Revisiting the ISO-TimeML abstract syntax

Harry Bunt
Tilburg U., Netherlands
harry.bunt@uvt.nl

Alex Fang
City U. of Hong Kong
alex.fang@cityu.edu.hk

Kiyong Lee
Korea U. / Seoul
ikiyong@gmail.com

Volha Petukhova
Saarland U., Germany
V.V.Petukhova@gmail.com

Purificação Silvano
Univ. of Porto, Portugal
puri.msilvano@gmail.com

James Pustejovsky
Brandeis U., Waldham
jamesp@brandeis.edu

Abstract

This paper describes some of the ongoing work within the ISO preliminary work item PWI 254617-17, 'Interlinking of annotations'. This PWI investigates the possibilities and problems of combining annotations made with different annotation schemes. using the 'interlinking' approach (Bunt, 2024) applied to different parts of the multi-part standard ISO 24617, 'Semantic annotation framework'. This paper focuses on the combination of ISO-TimeML and QuantML at the level of abstract syntax. A new version is defined for the ISO-TimeML abstract syntax specification and how it relates to the concrete (XML-based) syntax as a basis for this combination. As a side-effect, some issues in the use of ISO-TimeML come to light that could be relevant for a possible future second edition of this standard.

1 Introduction

1.1 Background

Existing semantic annotation schemes are often focused on a specific type of semantic information, such as TimeML (Pustejovsky, 2003) on time and events, SpatialML (Mani et al., 2010) on spatial information, DAMSL (Allen & Core, 1997) and DIT++ (Bunt, 2007) on dialogue acts, and PDTB (Prasad et al, 2008; 2019) on discourse relations. The ISO Semantic Annotation Framework (ISO 24617, 'SemAF') was set up as a multi-part standard, with different parts focusing on different semantic domains.

Developing the SemAF standard as a set of separate sub-standards has proved useful, as it is better feasible to develop an annotation schema for a well-delineated semantic domain. The first two parts of SemAF, informally known as 'ISO-TimeML' and 'DiAML', are successful examples of the application of this approach, as the annotation of time and events is clearly separable from the annotation of dialogue acts. However, some of the semantic domains are not entirely disjoint; some semantic phenomena play a role in more than one sub-standard.

For example, the expression *"every Monday"* quantifies over mondays. Being a temporal expression, ISO-TimeML provides an annotation of this expression, including an indication of its quantifying character. ISO-TimeML has only a rudimentary treatment of quantification, however (Bunt & Pustejovsky, 2010), while it is the focus of SemAF part 12, QuantML This paper reports on activities within the ISO preliminary work item PWI 254617-17, Interlinking of annotations. This PWI investigates the possibilities and problems of combining annotations made with different annotation schemes, using the interlinking approach introduced in (Bunt, 2024). In particular this approach seems interesting for combining annotations made according to different parts of SemAF, which focus on different types of semantic information. On this approach, links are added between elements of different annotations for indicating that these elements correspond to the same entities mentioned in the primary data. This allows annotations of the same entities with different types of information, and therefore facilitates the merge of the semantic information in the respective annotations.

When considering the combination of annotations from different SemAF parts, we have to consider all three interrelated levels distinguished in the architecture of a SemAF scheme (see Fig. 1): (1) the concrete syntax, conventionally with an XML-based reference format, (2) the abstract syntax, expressing the semantically relevant information of the annotations in the form of pairs, triples, and other set-theoretical structures (and interrelated with the concrete syntax through encoding and decoding functions), and (3) the semantics of the annotations.

At the level of concrete syntax, interlinking consists of adding identity links between components of representations from different schemes, indicating that the same stretch of primary data is annotated

Proceedings of the 21st Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-21), pages 12–20
September 24, 2025,

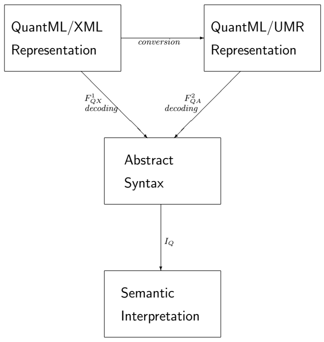

Figure 1: Levels and interrelations in SemAF annotation schemes.

from different points of view. At the level of abstract syntax, the structures of the interlinked annotations are combined into a single set-theoretical structure. At the level of semantics, finally, the semantic interpretation function describes the meaning of the joint abstract syntax expressions.

