4” from=“#w4”/>
    <span xml:id=“sp5” from=“#w5”/>
<spanGrp/>
```

b.

```
<dialogueAct xml:id=“da1” sender=“#u” communicativeFunction=“inform”
    dimension= “task” target=“#fs1”/>
<dialogueAct xml:id=“da1” sender=“#u”
    communicativeFunction=“selfCorrection”
    dimension=“ownCommunicationManagement” target=“fs2”
    feedbackDependence=“fbs1”/>
```

(26) a. fs1 = w1, w2, w3, w4, w5, w6, w7
fs2 = w8, w9
fbs1 = w4, w5

b1. Simplified DiAML-MultiTab

markables	FS text	sender	task	OCM
fs1	that's where the orange juice factory is	u	da1:inform	
fs2	orange factory	u		da2:selfCorrection (Fe:fbs1)

b2. Simplified DiAML-TabSW

markables	FS text	sender	Da-id's	dialogue acts
fs1	that's where the orange juice factory is	u	da1	Ta:inform
fs2	orange factory	u	da2	OCM:selfCorrection (Fe:fbs1)

5.1.2 Solution I to Problem II.

In the current section and Section 5.1.3 two solutions are described to improve the annotation of rhetorical relations. The first solution is the most straightforward one. However, it only deals with the problem related to the argument roles, and only to a certain extent. The second solution is more complex as it includes the integration of the DiAML and DRelML annotation languages. This solution deals with the problem of the annotation of argument roles, and allows for more information on rhetorical relations to be annotated. Moreover, this second solution facilitates the interoperability of SemAF annotations as the languages of the standards ISO 24617-2 and ISO-DR Core are integrated.

The first proposed solution has already been incorporated in most of the DialogBank annotations. The current DiAML structure for the annotation of rhetorical relations is left unchanged. The only change is the use of more fine-grained rhetorical relations by adding the role of that argument of a rhetorical relation that occurs first in the dialogue. The second argument is typically where it becomes apparent in a dialogue that such a relation with a previous dialogue act exists (DB16). In other words, in DiAML the argument role of the dialogue act corresponding to the ‘rhetoAntecedent’ value is included in the name of the relation that is the value of the ‘rhetoRel’ attribute. The annotation in (27) contains the new DiAML-XML representation of the annotation in (22). Note that the only difference between (22) and (27) is that the value ‘cause’ has been replaced by ‘cause_result’.

(27)

```
<diaml xmlns: “http://www.iso.org/diaml”>
<dialogueAct xml:id=”da1” target=”#fs1” sender=”#p1” addressee=”#p2”
        communicativeFunction=”inform” dimension=”task”/>
<dialogueAct xml:id=”da2” target=”#fs2” sender=”#p1” addressee=”#p2”
        communicativeFunction=”inform” dimension=”task”/>
<rhetoricalLink xml:id “da2” rhetoAntecedent=”da1” rhetoRel=”cause_result”/>
<diaml/>
```

If the roles of the dialogue acts would be the other way around – *"(implicit = (because)) High cash positions help buffer a fund when the market falls. Some have raised their cash positions to record levels*" – the value would be 'cause_reason'. In the cases of 'elaboration' and 'asynchrony' relations for instance the values would include 'elaboration_specific' and 'elaboration_broad', and 'asynchrony_before' and 'asynchrony_after'. More generally, every asymmetrical rhetorical relation defined in the ISO DR-Core standard is split into two relations that incorporate one of the two argument roles defined for the relations. In the DiAML-MultiTab and DiAML-TabSW formats the solution would be implemented as in (28). The example contains potential contents of a cell in one of these tabular formats, where dialogue act 7 is the 'result' argument and dialogue act 6 the 'reason' argument.

(28) da7: inform (Cause_Result da6)

In the DiAML abstract syntax, and more specifically in the conceptual inventory, asymmetrical rhetorical relations such as 'cause' and 'elaboration' need to be replaced with 'cause_reason' and 'cause_result', and 'elaboration_broad' and 'elaboration_specific' to specify their argument roles, respectively.

Note that, in anticipation of the ISO DR-Core standard, this is how rhetorical relations have been represented in DialogBank annotations in the DiAML-MultiTab and DiAML-TabSW formats. The limitations of this solution are discussed together with a more complete solution in the next section.

5.1.3 Solution II to Problem II.

While DiAML annotation of a rhetorical relation contains its two arguments, the rhetorical relation itself, and (potentially) the role of the second argument (solution I), DRelML annotation additionally contains (1) the roles of both arguments, (2) the specification of a rhetorical relation as an explicit or implicit one, (3) in case of an explicit relation possibly also the discourse connective, and most importantly (4) the identification of rhetorical relations as either a semantic or a pragmatic rhetorical relation.

Similarly to DiAML, the abstract syntax of DRelML consists of a specification of the concepts from which annotations are built up, and a specification of the possible ways of combining these into annotation structures (see Prasad & Bunt, 2016). An annotation structure in DRelML is also a set of entity structures and link structures, where entity structures contain semantic information about stretches of primary data and link structures describe semantic relations between stretches of primary data. Two types of entity structures are distinguished in DRelML: (1) a 'relation entity structure', which is a pair, $<m_i, r_j>$, composed of a markable and a rhetorical relation, and (2) an 'argument entity structure', which is a pair, $<m_k, t>$, composed of a markable and an argument type (including subtypes such as 'event', 'state', and 'proposition'). A link structure is a triple such as $<\rho, \varepsilon, \alpha>$ composed of a relation entity structure or in case of an implicit relation only a rhetorical relation, an argument entity structure, and an argument role. Such a link structure thus captures an argument's role in a rhetorical relation.

DRelML also specifies an XML-based reference format for its representation. This concrete syntax contains four XML elements: dRel, drArg, explDRLink, and implDRLink. The dRel element is defined to represent rhetorical relations (e.g. 'Cause') while the drArg element is defined to represent the events, propositions, or other types of semantic objects that form the arguments of a rhetorical relation. The last two elements are defined to represent explicit and implicit rhetorical relations. Take the following DRelML annotation examples, illustrating an explicit semantic cause relation (29) and an implicit pragmatic cause relation (30) (Prasad & Bunt, 2016). Note that for (29) m1 refers to "John Fell", m2 refers to the discourse connective "because", and m3 refers to "Bill pushed him". For (30) m1 refers to "Carl is crazy" and m3 refers to “he beats his wife”. The latter example shows a pragmatic relation, as the situation described in the second part is the motivation for the statement uttered in the first part. The beating did not cause Carl’s craziness; but rather it caused the speaker to say/believe that Carl is crazy.

(29) John fell because Bill pushed him.
<drArg xml:id=“a1” target=“#m1” type=“event”/>
<dRel xml:id=“r1” target=“#m2” rel=“cause”/>
<drArg xml:id=“a2” target=“#m3” type=“event”/>
<explDRLink rel=“#r1” result=“#a1” reason=“#a2”/>

(30) Carl is crazy; he beats his wife.
<drArg xml:id=“a1” target=“#m1” type=“dialogue act”/>
<drArg xml:id=“a2” target=“#m3” type=“situation”/>
<implDRLink xml:id=“r1” rel=“cause” disConn=“because”
result=“#a1” reason=“#a2”/>

Note that in the examples the following is specified: (1) the arguments and their markable (‘xml:id’ and ‘target’ attributes in ‘drArg’ elements), (2) the roles that these arguments play (‘reason’ and ‘result’), (3) whether the relation is explicit or implicit (‘explDRLink’ or ‘implDRLink’), and (4) the type of the arguments (‘type’ attributes). The latter attributes are used by ISO DR-Core on the arguments to label the relations as either semantic or pragmatic; when at least one of the ‘type’ attributes has the value ‘dialogue act’ the relation is interpreted as a pragmatic one, in all other cases as a semantic one.

The DiAML language is designed for the annotation of dialogue acts. As a result, DiAML annotations can only contain rhetorical relations between dialogue acts, using dialogueAct and rhetoricalLink elements as in (27). The revised DiAML version in (27) also contains an argument role, which in turn implies the role of the other argument. DiAML on its own is not able to refer to the semantic contents of dialogue acts. DRelML annotations, on the other hand, can represent both semantic and pragmatic rhetorical relations by manipulating the 'type' attributes, see (29) and (30), respectively. However, it has no means to annotate any information about dialogue acts (i.e. it just 'says' type='dialogue act'). Thus, DiAML and DRelML each have their limitations for representing semantic relations and pragmatic relations, respectively. An integration of the DiAML and DRelML languages could potentially eliminate these limitations.

Such an integration could make it possible to represent a semantic cause relation as in (31), where the content of P2's statement mentions a cause for the content of P1's statement. Instead of changing the 'type' attribute, the DiAML-DRelML integration allows for the annotation of semantic relations by referring to the semantic content of dialogue acts. Note that e1 and e2 represent the semantic contents of a1 and a2, respectively. The markable s1 refers to P1's utterance, s2 to P2's utterance, and s3 to the discourse connective "because".

(31) a. P1: I can never find my remote control.
P2: That's because they don't have a fixed place.
b. <dialogueAct xml:id="a1" target="#s1" sender="p1" addressee="p2"
dimension="task" communicativeFunction="inform"/>
<dialogueAct xml:id="a2" target="#s2" sender="p2" addressee="p1"
dimension="task" communicativeFunction="inform"/>
<drArg xml:id="e1" target="#s1"/>
<dRel xml:id="r1" target="#s2" rel="cause"/>
<drArg xml:id="e2" target="#s3"/>
<explDRLink rel="#r1" arg1="#e1" arg1Role="result"
arg2="#e2"arg2Role="reason"/>

Moreover, a DiAML-DrelML integration could represent a pragmatic relation as in (32), where J gives a motivation for his question. This example is more complex than the pragmatic DiAML example in (27). However, note that it contains more information as, among other things, both roles are specified.

(32) J: Is it safe to put my camera through here? Because it's a very expensive camera.
<dialogueAct xml:id="a1" target="s1" sender="j" addressee="k"
dimension="task" communicativeFunction="propositionalQuestion"/>
<dialogueAct xml:id="a2" target="s2" sender="j" addressee="k"
dimension="task" communicativeFunction="inform"/>
<dRel xml:id="r1" target="#s2" rel="cause"/>
<explDRLink rel="#r1" arg1="#a1" arg1Role="result"
arg2="#a2"arg2Role="reason"/>

Similar to the integration of DRelML into DiAML-XML as shown in (32), the DiAML-MultiTab and DiAML-TabSW formats can also be modified. Take example (33), with cell representations of an implicit semantic, explicit semantic, implicit pragmatic, and explicit pragmatic relation, respectively.

(33) da7: inform (imseCause:result da6:reason)
da7: inform (exseCause:result da6:reason)
da7: inform (imprCause:result da6:reason)
da7: inform (exprCause:result da6:reason)

The DiAML-DRelML integration leads, in addition to modified concrete syntaxes, also to a new abstract syntax which combines the DiAML and DRelML abstract syntaxes. First, the integration of both languages leads to a new conceptual inventory. This inventory consists of eleven sets: the six sets specified in the DiAML abstract syntax and the five sets in the DRelML abstract syntax. As mentioned before, DiAML consists of entity structures $\varepsilon = <m, \alpha>$ (where m is a markable and where α is a dialogue act structure $\alpha = <S, A, H, d, f, Q, (\Delta)>$), and link structures $L = <\varepsilon, E, \rho>$ (where ε is an entity structure, E is a set of entity structures, and ρ is a rhetorical relation), while DRelML consists of relation entity structures $<m, r>$ (where m, is a markable and r is a rhetorical relation), argument entity structures $<m, t>$ (where m is a markable and t is an argument type) and link structures $< \rho, \varepsilon, \alpha >$ (where ρ is relation or relation entity structure, ε is an argument entity structure, and α is an argument role).

