INTERNATIONAL STANDARD

ISO 24617-1

First edition
2012-01-15

Language resource management — Semantic annotation framework (SemAF) —

Part 1: Time and events (SemAF-Time, ISO-TimeML)

Gestion des ressources langagières — Cadre d'annotation sémantique (SemAF) —

Partie 1: Temps et événements (SemAF-Time, ISO-TimeML)

Reference number
ISO 24617-1:2012(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 24617-1 was prepared by Technical Committee ISO/TC 37, *Terminology and other language and content resources*, Subcommittee SC 4, *Language resource management*.

ISO 24617 consists of the following parts, under the general title *Language resource management — Semantic annotation framework (SemAF)*:

— *Part 1: Time and events (SemAF-Time, ISO-TimeML)*

— *Part 2: Dialogue acts*

The following parts are under preparation:

— *Part 4: Semantic roles (SemAF-SRL)*

— *Part 5: Discourse structure (SemAF-DS)*

The following parts are planned:

— *Part 3: Named entities (SemAF-NE)*

— *Part 6: Principles of semantic annotation*

— *Part 7: Spatial information (ISO-Space)*

— *Part 8: Relations in Discourse (SemAF-DRel)*

Introduction

This part of ISO 24617 results from the agreement between the TimeML Working Group and the ISO Working Group, ISO/TC 37/SC 4/WG 2, *Language resource management – Semantic annotation*, that a joint activity should take place to accommodate the two existing documents for annotating temporal information, *TimeML 1.2.1* and *TimeML Annotation Guidelines*, into ISO international standards. This work should lead to the achievement of two objectives:

- modification of the two documents in conformance to the ISO International Standards;
- verification of the annotation guidelines for a wide coverage of multilingual resources.

It should be noted that this part of ISO 24617 provides normative guidelines not just for temporal information, but also for information content in various types of events in English as well as other languages.

Language resource management — Semantic annotation framework (SemAF) —

Part 1: Time and events (SemAF-Time, ISO-TimeML)

1 Scope

Temporal information in natural language texts is an increasingly important component to the understanding of those texts. This part of ISO 24617, *SemAF-Time*, specifies a formalized XML-based markup language called *ISO-TimeML,* with a systematic way to extract and represent temporal information, as well as to facilitate the exchange of temporal information, both between operational language processing systems and between different temporal representation schemes. The use of guidelines for temporal annotation has been fully attested with examples from the TimeBank corpus, a collection of 183 documents that have been annotated by TimeML before the current version of *ISO-TimeML* was formulated.

NOTE Throughout this document, *SemAF-Time* refers to the ISO 24617-1, while *ISO-TimeML* refers to the annotation language specified in this document.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE The first reference shows how dates and times are represented and the second provides a format for the standoff representation of *ISO-TimeML* annotation presented here.

ISO 8601:2004, *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO 24612:2011, *Language resource management — Linguistic annotation framework* (*LAF*)

3 Terms and definitions

For the purposes of this document, the terms and definitions in ISO 8601:2004 and the following apply.

NOTE The terms and definitions provided below are provided to clarify the terminology relating to the metamodel, specification, and semantics of *ISO-TimeML*. Terminology derived from XML and other formal languages as well as from general temporal logics is not defined here.

3.1
ALINK
linking tag that represents a phase relation between an aspectual verb (or morpheme) and a predicate denoting an **event** (3.5)

3.2
annotation
process of adding information to segments of language data or that information itself

3.3
beginning
instant (3.6) at which a **temporal interval** (3.17) begins

NOTE Adapted from Hobbs and Pan (2004).

3.4
end
instant (3.6) at which a **temporal interval** (3.17) ends

NOTE Adapted from Hobbs and Pan (2004).

3.5
event
eventuality
something that can be said to obtain or hold true, to happen or to occur

NOTE The term “event” is used here with a very broad notion of event, which includes all kinds of actions, states, processes, etc. It is not to be confused with the more narrow notion of event as something that happens at a certain point in time (such as the clock striking 2, or waking up) or during a short period of time (such as laughing).

3.6
instant
point in time with no interior points

NOTE Time is often viewed as a straight line from minus infinity to plus infinity. In this view, time is formed by an infinite sequence of points. An instant can also be seen as an infinitesimally small interval. Cf. OWL-Time Ontology for “instant”: http://www.w3.org/TR/owl-time/.