At the level of concrete syntax, the addition of identity links between two (or more) representations is a straightforward matter, although there may be some issues in the identification and use of markables, but the real challenges lie at the levels of abstract syntax and semantics. In particular, sitting in between the levels of concrete representation and semantics, the combination of annotations at the level of abstract faces a dual challenge.

On the one hand, the expressions at that level should have a systematic encoding-decoding relation to each of the respective concrete representations, and on the other hand they should capture the information contained in the combined annotations in a way that allows their joint semantic interpretation.

Since ISO-TimeML (ISO-24617-1:2012 Time and events) and QuantML (ISO 24617-12:2025 Quantification) are two of the best developed and most complex SemAF parts, a sensible strategy would seem to first explore the possibilities of combining their respective annotations, in particular at the level of abstract syntax. QuantML has a fully developed abstract syntax, but ISO-TimeML, being the oldest SemAF part, has an abstract syntax that is not fully specified and at some points lacks conceptual clarity.

This paper therefore revisits the ISO-TimeML abstract syntax, aiming to develop a full, conceptually clear specification for the concrete (XML-based) representations as they are. Section 2 takes a step in that direction. Since the abstract syntax is required to allow systematic decoding of concrete representations, the adequacy of any revised version can be tested by specifying the decoding function. Section 3 is therefore devoted to the mapping of concrete representations to expressions of the abstract syntax. Section 4 indicates the next steps towards fully specified interlinked ISO-TimeMML- and QuantML-annnotations. formulating a version of the abstract syntax and the semantics of ISO-TimeML in the same style as QuantML

1.2 ISO-TimeML

ISO-TimeML distinguishes three types of temporal objects: instants, dates, and periods. With respect to instants, the ISO 24617-1:2012 specification document notes that in reality, nothing happens in infinitesimally small time; every event or state that occurs in reality (or in someone's mind) requires more than zero time, although natural languages offer speakers the possibility to express themselves as if something occurs at a precise instant (as in *"I will call you at twelve oclock"*. Such an instant is often associated with the beginning of an event, as in this example. The explicit mentioning of the start of an event, as in *"I was sad when Mary started to cry"*, illustrates the same phenomenon. Punctual events are associated with precise instants, as in the example *"Gates will close at 9:25."*

The notion of a precise instant is similar to that of a point in mathematics. Euclid defined a point as a spatial entity that which has no parts. In other words, a point is an indivisible spatial object with zero length, breadth, and height. Natural language speakers refer to instants as points on a timeline, as intervals of zero length, even though they probably know that such intervals do not really exist. In everyday language, instants are referred to with the precision of minutes, as in *"Its five past twelve"*. It is therefore appropriate to consider such intervals as instants in the ISO-TimeML abstract syntax.

Fully specified references to instants consist of a (fully specified) date and time. A fully specified date contains the specification of (1) a year, (2) a month and (3) a day number, or (2) a week number and a (3) day name. In practice, reference to instants is often underspecified, such as *"Monday at two"*, intended to be understood as next Monday at two p.m. or as last Monday at two p.m. depending on the context (which also allows to infer the year and the week). Underspecification is represented in ISO-TimeML by using the character 'X' in values of the @value attribute. (Examples below.)

Instants are annotated in the ISO-TimeML reference representation format by <TIMEX3> expressions with @type="TIME"; dates by expressions of type "DATE"; periods by expressions with @type="DURATION".

From a semantic point of view, year numbers, calendar month names, and day names function like proper names. Just like *"James"* refers to a contextually particularly salient person named James, *"Tuesday"* refers to the contextually most salient day named 'Tuesday'. Year numbers (*"1984"*) refer to certain time intervals independent of context, just like country names (*"Denmark", "Japan"*) refer to geopolitical regions independent of context.

The specification of the ISO-TimeML abstract syntax is best done with (a) the semantics in mind and (b) specifying the decoding function that relates it to the concrete syntax - which in turn calls for a precise specification of optional attributes and default values in the concrete syntax. In (Bunt, 2018) several forms of optionality are distinguished: (a) semantic, i.e. a certain type of annotation structure may contain such a component, but does not have to for being interpretable; (b) a component that does not have to be specified in the concrete syntax, since it has a default value in the abstract syntax; (c) a component in the concrete representation that has no semantic interpretation. These distinctions are useful for a clear formulation of the abstract syntax in relation to the concrete syntax and semantics.