A DiAML-DRelML abstract syntax defines two kinds of entity structures and a link structure. The entity structures are the 'DA entity structure' $<m, \alpha>$ from DiAML, and the 'Non-DA argument entity structure' $<m, t>$ from DRelML. The link structure $<\rho, < m, DA/AT, \text{role}, >\{< m, DA/AT, \text{role}>\}>$ consists of a relation or relation entity structure, a DA or non-DA entity structure, and a set of DA or non-DA entity structures. Note that in DiAML a rhetorical relation can be between more than two entity structures (e.g. giving more than one reason for the occurrence of a certain event), while in DRelML a rhetorical relation is always binary. In the DiAML-DRelML integration a relation between more than two entity structures is made possible by introducing the latter element in the link structure, the set of DA or non-DA act entity structures. So, by including this set of arguments the richer idea of DiAML is adopted. Moreover, the DRelML idea of annotating the roles that arguments play in a relation is adopted by means of the specification of the roles in the link structures.

5.2 Conversion Program

In the following sections various topics related to the conversion program are explained and discussed. First, all possible steps a user of the program may need to perform – depending on the chosen conversion – are explained. Secondly, the program's script is discussed. This includes the structure of the script and a global overview of all the conversions (see also Section 4.2.1). The individual functions that constitute the script are explained in accompanying comments in the script itself. Finally, a second less-extensive conversion program is introduced that allows the conversion of some XML annotations from the HCRC Map Task and DBOX corpora that deviate slightly from the ISO 24617-2 standard into slightly customized DiAML-MultiTab and DiAML-TabSW representations.

5.2.1 Procedure.

The conversion program first asks the user which conversion should be performed. The user should enter '1' for DiAML-MultiTab to DiAML-TabSW, '2' for DiAML-MultiTab to DiAML-XML, and so on. Subsequently, the user is asked to enter the path to the location of the annotation file, and if the input format is DiAML-MultiTab or DiAML-TabSW and the output format is DiAML-XML to enter the paths to the level-1 and level-2 files. In case anything goes wrong – for instance finding the particular files or parsing the files – the user will be instructed how to proceed. Finally, the user is asked to enter the name of the dialogue (e.g. 'Dialogue_one'). This name will be used to name the output file(s) of the conversion. If the user has reached this point, his or her work is done. The rest of the work is done by the program. How the program executes these conversions is described in the following section.

5.2.2 Script.

The script can be found at https://git.io/v6Dsv. The script includes a main function that is executed whenever the program is run. Depending on the conversion that was chosen the main function calls other functions that are included in the script. These functions can be divided into six groups. The first three groups correspond to the mappings of DiAML-MultiTab, DiAML-TabSW, and DiAML-XML representations to the DiAML abstract syntax. The three other groups of functions correspond to the mappings of the DiAML abstract syntax to DiAML-MultiTab, DiAML-TabSW, and DiAML-XML representations. The structure of the script follows this division.

First, the script includes a number of general functions that are included in a number of other functions. Next, in order, the script includes the functions that enable the DiAML-MultiTab to abstract syntax, DiAML-TabSW to abstract syntax, DiAML-XML to abstract syntax, abstract syntax to DiAML-MultiTab, abstract syntax to DiAML-TabSW, and abstract syntax to DiAML-XML mappings. These mappings are discussed below at a fairly high level, meaning that individual functions are not discussed. Again, note that in the script the functions are accompanied with explanatory comments. As mentioned before, all six conversions between the three representation formats use the abstract syntax as interlingua. A representation of such an abstract syntax looks – in Python language – as in (35), which shows the abstract annotation structure of the dialogue fragment in (34).

(34) Fs1: A. Hey.
Fs2: A. Are you going on holiday?
Fs3: B. Yes I am!
Fs4: B. The 26th.

(35) ([{'markable': 'dialogue1-fs1', 'sender': 'A', 'addressee': 'B', 'other Ps': 'NA', 'communicativeFunction': 'opening', 'dimension': 'discourseStructuring', 'qualifiers': ['NA', 'NA', 'NA'], 'dependences': ['NA', 'NA']}, {'markable': 'dialogue1-fs2', 'addressee': 'A', 'sender': 'B', 'other Ps': 'NA', 'communicativeFunction':'propositionalQuestion', 'dimension': 'task', 'qualifiers': ['NA', 'NA', 'NA'], 'dependences': ['NA', 'NA']}, {'markable': 'dialogue1-fs3', 'sender': 'B', 'addressee': 'A', 'other Ps': ‘NA’, 'communicativeFunction': 'answer', 'dimension': 'task', 'qualifiers': ['NA', 'NA', 'excitement'], 'dependences': ['da2', 'NA']}, {'markable': 'dialogue1-fs4', 'sender': 'B', 'addressee': 'A', 'other Ps': 'NA', 'communicativeFunction': 'inform', 'dimension': 'task', 'qualifiers': ['NA', 'NA', 'NA'], 'dependences': ['NA', 'NA']}, [{'rel': 'Elab:Specific', 'rhetoRelatum': ['da3'], 'rhetoDact': 'da4'}])

(36) <dialogue-fs1, <A, {B}, {}, opening, discourseStructuring, {}>>
<dialogue-fs2, <A, {B}, {}, propositionalQuestion, task, {}>>
<fialogue-fs3, <B,{A},{}, answer, task,{excitement},{<sw02-0224-fs2, <A,{B}, {}, propositionalQuestion, task, }>>}>>
<dialogue-fs4, <B, {A}, {}, inform, task, {}>>
<<dialogue-fs4, <B,{A},{}, inform, task,{<fialogue-fs3, <B,{A},{}, answer, task, {excitement},{<sw02-0224-fs2, <A, {B}, {}, propositionalQuestion, task, }>>}>>
}>>, {}, Elab:Specific>

The example in (35) consists of four entity structures and one link structure, that contains information on a semantic link between the dialogue acts in two entity structures. The first 'python list' – between the first pair of square brackets – stores the entity structures, with each entity structure stored in a 'python dictionary' (between curly brackets), separated by commas. The 'dictionary keys' all have 'values' attached to them (i.e. 'key': 'value'). The keys denote the entity structure elements: the markable, sender, addressee, possible other participants, communicative function, dimension, possible qualifiers, and possible dependences. Note that the 'dependences' and 'qualifiers' keys have lists as values. The values of the 'dependences' key point towards another entity structure. The values of the 'qualifiers' key are none ('NA'), or 'certain'/'uncertain', 'conditional'/'unconditional', or a sentiment value (e.g. 'happiness'), respectively. The second python list – between the second pair of square brackets – stores the link structure(s), with each link structure stored in a python dictionary. The keys in these dictionaries denote the rhetorical relation ('rel' key:value), the dialogue act under consideration ('rhetoDact' key:value), and the rhetorically related dialogue act(s) ('rhetoRelatum' key:value). The 'rhetoDact' key has an entity structure identifier as value, while the 'rhetoRelatum' key has a list with one or more of these identifiers as value. Note that both of these values, as well as the values of the 'dependences' key, are entity structures. To facilitate the conversions it was decided to instead make use of the identifiers of these entity structures. In the entity structures there are no such identifiers; they are created during the encoding of the abstract syntax in any of the DiAML representation formats, using the ordering based on the entity structures' markables and dimensions. This means that, formally, the 'da' values in the abstract syntax always are the intended entity structures. A Python version of the abstract syntax as above stores, essentially, the same information as would a formal – Python independent – version of the abstract syntax as in (36).

Before the program reaches this point – where the entity and link structures are captured in the abstract syntax – however, the DiAML-MultiTab, DiAML-TabSW, or DiAML-XML input annotation data has been processed and transformed into this intermediate Python format by means of various functions. These functions are part of one of the decodings of the DiAML representation formats to the abstract syntax. The first two decodings have some similarities and are discussed first, followed by the decoding of DiAML-XML to the abstract syntax, and the encodings of the abstract syntax to the DiAML representation formats.

Decodings 1 & 2: DiAML-MultiTab & DiAML-TabSW to Abstract syntax.

For these decodings the user provides the program with an annotation representation in of the two formats and the level-1 and level-2 files. The tabular annotation is then read into a 'Pandas DataFrame' which is a particular data structure from the python module called 'Pandas'. These DataFrames allow for working with two-dimensional arrays, such as the DiAML-MultiTab and DiAML-TabSW structures of rows and columns. The data in the DataFrame columns – which mirror the columns of the DiAML-MultiTab or DiAML-TabSW input annotation – are then stored in separate python lists. An initial list of entity structures is created, with each entity structure stored in a python dictionary with key and value pairs such as 'sender':'p1', 'addressee':'p2', 'task':'da4:answer [certain] {Fu: da3)', when the input is in DiAML-MultiTab format, or 'sender':'p1', 'addressee':'p2', 'dactID':'da4', 'dacts':'answer [certain] (Fu: da3)', when the input is in DiAML-TabSW format ('dactID' and 'dacts' store the data from the 'Da-ID' and 'Dialogue acts' columns, respectively). Notice that – at this point – the entity structure list still resembles the rows in the DiAML-MultiTab and DiAML-TabSW annotations. After this, the key and value pairs that ultimately should be part of the entity structures and link structures in the abstract syntax are extracted by applying transformations to the data in the dictionaries by means of various functions. Remember to consult the script and accompanying comments to learn more about the specific roles of the functions that perform these transformations.

Decoding 3: DiAML-XML to Abstract syntax.

First, a DiAML-XML file is parsed using the 'ElementTree' module, which is often used to parse XML files. It views an XML file as a tree with each XML element representing a single node in the three. The output annotation file is either in DiAML-MultiTab or DiAML-TabSW format, meaning that a level-1 file and a level-2 file need to be constructed. A number of functions are designed specifically for this purpose. The data required to create these files is extracted from the first and second "<div>" XML elements in the DiAML-XML file. The elements that should be part of the entity structures in the abstract syntax are retrieved from the <dialogueAct> XML elements, while the elements in the link structures are retrieved from the <rhetoricalLink> XML elements. Again, various functions are designed to make sure that the abstract syntax output looks as in (35).

Encodings 4 & 5: Abstract syntax to DiAML-MultiTab & DiAML-TabSW.

The abstract syntax is first read into two Pandas DataFrames: one that stores the entity structures and one that stores the link structures. The subsequent encoding functions all make use of specific Pandas code. This means that all transformations to the data as a result of these functions – for instance combining the elements from the link structures with the correct elements in the entity structures (e.g. "da4:inform{elab:specific}") – are performed while the data is stored in these Pandas DataFrames. Since the 'Turn transcription' and 'FS text' columns – whose primary goal is to improve readability – and the 'Comments' column are not part of the abstract syntax, when the input format is DiAML-MultiTab and the output DiAML-TabSW, or vice versa, the input versions of these columns are inserted into the DataFrame. If the input annotation is in DiAML-XML format these columns are constructed based on the level-1 and level-2 data in the DiAML-XML annotation. When the functions have transformed the

DataFrame to the point that it is basically a mirror image of the final output annotation, the DataFrame is converted into an Excel (.xlsx) format using the Pandas module together with the XlsxWriter module. This latter module allows additional formatting of the output file, such as setting the start row, inserting a title specifying the dialogue name and representation format, and making the column headers bold.

Encoding 6: Abstract syntax to DiAML-XML.