3.7
markable
entity in general, or segment of a text in particular, that is subject to an annotation (3.2)

3.8
MLINK
linking tag that represents the measurement of the duration of an **event** (3.5) or the measurement of the length of a (possibly discontinuous) time span

3.9
point of event
instant (3.6) at which the **event** (3.5) mentioned in a given utterance occurs

NOTE Next to a point of speech, a point of event also needs to be defined in order to interpret tense. For example, in “Arthur smiled”, the temporal location of the point of event can be defined as being prior to the point of speech.

3.10
point of reference
instant (3.6) of temporal perspective on the **event** (3.5) in a given utterance

NOTE 1 “Arthur will have gone by tomorrow”, where the point of speech is now, the point of event is some time in the future, but before the point of reference referred to by “tomorrow”.

NOTE 2 To locate certain tenses in time, a third anchor point is also required, defined as the point of reference.

3.11
point of speech
time unit (3.17) at which a given utterance occurs

NOTE 1 The notion of point of speech is needed in order to interpret tense. This requires the use of anchor points in time, of which the point of speech is one (point of text, see 3.12, is another one). For example, in “Arthur smiled”, the point of speech is the time that the utterance is made.

NOTE 2 For a document as a whole, this may be considered to be the same as the document creation time.

3.12
point of text
instant (3.6) at which reported speech is anchored

NOTE It is the point of time considered in the text of the speech. So for example, when a person is telling a story, it is not enough to know the point of the speech itself (the document creation time), but the point at which the speech in the story is taking place.

3.13
representation
format in which an **annotation** (3.2) is rendered, for instance in XML, independent of its content

3.14
SLINK
linking tag that represents a subordinating relation between two **events** (3.5)

3.15
temporal interval
period
uninterrupted stretch of time, with internal point structure.

NOTE 1 Adapted from WordNet.

NOTE 2 Time is often viewed as a straight line from minus infinity to plus infinity. A temporal interval is a part of that line without any holes, containing all the points between its beginning and its end.

NOTE 3 In mathematics, an important issue is whether an interval includes its beginning and its end (is “closed”) or not (is “open” or “half-open”). In natural language descriptions of intervals this may also be relevant, as when describing an interval in terms of a number of days, but not with the same granularity as in mathematics. Cf. OWL-Time Ontology for “interval”: http://www.w3.org/TR/owl-time/.

3.16
temporal ordering relation
relation that determines how objects are ordered in time

EXAMPLE precedence, simultaneity.

NOTE There is a limited number of ways to order objects which are collectively called ordering relations.

3.17
temporal unit
element in a **time amount** (3.18) that quantifies the length of a **temporal interval** (3.15) or a set of **temporal intervals** (3.15)

NOTE 1 Adapted from Bunt (1985).

NOTE 2 In measurement systems, various units are defined for different purposes. Small units such as seconds and minutes are defined to measure small temporal intervals; as one may want to avoid working with big numbers, for larger temporal intervals, units such as week, year, decade, and century are defined.

NOTE 3 The amount of a temporal unit is called a measure.

3.18
time amount
quantity of time, measured by **temporal units** (3.17) over **temporal intervals** (3.15)

NOTE 1 Adapted from Bunt (1985).

NOTE 2 A time amount is a measure of time that can be expressed in terms of a number of temporal units, such as “half an hour” or “30 minutes”.

3.19
tense
way that languages express the time at which an **event** (3.5) described by a sentence occurs

NOTE This is characterized as a property of a verb form. Noun forms will not be said to exhibit tense but rather temporal markers.

3.20
TLINK
linking tag that represents a temporal relation between two temporal entities: namely, between two **events** (3.5), two temporal expressions, or between a temporal expression and an event

NOTE 1 Adapted from Pustejovsky *et al*. (2004).

NOTE 2 Some ordering relations cannot be expressed by an ordering relation between two events because a signal, like a temporal preposition, complicates the ordering or there is an ordering relation between a temporal signal and an event.

4 Overview

An understanding of temporal information is needed to better understand natural language texts in general. Previous work in time stamping is a step in the right direction, but to fully appreciate the complexity of a text with respect to time, the ability to order events and temporal expressions is needed. This part of ISO 24617 defines *ISO-TimeML*, a markup language for time and events, which has been specifically designed for this task.