The <EVENT>, <TIMEX3> and <TLINK> elements all pose problems for the distinction between required and optional attributes. For example, the @relatedToEvent attribute in <TLINK> is not applicable if a @relatedToTime value is specified, and vice versa. Also, the attributes @tense and @aspect in <EVENT> elements are applicable only if @pos="VERB", and @beginPoint is applicable only if @type="DURATION".

The possible values of the @relType attribute in <TLINK> elements specify temporal relations between events and/or temporal entities. The value "IDENTITY" is unusual in this respect, as it designates the identity of two events, rather than a temporal relation; it would seem to entail the temporal relation SIMULTANEOUS. It may be noted that SIMULTANEOUS, AFTER and BEFORE are all instances of the discourse relations Synchrony and Asynchrony, defined in the ISO standard for annotating discourse relations (ISO 24617-8:2016).

The conditional applicability of various attributes in elements of the concrete syntax means, in the 3-layer architecture of SemAF parts, that alements like <TIMEX3>, can correspond to several different structures in the abstract syntax.

'

2 Abstract Syntax

2.1 Overview

As in the case of other SemAF parts, the abstract syntax of ISO-TimeML has two components: (1) the specification of a store of primitive concepts, called the Conceptual Inventory, and (2) the recursive specification of the annotation structures that may be formed by combining primitive concepts or annotation structures to form set-theoretic structures like pairs and triples.

The structures defined by the abstract syntax come in two forms: (a) *entity structures*, i.e., structures that contain semantic information about a stretch of source data (a *markable*), and (b) *link structures*, which express semantic relations between two or more entity structures. An entity structure has the form of a pair ⟨markable, semantic information⟩; a link structure has the form ⟨'entity structure 1, entity structure 2, .. entity structure n⟩, semantic relation⟩. Entity structures are represented in the concrete syntax by XML elements that have a @target attribute whose value refers to the relevant stretch of source data.

2.1.1 Conceptual inventory

The minimal building blocks of ISO-TimeML annotation structures are constants. These fall into one of the five categories listed below. Constants that denote properties are unary predicates characterizing event types, event classes, tenses, aspects, polarity, and set-theoretic type. Natural numbers are used for capturing the information expressed in examples such as *"twice", "three times",* and *"double*". Rational numbers are needed for examples like *"half a day*".

1. Linguistic semantic properties: unary predicates, like 'occurrence', 'process', and 'past'.

2. Relations: binary predicates for expressing temporal relations, durations, numerical relations, subordination relations, and aspectual relations.

3. Named temporal concepts: calendar years, calendar months, calendar days, month numbers, week numbers, weekday numbers, and clock times.

4. Temporal units, like hours, days, weeks, months, and years.

5. Natural numbers and rational numbers.

2.1.2 Entity structures

The abstract syntax has entity structures for events and for temporal entities. An event structure is a 6-tuple <E, T_y, C, T, A, V>, consisting of an event predicate, an event type, an event class, a tense, an aspect, and a veracity.

Temporal entity structures fall into 6 categories: (1) instant, (2) date, (3) period, (4) set of any of these, (5) amount of time, and (6) a frequency. Items in these categories are all represented in the concrete syntax by <TIMEX3> expressions with different values of @type.

2.1.3 Instants

An instant structure, corresponding to a <TIMEX3> element of type TIME, is one of the following.

1. a pair ⟨ day, clock time⟩ Clock times are predicate constants designating a time on the clock, for example annotating '*"four p.m.*, which is represented in XML as a <TIMEX3> element with @value=T16:00. These predicate constants take the form of sequences of two numbers, followed by a colon symbol (':') followed by another sequence of two numbers. The first two numbers are 00, 01, ...24 and the last two 00, 01, ...59 (as in 16:00).

2. a triple ⟨instant, time amount, begin/end relation⟩ (*"half an hour before midnight"*).

3. a triple ⟨event, time amount, begin/end relation⟩ (*"ten minutes after the explosion"*)

4. a single clock time.

5. a pair ⟨ date structure, clock time ⟩.

2.1.4 Dates

A date structure is any (complete or incomplete) specification of a time interval of a length of one day by means of concepts related to the calendar, corresponding to a <TIMEX3> element of type DATE and is one of the following.