For this particular encoding the program makes use of the lxml module, which is based on the ElementTree module and works well for writing XML (and parsing XML). The keys and values in the abstract syntax are used to create the <dialogueAct> and <rhetoricalLink> XML elements, which represent the level-3 DiAML annotation. The level-1 and level-2 representations in a DiAML-XML file are constructed based on the level-1 and level-2 files whose paths are provided by the user. First, a so-called root element is created ("TEI" element) which encloses all other elements. This element is also called the parent element and has various child elements – "profileDescr" and "text" – which have child elements themselves, and so forth. After creating these elements the data that belongs inside these elements (e.g. XML attributes) is added.

5.2.3 HCRC Map Task and DBOX.

Currently, three dialogue annotations from the HCRC Map Task corpus and five dialogue annotations from the DBOX corpus are present in the DialogBank. These annotations are all in DiAML-XML format, however they do not comply with ISO 24617-2 in some respects, making them unsuitable for conversion into either DiAML-MultiTab or DiAML-TabSW via the conversion program as described.

According to the ISO 24617-2 standard, feedback antecedents in feedback dependence relations should either be a dialogue act or functional segment, as discussed. In both the Map Task and DBOX annotations functional segments are introduced for feedback antecedents, even if these segments are not assigned any communicative function. The introduction of segments in cases where feedback antecedents are a non-functional stretch of primary data is not wrong, and even desirable. In this thesis the notion of a feedback segment is proposed for this purpose (see Section 5.1.1). Currently, ISO 24617-2 allows only the use of functional segments and dialogue acts as feedback antecedents. Therefore, this is also how the conversion program works: a feedback antecedent can only be a dialogue act or an actual functional segment. In annotations of both corpora additional functional segment names are introduced if more than one dialogue act is anchored to a segment, e.g. "fsp1TUMCV0" and "fsp1DISCV0" for a functional segment of participant 1 with dialogue acts in the 'turn management' and 'discourse structuring' dimensions. This is not incompatible with the ISO standard, but it does mean a complication for XML to abstract syntax decoding, since one object in the abstract annotation structures (a markable) may not have multiple names in the XML representations. Finally, annotations in both corpora make use of dimensions – 'contact management' (both corpora) and 'task management' (DBOX) – that are not specified by ISO-24617-2.

Taking into account these deviations and extensions compared to ISO 24617-2 a second conversion program has been created particularly aimed at these Map Task and DBOX dialogue annotations, allowing for the conversion of these original annotations into equivalent DiAML-MultiTab and DiAML-TabSW representations. The structure of the script and most of the functions used in the script are similar to those in the main program, so these will not be discussed here.

The DiAML-TabSW and DiAML-MultiTab formats have been slightly adjusted to enable the representation of the information in the DiAML-XML annotations with additional dimensions. All functional segments and other segments (feedback antecedents) that are specified in the input annotations are included in the level-2 segmentation file as output of the conversion. Moreover, in both tabular formats, all named segments that are assigned a dialogue act are placed in the 'Markables' column. When two or more of these functional segments consist of the same primary data they are separated by a semicolon, similar to how two or more dialogue acts are separated in the DiAML-TabSW format (e.g. "fsp1TUMCV0 ; fsp1DISCV0"). Finally, for dialogue acts in non ISO 24617-2 dimensions the DiAML-MultiTab format includes an extra column ('other') in which these acts are specified, while in the DiAML-TabSW format they – like any other dialogue act – are specified in the 'Dialogue acts' column. In both formats a representation of such a dialogue act consists of its dimension and communicative function, for instance: "contactManagement:ContactIndication".

For an original DiAML-XML Map Task annotation see: https://git.io/v6DsO (dialogue "q1ec6"). To view the dialogue in the adjusted DiAML-MultiTab and DiAML-TabSW formats see: https://git.io/v6DsZ and https://git.io/v6DsC, respectively. More annotations in the three representation formats from the HCRC Map Task and DBOX corpora can be found on the DialogBank website. The script of this second conversion program– with explanatory comments – can be found at https://git.io/v6DsI.

5.3 Experiment

In the following sections the analyses are described of the data collected in the experiment. First, the analyses of the total score, search score, and inaccuracy score (see Section 4.3.2) are presented. Secondly, the analyses of the time spent per question are presented. Thirdly, the analyses of the responses to the fifteen statements are presented. Finally, the (interaction) effects are presented of other variables on the scores, time spent, and the responses to the statements, including: age, education, gender, previous experience with annotations, and previous experience with XML.

5.3.1 Scores.

In this section data analyses are described that are relate to the first hypothesis – more information retrieval needs are expected to be satisfied in the DiAML-MultiTab and DiAML-

TabSW conditions compared to the DiAML-XML condition – and the second hypothesis – more annotation inaccuracies are expected to be uncovered in the DiAML-MultiTab and DiAML-TabSW conditions compared to the DiAML-XML condition – are described. Three analyses are presented examining the effect of representation format on total score (questions 1-20), search score (questions 1-15), and inaccuracy score (questions 16-20). Finally, not directly related to one of the hypotheses, the effect of representation format on correctness of answers is described (i.e. number of correct and incorrect answers per question between conditions).

For each of the three score analyses the assumptions of normality and homogeneity of variance were assessed. The assumption of normality was assessed with the Kolmogorov-Smirnov test, the Shapiro-Wilk test, and the inspection of Q-Q plots and histograms. The assumption of homogeneity of variance was assessed with Levene's test. Violations of the normality assumption were found for both the search scores and inaccuracy scores in the different conditions. In these cases the non-parametric Kruskal-Wallis test was conducted since this test does not require the assumption of normality.

A one-way analysis of variance (ANOVA) was conducted to compare the effect of representation format on total score. The means and standard deviations of the total scores in the three conditions are presented in Table 14. The analysis showed a statistically significant difference between groups; $F(2,44) = 4.71$, $p = .014$, $\omega = .36$. Post hoc comparisons (Tukey HSD) revealed that the mean total score in the DiAML-TabSW condition was higher than in the DiAML-XML condition, $p = .010$. The mean total score in the DiAML-MultiTab condition did not differ from the DiAML-TabSW and DiAML-XML conditions.

Table 14

Total Scores per Condition (Mean and Standard Deviation)

Condition	*M*	*SD*
DiAML-MultiTab	61.88	16.14
DiAML-TabSW	70.63	16.32
DiAML-XML	53.33	14.23

Note. Scores between 0-100

Related to the first hypothesis, a Kruskal-Wallis test was conducted to compare the effect of representation format on search score. Note that these scores are on a scale of 0-15, i.e. minimum score = 0, maximum score = 15 (N.B. a scale of 0-75, similar to the total score, would give identical results). The means and standard deviations are presented in Table 15. The analysis revealed a statistically significant difference between groups; $H(2) = 3.44$, $p = .038$. Post hoc comparisons (pairwise comparisons with adjusted *p*-values) revealed that the mean search score was higher in the DiAML-TabSW condition compared to the DiAML-XML condition, $p = .037$. The mean search score in the DiAML-MultiTab condition did not differ from to the DiAML-TabSW and DiAML-XML conditions.

Table 15

Search Scores per Condition (Mean and Standard Deviation)

Condition	*M*	*SD*
DiAML-MultiTab	10.12	2.53
DiAML-TabSW	11.56	2.50
DiAML-XML	9.20	2.57

Note. Scores between 0-15

Finally, related to the second hypothesis, a Kruskal-Wallis test was conducted to compare the effect of representation format on inaccuracy score. Note that these scores are on a scale of 0-5 (i.e. minimum score = 0, maximum score = 5). The means and standard deviations are presented in Table 16. The analysis revealed an effect between groups that approached statistical significance; $H(2) = 5.99$, $p = .050$.

Table 16

Inaccuracy Scores per Condition (Mean and Standard Deviation)

Condition	*M*	*SD*
DiAML-MultiTab	2.31	1.49
DiAML-TabSW	2.56	1.25
DiAML-XML	1.47	0.99

Note. Scores between 0-5

In addition to analysing the three scores, chi-square tests were conducted to examine the relation between representation format and correctness of answers (i.e. number of correct and incorrect answers per question compared between conditions). Table 17 contains the number of correct answers per question in each of the three conditions. For six of the twenty questions a statistically significant association was found between type of representation format and correctness of answer: question four, $\chi^2 = 6.66$ (2, N = 47), $p = .036$; question six, $\chi^2 = 7.38$ (2, N = 47), $p = .025$; question fourteen, $\chi^2 = 7.10$ (2, N = 47), $p = .029$; question fifteen, $\chi^2 = 13.39$ (2, N = 47), $p = .001$; question sixteen, $\chi^2 = 6.62$ (2, N = 47), $p = .037$; and question eighteen, $\chi^2 = 16.32$ (2, N = 47), $p < .001$. These significant chi-square tests were broken down further by taking into account their respective adjusted (standardized) residuals, which may be interpreted as z-scores (Field, 2013; Test, 2015). For all questions statistically significant differences were found (i.e. adjusted residuals (z) greater than -2/+2). First, more incorrect (z =

2.3) and fewer correct (z = -2.3) answers were given to question four in the DiAML-XML condition than expected. Secondly, more correct (z = 2.7) and fewer incorrect (z = -2.7) answers were given to question six in the DiAML-TabSW condition than expected. Thirdly, more incorrect (z = 2.6) and fewer correct (z = -2.6) answers were given to question fourteen in the DiAML-MultiTab condition than expected. Fourthly, more incorrect (z = 3.6) and fewer correct (z = -3.6) answers were given to question fifteen in the DiAML-XML condition than expected, and more correct (z = 2.4) and fewer incorrect (z = -2.4) answers were given to question fifteen in the DiAML-MultiTab condition than expected. Fifthly, more correct (z = 2.6) and fewer incorrect (z = -2.6) answers were given to question sixteen in the DiAML-MultiTab condition than expected. Finally, more incorrect (z = 4.0) and fewer correct (z = -4.0) answers were given to question eighteen in the DiAML-XML condition than expected.

Table 17

Number of Correct-Incorrect Answers per Question per Condition

Q1-Q10	DiAML-MultiTab	DiAML-TabSW	DiAML-XML	Q11-Q20	DiAML-MultiTab	DiAML-TabSW	DiAML-XML
Q1	15-1	14-2	13-2	Q11	13-3	12-4	9-6
Q2	5-11	11-5	10-5	Q12	9-7	13-3	6-9
Q3	12-4	12-4	8-7	Q13	14-2	14-2	9-6
Q4*	12-4	15-1	8-7	Q14*	7-9	12-4	13-2
Q5	13-3	14-2	14-1	Q15**	11-5	9-5	1-14
Q6*	9-7	15-1	8-7	Q16*	9-7	3-13	3-12
Q7	13-3	16-0	12-3	Q17	9-7	14-2	9-6
Q8	11-5	15-1	12-3	Q18***	10-6	10-6	0-15
Q9	13-3	12-4	11-4	Q19	4-12	7-9	5-10
Q10	5-11	1-15	4-11	Q20	4-12	7-9	5-10

Notes. **p* < .05; ***p* < .01; ****p* < .00. Statistically significant sources of main effect are highlighted.

5.3.2 Time Spent.

The analyses that are described in this section are related to the third hypothesis – information retrieval needs are expected to be satisfied quicker in the DiAML-MultiTab and DiAML-TabSW conditions than in the DiAML-XML condition – and the fourth hypothesis – annotation inaccuracies are expected to be uncovered quicker in the DiAML-MultiTab and DiAML-TabSW conditions than in the DiAML-XML condition– and cover the timing data that was collected with the 'timing questions'. Note that these were not visible to the participants and measured the time spent on each question before advancing to the next question (see also Section 4.3.2). Two analyses were carried out on the timing data. In the first analysis only timing data of questions that were answered correctly were included, meaning timing data of incorrect answers were excluded from this analysis. In the second analysis timing data of all questions are included, answered correctly or incorrectly. The former analysis is described first.