ISO-TimeML annotates all expressions having temporal import, broadly categorized as temporal expressions and eventualities (situations, events, states, and activities). Temporal expressions and events participate in temporal relationships (e.g. “before”, “simultaneous”), subordinating relationships (e.g. “intensional”, “factive”), and aspectual relationships (e.g. “initiates”, “continues”). *ISO-TimeML* provides an additional expressive capability of capturing and representing the complexities of these relationships.

TimeML, the precursor of *ISO-TimeML*, is already in use in a number of applications focusing on analysis (manual and automatic) of news articles. The TimeBank corpus contains approximately 185 such documents and has been validated against the most recent version of TimeML. The resulting output of a TimeML annotated document is in XML, which allows for general XML validation methods to be used. In addition to supporting interoperability among different temporal representation schemes, TimeML has been shown adequate to support a mapping from the temporal information in a text to its formal representation in a Web Ontology Language such as OWL-Time.

Unlike prior event annotation schemes, *ISO-TimeML*'s somewhat unique definition of an event does not limit the standard's applicability to specific natural language genres. An *ISO-TimeML* event is simply something that can be related to another event or temporal expression using an *ISO-TimeML* relationship — thus an *ISO-TimeML*-compliant representation can be adapted (derived) from the full standard specification, appropriate to different genres, styles, domains, and applications. Future work will involve applying the standard in such different contexts, and formulating guidelines and principles for appropriate use of *ISO-TimeML* in a variety of language engineering environments.

5 Motivation and requirements

The identification of temporal and event expressions in natural language text is a critical component of any robust information retrieval or language understanding system, and recently this has become an area of intense research in computational linguistics and Artificial Intelligence. The importance of temporal awareness to question answering systems has become more obvious as current systems strive to move beyond keyword and simple named-entity extraction. Named-entity recognition has moved the fields of information retrieval and information exploitation closer to access by content, by allowing some identification of names, locations and products in texts. One of the major problems that has not been solved is the recognition of events and their temporal anchorings in text. Events are naturally anchored in time within a narrative. Without a robust ability to identify and extract events and their temporal anchoring from a text, the real aboutness of the text can be missed. Moreover, since entities and their properties change over time, a database of assertions about entities will be incomplete or incorrect if it does not capture how these properties are temporally updated. To this end, event recognition drives basic inferences from text.

As it happens, however, much of the temporal information in an article or narrative is left implicit in the text. The exact temporal designation of events is rarely explicit and many temporal expressions are vague at best. A crucial first step in the automatic extraction of information from such texts, for use in applications such as automatic question answering or summarization, is the capacity to identify what events are being described and to make explicit when these events occurred.

Another important point is that, although most of the information on the web is in natural language, it is unlikely that it will ever be marked up for semantic retrieval, if that entails hand annotation. Natural language programs will have to process the contents of web pages to produce annotations. Remarkable progress has been made in the last decade in the use of statistical techniques for analysing text. However, these techniques, for the most part, depend on having large amounts of annotated data, and annotations require an annotation scheme. Hence, in addition to developing the necessary tools for temporal analysis, it is important to enable for seamless integration into existing and emerging ontologies, such as OWL. Interest in temporal analysis and event-based reasoning has contributed to the development of a specification language for events and temporal expressions and their orderings (TimeML). Some issues relating to temporal and event identification have remained unresolved, however, and *ISO-TimeML* has been designed to address these issues. Specifically, four basic problems in event-temporal identification have been addressed in the design of *ISO-TimeML*:

- time anchoring of events (identifying an event and anchoring it in time);
- ordering events with respect to one another (distinguishing lexical from discourse properties of temporal ordering);
- reasoning with contextually underspecified temporal expressions (temporal functions such as “last week” and “two weeks before”);
- reasoning about the persistence of events (how long does an event or the outcome of an event last).

The specification language, *ISO-TimeML*, is designed to address these issues, in addition to handling basic tense and aspect features.

Linking a formal theory of time with an annotation scheme aimed at extracting rich temporal information from natural language text is significant for at least two reasons. It will allow us to use the multitude of temporal facts expressed in text as the ground propositions in a system for reasoning about temporal relations. It will also constitute a forcing function for developing the coverage of a temporal reasoning system, as we encounter phenomena not normally covered by such systems, such as complex descriptions of temporal aggregates.