1. a triple ⟨year, month name or number, day name or number⟩ (*"December 25, 2024"* or *"2024-52-3"*).

2. a pair ⟨year, month⟩ (*"December 2024"*) or or ⟨year, season⟩ (*Spring 2025"*).

3. a pair ⟨month, day number⟩ (*"December 25"*).

4. a pair ⟨week, day name⟩ (*"Friday next week"*).

5. a predicate constant denoting a year, a month, or a day.(*"1984", "May", "Sunday", "labour day", "leap day", "the 25th"*).

Named temporal entities like *"Wednesday"* work as other proper names and definite descriptions; they refer to a contextually uniquely determined entity.

2.1.5 Periods

A period structure is one of the following structures, which specify a time interval that does not form a date.

1. a pair of two structures indicating the beginning and end points of a contiguous time interval, viz. ⟨instant, instant⟩ or ⟨date, date⟩, or ⟨period⟩(*"between two and five on January 1, 2025", "from May through September"*).

2. a triple ⟨t, t_A, R⟩ where t is an instant structure, indicating the beginning or end of a period, t_A is a time-amount structure indicating the length of the period, and R is 'before' or 'after' (*"the week before Christmas", "the week following May 1"*).

3. a triple ⟨e, t_A, R⟩ where e is an event structure, indicating the beginning or end of an event, t_A is a time-amount structure indicating the length of the period, and R is either "before" or"after" (*"two days before the attack", "a month after the cease-fire"*).

2.1.6 Time-amount structures

A time-amount structure is a triple ⟨'numerical relation, rational number, temporal unit⟩ (*"less than two hours"*).

2.1.7 Frequency structures

A frequency structure is a natural number or a pair ⟨natural number, temporal unit⟩.

2.1.8 Quantification structures

A quantification structure corresponds to a <TIMEX3> element of type SET. From a semantic point of view, such elements contain information about three aspects of a quantification: (1) a quantifier in the sense of classical logic, expressed by @quant values like EVERY and SOME, (2) a domain that the quantifier ranges over, indicated by the @value attribute, and (3) repetitions of an event indicated by the optional attribute @freq.

For the abstract syntax this means that a quantification structure is one of the following, where a domain is a set of days, weeks, months, or years:

1. a pair ⟨domain, quantifier⟩.
2. a triple ⟨domain, quantifier, frequency⟩.

2.2 Link structures

ISO-TimeML has link structures for (1) anchoring events in time; (2) temporal ordering of events, (3) ordering of periods, dates or instants relative to each other; (4) measuring a time interval; (5) specifying subordination relations between events; and (6) indicating aspectual relations between events.

a. Temporal anchoring: a triple ⟨event structure, temporal entity structure, anchoring relation⟩. The anchoring relation corresponds to a natural language expression like *"at", "in", "during"*.

b. Temporal event relations: a triple ⟨event structure, event structure, temporal relation⟩. Temporal relations are predicate constants corresponding to natural language expressions like *"while", "after", "just before"*.

c. Intra-time relations: a triple ⟨temporal entity, temporal entity, temporal relation⟩.

d. Time measurement, corresponding to the use of MLINK in the concrete syntax: a pair ⟨event structure, time-amount structure⟩ or a pair ⟨period structure, time-amount structure⟩.

e. Subordination structures, corresponding to the use of SLINK in the concrete syntax: a triple ⟨event structure, event structure, subordination relation⟩.

f. An aspectual link structure, corresponding to the use of ALINK, is a triple ⟨event structure, event structure, aspectual relation⟩.

3 Completeness and semantic adequacy

3.1 Requirements on abstract syntax

The abstract syntax of a markup language should meet two fundamental requirements (ISO 224617-5:2016, Principles of semantic annotation), First, it should be *complete* in the sense that for every representation structure of the concrete syntax an abstract annotation structure is defined. In other words, a decoding function (see Fig. 1) is a total function. Second, every abstract annotation structure should have a well-defined semantics. Regarding the first requirement, in this paper we present a specification of the decoding function of ISO-TimeML. Regarding the second requirement, we indicate the direction in which the semantic interpretation will go. Notes from the PWI 24617-17 project containing more details which will be made available in future project reports and follow-up papers.