Prior to any statistical tests, the assumptions of normality and homogeneity of variance were assessed for all questions. Again, the assumption of normality was assessed with the Kolmogorov-Smirnov test, the Shapiro-Wilk test, and the inspection of Q-Q plots, and the assumption of homogeneity of variance was assessed with Levene's test. Analysis showed that these assumptions were often violated. The non-parametric Kruskal-Wallis test (with post hoc pairwise comparisons with adjusted *p*-values) was conducted in case of a violation of the normality assumption or both assumptions, since it is does not require the assumption of normality and is a well-known alternative when the assumption of homogeneity of variance is violated. If only the homogeneity of variance assumption was violated the Welch F test was conducted, potentially with Games-Howell for post hoc comparisons. In case of no violations a regular ANOVA was conducted. The following paragraph elaborates on the tests that found statistically significant results. Table 18 shows, among other things, the mean time spent on the twenty questions.

A Kruskal-Wallis test was conducted to evaluate differences in time spent on question one (Q1) among the three representation formats (DiAML-MultiTab, DiAML-TabSW, DiAML-XML). Time spent on this question was statistically significantly affected by representation format, $H(2) = 24.40$, $p < .001$. Pairwise-comparisons with adjusted p-values showed that more time was spent in the DiAML-XML condition compared to both the DiAML-TabSW ($p < .001$, r = .77) and DiAML-MultiTab ($p < .001$, r = .77) conditions. A statistically significant effect of representation format on time spent was also found for: question two (Q2), $H(2) = 13.93$, $p = .001$, with more time spent in the DiAML-MultiTab condition compared to the DiAML-TabSW condition ($p = .001$, r = .64); and question three (Q3), $H(2) = 10.60$, $p = .005$, with more time spent in the DiAML-TabSW condition compared to the DiAML-MultiTab condition ($p = .006$, r = .54). For three questions it was not possible to compare the effect of representation format on time spent with the above tests due to one of the three conditions containing fewer than two cases. Therefore, independent t-tests were conducted for question ten (Q10) (DiAML-MultiTab vs DiAML-XML), question fifteen (Q15) (DiAML-MultiTab vs DiAML-TabSW), and question eighteen (Q18) (DiAML-MultiTab vs DiAML-TabSW) to compare the mean time spent in the two remaining conditions. No statistically significant results were found.

Table 18
Mean Time Spent, Standard Deviation, and Number of Correct Answers for all Questions per Condition

	DiAML-MultiTab			DiAML-TabSW			DiAML-XML		
	M	SD	N	M	SD	N	M	SD	N
Q1***	29.87	17.65	15	26.83	10.02	14	133.95	127.66	13
Q2**	125.04	84.20	5	15.60	8.33	11	42.24	42.57	10
Q3**	42.10	34.03	12	73.82	14.33	12	94.95	89.25	8
Q4	39.60	27.65	12	53.91	62.27	15	105.52	74.41	8
Q5	33.60	18.72	13	25.84	24.18	14	36.63	23.45	14
Q6	95.53	82.77	9	73.82	47.35	15	124.91	134.04	8
Q7	16.52	6.13	13	27.44	38.90	16	45.20	34.60	12
Q8	27.52	9.70	11	42.05	43.16	15	29.79	24.38	12
Q9	85.94	58.96	13	69.36	26.97	12	93.29	50.38	11
Q10	54.76	22.74	5	97.69	0	1	79.60	38.06	4
Q11	35.28	15.14	13	34.92	22.62	12	50.71	35.71	9
Q12	101.07	39.15	9	65.41	57.69	13	85.83	60.71	6
Q13	91.06	45.66	14	94.26	53.10	14	77.71	59.83	9
Q14	35.96	8.57	7	50.21	45.61	12	45.05	28.19	13
Q15	79.89	49.02	11	58.19	15.27	9	286.61	0	1
Q16	75.71	28.46	9	60.96	13.00	3	118.28	93.77	3
Q17	35.79	22.92	9	36.60	19.65	14	23.33	12.03	9
Q18	59.69	35.21	10	93.74	40.46	10	-	-	0
Q19	36.63	24.88	4	72.86	18.58	7	51.37	18.28	5
Q20	26.56	24.72	4	45.32	11.64	7	29.19	18.59	5

Notes. **p*<.05; ***p*<.01; ****p*<.001. Times are in seconds. N shows the number of correct answers for a question in a condition. N max for DiAML-MultiTab = 16, DiAML-TabSW = 16, DiAML-XML = 15. *SD* is zero in case of only one correct answer.

The second analysis, examining the effect of representation format on time spent on questions that were answered correctly or incorrectly, is described in the following. The normality and homogeneity of variance assumptions were assessed as described in the first analysis. Also, the statistical tests that were carried out based on these violations are similar. Table 19 shows, among other things, the mean time spent on the twenty questions.

A Kruskal-Wallis test was conducted to evaluate differences in time spent on question one (Q1) among the three representation formats. Time spent on this question was statistically significantly affected by representation format, $H(2) = 25.51$, $p < .001$. Similar to the previous analysis (on correctly answered questions), pairwise-comparisons with adjusted p-values showed that more time was spent on this question in the DiAML-XML condition compared to both the DiAML-TabSW ($p < .001$, $r = .81$) and DiAML-MultiTab ($p < .001$, $r = .76$) conditions.

A statistically significant effect of representation format on time spent was found for question two (Q2), H(2) = 14.20, p = .001, again with more time spent in the DiAML-MultiTab condition compared to the DiAML-TabSW condition (p = .001, r = .65), and additionally also more time spent than in the DiAML-XML condition (p = .038, r = .45). Compared to the previous analysis, significant effects were also found for question three (Q3), H(2) = 7.23, p = .027, with more time spent in the DiAML-TabSW condition compared to the DiAML-MultiTab condition (p = .027, r = .46); question four (Q4), H(2) = 6.86, p = .032, with more time spent in the DiAML-XML condition compared to the DiAML-TabSW condition (p = .037, r = .45); question seven (Q7), H(2) = 7.40, p = .025, with more time spent in the DiAML-XML condition compared to the DiAML-TabSW condition (p = .034, r = .46); and question seventeen (Q17), H(2) = 7.03, p = .030, with pairwise comparisons showing no statistically significant differences between conditions.

Table 19

Mean Time Spent (and Standard Deviation) for all Questions per Condition

	DiAML-MultiTab		DiAML-TabSW		DiAML-XML	
	M	SD	M	SD	M	SD
Q1***	32.97	21.09	29.37	17.70	139.70	122.96
Q2**	89.64	59.38	33.82	34.14	43.83	43.80
Q3*	52.63	43.80	81.67	36.46	94.52	87.19
Q4*	49.61	31.11	55.06	60.33	108.30	69.43
Q5	46.24	35.58	31.58	34.60	36.27	22.64
Q6	94.69	67.39	87.75	72.09	119.16	107.31
Q7*	17.43	8.11	27.44	38.90	50.22	37.77
Q8	32.80	15.93	45.63	44.09	44.06	41.44
Q9	86.60	53.93	90.04	87.46	89.77	44.24
Q10	65.75	46.37	60.74	47.43	70.77	32.83
Q11	36.06	16.31	41.27	29.71	53.26	38.78
Q12	94.61	49.12	63.39	52.82	100.99	67.31
Q13	85.92	45.10	91.16	50.26	82.09	55.24
Q14	46.73	18.47	54.03	40.30	53.55	35.97
Q15	81.03	45.70	69.92	32.32	101.98	79.21
Q16	75.96	40.93	81.95	28.35	92.30	70.38
Q17*	37.97	20.67	38.25	19.78	24.12	12.18
Q18	72.20	40.50	73.24	43.94	75.53	45.07
Q19	71.62	43.96	73.24	43.94	49.92	30.06
Q20	42.84	35.98	49.49	20.50	38.92	33.61

Notes. *p<.05; **p<.01; ***p<.001. Times are in seconds.

5.3.3 Statements.

The analyses that are described in this section test the effect of representation format on the responses to the statements and are related to the fifth hypothesis, which says that users are expected to be more positive towards the DiAML-MultiTab and DiAML-TabSW formats than to the DiAML-XML format. For each statement the assumptions of normality and homogeneity of variance were taken into account and followed up with statistical tests as described in Section 5.3.2.

Table 20 shows the means and standard deviations of the responses to statements 1-5 in the three conditions. No significant results were found. A Kruskal-Wallis test examining the relation between representation format and response to statement 2 showed a *p*-value that approached significance, $H(2) = 5.55$ $p = .062$.

Table 20

Results (Mean and Standard Deviation) of Statement 1-5 in Each Condition

Statement	DiAML-MultiTab	DiAML-TabSW	DiAML-XML
1. it was easy to find the right answers.	3.00 (1.21)	3.50 (1.32)	2.60 (1.77)
2. I sometimes got disoriented.	4.31 (1.85)	4.38 (1.78)	5.47 (1.80)
3. I sometimes got frustrated.	3.00 (1.67)	3.31 (1.66)	4.20 (1.42)
4. I became better and better in finding the right answers.	4.94 (1.18)	4.13 (1.59)	4.47 (1.81)
5. I thought it was unnecessary complex to find the right answers.	4.38 (1.46)	4.31 (1.66)	5.13 (1.46)

Note: results between 1 and 7.

Next, statistical tests were conducted to measure the effect of representation format on the responses to statements 6-10. Table 21 shows the mean and standard deviation of the responses to these statements in the three conditions. A Kruskal-Wallis test showed a statistically significant difference between groups for statement 8, $H(2) = 9.38$, $p = 0.009$. Post hoc comparisons (pairwise comparisons with adjusted *p*-values) revealed that the responses in the DiAML-TabSW condition were higher than responses in the DiAML-XML condition, p = .009, r = .53. The difference between the DiAML-MultiTab and DiAML-XML conditions was not statistically significant, p = .092. Furthermore, a Kruskal-Wallis test showed a statistically significant difference between groups for statement 9; $H(2)$ = 11.06, p = 0.004. Post hoc comparisons (pairwise comparisons with adjusted p-values) revealed that the responses in both the DiAML-TabSW condition (p = .010, r = .53) and the DiAML-MultiTab condition (p = .013, r = .51) were higher than the responses in the DiAML-XML condition. Finally, a Kruskal-Wallis test showed a statistically significant difference between groups for statement 10: $H(2)$ = 9.43, p = .009. Post hoc comparisons (pairwise comparisons with adjusted p-values) revealed that the responses in the DiAML-TabSW condition higher than in the DiAML-XML condition (p = .009, r = .53). The difference between DiAML-MultiTab and DiAML-XML was not statistically significant, p = .078.

Table 21

Results (Mean and Standard Deviation) of Statement 6-10 in Each Condition

Statement	DiAML-MultiTab	DiAML-TabSW	DiAML-XML
6. The XML format was unclear.	3.88 (1.78)	4.31 (1.45)	4.60 (2.06)
7. The XML format was easy to understand.	4.06 (1.44)	4.87 (1.26)	3.50 (1.95)
8. The XML format was user-friendly.	4.38 (1.54)	4.88 (1.15)	3.13 (1.55)
9. I think I could work productively with the XML format soon.	4.13 (1.46)	4.19 (1.28)	2.60 (1.24)
10. I think that most people quickly master the XML format.	4.25 (1.61)	4.69 (1.49)	2.93 (1.44)

Note. Results between 1 and 7.