6 Basic concepts and metamodel

Regarding the temporal information in a document, a distinction can be made between (1) the temporal metadata, regarding when the document was created, published, distributed, received, revised, etc., and (2) the temporal properties of the events and situations that are described in the document. The former type of information is associated with the document as a whole; information of the latter type will be associated in annotations with parts of the text in the document, “markables” such as words and phrases.

Temporal objects and relations have been studied from logical and ontological points of view; well-known studies include those by Allen (1984), Prior (1967), and more recently Hobbs and Pan (2004); see also the collection of papers in Mani *et al.* (2005). The most common view of time, which underlies most natural languages, is that time is an unbounded linear space running from a metaphorical “beginning of time” at minus infinity to an equally metaphorical “end of time” at plus infinity. This linear space can be represented as a straight line, the points of which correspond to moments in time; following Hobbs and Pan (2004), we will also use the term “instant” to refer to time points. From a mathematical point of view, the points on the time line are line segments of infinitesimally small size, corresponding to the intuition that a moment in time can, in principle, be determined with any precision that one may wish.

For linguistic and philosophical reasons, several classifications have been proposed of verbs describing various types of states or events, the Vendler classification being the best known (Vendler, 1967). For the annotation of temporal information in text, not only verbs with their tenses and temporal modifications should be considered, but also nouns, since nouns may also denote events and situations ("The meeting tomorrow"; "The six o'clock news"). In TimeML, Pustejovsky *et al.* (2007) have proposed a classification of states and events into seven categories. In the literature, a distinction is often made between events and states, where events are commonly characterized as occurring at a point in time or during a certain definite interval, whereas states may obtain for any indefinite stretch of time ("The Mediterranean Sea separates Europe from Africa"). On a terminological note: the term "event" will henceforth be used as a generic term that also covers such notions as "state", "situation", "action", "process", etc.; this broad notion of event has also been termed "eventuality" (Bach, 1986).

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 (such as "I will call you at twelve o'clock"). Since instants are formally a special kind of interval, a consistent approach to modelling the time that an event occurs is to always use intervals, where it may happen that the interval associated with a particular event is regarded as having zero length, and thus being an instant. This is reflected in the metamodel presented in Figure 1, which uniformly relates events with temporal intervals.

The length of an interval can also occur as temporal information in a text, as in "I used twelve hours to read that book" and "It takes seven minutes to walk to the station". An expression such as "seven minutes" does not denote an interval, but the length of an interval. It is the temporal equivalent of spatial distance ("seven miles"). To describe the length of a temporal interval, one needs a unit of measurement, which may be combined with a numerical expression to obtain an amount of time. The metamodel presented below therefore includes the concept of an amount of time, related to intervals through the function *length*, and the auxiliary concepts of temporal units and real numbers. (Moreover, in the *ISO-TimeML* semantics, different temporal units are related through a conversion function, stipulating such things as 1 hour = 60 minutes; 1 day = 24 hours, etc. An amount of time can be characterized equivalently by as many pairs <*numeral, temporal unit*> as there are temporal units, the equivalence being defined through the numerical conversions between units [see Bunt (1985)].

Regarding the temporal anchoring of events in time, it may be noted that the association of a temporal interval with an event does not necessarily mean that the event took place during every moment within that interval. When someone says "I've been working on my presentation from 8.30 to 12 o'clock", that presumably does not mean that the speaker has been working on his presentation for every single moment between 8.30 and 12 o'clock; there must have been interruptions for having some coffee, going to the bathroom, etc. In such a case it is more accurate to anchor the event at the time span starting at 8.30 and ending at 12 o'clock, a "time span" being understood as a period of time that may have "holes", where the event was interrupted. The metamodel shown in Figure 1 does not distinguish time spans, but reflects the assumption that whether an event occurs during an interval, with or without any interruptions, can only be decided on a case by case basis, and is best modelled as a property of the temporal anchoring relation applied to a specific event.

ISO 24612:2011 insists on the use of stand-off annotation, i.e. the construction of annotations in separate files, separate from the document containing the primary language data, as contrasted with in-line annotation. Stand-off annotations refer to specific locations in the primary data by addressing byte offsets, linguistic elements such as words, or times associated with recorded data, to which the annotation applies. Compared to in-line annotation, stand-off annotation has the advantages of respecting the integrity of the primary data and of allowing multiple annotations to be layered over a given primary document. Since semantic annotations typically occur at a relatively high level in a layered annotation structure, they do not necessarily refer directly to segments in the primary data, but may also refer to structures in other annotation layers. The generic term "markable" is used to refer to the entities that the annotations are associated with.