3.1.1 Decoding: events and participants

The decoding function dF computes the entity structure of the abstract syntax that contains the semantically relevant information in a given concrete representation, abstracting away from other than semantic elements. Since an entity structure provides semantic information about a certain stretch of primary data, it always has the form ⟨m,s⟩, where m is a markable and s is semantic information. The use of markables in entity structure allows us to attach different semantic information to different occurrences of the same source words. This provides an opening for dealing with lexical ambiguities. In this paper we are not concerned with lexical disambiguation and simplify the presentation of abstract annotation structures by suppressing markables in the abstract syntax.

3.1.2 Decoding events

The decoding function *dF* is defined for <EVENT> elements as follows.

```
dF(<EVENT xml:id=e1 target=m1
     pred=P1 type=T1 class=C1
     tense=t1 aspect=a1/>)
= ⟨dF(P1), dF(T1), dF(C1), dF(t1),
   dF(a1)⟩
```

Example: *"Mary laughed"*.

```
dF(<EVENT xml:id="e1" target="#m1
     pred="laugh" type="occurrence"
     class="process" tense="past"
     aspect="none"/>)
= ⟨laugh, occurrence, process, past⟩
```

3.1.3 Decoding <TIMEX3> elements

a. Instants

A complete specification of an instant is formed by the complete specification of a date, which is formed by (1) a year plus (2) a month and a day number, or a week number and a weekday plus (3) a clock time. In ISO-TimeML these components are represented as parts of the string that forms the value of the @value attribute in a <TIMEX3> element of type TIME. The decoding function, which extracts the components from such strings, is defined as follows.

Example: *"July 5, 2012, at 4 p.m."*

```
dF(<TIMEX3 xml:id=t1 target="#m1 type="TIME"
       value="2012-07-05T16:00"/>)
= ⟨⟨ 2012, july, 5 ⟩, 16:00⟩
```

An instant can also be specified by describing its distance from another instant, as in *"Two hours before (December 31, 2024,) midnight"*. The definition of the decoding function for this type of specification is defined as follows:

```
dF(<TIMEX3 xml:id="t1" target="#m1"
          type="DURATION" value="PkU"
          beginPoint="#t2" endpoint="#t3"/>
       <SIGNAL xml:id="s1" target="#m2" pred="R"/>
       <TIMEX3 xml:id="t2" type="TIME"/>
       <TIMEX3 xml:id="t3" target="#m3"
          type="TIME" anchorTime="#t1"
       value="yvwz-mn-d1Tij:kl"/>)
= ⟨dF(#t3), dF(#t1), dF(R)⟩
= ⟨dF(yvwz-mn-d1Tij:kl), dF(PkU), dF(R)⟩
```

Example: *"Ttwo hours before December 31, 2024, midnight."*

```
Markables: m1 = two hours, m2 = before,
              m3 = December 31, 2024, midnight
dF(<TIMEX3 xml:id="t1" target="#m1"
        type="DURATION" value="P2H"
        beginPoint="#t2" endPoint="#t3"/>
     <SIGNAL xml:id="s1" target="#m2"
        pred="BEFORE"/>
     <TIMEX3 xml:id="t2" type="TIME"
        value="2024-12-31:T22:00"
        anchorTime="#t1"/>
     <TIMEX3 xml:id="t3" target="#m3"
        type= "TIME"
           value="2024-12-31:T24:00"/>)
= 2024, december, 31 ⟩, 24:00⟩, ⟨ 2, hour⟩, before⟩⟩
```

The XML representation used here follows the ISO 24617-1:2012 document, where DURATION expressions have both a value of the @value attribute, specifying the length of a time period, and values of the @beginPoint and @endPoint attributes. Specifying a temporal distance in this way may run into two problems: (1) if two of these three attributes have values, then the value of the third can be inferred, therefore assigning values to all three results in expressions which are either redundant or potentially inconsistent; (2) the expressive power is insufficient for representing durations such as *"less than two hours"*.

To resolve the latter problem, an attribute @length, could be introduced, whose value refers to the specification of an amount of time. To resolve the former pproblem, it would seem best to require only two of the three DURATION attributes to be specified, not all three.

Without changing the use of the @value attribute, it would seem preferable to use the <TIMEX3> and <TLINK> elements in combination with the relation I-BEFORE (immediately before).

b. Dates

The decoding of XML representations of dates is for the most part very similar to that of instants. A complete specification of a date consists either of the specification of a year, a month and a day number (as in *"December 25, 2024"*) or a year, a week number and a weekday number (as in *"2024-52-3"*). Such a structure denotes a specific, unique date in a context-independent fashion. As in the case of a complete explicit instant description, the decoding of the XML representation rests on the decoding of the value of the @value attribute.