Finally, Kruskal-Wallis tests were conducted to measure the effect of representation format on the responses to statements 11-15. Table 22 shows the mean and standard deviation of the responses to these statements in the three conditions. The tests revealed no statistically significant differences between the responses in the three conditions.

Table 22

Results (Mean and Standard Deviation) of Statement 11-15 in Each Condition

Statement	DiAML-MultiTab	DiAML-TabSW	DiAML-XML
11. The presentation of the three levels was clear	4.87 (1.31)	4.25 (1.53)	3.93 (2.15)
12. Finding the right information in the three levels was easy.	3.69 (1.40)	3.75 (1.61)	3.00 (1.36)
13. I thought it was difficult to determine which level to search.	4.25 (1.39)	3.94 (1.84)	4.07 (1.77)
14. The presentation of the three levels was pleasant	4.06 (1.39)	4.13 (1.15)	3.67 (1.80)
15. It was difficult to navigate between the three levels.	3.94 (1.29)	4.25 (1.39)	4.47 (1.46)

Note. Results between 1 and 7.

5.3.4 Other Variables.

Finally, analyses were carried out testing the (interaction) effects of the variables ‘age’, ‘education’, ‘gender’, and ‘previous experience with annotations’ with representation format on total score, search score, inaccuracy score, time spent, and the responses to the statements. No statistically significant results were found for these variables. Concerning the latter variable, note that only 5 of the 47 participants reported that they had previous experience with annotations. Furthermore, there was little variation in the data distribution of the ‘age’ and ‘education’ variables.

Also, the effect of the variable ‘previous experience with XML’ on the three scores, time spent, and the responses to the statements in the DiAML-XML condition was examined, showing some interesting results. Participants with XML experience showed similar search, inaccuracy, and total scores to participants without XML experience. Moreover, time spent on questions showed no statistically significant differences between participants with and without XML experience. Concerning the statements it was found, using independent t-tests, that participants with XML experience (N = 9) (M = 4.38, SD = 1.51) more strongly agreed that the

XML format was easy to understand compared to participants without XML experience (N = 6) (*M* = 2.33, *SD* =1.97), *t*(12) = 2.207, *p* = .048, r = .51 (statement 7); and they more strongly agreed that the XML format was user friendly, *t*(13) = 2.232, *p* = .044, r = .50 (*M* = 3.78 and *SD* = 1.30, M = 2.17 and *SD* = 1.47, respectively) (statement 8). Moreover, participants with XML experience more strongly agreed that they could work productively with the XML format soon (*M* = 3.11, *SD* = 1.27) compared to participants without XML experience (*M* = 1.83, *SD* = 0.75), t(13) = 2.205, *p* = .046, r = .52 (statement 9). Finally, participants with XML experience more strongly agreed that finding the right information in the three levels was easy (*M* = 3.56, *SD* = 1.24) compared to participants without XML experience (*M* = 2.17, *SD* = 1.17), *t*(13) = 2.177, *p* = .049, r = .50 (statement 12). The other statements showed no statistically significant differences.

6. Discussion

6.1 Research Question 1

The first research question – "What theoretical and practical shortcomings of the ISO 24617-2 standard have been discovered during the construction of the DialogBank, and how can these be solved?" – has been answered by presenting two problems that were encountered in using ISO 2461-2 during the construction of the DialogBank and proposing solutions to these particular problems. For the first problem, which emerges when the feedback antecedent of a feedback dependence relation is a stretch of primary data that does not form a functional segment or dialogue act, a solution has been proposed which includes the introduction of "feedback segments". These new segments are defined as stretches of source text which are neither functional segments nor dialogue acts but which are relevant to the annotation process and more specifically to annotating antecedents in dependence relations for feedback, own- and partner communication acts. The effects of a potential introduction of feedback segments on the ISO 24617-2 abstract syntax and concrete representation formats have been demonstrated.

For the second problem, concerning the inadequate ISO 24617-2 annotation of rhetorical relations, two solutions have been proposed. With the first solution the annotation of rhetorical relations is enriched by adding the role of that argument of a rhetorical relation that occurs first in the dialogue. The second solution is more attractive as it additionally allows for making the semantic-pragmatic distinction. As demonstrated, however, this solution is more complex as it requires the integration of the annotation languages DiAML and DRelML and affects, besides the concrete syntax, also the abstract syntax and semantics of ISO 24617-2.

In the current thesis the aim of identifying ISO 24617-2 aspects that require extra attention is to stimulate the process of improving the ISO scheme. The solutions to these problems will hopefully contribute to the improvement of the scheme, potentially by incorporating (aspects of) the solutions to a revised version of the ISO 24617-2 scheme, which is foreseen in 2017.

6.2 Research Question 2

The second research question – "To what extent are the new representation formats, DiAML-MultiTab and DiAML-TabSW, *ideal* representation formats?" – has been answered in the current thesis by means of theoretical mappings and a conversion program. First, by describing the mappings from DiAML annotation structures to representations in the DiAML-MultiTab and DiAML-TabSW format, the completeness of the two tabular formats has been demonstrated. Similarly, their unambiguity has been demonstrated by describing their inverse functions. These mappings indicate that, theoretically, the DiAML-MultiTab and DiAML-TabSW formats are ideal formats.

Secondly, by the construction of a conversion program which successfully allows for automatic meaning-preserving conversions between the DiAML-MultiTab, DiAML-TabSW, and DiAML-XML formats, via the abstract syntax, the one-on-one interoperability and thus the ideal character of the formats has been demonstrated in practice. In addition, where beforehand dialogue annotations were present in the DialogBank in only one of three representation formats, the DialogBank from now on contains for each dialogue their annotations in all three representation formats. This means that the proven inter-convertibility of the three formats allows the user to view the annotations in his or her preferred format. In addition to the conversion program that allows for the conversion of fully ISO 24617-2 compliant annotations, a second conversion program has been introduced that allows for the conversion of HCRC Map Task and DBOX annotations in the DiAML-XML format, which are not fully compliant with the ISO standard, to slightly adjusted DiAML-MultiTab and DiAML-TabSW formats.

6.3 Research Question 3

The third research question - "How are the new representation formats an improvement over the DiAML-XML representation format for users of these formats?" - can be answered on the basis of the results of the experiment. Related to this experiment five hypotheses were proposed.

The first hypothesis – more information retrieval needs are expected to be satisfied in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format – was partially supported. Users of the DiAML-TabSW format achieved higher 'search scores' than DiAML-XML users, indicating that, in line with the hypothesis, more information retrieval needs are satisfied while using the DiAML-TabSW format. Users of the DiAML-MultiTab format, however, did not satisfy more information retrieval needs than DiAML-XML users.

No direct support was found for the second hypothesis, according to which more annotation inaccuracies are expected to be uncovered in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format. However, it must be noted that there was a near statistical significance. It is possible that the study was underpowered to detect the association(s) between the representation format(s) and inaccuracy score, most likely due to the relatively small sample size. Therefore, the fact that the inaccuracy score was highest in users of the DiAML-TabSW format and lowest in users of the DiAML-XML format may be cautiously interpreted as supportive evidence for the second hypothesis. Together, the findings related to the first two hypotheses show support for the DiAML-TabSW format as an improvement over the DiAML-XML format for information retrieval purposes and cautious support for the uncovering of annotation inaccuracies. This is further supported by the total score, a combination of the search score and inaccuracy score, which was significantly higher for users of the DiAML-TabSW format than for users of the DiAML-XML format. Again, the findings do not show direct support for the DiAML-MultiTab format as an improvement over the DiAML-XML. Finally, it is important to highlight that all three scores were highest for users of the DiAML-TabSW format and lowest for users of the DiAML-XML format - with search score and total score being significantly higher in the DiAML-TabSW format than the DiAML-XML format – justifying the assumption that future research with a larger sample size would result in more conclusive positive results for the DiAML-TabSW format and the DiAML-MultiTab format.

For the third hypothesis – retrieval needs are expected to be satisfied quicker in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format – some supportive evidence was found. It was found that users of the DiAML-XML format spent more time than users of both the DiAML-MultiTab and DiAML-TabSW formats on one of the fifteen information retrieval questions to ultimately satisfy the retrieval need. However, it was also found that users of the DiAML-MultiTab format spent more time on one question than users of the DiAML-TabSW format, and users of the DiAML-TabSW format spent more time on one question than users of the DiAML-MultiTab format. These findings show some but no overwhelming support for the third hypothesis. For the fourth hypothesis – annotation inaccuracies are expected to be uncovered quicker in the DiAML-MultiTab and DiAML-TabSW formats compared to the DiAML-XML format – no support was found. For both the third and fourth hypothesis, however, it is important to note that only correctly answered questions were taken into account to test the effect of representation format on time spent. This means that for many questions a limited amount of data was available, negatively affecting the ability to find statistically significant differences between formats. For this reason a second analysis was conducted that took into account time spent on correctly and incorrectly answered questions. It was found that users of the DiAML-XML format spent more time than users of both the DiAML-MultiTab and DiAML-TabSW formats on one of the fifteen information retrieval questions, and more time than users of the DiAML-TabSW format on two of these questions. For one question more time was spent in the DiAML-MultiTab format compared to the DiAML-TabSW format, and for one question more time was spent in the DiAML-TabSW format compared to the DiAML-MultiTab format. Time spent did not differ between formats for any of the five annotation inaccuracy questions. This second analysis shows that users of the DiAML-XML condition spent more time than users of the other formats on three of the information retrieval questions, providing some additional evidence for the third hypothesis.

Also, some supportive evidence was found for the fifth hypothesis – users of the DiAML-MultiTab and DiAML-TabSW formats are expected to be more positive towards the formats compared to users of the DiAML-XML format. The five statements about the users' experience during the answering of the questions showed no significant differences between formats. However, for all five statements the most positive results were found for the DiAML-MultiTab and DiAML-TabSW formats and not in the DiAML-XML format, potentially leading to statistical significance between formats when a larger sample size is used. The five statements about the XML- and tabular formats showed supportive results for the hypothesis. First, the DiAML-XML format was experienced as more unclear than the DiAML-TabSW format. Secondly, users of the DiAML-MultiTab format and users of the DiAML-TabSW format more strongly agreed than users of the DiAML-XML format that they would be able to work productively with their respective format soon. Finally, users of the DiAML-TabSW format more strongly agreed that most people would quickly master their respective format than users of the DiAML-XML format. Finally, no significant differences were found between formats for the five statements about the three-level-architecture. Again, however, responses to four of the five statements were least positive for the DiAML-XML format. For all findings related to the five hypotheses it is important to note that the annotation in the experiment is a relatively short one with various concepts omitted. A more extensive annotation with more complex concepts used might have resulted in different scores, time spent on the questions, and feelings towards the formats.

In addition to the results related to the hypotheses some other interesting results were found. The three representation formats were compared by analysing the number of correct and incorrect answer per question. It was found that for three questions more incorrect answers and fewer correct answers were given than expected by users of the DiAML-XML format compared to users of the other formats. Moreover, users of the DiAML-MultiTab format and users of the DiAML-TabSW format gave more correct and fewer incorrect answers than expected for three questions and one question, respectively. These findings show additional support for both formats as attractive alternatives to the DiAML-XML format. Also, five additional variables were examined for their effect on total score, search score, inaccuracy score, time spent, and the responses to the statements in the three representation format conditions. These variables were gender, age, education, previous experience with annotations, and – only for the DiAML-XML condition – previous experience with XML (or HTML). No statistically significant results were found for the first four variables. However, it is important to note that for the age and education variables there was little variation between participants and for the two experience variables limited data was available as not many participants reported to have previous experience with annotations. Participants with XML experience more strongly agreed that the XML format was easy to understand and user friendly, and thought they could work productively with the XML format soon, compared to participants without XML experience. In addition, they more strongly agreed that finding the right information in the three levels was easy. However, for the other (eleven) statements no differences were found between participants with and without XML experience. Moreover, the three scores showed no differences. Finally, no differences were found in time spent on the questions, indicating that having XML experience does not result in lower time consumption and better performances in using the DiAML-XML format.