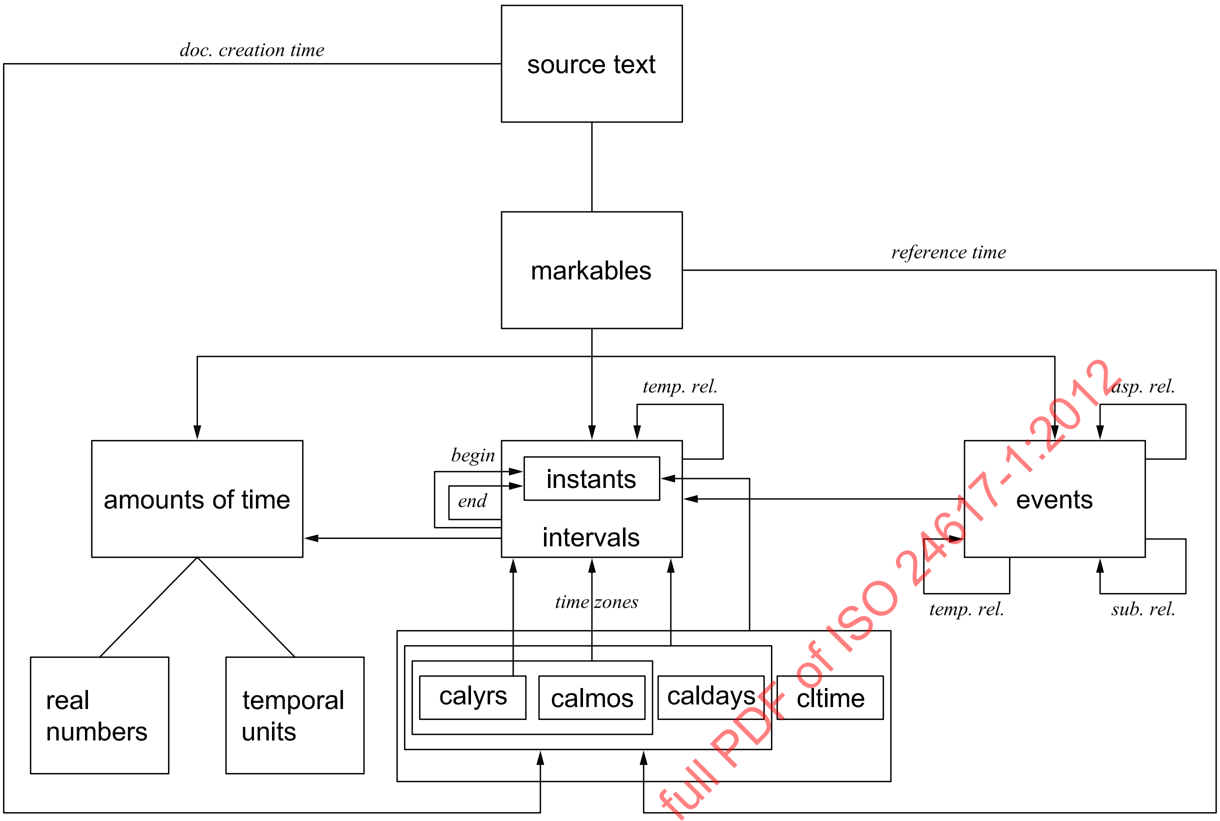

Figure 1 — Metamodel

Markables are derived from documents, which will have certain metadata that are particularly important for the interpretation of temporal annotations. For interpreting the tenses of verb forms and adverbial temporal deixis in a text (“yesterday”; “next week”), for instance, one must know when the text was produced. This will often be defined by the document creation time, and more precisely by the combination of a creation time and a creation location, since the latter defines the time zone within which the creation time is precisely defined. In many documents, the time and place of the document creation will be those of all the markables that may be derived from the document, but it may also happen that the text in a document introduces other times and places relative to which the annotations of the markables should be understood. A time zone, such as Greenwich Mean Time (GMT), can be seen as a way of segmenting the time line into named segments of particular lengths, such as (calendar) years, months, days, hours, and minutes. Accordingly, time zones show up in the metamodel as functions mapping a calendar year (“2008”), a combination of a calendar year and a calendar month (“May 2008”), a date (“May 25, 2008”), or a date plus a clock time (“May 25, 2008, 12.30 p.m.”) onto a temporal interval (in the latter case, an instant).