The decoding of a fully specified date is as follows.

```
   dF(<TIMEX3 xml:id="t1" target="#m1" type="DATE"
      value="yvwz-mn-d1"/>)
= dF(yvwz-mn-d1)
= ⟨dF(yvwz), dF(mn), dF(d1)⟩
```

Example: *"July 5, 2012"*

```
dF(<TIMEX3 xml:id="t1" target= "#m1" type="DATE"
        value="2012-07-05"/>)
=⟨ 2012, july⟩
```

c. Periods

A contiguous time interval can be defined by the specification of a begin- and an end point (*"From two to five"*, *"From May through September"*).

Example: *"On New Years day I biked from ten to five."*

```
dF(<TIMEX3 xml:id="t1" target= "#m1" type=
        "DURATION" beginPoint="#t2" endpoint="#t3"
        value="P7H"/>
     <SIGNAL xml:id="s1" target="#m4" pred="FROM/>
     <SIGNAL xml:id="s2" target="#m6" pred="TO"/>
     <TIMEX3 xml:id="t2" target="#m2" type="TIME"
        value="2025-01-01T14:00"/>
     <TIMEX3 xml:id="t3" target= "#m3" type="TIME"
```

value="2025-01-01T17:00"/>)
=⟨ dF(#t2), dF(#t3) ⟩
=⟨⟨january,1⟩,10:00⟩,⟨january,1⟩,17:00⟩⟩

(To link this interval to the biking event, two additional <TLINK> elements are needed as follows, where "e1" is the identifier of the event:
<TLINK eventID="e1" relatedToTime="#t2"
signalID="#s1" relType="BEGUN_BY"/>
<TLINK eventID="#e1" relatedToTime="#t3"
signalID="#s2" relType="ENDED_BY"/>
See also ISO 24617-1:2012, p. 84.))

A period can also be specified in a relative way by a beginning or an end point and the amount of time that separates that point from (a) a given instant or date (*"for two hours after midnight"* or *"the week beginning May 5"*) or (b) from the beginning or end of an event (*"two weeks before the attack"*).

Example: *"(for) two hours after midnight"*

dF(<TIMEX3 xml:id="t1" target= "#m2"
type="DURATION" beginPoint="#t2"
value="PT2H"/>
<SIGNAL xml:id="s1" target="#m4'
pred="AFTER"/>
<TIMEX3 xml:id="t2" target= "#m1" type=
"TIME" value="XXXX-XX-XXT00:00"/>)
= ⟨dF(#t2), dF(#t2), dF("AFTER")⟩
=⟨⟨2,hour⟩,00:00, after⟩

3.2.3 Time amounts

An amount of time is represented by a <TIMEX3> element of type DURATION that has neither a begin point nor an end point specified, but which has a @value attribute with a value of the form 'PnU', where'P' as before stands for 'period', 'n' for a real number, and 'U' for a temporal unit (like second, minute, hour, day,). Such a <TIMEX3> element does not identify a specific period and can be viewed as a case of underspecification, denoting the set of all periods of the specified length. Its use is to link an event to an amount of time through an <MLINK> element with @relType="MEASURES". The decoding of such a <TIMEX3> element is defined as follows.

dF(<TIMEX3 xml:id=ti target=mj
type=DURATION value=PnU
tense=t1 aspect="'NONE"/>)
= ⟨dF(n), dF(U)⟩

Example: *"John taught for three hours"*

dF(<EVENT xml:id="e1" target= "#m2" pred="teach"
class="OCCURRENCE" type="PROCESS"
<TIMEX3 xml:id="t1" target= "#m3" type=
"DURATION" value="P3H"/>
<MLINK eventID="e1" relatedToTime="#t1"
relType="MEASURES"/>)
= ⟨ dF(#e1), dF(#t1), dF(MEASURES) ⟩
= ⟨⟨ teach, occurrence, process, past,⟩
⟨ 3, hour ⟩, duration ⟩

3.2.4 Frequency structures

Example: *"twice a month"*

dF(<TIMEX3 xml:id="t3" type="SET" value=
"P1M" freq="2X"/>)
=⟨dF(XXXX-XX), dF(EVERY), dF(2X)⟩
=⟨month, all, 2⟩

For the treatment in ISO-TimeML of repetitions rather than frequencies, as in *"John kissed Mary twice"* see Section 4.