In conclusion, in all analyses the new formats showed results that were – at least – at the level of the DiAML-XML format. The DiAML-TabSW format, and the DiAML-MultiTab to a lesser extent, even showed more positive results than the DiAML-XML format on multiple occasions. Moreover, an overall trend in the analyses could be identified – with the DiAML-TabSW format coming out on top and the DiAML-XML format finishing last nearly all of the time – which potentially warrants further investigation into the three formats with a larger sample size. Furthermore, it is important to note that participants in the current experiment were students that had little or no experience working with annotations. Because the new representation formats are and will primarily be used by experienced experts in the domain of semantic (dialogue) annotation, future research could use experts to compare the three representation formats. Finally, while the current experiment examined the effect of representation format on the ability to read the annotations, future research could examine the effect of representation format on the ability to write annotations.

7. Concluding Remarks

The construction of the DialogBank has been motivated by the belief that it could be a valuable resource for the design of dialogue systems and for dialogue research. The DialogBank – even though still in its infancy – already appears to be a useful instrument for the latter purpose. While the original format of the ISO 24617-2 standard, DiAML-XML, is an ideal format and has been demonstrated to be convertible to other ideal formats, DiAML-MultiTab and DiAML-TabSW, it is not the most attractive format for dialogue research. The conversion program has shown that it is possible to switch between ideal formats, depending on the desired conversion. Moreover, the experiment supports the idea that users may prefer one format to the other, and that one format may – for a variety of purposes – be more convenient to use than the other. Finally, limitations of the ISO standard that came to light during the construction of the DialogBank have been identified and potential solutions have been proposed.

References

Allen, J., & Core, M. (1997). Draft of DAMSL: Dialog act markup in several layers. Unpublished manuscript, 2.

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., ... & Traum, D. R. (1995). The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental & Theoretical Artificial Intelligence*, *7*(1), 7-48.

Alexandersson, J., Buschbeck-Wolf, B., Fujinami, T., Maier, E., Reithinger, N., Schmitz, B., & Siegel, M. (1997). Dialogue acts in VERBMOBIL-2.

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G., Garrod, S., ... & Sotillo, C. (1991). The HCRC map task corpus. *Language and speech*, 34(4), 351-366.

Bunt, H. (1994). Context and dialogue control. Think Quarterly, 3(1), 19-31.

Bunt, H. (1999). Dynamic interpretation and dialogue theory. The structure of multimodal dialogue, 2, 1-8.

Bunt, H. (2000). Dialogue pragmatics and context specification. *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics. Amsterdam: Benjamins*, 81-150.

Bunt, H. (2009). The DIT++ taxonomy for functional dialogue markup. In *AAMAS 2009 Workshop*, Towards a Standard Markup Language for Embodied Dialogue Acts (pp. 13-24).

Bunt, H. (2011). The semantics of dialogue acts. In *Proceedings of the Ninth International Conference on Computational Semantics* (pp. 1-13). Association for Computational Linguistics.

Bunt, H. (2013). A methodology for designing semantic annotations. *Forthcoming in Language Resources and Evaluation*.

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

Bunt, H., Alexandersson, J., Carletta, J., Choe, J. W., Fang, A. C., Hasida, K., ... & Soria, C. (2010). Towards an ISO standard for dialogue act annotation. In *Seventh conference on International Language Resources and Evaluation (LREC'10)*.

Bunt, H., Alexandersson, J., Choe, J. W., Fang, A. C., Hasida, K., Petukhova, V., ... & Traum, D. R. (2012). ISO 24617-2: A semantically-based standard for dialogue annotation. In *LREC* (pp. 430-437).

Bunt, H., Kipp, M., & Petukhova, V. (2012). Using DiAML and ANVIL for multimodal dialogue annotations. In *LREC* (pp. 1301-1308).

Bunt, H., Petukhova, V., Malchanau, A., Fang, A., & Wijnhoven, K. (2016). The DialogBank. In *Proceedings 9th International Conference on Language Resources and Evaluation* (LREC), Portoroz, May 25-28.

Bunt, H., & Prasad, R. (2016). ISO DR-Core (ISO 24617-8): Core Concepts for the Annotation of Discourse Relations.

Bunt, H., Prasad, R., & Joshi, A. (2012). First steps towards an ISO standard for annotating discourse relations. In *Proceedings of the Joint ISA-7, SRSL-3, and I2MRT Workshop on Semantic Annotation and the Integration and Interoperability of Multimodal Resources and Tools* (pp. 60-69).

Bunt, H., & Romary, L. (2004). Standardization in multimodal content representation: Some methodological issues. In *4th International Conference on Language Resources and Evaluation-LREC'04* (pp. 28-p).

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., ... & Lathoud, G. (2005, July). The AMI meeting corpus: A pre-announcement. In *International Workshop on Machine Learning for Multimodal Interaction* (pp. 28-39). Springer Berlin Heidelberg.

Domain Adaptations for Dialog Agents (DADA). (2016). Riva del Garda, Italy. Retrieved from: https://sites.google.com/site/ecmldaworkshop/home.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. Sage.

Geertzen, J. (2007). DitAT: a flexible tool to support web-based dialogue annotation. In *Proceedings of the Seventh International Workshop on Computational Semantics* (IWCS) (pp. 320-323).

Geertzen, J., Girard, Y., Morante, R., Van der Sluis, I., Van Dam, H., Suijkerbuijk, B., ... & Bunt, H. (2004). The DIAMOND project. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue*, Catalog'04.

Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Van Leeuwen, D., ... & Wan, V. (2008). The 2007 AMI (DA) system for meeting transcription. In *Multimodal Technologies for Perception of Humans* (pp. 414-428). Springer Berlin Heidelberg.

Ide, N., & Suderman, K. (2014). The linguistic annotation framework: a standard for annotation interchange and merging. *Language Resources and Evaluation*, 48(3), 395-418.

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

ISO 24612:2012. Language Resource Management – Linguistic annotation framework (LAF). International Standard. International Organization for Standardization, Geneva.

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

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

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

ISO 24617-6:2016. Language Resource Management — Semantic Annotation Framework — Part 6: Principles of semantic annotation (SemAF Principles).

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

ISO (2015). ISO CD 24617-8:2015, Language Resource Management - Semantic Annotation Framework (SemAF) - Part 8: Semantic relations in discourse Geneva: ISO.

Jurafsky, D., Shriberg, E., & Biasca, D. (1997). Switchboard-damsl labeling project coder's manual. *Technická Zpráva*, 97-02.

Kipp, M. (2001). Graduate College for Cognitive Sciences University of the Saarland, Germany.

Lee, K., & Romary, L. (2010). Towards interoperability of ISO standards for Language Resource Management. In *ICGL* 2010 (p. 9p).

Lohr, S. (2012). The age of big data. *New York Times*, 11.

Miltsakaki, E., Prasad, R., Joshi, A. K., & Webber, B. L. (2004). The Penn Discourse Treebank. In *LREC*.

Petukhova, V. V. (2011). Multidimensional dialogue modelling. Promotor: H. Bunt. Tilburg, 1.

Petukhova, V., & Bunt, H. (2012). The coding and annotation of multimodal dialogue acts. In *LREC* (pp. 1293-1300).

Petukhova, V., Gropp, M., Klakow, D., Schmidt, A., Eigner, G., Topf, M., ... & Deroo, O. (2014). The DBOX corpus collection of spoken human-human and human-machine dialogues. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC" 14)* (No. EPFL-CONF-201766). European Language Resources Association (ELRA).

Petukhova, V., Malchanau, A., & Bunt, H. (2014). Interoperability of Dialogue Corpora through ISO 24617-2-based Querying. In *LREC* (pp. 4407-4414).

Petukhova, V., Prévot, L., & Bunt, H. (2011). Multi-level discourse relations between dialogue units. In *Proceedings 6th Joint ACL-ISO Workshop on Interoperable Semantic Annotation* (ISA-6), Oxford (pp. 18-27).

Prasad, R., & Bunt, H. (2015). Semantic relations in discourse: The current state of ISO 24617-8. In *Proceedings 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation* (ISA-11) (pp. 80-92).

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., & Webber, B. L. (2008). The Penn Discourse TreeBank 2.0. In *LREC*.

Reithinger, N., & Klesen, M. (1997). Dialogue act classification using language models. In *EuroSpeech*.

Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., & Carvey, H. (2004). The ICSI meeting recorder dialog act (MRDA) corpus. INTERNATIONAL COMPUTER SCIENCE INST BERKELEY CA.

Test, O. (2015). Your Chi-Square Test is Statistically Significant: Now What?. *Practical Assessment, Research & Evaluation*, *20*(8), 2.

Traum, D. R., & Larsson, S. (2003). The information state approach to dialogue management. In *Current and new directions in discourse and dialogue* (pp. 325-353). Springer Netherlands.

Appendix A

Dialogue Fragment from "TRAINS 2" in DiAML-MultiTab format.

First part from dialogue "TRAINS 2", Gold standard ISO 24617-2 annotation represented in DiAML-MultiTab format.

Markables	Sender	Addressee	Other P's	FS text	Turn transcript	Task	autoFeedback	alloFeedback	turnManagement	timeManagement	ownCommunicationManagement	partnerCommunicationManagement	discourseStructuring	socialObligationsManagement	Comments
trains2.fs1	S	U		hello	hello can I help you								da1:opening	da2:initial Greeting	
trains2.fs2	S	U		can I help you									da3:offer		
trains2.fs3	U	S		yes	yes, I have a problem I need to transport two tankers of OJ to Avon and three boxcars of bananas to Elmira, the bananas must arrive in Elmira by nine p.m.								da4:acceptOffer (Fu:da3)		
trains2.fs4	U	S		I have a problem I need to transport two tankers of OJ to Avon and three boxcars of bananas to Elmira		da5:inform									
trains2.fs5	U	S		the bananas must arrive in Elmira by nine p.m.		da6:inform (Expand:Expander da5)									
trains2.fs6	S	U		okay	okay		da7:autoPositive (Fe:da6)								

Appendix B

Dialogue Fragment from “TRAINS 2” in DiAML-TabSW format.

First part from dialogue "TRAINS 2", Gold standard ISO 24617-2 annotation represented in DiAML-TabSW format

Markables	Da-ID	Dialogue acts	Sender	Addressee	Other P's	FS text	Turn transcript	Comments
trains2.fs1	da1; da2	Ds:opening; SoM:initialGreeting	S	U		hello	hello can I help you	
trains2.fs2	da3	Ds:offer	S	U		can I help you		
trains2.fs3	da4	Ds:acceptOffer (Fu:da3)	U	S		yes	yes, I have a problem I need to transport two tankers of OJ to Avon and three boxcars of bananas to Elmira the bananas must arrive in Elmira by nine p.m.	
trains2.fs4	da5	Ta:inform	U	S		I have a problem I need to transport two tankers of OJ to Avon and three boxcars of bananas to Elmira		
trains2.fs5	da6	Ta:inform (Expand:Expander da5)	U	S		the bananas must arrive in Elmira by nine p.m.		
trains2.fs6	da7	Auf:autoP (Fe:da6)	S	U		okay	okay	

Appendix C

Customized transcript of dialogue "sw01-0105".

