An Extensible Compositional Semantics for Temporal Annotation

Harry Bunt, Chwhynny Overbeeke

Tilburg University, Department of Communication and Information Sciences
P.O. Box 90153, 5000 LE Tilburg, Netherlands,
harry.bunt@uvt.nl., info@chwhynny.nl

Abstract

In this paper we present an event-based formal semantics for temporal annotation, in particular for the ISO-TimeML annotation language under development in the International Organization for Standardization. This semantics has the form of a *compositional translation* into First-Order Logic (FOL) using terms that denote concepts in an extended OWL-Time. Given the fact that FOL has a compositional semantics, our ISO-TimeML semantics is compositional because its translation into FOL is compositional in the sense that the translation of the annotation of a text is a function of the translations of its subexpressions (where any well-formed subexpression can be translated independently of other subexpressions) and the structure of the annotation, as encoded in its linking tags. The approach presented here has been designed to be extensible to the semantic annotation of other than temporal information.

1. Introduction

Linguistic annotation, according to Ide & Romary (2004), is the process of adding linguistic information to language data, or that information itself. The primary aim of annotation is usually the identification of certain linguistic patterns, in order to support the investigation of linguistic phenomena illustrated by such patterns, in particular for applying machine learning algorithms. As such, syntactic annotation as well as morphosyntactic, prosodic and pragmatic annotation have been useful in the development of data-driven linguistic models and theories.

Semantic annotations are meant to capture some of the meaning in the annotated text. This is not only potentially useful for identifying certain linguistic semantic patterns, but the meaning that is captured by the annotation should also support the exploitation of that semantic information in language processing tasks. For instance, Pustejovsky et al. (2003) argue that their annotation language TimeML, designed to support the automatic recognition of temporal and event expressions in natural language text, should also support *"temporal and event-based reasoning in language and text, particularly as applied to information extraction and reasoning tasks"*. (See also Han & Lavie (2004).) Bunt & Romary (2002) argue that any adequate semantic annotation formalism should have a well-defined semantics. Existing approaches to semantic annotation, by contrast, tend to take the semantics of the annotations for granted.

A current development in the area of semantic annotation is the design of an international standard for the annotation of temporal information, undertaken in the project "Semantic Annotation Framework, Part 1: Time and Events", which is carried out by an expert group within the International Organisation for Standardisation ISO. The annotation language that is defined in this project is based on TimeML and is therefore called ISO-TimeML. This project includes an effort to provide a formal semantics for the annotation language based on Pratt-Hartman's proposal of a formal semantics for TimeML (Pratt-Hartman, 2007) using Interval Temporal Logic, a first-order logic for reasoning about time. In this framework, the annotations are interpreted as statements about time intervals associated with events; events are not represented explicitly. While representing a substantial step forward, this semantics, described in the ISO (2007) document, has certain important limitations:

1. it applies only to a rather limited fragment of the annotation language, not including for instance tense, aspect, and durations;
2. it is not compositional, in the sense that it involves a translation from ISO-TImeML to ITL in such a way that the translation of a subexpression of an annotation structure is dependent on that of other subexpressions;
3. it is applicable to temporal information only, and not extensible to other kinds of semantic information, such as the identification of the participants in the events of which the temporal properties are considered.

In this paper we present an alternative, event-based formal semantics for ISO-TImeML, which applies to a substantially greater part of the annotation language, which is fully compositional, and which is not limited to dealing with temporal information. This approach follows the familiar 'interpretation-by-translation' paradigm, translating ISO-TimeML annotations, as represented in XML, into First-Order Logic (FOL). The compositionality of the approach rests on making this translation compositional.

In discussing this approach we will follow the TimeML terminology and speak of *'events'* in the generalized sense for which Bach (1981) introduced the term *eventualities*, as covering both states and events, where events may be subcategorized in various ways, for instance in processes and transitions.

This paper is organized as follows. In section 2 we briefly look at temporal information from a (onto-)logical and a linguistic point of view, and at the role that temporal annotation has to play. In section 3 we describe the translation of ISO-TimeML tags into formal representations. In section 4 we discuss the problem of making a formal semantics for XML-based annotations compositional, and present our solution to the problem. We end with concluding remarks in section 5.

2. Temporal Information

From a (onto-)logical point of view, the fundamental concepts relating to time are *time point*; the *ordering* relation between time points (*'before'*); *temporal interval*; the *begin* and *end* points of an interval; the relation *'inside'* between points of time and temporal intervals; and the *length* of a temporal interval, which requires the notion of a *temporal unit* of measurement. The general framework of Allen (1984), which has been very influential in the computational modelling of time, distinguishes 7 relations (and their inverses) between temporal intervals: *equals, before/after, meets, overlaps, starts, finishes, during/contains*. These relations can all be defined in terms of the *before* relation among time points and the begin- and end points of intervals. In our FOL translations of ISO-TimeML annotations we will use polymorphic versions of Allen's relations, applying them both to time points and temporal intervals where appropriate. (For instance, we will use a predicate 'Before' which can apply to two temporal intervals, to two time points, to a time point and a temporal interval, or to a temporal interval and a time point, with the obvious interpretations.)

From a linguistic point of view, the issue is in what way these temporal objects and relations are described by linguistic expressions, and how language relates temporal objects to other concepts; in particular to states and events.