3.2.5. Quantification structures

<TIMEX3> expressions with type="SET" have a @value and a @quant attribute, and optionally a @freq attribute.

dF(<TIMEX3 xml:id="t1" type="SET"
value="PnU" quant="q1"/>)
= ⟨dF(PnU), dF(q1) ⟩
=⟨⟨dF(n), dF(U)⟩, dF(q1)⟩

Example: *"Every Monday"*

dF(<TIMEX3 xml:id="t1" type="SET" value=
"XXXX-WXX-1 quant="EVERY"/>)
= ⟨dF(XXXX-WXX-1), dF(EVERY)⟩
= ⟨monday, all⟩

3.2 Link structures

a. Event time relations

<TLINK> elements that specify information about the time of occurrence of an event have the attributes @eventID, @relatedToTime, and @relType. Their decoding is defined as follows:

dF(<TLINK eventD="#e" relatedToTime="#t"
signalID="#s" relType="R"/>
<SIGNAL xml:id="s" pred="R"
= ⟨dF(#e), dF(#t), dF(R)⟩

b. Temporal discourse relations

<TLINK> elements that specify information about the temporal relation between two events have the attributes @eventID and @relatedToEvent, plus a @relType attribute whose value represents the relation. Decoding:

dF(<TLINK eventD= "#e1" relatedEvent="#e2"
signalID="#s" relType="R"/>

<SIGNAL xml:id="*s*" pred="R1"/>
= ⟨dF(#e1), dF(#e2), dF(R)⟩

c. Relations between temporal entities
Relations between two times, dates, or periods, possibly quantified, as in *"twenty minutes every Monday"*, are represented by <TLINK> elements with the attributes @timeID, @relatedToTime, and @relType. Their decoding is like in the above cases a and b.

4 Next steps and issues for further study

Revisiting the ISO-TimeML abstract syntax, we are in fact applying the CASCADES method for developing or improving an annotation schema (Pustejovsky, Bunt & Zaenen, 2017). This means that the specification of an annotation scheme consists of four consecutive stages: (1) establishment of a metamodel, (2) - (3) specification of concrete and abstract syntax and the decoding function that connects them, (4) definition of semantic interpretation. Feedback loops go back from any stage to any previous stage.

In reverse engineering mode, one can start at any of the four stages and follow steps forward or backward to ensure inter-stage consistency. In the case of revisiting the ISO-TimeML abstract syntax, we take the existing concrete syntax and the metamodel on which it is loosely based for granted. Starting at the abstract syntax stage (3), the next step forward is the specification of the semantics; the most relevant feedback step is ensuring the consistency between abstract and concrete syntax, which is accomplished by specifying a decoding function.

Regarding the next step forward, we have started the definition of a revised compositional semantics for the expressions of the (revised) abstract syntax as outlined above. Inspired by the specification of the semantics of QuantML (see Bunt, 2023), this semantics takes the form of a recursive function that interprets annotation structures as second-order DRSs (Kamp & Reyle, 1993).

Regarding the next step backward, while specifying the decoding function we have noted some unclear aspects, limitations and gaps in the concrete syntax as specified in ISO 24617-1:2012 and in the guidelines for its use. This could be of interest for a possible future update of the specification and the guidelines. Some these issues are the following.

1. In ISO-TimeML, periods are assumed to be contiguous. This is not always realistic, in view of sentences such as *"I studied from nine to five"* - there must have been interruptions in this interval. This calls for the introduction of possibly discontinuous intervals.

2. <TIMEX3> elements of type DURATION have a length which is specified by the @value attribute. This suggests that the duration of an event is conceived as a period of a certain length. Although technically possible in many cases, this does not seem general enough. It is not clear how an example like *"In Hong Kong it's 7 hours later than in Amsterdam* could be annotated.

3. The @quant attribute in <TIMEX3> elements of type SET is limited in the variety of generalized quantifiers that it allows. Examples like *"More than 3000 students protested"* cannot be annotated. See also Bunt & Pustejovsky (2010) for a discussion of the annotation of quantification over times and events.