A: Hi Jimmy.

J: Hello Anne.

A: So how do you get most of your news?

J: Well, I uh watch the uh national news every day. I read one or two papers a day. I also tune into CNN a lot.

A: Wow. So, at what time do you watch the national news?

J: Uh, every evening at six thirty, I believe, I watch the national news.

A: Oh, okay. I would probably be a junkie or watch <laughter> CNN a lot, but I do don't uh subscribe to cable, because of the poor service and also, well, because I give to the United way. So I figured that amount of money I just donate that.

J: Uh-huh. As opposed to paying for cable?

A: Yeah. I take away, uh, an addiction <laughter>.

J: <Laughter> yeah.

A: Well I, -

J: Uh, overall, the quality of the news you get o off of most, uh, sources I would say is pretty low. Usually you get pretty incomplete coverage. That's one of the reasons where I, why I try and get as many sources as possible.

A: Right.

J: Because if you hear the same story from three different sources, then you get a much clearer picture of of what is going on, you would hope.

A: Yeah.

J: But, still, you know, hard to say that what you've heard is what really is.

A: What, what, uh, newspapers do you read?

J: Well, I read, uh, the local newspaper and I also try and read one of the, uh, major dailies like the CHICAGO TRIBUNE, or the NEW YORK TIMES or something like that.

A: Okay. For a while there I, I, I, uh subscribed to NEW YORK TIMES, a- actually a couple of newspapers because, uh you know, my fiancé, he was unemployed for a while. So, we really needed to look at the <inhaling> the want- help wanted ads.

J: Uh-huh.

A: And so often they, they tr- the newspapers are trying to compete with a lot of other sources of newspape, uh, news.

J: Right.

A: And, I don't know, NEW YORK TIMES is okay, but, uh, when you read a lot of this stuff, the, the quality of the writing has definitely gone down in the last ten years or so, but --

J: Uh-huh.

A: but uh, I mean, they, they say it's like the sixth grade reading level --

J: Right.

A: but, I swear it's at least second or third grade –

J: <laughter>

A: sometimes.

J: Pretty low. I really like NRP radio.

A: NPR.

J: Yeah.

A: That's pretty good. I like listening to that when I'm in the car a long enough time to, to listen to it, because I

J: Right. I like NPR a lot because, uh, instead of being as your newspaper or your TV news where you get a five minute blurb or a ten paragraph blurb, they go really in depth on topics which I think is good.

A: Uh --

J: And, their editorial stand point seems to be a lot better than say your major network news and that kind of thing.

A: Right. Sorry, I have to go now. See you Jimmy.

J: Okay. Goodbye Anne.

Appendix D

'General information' in DiAML-XML condition (English).

General information
This documents contains information about the documents which you have access to. Do not worry if you do not understand something. Try to answer the questions. If you really do not understand something you can ask the researcher a question.
On one of your screens you can find an XML file. This is the annotation of the dialogue. An annotation consists of three levels.

- **Level 1**. All **words** that are present in the dialogue are coupled with word numbers
    - w1 = hi, w2= Jimmy, w3 = hello, …
- **Level 2**. In this level all **dialogue segments** are specified (consisting of one or more words). Dialogue segments are built up of above word numbers.
- **Level 3**. In this level information is assigned to the dialog segments. So-called *dialogue acts* are attached to dialogue segments. A dialogue act is a combination of (**1**) a *function* and (**2**) a *dimension*. Moreover, (**3**) a relation with one or multiple other dialogue acts may exist (e.g. question-answer).

A ***function*** describes the intention of a speaker's utterance. Examples are 'stalling' (stretching time), 'turnTake' (take the speaker role), and 'answer'. A function is always partnered by a dimension.

A ***dimension*** describes the information-type of a dialogue act. A dialogue act can, for instance, be about the subject of the dialogue (*'subject'* or *'subj'* dimension), about the discussing or reporting of (successful or unsuccessful) processing of previous utterances (*'feedback'* or *'fb'* dimension), about the distribution of the speaker role (*'turn management'* or *'TuM'* dimension), about the distribution of time (*'time management'* or *'TiM'* dimension), about social obligations such as greeting each other (*'social'* or *'soc'* dimension), or about the correcting of a previous utterance (*'speech edit'* or *'spEdit'* dimension).

Example:

"Hello, what is the problem?"

- Level 1: w1=hello, w2=what, w3=is w4=the, w5=problem.
- Level 2: dialogue segment 1 (ds1) = w1.
- Level 2: dialogue segment 2 (ds2) = w2, w3, w4, w5.
- Level 3: dialogue segment 1 (ds1) = 'initialGreeting' function in 'social' dimension.
- Level 3: dialogue segment 2 (ds2) = 'openQuestion' function in 'subject' dimension.

The following correct function-dimension combinations are present in your annotation:

Functie	Dimensie	Functie	Dimensie	Functie	Dimensie
Inform	Subj	turnKeep	TuM	Apology	Soc
openQuestion	Subj	turnTake	TuM	initialGoodbye	Soc
answer	Subj	turnGrab	TuM	returnGoodbye	Soc
agreement	Subj	turnAssign	TuM	selfCorrection	spEdit
positiveFb	fb	stalling	TiM	correctMisspeaking	spEdit
checkQuestion	fb	initialGreeting	Soc		
confirm	fb	returnGreeting	Soc		

There may be a **relation** between dialogue acts. A *feedback dependence relation* is a relation between a dialogue act in the feedback dimension and one or more other dialogue acts. A *functional dependence relation* is a relation between two or more dialogue acts that are not in the feedback dimension (e.g. question and answer).

XML format

The annotation that you are about to use is represented in an XML format. Like any XML file your file is structured in *elements*. Elements have an opening tag (e.g. <country> The Netherlands </country>). Data can be present as content of an element (The Netherlands is the content of the ‘country’ element), or as *value* of an *attribute*. In the example below ‘Greenwich’ is the value of the ‘time-zone’ attribute.

```
<dateTime time-zone="Greenwich">
   <date>08-sep-2005</date>
   <localTime>10:30 AM</localTime>
 </dateTime>
```

Your XML file is structured in three parts that each correspond to one of the annotation levels: annotatielevels:

```
<TEI>
  <text>
    <div>
          Level 1 - Words
    </div>
    <div>
          Level 2 - Dialogue segments
    </div>
    <diaml>
          Level 3 – Annotation (with, among other things, functions, dimensions and
          relations)
    </diaml>
  </text>
</TEI>
```

Level 1. First part of the XML file. First, Anne’s words are coupled with word numbers. Then, Jimmy’s words are coupled with word numbers. This first part ends right after Jimmy’s last word; w502 “Anne”. With the opening of a new <div> element Level 2 starts.

Level 2. Middle part of the XML file. In this level all dialogue segments are specified. They are composed of above word numbers.

```
 <spanGrp xml:id="spGr1" type="dialogueVerbalSegment">
          <span from="#w1" />
          <span from="#w2" />
 </spanGrp>
 <fs xml:id="ds1" type="dialogueSegment">
          <f name="verbalComponent" fVal="#spGrp1" />
 </fs>
```

The word numbers are given as values of the “from” attributes. The “xml:id” attribute below (within the “fs” element) gives the name of the dialogue segment. See above example, which is about dialogue segment 1 (because xml:id=“ds1”) and is composed of w1 and w2 (“Hello Jimmy”).

Level 3. Last part of the XML file. This part is composed of ‘DialogueAct’ elements. You can recognize the dialogue segments by the values of the ‘dialogSegment’ attributes. For instance, see that the first dialogueAct element (xml:id=”da1”) covers the first dialogue segment (dialogSegment=“#ds1”), that p1 is the sender and p2 the addressee, and that an “initialGreeting” function and “social” dimension are annotated. Possible relations with other dialogue acts annotated with ‘feedbackDependence’ and ‘functionalDependence’ attributes.

Appendix E

‘General information’ in DiAML-MultiTab condition (English).

General information

This documents contains information about the documents which you have access to. Do not worry if you do not understand something. Try to answer the questions. If you really do not understand something you can ask the researcher a question.

You have received three documents that each correspond to one annotation ‘level’.

- **Level 1 (paper)**. This is the document where all **words** that are present in the dialogue are coupled with word numbers (“Level 1 Words”) as follows:
    - w1 = hi, w2= Jimmy, w3 = hello, …
- **Level 2 (paper)**. This is the document where all **dialogue segments** (consisting of one or more words) are specified (“Level 2 Dialogue Segments”). Dialogue segments are built up of above word numbers.
- **Level 3**. This is the Excel file where information is assigned to the dialogue segments. So-called *dialogue acts* are attached to dialogue segments. A dialogue act is a combination of (**1**) a *function* and (**2**) a *dimension*. Moreover, (**3**) a relation with one or multiple other dialogue acts may exist (e.g. question-answer).

A ***function*** describes the intention of a speaker’s utterance. Examples are ‘stalling’ (stretching time), ‘turnTake’ (take the speaker role), and ‘answer’. A function is always partnered by a dimension.

A ***dimension*** describes the information-type of a dialogue act. A dialogue act can, for instance, be about the subject of the dialogue (*‘subject’* or *‘subj’* dimension), about the discussing or reporting of (successful or unsuccessful) processing of previous utterances (*‘feedback’* or *‘fb’* dimension), about the distribution of the speaker role (*‘turn management’* or *‘TuM’* dimension), about the distribution of time (*‘time management’* or *‘TiM’* dimension), about social obligations such as greeting each other (*‘social’* or *‘soc’* dimension), or about the correcting of a previous utterance (*‘speech edit’* or *‘spEdit’* dimension).

Example:

“Hello, what is the problem?”

- Level 1: w1=hello, w2=what, w3=is w4=the, w5=problem.
- Level 2: dialogue segment 1 (ds1) = w1.
- Level 2: dialogue segment 2 (ds2) = w2, w3, w4, w5.
- Level 3: dialogue segment 1 (ds1) = ‘initialGreeting’ function in ‘social’ dimension.
- Level 3: dialogue segment 2 (ds2) = ‘openQuestion’ function in ‘subject’ dimension.

The following correct function-dimension combinations are present in your annotation:

Functie	Dimensie	Functie	Dimensie	Functie	Dimensie
Inform	Subj	turnKeep	TuM	Apology	Soc
openQuestion	Subj	turnTake	TuM	initialGoodbye	Soc
answer	Subj	turnGrab	TuM	returnGoodbye	Soc
agreement	Subj	turnAssign	TuM	selfCorrection	spEdit
positiveFb	fb	stalling	TiM	correctMisspeaking	spEdit
checkQuestion	fb	initialGreeting	Soc		
confirm	fb	returnGreeting	Soc		

There may be a **relation** between dialogue acts. A *feedback dependence relation* is a relation between a dialogue act in the feedback dimension and one or more other dialogue acts. A *functional dependence relation* is a relation between two or more dialogue acts that are not in the feedback dimension (e.g. question and answer).

Tabular format (Level 3 – digital)

The dialogue annotation is structured in rows and columns as follows (see the Excel file):

Rows: Each row represents the information about one dialogue segment.

Column A: "Dialogue Segment IDs"; contains the numbering of the dialogue segments.

Column B and **C**: 'Sender' and 'Addressee' of the dialogue segment.

Column D: "Dialogue segments"; contains the words of the dialogue segments.

Column E: "Turns"; contains the dialogue segment transcripts.