Temporal annotation, when endowed with a formal semantics, can be viewed as a bridge between the linguistic encoding of temporal information and the logical modeling of temporal structures and relations. For the formal semantics of ISO-TimeML (ISO, 2007), we will use an extension of the OWL-Time ontology (Hobbs & Pan, 2004). To the basic concepts of OWL (*interval, instant, beginning, end, inside, time zone*) we add the concepts of *temporal unit* and *duration*; and concepts needed for interpreting tense: *event time, speech time, and reference time*.[1]

2.1. Dates, Times and Periods

To represent dates, ISO-TimeML follows ISO standard 8601 and uses the format `yyyy-mm-dd` to encode year, month and day. This representation is unsatisfactory from a logical point of view, as it does not make the components of this information available for reasoning. For specifying a point in time we will use functions like *calYear, calMonth, calDay*, and *clockTime* (which specifies a time as shown on the clock in a given time zone):

(1) *March 16th 2007 at 10:15 a.m. CET*

$\lambda t : \text{INSTANT}(t) \wedge \text{calYear}(t,cet) = 2007 \wedge \text{calMonth}(t,cet) = \text{march} \wedge \text{calDay}(t,cet) = 16 \wedge \text{clockTime}(t,cet) = 10{:}15$

It is rather unusual to be as explicit about a time zone as in (1); the time zone in which a clock time is considered is usually assumed to be obvious from the context in which the text fragment occurs that mentions the time. We will use the constant z_c to indicate the contextually relevant time zone in which a clock time is intended.

We use predicates like DAY and MONTH to represent intervals such as days, weeks, months, and years. The predicate DAY, for instance, is true of an interval starting at twelve midnight in some time zone, ending 24 hours later.

Again using ISO standard 8601, ISO-TimeML represents weekdays according to the format `xxxx-wxx-d`, where *d* is the number of the weekday. Thus, Monday would be `xxxx-wxx-1`, and Friday would be `xxxx-wxx-5`. We will use predicates of the weekdays and Allen's relations between temporal intervals to interpret the ISO-TimeML annotation of such expressions:

(2) (a) *Friday*
$\lambda t \,.\, \exists T : \text{FRIDAY}(T) \wedge \text{Inside}(t,T)$

(b) *every Friday*
$\lambda P \,.\, \forall T : \text{FRIDAY}(T) \rightarrow P(T)$

(c) *each year in March*
$\lambda P \,.\, \forall T_1 : (\text{YEAR}(T_1) \wedge \exists T_2 : \text{calMonth}(T_2,z_c) = \text{march} \wedge \text{Before}(\text{Start}(T_1),\text{Start}(T_2)) \wedge \text{Before}(\text{End}T_2),\text{End}(T_1)) \rightarrow P(T_1)$

We will use the constant *today* to refer to an interval that is a day inside which lies the speech time: *today* $\Leftrightarrow$ $\text{DAY}(T) \wedge \text{Inside}(T_0,T)$:

(3) (a) *yesterday*
$\text{DAY}(T) \wedge \text{END}(T) = \text{START}(\text{today})$

(b) *the day before yesterday*
$\text{DAY}(T_1) \wedge \text{START}(\text{today}) = \text{END}(T_1) \wedge \text{DAY}(T_2) \wedge \text{END}(T_2) = \text{START}(T_1)$

2.2. Durations

To define durations we introduce the function *TimeAmount*, which constructs an amount of time from a numerical specification and a temporal unit, as illustrated in (4a).

(4) *for 2 hours*
$\lambda T : \text{DURATION}(T) = \text{TimeAmount}(2,\text{hour})$

A conversion function which specifies a numerical relation between temporal units, such as Conversion(hour, minute) = 60 explain equivalences like TimeAmount(1,day) = TimeAmount(24,hour) (see further Bunt (1985) for a calculus of amounts).

2.3. Tense and Aspect

Following Reichenbach (1947), we analyse tenses in terms of *event time, speech time*, and *reference time* (ET, T_0, and RT in the formal representations). ISO-TimeML uses PAST, PRESENT, and FUTURE as values of the `tense` attribute. If

[1] Hobbs & Pan (2004) use the term 'duration' to indicate a time span during which an event or state occurs. This is to be distinguished from our use of the term as indicating the length of a time span.

an utterance applies to an event in the past, the event time lies before the speech time; if it applies to an event in the present, the speech time is contained in the event time; if it applies to an event in the future, its event time is after the speech time. We can therefore conclude that:

PAST(e)	$\Leftrightarrow$	Before(ET(e),T_0)
PRESENT(e)	$\Leftrightarrow$	Inside(T_0,ET(e))
FUTURE(e)	$\Leftrightarrow$	Before(T_0,ET(e))

Some examples::

(5) (a) *Igor coughed.*
$\exists e\ \exists x$: SLEEP(e) $\wedge$ AGENT(x,e) $\wedge$ IGOR(x) $\wedge$ Before(ET(e),T_0)

(b) *Igor coughs.*
$\exists e\ \exists x$: SLEEP(e) $\wedge$ AGENT(x,e) $\wedge$ IGOR(x) $\wedge$ Inside(T_0,ET(e))

(c) *Igor will cough.*
$\exists e\ \exists x$: SLEEP(e) $\wedge$ AGENT(x,e) $\wedge$ IGOR(x) $\wedge$ Before(T_0,ET(e))

Note that in these examples we consider a literal interpretation of tenses, treating tense as an indicator of the temporal ordering relation between event time, speech time and reference time. Tense information should not always be taken literally, however. For instance, in (6) the event time lies after the speech time, in spite of the present tense of the verb:

(6) *I am at the office tomorrow.*

The temporal adverb *tomorrow* determines this, even though the present tense of the verb would suggest that the event time includes the speech time. This is a complication for any semantic interpretation of temporal annotation. One way to handle this problem could perhaps be to assign a different value to the `tense` attribute in such cases when annotating the text (e.g., Lee (2008) uses *'future present'*), but this has the drawback altering the linguistic concept of tense. Similar problems may arise with the interpretation of other syntactic attributes like `gender` and `number`.