4. Similarly, the use of values like "2X" for the @value attribute in order to represent *"twice"* does not permit to represent 'generalized' repetitions and frequencies like *"more than three times", "at least twice a week"*.

5. Some attributes in <EVENT> are either syntactic or lexical in nature. The attribute @pos (part-of-speech) is obviously not semantic; @tense and @aspect are only partly semantic, as seems to be reflected in the fact that they apply only to events expressed by verbs. Moreover, for verbs the values of the attributes @type and @class can typically be obtained from the lexical information of the verb and do not need to be annotated.

6. A similar issue concerns the use of <SIGNAL> elements. As the examples in Section 3.3 illustrate, <TLINK> elements that express a temporal relation have @signalID as one of their attributes, whose value refers to a <SIGNAL> element. This attribute seems semantically superfluous, since its @pred value, which specifies the temporal relation, is also expressed in the value of the @relType attribute in the <TLINK> element.

7. An <MLINK> element always has @relType = "MEASURES". Since this is always the

case, it would seem superfluous to annotate it as such.

8. According to the ISO 24617-1 document, the possible values of the @value attribute for dates, times, pertiods, frequencies and quantifications are taken from the TIDES scheme (Ferro et al., 2003), which follows the representation of dates and times in the ISO 8601 standard. Whether this Is desirable also for annotating period lengths, quantifications, frequencies and repetitions deserves further study.

Most importantly, the outcome of this study is a version of the ISO-TimeML abstract syntax and the decoding function from concrete to abstract syntax that promise to provide a solid basis for the further interlinking of ISO-TimeML annotations and QuantML annotations.

References

J. Allen and M. Core. 1997. *DAMSL: Dialogue Act Markup in Several Layers (Draft 2.1). Technical Report*. University of Rochester, Rochester, NY.

H. Bunt. 2009. The DIT++ taxonomy for functional dialogue markup. In *Proceedings of AAMAS 2009 Workshop 'Towards a Standard Markup Language for Embodied Dialogue Acts'*, pages 13–24, Budapest.

H. Bunt. 2019. An annotation scheme for quantification. In *Proceedings 14th International Conference on Computational Semantics (IWCS 2019)*, pages 31–42, Gothenburg, Sweden.

H. Bunt. 2020. The annotation of quantification: The current state of ISO 24617-12. In *Proceedings of the 16th Joint ISO-ACL Workshop on Interoperable Semantic Annotation (ISA-16)*, Marseille.

H. Bunt. 2024. Combining Annotation schemes through Iinterlinking. In *Proceedings of the 20th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-20)*, pages 83 – 95, Turin, Italy.

H. Bunt and J. Pustejovsky. 2010. Annotating temporal and event quantification. In *Proceedings ISA-5, Fifth Joint ACL-ISO Workshop on Interoperable Semantic Annotation*, pages 15–22. City University of Hong Kong.

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

Harry Bunt. 2018. Downward Compatible Revision of Dialogue Annotation. In *Proceedings 14th Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-9)*, page 241, Santa Fe, New Mesico, USA.

R. Cooper. 1983. *Quantification and syntactic theory*. Reidel, Dordrecht.

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

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

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

ISO. 2024. *Preliminary Work Item proposal PWI 24617-17, Interlinking*. International Organisation for Standardisation ISO, Geneva.

ISO. 2025. *ISO24617-12:2025, Language Resource Management: Semantic Annotation Framework (SemAF) - Part 12: Quantification*. International Standard. International Organisation for Standardisation ISO, Geneva.

H. Kamp and U. Reyle. 1993. *From Discourse to Logic*. Kluwer Academic Publishers, Dordrecht.

K. Lee, H. Bunt, J. Pustejovsky, A. Fang, and C. Park. 2025. Representing ISO Annotated Dynamic Information in UMR. In *Proceedings 6th International Workshop on designing meaning representations (DMR 2025)*, Prague.

I. Mani, C. Doran, D. Harris, J. Hitzeman, R. Quimby, J. Richer, B. Wellner, S. Mardis, and S. Clancy. 2010. SpatialML: Annotatopn, Resources, and Evaluation. *Language Resources and Evaluation*, 44 (3):263–280.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The Penn Discourse Treebank 2.0. In *Proceedings 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco.

J. Pustejovsky, J. Castano, R. Ingria, R. Gaizauskas, G. Katz, R. Saurí, and A. Setzer. 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.