Columns F till **K**: represent the six dimension. The contents of a cell in these columns contains (1) the numbering of the dialogue act, (2) the function and possibly (3) a relation with one or more other dialogue acts. For instance; *da7:answer(funcDep:da3)*, where there is a functional dependence relation between da7 and da3.

Appendix F

‘General information’ in DiAML-TabSW condition (English).

General information

This documents contains information about the documents which you have access to. Do not worry if you do not understand something. Try to answer the questions. If you really do not understand something you can ask the researcher a question.

You have received three documents that each correspond to one annotation ‘level’.

- **Level 1 (paper)**. This is the document where all **words** that are present in the dialogue are coupled with word numbers (“Level 1 Words”) as follows:
    - w1 = hi, w2= Jimmy, w3 = hello, …
- **Level 2 (paper)**. This is the document where all **dialogue segments** (consisting of one or more words) are specified (“Level 2 Dialogue Segments”). Dialogue segments are built up of above word numbers.
- **Level 3**. This is the Excel file where information is assigned to the dialogue segments. So-called *dialogue acts* are attached to dialogue segments. A dialogue act is a combination of (**1**) a *function* and (**2**) a *dimension*. Moreover, (**3**) a relation with one or multiple other dialogue acts may exist (e.g. question-answer).

A ***function*** describes the intention of a speaker’s utterance. Examples are ‘stalling’ (stretching time), ‘turnTake’ (take the speaker role), and ‘answer’. A function is always partnered by a dimension.

A ***dimension*** describes the information-type of a dialogue act. A dialogue act can, for instance, be about the subject of the dialogue (*‘subject’* or *‘subj’* dimension), about the discussing or reporting of (successful or unsuccessful) processing of previous utterances (*‘feedback’* or *‘fb’* dimension), about the distribution of the speaker role (*‘turn management’* or *‘TuM’* dimension), about the distribution of time (*‘time management’* or *‘TiM’* dimension), about social obligations such as greeting each other (*‘social’* or *‘soc’* dimension), or about the correcting of a previous utterance (*‘speech edit’* or *‘spEdit’* dimension).

Example:

“Hello, what is the problem?”

- Level 1: w1=hello, w2=what, w3=is w4=the, w5=problem.
- Level 2: dialogue segment 1 (ds1) = w1.
- Level 2: dialogue segment 2 (ds2) = w2, w3, w4, w5.
- Level 3: dialogue segment 1 (ds1) = ‘initialGreeting’ function in ‘social’ dimension.
- Level 3: dialogue segment 2 (ds2) = ‘openQuestion’ function in ‘subject’ dimension.

The following correct function-dimension combinations are present in your annotation:

Functie	Dimensie	Functie	Dimensie	Functie	Dimensie
Inform	Subj	turnKeep	TuM	Apology	Soc
openQuestion	Subj	turnTake	TuM	initialGoodbye	Soc
answer	Subj	turnGrab	TuM	returnGoodbye	Soc
agreement	Subj	turnAssign	TuM	selfCorrection	spEdit
positiveFb	fb	stalling	TiM	correctMisspeaking	spEdit
checkQuestion	fb	initialGreeting	Soc		
confirm	fb	returnGreeting	Soc		

There may be a **relation** between dialogue acts. A *feedback dependence relation* is a relation between a dialogue act in the feedback dimension and one or more other dialogue acts. A *functional dependence relation* is a relation between two or more dialogue acts that are not in the feedback dimension (e.g. question and answer).

Tabular format (Level 3 – digital)

The dialogue annotation is structured in rows and columns as follows (see the Excel file):

Rows: Each row represents the information about one dialogue segment.

Column A: “Dialogue Segment IDs”; contains the numbering of the dialogue segments.

Column B: “Dialogue Act IDs”; contains the numbering of the dialogue acts.

Column C: “Dialogue Acts”; represents the contents of dialogue acts as follows: “dimension:function (possible relation with one or more other dialogue acts)”. For instance, “subj:answer(funcDep da12)”, where the dialogue act includes the function ‘answer’ and the dimension ‘subject’, and has a ‘functional dependence relation’ with dialogue act 12.

Columns D and **E**: ‘Sender’ and ‘Addressee’ of the dialogue segment.

Column F: “Dialogue segments”; contains the words of the dialogue segments.

Column G: “Turns”; contains the dialogue segment transcripts.

Appendix G

DiAML-XML survey in English (correct answers in green).

Thank you for participating in this experiment.

Before you start the experiment you are first asked a number of general questions.

--------------------- (= new page)

What is your gender?

Man

woman

What is your age?

Textbox

What is the highest degree or level of school you have completed?

VMBO

HAVO/VWO

Bachelor (premaster)

Master

Do you have previous experience with annotations?

Yes

No

Do you have previous experience with XML (or HTML)?

Yes

No

Make sure you carry out the next steps:

Step 1: Read the dialogue transcript. The dialogue is between Anne and Jimmy and is about sources of news. The transcript and annotation are in English.

Step 2: Read and study the 'general information'. You can consult this information during the entire experiment.

Have you carried out both steps? If so, you are ready to start the experiment.

The experiment contains 20 questions. Try to answer these questions correctly.

Remember for some of the questions to take into account all three levels (words, dialogue segments, annotation)!

Good luck!

Q1. How many dialogue segments are present in the annotation?

112

115

104

107

Q2. How many dialogue acts (combinations of functions and dimensions) are present in the annotation?

107

104

115

112

Q3. How often does the 'social' dimension occur in the annotation?

3 times

4 times

5 times

6 times

Q4. What is the function of the sixth dialogue segment?

Stalling

turnTake

inform

turnKeep

Q5. What dimensions is annotated at dialogue act 14?

Subject

Time management

Feedback

Turn management

Q6. Jimmy says that he likes "NRP" radio. Anne, however, speaks about "NPR" radio. Which function and dimension are assigned to Anne's utterance?

Inform and feedback

CorrectMisspeaking and speechEdit

PositiveFb and feedback

selfCorrection and speechEdit

Q7. What function is assigned to the dialogue segment “I really like NRP radio”?

Answer

Inform

Confirm

positiveFb

Q8. What dimension is assigned to dialogue segment 100?

Time management

Feedback

Turn management

Correction

Q9. Jimmy says that he watches the national news every day; “I watch the national news every day”. Which word numbers together constitute this utterance?

W15,w16,w17,w18,w19,w20,w22

W15,w17,w19,w20,w21,w22,w23

W15,w17,w18,w19,w20,w21,w22

W15,w17,w18,w20,w21,w22,w23

Q10. What function is assigned to the dialogue segment that consists of the following word numbers? w119,w120,w121,w122,w123,w124,w125

turnTake

positiveFb

inform

this segment does not exist

Q11. What is the function of the dialogue segment that consists of the following word numbers? W78,w80,w82,w83,w84

Stalling

turnKeep

inform

this segment does not exist

Q12. How many dialogue segments have **more** than one function and dimension? (so more than one dialogue act)

4 dialogue segments

6 dialogue segments

8 dialogue segments

10 dialogue segments

Q13. At a certain point Jimmy laughs (‘laughter’). This dialogue act has a feedback dependence relation with one of Anne’s dialogue acts. Which function is assigned to (1) Jimmy’s dialogue act and to (2) Anne’s dialogue act?

(1) Answer – (2) openQuestion
(1) positiveFb – (2) inform
(1) confirm – (2) checkQuestion
(1) agreement – (2) answer

Q14. Which function and dimension are annotated at (1) dialogue act 100 and (2) dialogue act 102?

(1) positiveFb and feedback, (2) turnTake and turnManagement

(1) inform and subject, (2) stalling and timeManagement

(1) turnKeep and turnManagement, (2) apology and social

(1) stalling and timeManagement, (2) inform and subject

Q15. Dialogue act 114 has a feedback dependence relation with one of Anne’s previous dialogue acts. Which word numbers together make up the dialog segment that belongs to this previous dialogue act?

W492,w493,w494,w495,w496

W491

W497,w498,w499

W491,w492,w493,w494,w495,w496

You have now completed fifteen questions. The last five questions are about inaccuracies in the annotations.

Q16. The annotation of dialogue act 15 includes an inaccuracy. What inaccuracy?

An incorrect function is annotated

An incorrect dimension is annotated

An incorrect sender is annotated

A relation is annotated

Q17. The annotation of dialogue act 24 is incomplete. What is missing?

A function

A sender

A dimension

An addressee

Q18. Dialogue act 59 has a functional dependence relation with another dialogue act. With which dialogue act? Give the dialogue act number (ID).

Textbox (da56 / 56)

Q19. Dialogue act 80 contains a feedback function. What is wrong about the annotation of dialogue act 80?

A feedback dependence relation is annotated instead of a functional dependence relation

A functional dependence relation is annotated instead of a feedback dependence relation

A wrong dialogue segment number is annotated

No mistake

Q20. What is incorrect about the annotation of dialogue act 87?

An incorrect function

No mistake

There is no functional dependence relation

There is no feedback dependence relation

You have now answered all twenty question and have almost reached the end of the experiment.

Please read the next statements about your experiences with working with the various annotation materials.

Indicate to what extent you agree with the following statements. During the answering of the questions…

it was easy to find the right answers. (1 = strongly disagree – 7 = strongly agree)

I sometimes got disoriented. (1-7)

I sometimes got frustrated. (1-7)

I became better and better in finding the right answers. (1-7)

I thought it was unnecessary complex to find the right answers. (1-7)

Indicate to what extent you agree with the following statements.

The XML format was unclear. (1-7)

The XML format was easy to understand. (1-7)

The XML format was user-friendly. (1-7)

I think I could work productively with the XML format soon. (1-7)\

I think that most people quickly master the XML format. (1-7)

Indicate to what extent you agree with the following statement about the three levels (words, dialogue segments, semantic annotation).

The presentation of the three levels was clear. (1-7)

Finding the right information in the three levels was easy. (1-7)

I thought it was difficult to determine which level to search. (1-7)

The presentation of the three levels was pleasant. (1-7)

It was difficult to navigate between the three levels. (1-7)

Do you have any remarks about the annotation, documents, or other aspects of the experiment? Leave them here.

Textbox

This is the end of the experiment. Thank you for your participation.

Appendix H

Consent form (English).

Consent form
Title experiment: "The Dialogue Quiz"

Goal Research: In this experiment you will be asked to answer questions related to the annotation of a dialogue. This annotation contains information on the communicative acts of the two people in the dialogue. When you start the experiment you will receive more information on the contents of the dialogue and the annotation. It is your job to answer the questions the best you can.

Duration and compensation research: The experiment will take approximately 30-40 minutes and you can earn 1 'participants pool credit'.

Privacy: Your information is highly confidential and your privacy is guaranteed. Your name will not be connected to the (results of the) research in any way.

Voluntary participation: Participation in this experiment is not compulsory. You can cancel your participation in the experiment at any time. You will, however, not receive a credit in that case.

Contact: If you have any questions about the research afterwards contact Kars Wijnhoven (k.wijnhoven@tilburguniversity.edu).

Consent

I have read this consent form and the research was explained to me. I am willing to participate in the current experiment.

____
Signature participant

____
Date

____
Name participant

____
Signature researcher

____
Date

____
Name researcher