The *progressive aspect* indicates that an event is occurring over a certain period of time and has not yet ended. That is, the speech time lies between the starting point and the end point of the event time.
Similarly, the *perfective aspect* indicates that an event has been ended, or refers to a state resulting from an event that has occurred:

(7) *Igor had already slept.*
$\exists e\ \exists x$: SLEEP(e) $\wedge$ AGENT(x,e) $\wedge$ IGOR(x) $\wedge$ Before(ET(e),RT) $\wedge$ Before(RT,T_0)

2.4. Temporal Anchoring

The Reichenbach (1947) notion of 'event time', originally introduced to interpret tenses, can obviously also be used for describing the temporal anchoring of an event to a time point or a temporal interval:

(8) *Igor died between 10 and 11 AM.*
$\exists e\ \exists x\ \exists T\ \exists t_1\ \exists t_2$: DIE(e) $\wedge$ PIVOT(x,e) $\wedge$ IGOR(x) $\wedge$ Interval(t_1,t_2) = T $\wedge$ clockTime(t_1,z_c) = 10:00 $\wedge$ clockTime(t_2,z_c) = 11:00 $\wedge$ Inside(ET(e),T) $\wedge$ Before(T,T_0)

ISO-TimeML also supports the temporal anchoring of an event with a specification of frequency, which may involve several temporal elements, such as *two hours a day* and *three days every month*. The ISO-TimeML annotation of such cases and our formal representations are as follows:

(9) `<TIMEX3 tid="t1" type="SET" value= "P1M" quant="EVERY" freq="3D">`
three days every month `</TIMEX3>`
λ P. $\forall T_1$: MONTH(T_1) $\wedge$ $\exists^{!3} T_2$: DAY(T_2) $\wedge$ Inside(T_2,T_1) $\wedge$ P(T_1)

2.5. Relations between events

ISO-TimeML distinguishes three types of relation linking events to temporal elements or other events.

First, TLINK relates two temporal elements to one another, temporal elements to events, or eventualities to events, like for instance *20 minutes* to *every Friday* and *every Friday* to RUN in *Igor runs 20 minutes every Friday*, and LEAVE to ARRIVE in *Amy left before Igor arived*.

Second, SLINK is a subordination link between events for cases like *Igor wants to run* and *Amy believes that Igor loves her*. There are six types of SLINK relations: modal (e.g. PROMISE, WANT), evidential (e.g. SEE), negative evidential (e.g. DENY), factive (e.g. REGRET), counter-factive (e.g. PREVENT), and conditional (e.g. *if*). SLINK is not a temporal relation, and its interpretation is thus outside the scope of this paper (but see Bunt, 2007).

Third, ALINK indicates an aspectual relation between two eventualities: initiation, culmination, termination, continuation, or re-initiation, as exemplified by *Igor started to run*. These relations are more than just temporal relations. They can be viewed as a thematic relation (notably a THEME relation) plus certain specific properties. In the case of initiation, the specific property is that the starting point of the initiating event equals the starting point of the initiated event. Culmination means that the subordinate event has been completed, whereas termination implies that the subordinate event has not been completed.

3. From Annotations to Formal Representations

We follow the "interpretation through translation" approach for interpreting ISO-TimeML annotations, and formulate a compositional translation from the XML representations of ISO-TimeML annotations into formulas of First-Order Logic. The translation is defined by a set of rules for translating ISO-TimeML subexpressions and a set of operations for combining these translations, ultimately leading to the construction of a formal representation of the annotated text.

We mentioned in the beginning of this paper that the proposed ISO-TimeML semantics in terms of Interval Temporal Logic (see Pratt-Hartman (2007) and the ISO (2007) document) is not fully compositional. In a nutshell, the problem of translating (XML-representations of) ISO-TimeML annotations into formulas of a logical language in a compositional way is the following.

Compositional translation means that every well-formed subexpression of the source language is translated into the target language independently of other subexpressions; these translations are subsequently combined in a way that is determined by the structure of the source expression as a whole, as encoded in the TLINK, ALINK and SLINK tags that link the various subexpressions. ISO-TimeML annotations contain two kinds of subexpressions: on the one hand the expressions corresponding to events and temporal objects (`<EVENT .../EVENT>` and `<TIMEX3 ... /TIMEX3>` subexpressions) and on the other hand subexpressions that indicate temporal, aspectual, or subordinate relations (`TLINK, ALINK`, and `SLINK` expressions). The latter type of expressions contain attributes whose values are identifiers in the subexpressions denoting events or temporal objects, thereby 'linking' these subexpressions. Now when the various types of subexpressions are translated into logical formulas, this linking information is lost because the logical formulas do not have identifiers like the XML structures of the ISO-TimeML annotation. The following example illustrates the problem for the ITL-based semantics of ISO-TimeML provided in the ISO (2007) document.

(10) John
<EVENT eiid="ei" type="OCCURRENCE">
drove /EVENT>
to Boston on
<TIMEX3 tid="t1" >
Saturday TIMEX3>
<TLINK eventInstanceID="ei"
relatedToTime="t1"
relType="DURING">

The event tag is translated into $\exists$ I_{ei}: $P_{ei}(I_{ei})$, which says that there is a temporal interval I_{ei} for which the predicate P_{ei} holds, i.e. for which it is true that John drove to Boston during that interval:

<EVENT eiid="ei" type="OCCURRENCE" >
drove /EVENT>
$\leadsto \exists\ I_{ei}: P_{ei}(I_{ei})$

The TLINK structure is subsequently translated in such a way that it takes this latter formula and conjoins it with a formula expressing that the interval I_{ei} is related to another interval I_{t1} (corresponding to Saturday) through the relation specified as the `relType` value in the TLINK expression:

<TLINK eventInstanceID="ei"
relatedToTime="t1" relType="DURING">
$\leadsto \exists\ I_{ei}: P_{ei}(I_{ei}) \wedge \exists\ I_{ei'}: \mathrm{DURING}_r(I_{ei}, I_{ei'})$

Now note that this formula has not been constructed by indepently translating the TLINK structure into a formula which is combined with the formula that translates the event, but in fact the translation rule operating here says: *When translating a* TLINK *expression, find the* EVENT *expression that is identified by the value of the* `eventInstanceID` *attribute; take the translation of that structure, and build within the scope of the existential event quantifier of that formula a conjunction which adds the temporal relation encoded in the* TLINK *structure.*

Kiyong Lee (2007), in trying to provide an alternative semanticss for ISO-TimeML, struggled with the same problem, and adopted the solution that is described below. Katz' (2007) attempt to give a denotational semantic to ISO-TimeML also runs into scoping problems.

We present a solution to this problem and specify a fully compositional translation at the price of having to deal with more complex intermediate representational structures during the translation process. These intermediate representations are triples consisting of a FOL formula plus two components, that we call a *'combination index'* and a *'derivation index'*. The first of these is a list containing the ISO-TimeML identifiers of the subexpressions whose translations are to be combined with the present representation; the second is another list of ISO-TimeML identifiers, indicating the subexpressions whose translations have been used to construct the present representation. As such, they act as a kind of storage which allows to keep track of (a) which pieces of semantic information should be combined, according to the links in the ISO-Timeml/XML representations, and (b) which pieces have already been combined. With the help of these devices, we can make sure that those and only those translations of the ISO-TimeML subexpressions which are linked through TLINK, SLINK or ALINK structures are combined, and in a correct way.

3.1. Translating ISO-TimeML Subexpressions

Here we will deal with the translation of each type of ISO-TimeML tag. (We will not take into account the SIGNAL tag of ISO-TimeML, which has been left out of consideration in this paper, since all it does is assign an index to a signal word such that it can be referred to in other tags.)

3.1.1. The EVENT Tag

The translation of event tags is determined by their polarity. There are two translation rules, one for each polarity value. The notation $\exists e \in E$ is used here and throughout as a shorthand for $\exists e$: E(e).

<EVENT eiid="e" tense=T aspect=A
polarity="POS">
$\leadsto \lambda E\ .\ \lambda P\ .\ \exists e \in E : P(e) \wedge T'(e) \wedge A'(e)$

<EVENT eiid="e" tense=T aspect=A
polarity="NEG">
$\leadsto \lambda E\ .\ \lambda P\ .\ \neg\exists e \in E : P(e) \wedge T'(e) \wedge A'(e)$

The translations of the time and aspect values are given in Table 1. and Table 2, respectively.

`tense` **value**	**Translation**
`tense="PAST"`	λe . Before(ET(e),T_0)
`tense="PRESENT"`	λe . Inside(T_0,ET(e))
`tense="FUTURE"`	λe . Before(T_0,ET(e))

Table 1: Translation table for the EVENT tag attribute `tense`.

`aspect` **value/ Translation**
`aspect="PROGRESSIVE"` λe . Before(START(e),T_0) $\wedge$ Before(T_0,END(e))
`aspect="PERFECTIVE"` λe . Before(END(e),RT)
`aspect="PERFECTIVE_PROGRESSIVE"` λe . Before(START(e),T_0) $\wedge$ Before(T_0,END(e)) $\wedge$ Before(END(e),RT)

Table 2: Translation table for the EVENT tag attribute `aspect`.

3.1.2. The TIMEX3 Tag

ISO-TimeML uses an adapted form of the TIDES 2002 standard (Ferro et al., 2002), called TIMEX3, for marking up descriptions of time points and intervals. In natural language, events are often temporally anchored to an underspecified moment or period. The temporal anchoring of events can be represented in such cases with the (polymorphic) Inside relation (where T2 stands for the underspecified moment or period):

```
<TIMEX3 tid="t2" type=TYPE value=VALUE
temporalFunction="TRUE" anchorTimeID="t1">
```
$\rightsquigarrow \lambda P . \lambda t_1 . \exists T_2 : \text{Inside}(t_1,T_2) \wedge P(T_2)$

The translation of TIMEX3 tags with specified starting points and end points is quite straightforward:

```
<TIMEX3 tid="t1" type=TYPE value=VALUE
beginPoint="t2" end="t3">
```
$\rightsquigarrow \lambda P . \lambda t_2 . \lambda t_3 . \exists T_1 : \text{START}(T_1) = t_2 \wedge \text{END}(T_1) = t_3 \wedge P(T_1)$

3.1.3. The TLINK Tag

A TLINK tag, used to anchor an event in time, is structured in ISO-TimeML as follows:

(11)
```
<TLINK eventInstanceID=e1 signalID=s1
relatedToTime=t1 relType=R />
```

Here, the attribute `relType` has values corresponding to the use of temporal prepositions such as *at, before, in, during*; these values correspond to temporal relations in the underlying temporal ontology. The translation of such a TLINK tag has the following form:

λ e. λ t. R'(ET(e),t)

where R' is the translation of the *relType* value. Table 3 exemplifies the translation of these values. 'Before' is the polymorphic temporal ordering relation between instants and intervals.

In its other main use in ISO-TimeML, to represent a temporal relation between two events, a TLINK tag is translated as:

λ e_1. λ e_2. R'(e_1, e_2)

where e_1 and e_2 correspond to the two related events and R' translates the value of the `relType` attribute (which has values like *when, while, after*).

relType value	**Translation**
BEFORE	$\lambda x . \lambda y$. Before(x,y)
AFTER	$\lambda x . \lambda y$. Before(y,x)
AT	$lambda$x. λy. x=y
INCLUDES	$\lambda T . \lambda e$. Before(START(T),START(e)) $\wedge$ $\wedge$ Before(END(e),END(T))
IS_INCLUDED	$\lambda T . \lambda e$. Before(START(e),START(T)) $\wedge$ $\wedge$ Before(END(T),END(e))
DURING	$\lambda e_1 . \lambda e_2$. Before(START(e_2),START(e_1)) $\wedge$ $\wedge$ Before(END(e_1),END(e_2))

Table 3: Translation table for some `relType` values of the TLINK tag.

3.1.4. The ALINK Tag

The different possible aspectual relations that can be marked up in an ALINK tag are encoded in the values of its `relType` attribute. Since an aspectual relation always seems to correspond to a thematic relation plus a temporal relation, we translate all ALINK tags to a formal representation of the form:

$\lambda e_1.\lambda e_2$. THEME(e_1,e_2) $\wedge$ τ

where τ is the temporal component that depends on the value of the `relType` attribute. Table 4 specifies the translations of the various `relType` values.

relType value	**Translation component**
INITIATES	ET(e_1) = START(e_2)
TERMINTATES	ET(e_1) = END(e_2) $\wedge$ $\neg$COMPLETED(e_2)
CULMINATES	ET(e_1) = END(e_2) $\wedge$ COMPLETED(e_2)
CONTINUES	Before(START(e_2),ET(e_1)) $\wedge$ $\wedge$ Before(ET(e_1),END(e_2))

Table 4: Translation table for the ALINK tag.

4. Combining Translations

In order to compositionally translate an entire ISO-TimeML annotation into FOL, we need to combine the translations of its subexpressions. This poses a problem, as he following example shows.

(12) Igor arrived at 11 AM.
Igor
`<EVENT eiid="e1" tense="PAST"`
`polarity="POS">`
arrived `</EVENT>`
`<SIGNAL sid="s1">`
at `</SIGNAL>`

```
<TIMEX3 tid="t1" type="TIME"
value="T11:00">
11 AM </TIMEX3>
<TLINK eventInstanceID="e1"
signalID="s1" relatedToTime="t1"
reltype="AT" />
```

The respective translations of the event tag, the TIMEX3 tag, and the TLINK tag are as follows (where z_c as before indicates the contextually relevant time zone for the clock time):

λP. $\exists e_1 \in$ARRIVE: Before(ET(e_1),T_0) $\wedge$ P(e_1)
λQ. $\exists t_1$:clockTime(t_1,z_c)=11:00 $\wedge$ Q(t_1)
λe_1. λt_1. ET(e_1) = t_1

We would like to combine these representations, and in this case that's quite simple. However, the simplicity of the example is deceptive. When we consider a more complex example, such as *Amy was happy when Igor arrived before 11 AM*, then we get two translations of event tags and we must make sure that the translation of the TLINK tag is combined with that of the ARRIVE event, not with that of the REJOICE event. This is an instance of the problem of defining a compositional translation, pointed out above. Here, the problem is that the translations of the event- and TLINK tags have lost the linking information captured in the XML tags by the values of the eventInstance and relatedToTime attributes; the use of the same variables e_1 and t_1 in the translations of the tags only optically preserves the linking information; formally the names of these variables are insignificant.

We resolve this problem by keeping track of the linking information in the annotations and reformulating all translations as using intermediate representations in the form of triples

$$< ci, di, \varphi>$$

where *ci* (the 'combination index') contains XML identifiers such as the values of the eventInstance and relatedToTime attributes, for keeping track of the ISO-TimeML tags whose translations should be combined with the present representation, and where *di* (the 'derivation index') contains XML identifiers like the value of the eiid attribute in an event tag; this keeps track of which translations of ISO-TImeML subexpressions have already been used in the translation.

After translating the various tags in terms of such triples, the rest of the translation process consists of combining these triples, until a triple has been constructed whose combination index is empty and whose derivation index indicates that all the ISO-TimeML subexpressions have been linked together. For the combination of these triples we use a number of formal operations which are defined in the next subsection.

4.1. Combination operations

The operations that we use for combining the translations of ISO-TimeML subexpressions involve a few formula-manipulation operations defined in (Bunt, 2007). The most important one is a type of function application called *late unary application*, where a one-argument function is applied to an argument expression of the form $\lambda x_1,...,x_k$. E(x_1,...,x_k). The definition of this operation, designated by '$\Box$', is as follows:

$$F \Box \lambda x_1,...,x_k\ \lambda a . E = \lambda x_1,...,x_k\ F(\lambda a . E)$$

This operation and the others that we will describe below have to be extended to triples. In what follows, we will use the same symbols for the operations when applied to triples as when applied to formulas, except in the definitions where the subscript '3' is used to make clear that an operation is applied to triples. (We will use '·' to indicate concatenation of lists, and' -' subtraction of lists.) For late unary application the triple-definition is:[2]

$<ci_1, li_1, \varphi_1> \Box_3 <ci_2, li_2, \varphi_2> =$
$< ci_2\text{-}li_1, li_2 \cdot li_1, \lambda x_1,...,x_{k-1} . \varphi_1 (\lambda x_k \Box \varphi_2)>$

Second, an operation called *lambda insertion-application* (designated by $\oplus$) is defined, which combines a lambda abstraction λa.F, where F is a function expression, with an expression of the form $\lambda x_1,..,x_k$. E_1 $\exists$z : E_2 into $\lambda x_1,..,x_k$. λa . E_1 $\exists$z : F(z) $\wedge$ E(z).

In terms of triples:[3]

$<ci_1, li_1, \varphi_1> \oplus_3 <<>, li_2, \varphi_2> =$
$<ci_1\text{-}li_2, li_1 \cdot li_2, \varphi_1 \oplus \varphi_2>$

A variant of this operation, designated by $\oplus$', swaps the order of its arguments in application, and is defined as follows, with its obvious extensions to triples:

$$(\lambda x_1 . \lambda x_2 . F) \oplus' A = (\lambda x_2 . \lambda x_1 . F) \oplus A$$

A third operation, called *cross-application* (designated by $\otimes$), merges two expressions of the form λv . $\exists$x : E_1(v,x) $\wedge$ E_2 and λw . $\exists$y : E_1(y,w) $\wedge$ E_3 into $\exists$x $\exists$y : E_1(y,x) $\wedge$ E_2 $\wedge$ E_3.
In terms of triples:[4]

$<ci_1, li_1, \varphi_1> \otimes_3 < < >, li_2, \varphi_2> =$
$<<> li_1 \cdot li_2, \varphi_1 \otimes \varphi_2>$

Finally, an operation called *merge-application* (designated by $\odot$), is defined for any two representations E1 = $<ci_1, di_1, \alpha>$ and E2 = $<ci_2, di_2, \lambda z . \beta>$, where the set of first elements in the pairs constituting di_1 equals the set of identifiers in ci_1; β is not of the form λx..., and the length of the sequence of λ-abstractions in E2 equals the length of the list di_2. If α is a formula of the form γ Qz δ, where Q is a (generalized) quantifier, then the logical formula resulting from merge-application is γ Qz [λz . β](z) $\wedge$ δ.

In terms of triples: [5]

$$<<>, li_1, \varphi_1> \odot_3 <ci_2, li_2, \varphi_2>= <ci_2\text{-}li_1, li_2 \cdot li_1, \varphi_1 \odot \varphi_2>$$

These operations can be applied in any order to any triples that satisfy the properties required in the definitions of the operations, without any further constraints, thus ensuring the compositionality of the process. In the next subsection we will give some examples to illustrate the process.

4.2. Worked examples

(13) *Igor arrived at 11 AM.*

---

[2]A condition on the applicability of the operation $\Box_3$ is that the combination index ci_2 of the second operand has the form $ci_2' \cdot li_1$.

[3]A condition on the applicability of the operation $\oplus_3$ is that the combination index ci_1 of the first operand has the form $ci_1' \cdot li_2$.

[4]A condition on the applicability of the operation $\otimes_3$ is that the derivation index ci_1 of the first operand is a sublist of the derivation index li_2 of the second operand.

[5]See footnote 2.

We considered the ISO-TimeML annotation of this example in the previous subsection (see (11)). We describe the translation step by step. The TIMEX3 tag and the TLINK tag:

$\mathbf{T'} = <<>, <t1>, \lambda P.\ \exists t_1\text{: clockTime}(t_1,z_c)\text{=11:00} \wedge P(t_1)>$
$\mathbf{TL}_a\mathbf{'} = <<e1,t1>, <>, \lambda a.\ \lambda b.\ \text{ET}(a) = b>$

Combination of the two translations using late unary application:

$\mathbf{T'} \sqcup \mathbf{TL}_a\mathbf{'}$ =
$<<e1>, <t1>, \lambda a.\ \exists t_1\text{: clockTime}(t_1,z_c)\text{=11:00} \wedge \text{ET}(a)=t_1>$

Translation of the EVENT tag:

$\mathbf{E'} = <<>,<e1>,\lambda Q.\ \exists e_1 \in \text{ARR: Before(ET}(e_1),T_0) \wedge Q(e_1)>$

The EVENT translation is combined with that of the combination of the TIMEX3 tag and the TLINK tag using late unary application, which delivers the desired end result:

$\mathbf{E'} \sqcup (\mathbf{T'} \sqcup \mathbf{TL}_a\mathbf{'})$ =
$<<>, <t1,e1>, \exists e_1 \in \text{ARRIVE} : \exists t_1 : \text{clockTime}(t_1,z_c) = \text{11:00} \wedge \text{ET}(e_1) = t_1 \wedge \text{Before(ET}(e_1),T_0)>$

Next we consider an example with two temporally ordered events:

(14) *Amy left before Igor arrived.*
Amy
<EVENT eiid="e1" tense="PAST"
polarity="POS">
left </EVENT>
<SIGNAL sid="s1">
before </SIGNAL>
Igor
<EVENT eiid="e2" tense="PAST"
polarity="POS">
arrived </EVENT>.
<TLINK eventInstanceID="e1"
signalID="s1"
relatedToEventInstance="e2"
reltype="BEFORE" />

The two EVENT tags:

$\mathbf{E1'} = <<>, <e1>, \lambda Q.\ \exists e_1 \in \text{LEAVE: Before(ET}(e_1),T_0) \wedge Q(e_1)>$
$\mathbf{E2'} = <<>, <e2>, \lambda Q\ .\ \exists e_2 \in \text{ARRIVE: Before(ET}(e_2),T_0) \wedge Q(e_2)>$

The TLINK tag:

$\mathbf{TL}_e\mathbf{'} = <<e1,e2>, <>, \lambda a\ .\ \lambda b\ .\ \text{Before}(a,b)>$

Combination of the translation of the second EVENT tag with that of the TLINK tag using late unary application:

$\mathbf{E2'} \sqcup \mathbf{TL}_e\mathbf{'}$ =
$<<e1>, <e2>, \lambda a\ .\ \exists e_2 \in \text{ARRIVE} : \text{Before(ET}(e_2),T_0) \wedge \text{Before}(a,e_2)>$

Combination of the translation of the first EVENT tag (*Amy left*) with that of the second EVENT tag plus the TLINK tag (*before Igor arrived*) using late unary application, gives the desired end result:

$\mathbf{E1'} \sqcup (\mathbf{E2'} \sqcup \mathbf{TL}_e\mathbf{'})$ =
$<<>, <e1,e2>, \exists e_1 \in \text{LEAVE} : \text{Before(ET}(e_1),T_0) \wedge \exists e_2 \in \text{ARRIVE} : \text{Before(ET}(e_2),T_0) \wedge \text{Before}(e_1,e_2)>$

We finally consider an example with three related events, two of which have an aspectual relation and two a temporal ordering relation.

(15) *Amy started to laugh when Igor arrived.*

Amy
<EVENT eiid="e1" tense="PAST"
polarity="POS">
started </EVENT>
to
<EVENT eiid="e2" tense="NONE"
vform="INFINITIVE" polarity="POS">
laugh </EVENT>
<SIGNAL sid="s1">
when </SIGNAL>
Igor
<EVENT eiid="e3" tense="PAST"
polarity="POS">
arrived </EVENT>.
<ALINK eventInstanceID="e1"
relatedToEventInstance="e2"
reltype="INITIATES" />
<TLINK eventInstanceID="e3"
signalID="s1" relatedToEventInstance="e1"
reltype="IDENTITY" />

The translation of *Amy started to laugh*:

$\mathbf{E1'} \sqcup (\mathbf{E2'} \sqcup \mathbf{AL'})$ =
$<<>, <e1,e2>, \exists e_1 \in \text{START} : \text{Before(ET}(e_1),T_0) \wedge \exists e_2 \in \text{LAUGH} : \text{THEME}(e_2,e_1) \wedge \text{ET}(e_1) = \text{START}(e_2)>$

The ARRIVE event tag:

$\mathbf{E3'} = <<>, <e3>, \lambda Q\ .\ \exists e_3 \in \text{ARRIVE} : \text{Before(ET}(e_3),T_0) \wedge Q(e_3)>$

The TLINK tag:

$\mathbf{TL}_e\mathbf{'} = <<e1,e3>, <>, \lambda a\ .\ \lambda b\ .\ \text{ET}(a) = \text{ET}(b)>$

Combination of the translation of the third EVENT tag with the that of the TLINK tag using late unary application:

$\mathbf{E3'} \sqcup \mathbf{TL}_e\mathbf{'}$ =
$<<e1>, <e3>, \lambda a\ .\ \exists e_3 \in \text{ARRIVE} : \text{Before(ET}(e_3),T_0) \wedge \text{ET}(a) = \text{ET}(e_3)>$

Application of lambda-insertion application with swapping of variables:

$\mathbf{TL}_e\text{'} \oplus\text{'} (\text{E1'} \sqcup (\text{E2'} \sqcup \text{AL'}))$ =
$<<e3>, <e1,e2>, \lambda b\ .\ \exists e_1 \in \text{START} : \text{Before(ET}(e_1),T_0) \wedge \exists e_2 \in \text{LAUGH} : \text{THEME}(e_2,e_1) \wedge \text{ET}(e_1) = \text{START}(e_2) \wedge \text{ET}(e_1) = \text{ET}(b)>$

Application of cross-application to this representation for *Amy started to laugh* and the translation of *when Igor arrived* gives the desired end result:

$(\mathbf{E3'} \sqcup \mathbf{TL}_e\mathbf{'}) \otimes (\mathbf{TL}_e\mathbf{'} \oplus\mathbf{'}\ (\mathbf{E1'} \sqcup (\mathbf{E2'} \sqcup \mathbf{AL'})))$ =
$<<>, <e1,e2,e3>, \exists e_1 \in \text{START} : \text{Before(ET}(e_1),T_0) \wedge \exists e_2 \in \text{LAUGH} : \text{THEME}(e_2,e_1) \wedge \text{ET}(e_1) = \text{START}(e_2) \wedge \exists e_3 \in \text{ARRIVE}: \text{Before(ET}(e_3),T_0) \wedge \text{ET}(e_1) = \text{ET}(e_3)>$

5. Discussion and Conclusions

The method described in this paper enables a larger part of ISO-TimeML to be formally interpreted than the ITL approach, including the interpretation of tense and aspect, the treatment of durations, and that of calendar years, clock times, and so on. A treatment of calendar years and the like in an ITL-based semantics would probably not be hard, adding predicates applicable to certain temporal intervals as we have have done here. It would be more difficult to extend would be difficult to extend the ITL-based semantics with the interpretation of tense and aspect, since tense interpretation for instance requires the representation of event times (as temporally related to speech times and reference times), which is a property of events and thus necessitates the availability of events as such. Even more difficult would be the addition of durations, since this requires new concepts (temporal units and amounts of time, defining equivalence classes of pairs of a temporal unit and a numerical value) to be added to the underlying ontology.

More important from a theoretical point of view, is that we have specified a fully compositional interpretation of ISO-TimeML. This has been achieved at the price of making use of more complex intermediate representations, but has, besides the obvious theoretical importance, the advantage of allowing a very flexible translation process, which consists of a number of operations that can be applied in any order.

The attempt to formally interpret ISO-TimeML annotations has also revealed interesting interferences with the annotation of other semantic information, such as semantic roles and quantification. As long as semantic annotation is restricted to temporal annotation only, it may be reasonable to annotate the relations between events for which ISO-TimeML uses SLINK structures in the temporal annotation language, but these relations are not really temporal in nature and would be better treated as semantic role relations which have certain temporal implications. Also, aspectual relations, as captured in ALINK tags, are by their very nature a combination of thematic and temporal relations. Temporal quantification does not have a fully satisfactory treatment in ISO-TimeML, and indeed this only seems possible by taking quantification into account more generally.

For ISO-TimeML interpretation only, it might be feasible to cast the formal semantics in terms of a description logic like OWL-DL; however this would restrict the extensibility of the approach. An important aspect of the ISO-TimeML semantics outlined in this paper is that it has a richer underlying ontology than Interval Temporal Logic, including events and nontemporal individuals, which makes it possible to extend the approach to the semantic annotation of other information related to events. This would notably include the roles that the participants in an event play ('semantic roles'), as well as other properties of such participants, such as referential relations among participants in different events, and aspects of quantification for dealing with cases where sets of participants are involved in sets of events. The possibilities in this direction are explored in (Bunt, 2007) and Bunt & Overbeeke (2008).

References

Allen, J. (1984). A General Model of Action and Time. *Artificial Intelligence*, 23-2.

Bach, E. (1981). On Time, Tense, and Aspect: An Essay in English Metaphysics. In Cole, P., editor, *Radical Pragmatics*. Academic Press, New York.

Bunt, H. (1985). *Mass Terms and Model-Theoretic Semantics*. Cambridge University Press.

Bunt, H. (2007). The Semantics of Semantic Annotation. In *Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation (PACLIC21)*, pages 13–29, Korean Society for Language and Information.

Bunt, H. and Overbeeke, C. (2008). Towards formal interpretation of semantic annotation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech. ELRA.

Bunt, H. and Romary, L. (2002). Towards Multimodal Content Representation. In Choi, K. S., editor, *Proceedings of LREC 2002, Workshop on International Standards of Terminology and Language Resources Management*, pages 54–60, Las Palmas, Spain. Paris: ELRA.

Ferro, L., Gerber, L., Mani, I., Sundheim, B., and Wilson, G. (2002). *Instruction Manual for the Annotation of Temporal Expressions*. MITRE Washington C3 Center, McLean, Virginia.

Han, B. and Lavie, A. (2004). A Framework for Resolution of Time in Natural Language. *TALIP Special Issue on Spatial and Temporal Information Processing*, 3.

Hobbs, J. and Pan, F. (2004). An Ontology of Time for the Semantic Web. *TALIP Special Issue on Spatial and Temporal Information Processing*, 3-1:66–85.

Ide, N. and Romary, L. (2004). International Standard for a Linguistic Annotation Framework. *Natural Language Engineering*, 10:211–225.

ISO (2007). *Language Resource Management - Semantic Annotation Framework (SemAF) - Part 1: Time and Events*. Secretariat KATS. ISO Report ISO/TC37/SC4 N269 version 19 (ISO/WD 24617-1).

Katz, G. (2007). Towards a Denotatial Semantics for TimeML. In Schilder, F., Katz, G., and Pustejovsky, J., editors, *Annotation, Extraction, and Reasoning about Time and Events*. Springer, Dordrecht.

Lee, K. (2008). Against a Davidsonian Analysis of Copula Sentences. In Kadowaki, M. and Kawahara, S., editors, *NELS 33 Proceedings*.

Pratt-Hartmann, I. (2007). From TimeML to Interval Temporal Logic. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7)*, pages 166–180, Tilburg, Netherlands.

Pustejovsky, J., Castano, J., Ingria, R., Gaizauskas, R., Katz, G., Saurí, R., and Setzer, A. (2003). TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, pages 337–353, Tilburg, Netherlands.

Pustejovsky, J., Knippen, R., Littman, J., and Saurí, R. (2007). Temporal and Event Information in Natural Language Text. In Bunt, H. and Muskens, R., editors, *Computing Meaning*, volume 3, pages 301–346. Springer.

Reichenbach, H. (1947). *Elements of Symbolic Logic*. Macmillan, New York.