A markable may refer to more than one, related event, as in “She started to laugh” (two aspectually related events); “John drove to Boston after the concert” (two temporally related events); or “Will you attend the meeting on Tuesday?” (one event having a subordination relation to another). For expressing such relations, the metamodel includes the corresponding classes of relations showing up as inter-event links. Temporal relations between events may also be stated between intervals, hence they show up again in the metamodel.

7 Specification of ISO-TimeML

7.1 Overview

ISO-TimeML implements the fundamental distinction between the concepts of ***annotation*** and ***representation*** which is made in ISO 24612:2011 (*LAF*) [see also Ide and Romary (2003)]. The term ***annotation*** is used to refer to the process of adding information to segments of language data, or to refer to that information itself. This notion is independent of the format in which this information is represented. The term ***representation*** is used to refer to the format in which an annotation is rendered, for instance in XML, independent of its content. According to *LAF*, annotations are the proper level of standardization, not representations. This part of ISO 24617 (*SemAF-Time*) therefore defines a specification language for annotating documents with information about time and events at the level of annotations and then for representing these annotations in a specific way, namely XML. This language is called *ISO-TimeML*.

The distinction between annotations and representations is reflected in the specification of *ISO-TimeML* given below, where, following Bunt (2010), an ***abstract syntax*** is defined in addition to a ***concrete syntax***. The abstract syntax specifies the kinds of elements which make up the information in annotations, and describes how these elements may be combined to form complex annotation structures; these combinations are defined as set-theoretical structures, independent of any particular representation format. There are infinitely many ways in which these structures can be represented. In line with other ISO/TC 37/SC 4 standards, and in particular with *LAF*, *ISO-TimeML* has an XML-based concrete syntax for temporal and event-related information. *ISO-TimeML* has a semantics associated with its abstract syntax, which defines the meanings of *ISO-TimeML* annotation structures. All concrete representations of *ISO-TimeML* annotations inherit the semantics of the abstract syntax.

The specification of *ISO-TimeML* consists of three parts, mirroring the *LAF* distinction between the two levels of annotations and representations:

- Part 1 describes the abstract syntax of *ISO-TimeML* annotations (see 7.2);
- Part 2 describes a concrete syntax, which specifies a format for representing these annotations in XML [*ISO-TimeML* (see 7.3, Annex A and Annex H)];
- Part 3 describes the semantics of *ISO-TimeML* (see Clause 8) and is again split into two parts:
  - one part describes the semantics of the XML representation of the annotation of temporal expressions using Interval Temporal Logic, a first-order logic for reasoning about time;
  - the other part describes a more general, event-based semantics for the abstract *ISO-TimeML* syntax.

The annotations shall be implemented as specified in Annexes A and H.

7.2 Abstract syntax

7.2.1 Introduction

The abstract syntax of *ISO-TimeML* defines the set-theoretical structures that constitute the information about time and events that may be contained in annotations. The definition of the abstract syntax consists of

1) a specification of the elements from which these structures are built up, called a “conceptual inventory”, and

2) a set of syntax rules which describe the possible combinations of these elements. What these combinations mean, i.e. which information they capture, is specified by the semantics associated with the abstract syntax (see 8.4).

7.2.2 Conceptual inventory

The concepts that can be used to build *ISO-TimeML* annotations fall into five categories, all consisting of finite sets, along with the concepts of real number and natural number:

- a finite set of elements called “event classes”, “tenses”, “aspects”, “polarities”, and “set-theoretic types”;
- a finite set of elements called “temporal relations”, “duration relations”, “numerical relations”, “event subordination relations”, and “aspectual relations”;
- a finite set of elements called “time zones”;
- a finite set of elements called “calendar years”, “calendar months”, “calendar day numbers”, “clock times” (natural numbers ranging from 0000 to 0059; from 0100 to 0159; ... from 2300 to 2400);
- a finite set of elements called “temporal units”.

Natural numbers are needed for capturing the information expressed in English, for example, by “twice” and “three times”; real numbers are needed for cases such as “two and a half hours”.

7.2.3 Syntax rules

7.2.3.1 General principles

Annotation structures in *ISO-TimeML* come in two varieties, which we will refer to as ***entity structures*** and ***link structures***. Entity structures contain semantic information about a segment of source text; link structures describe semantic relations between segments of source text by means of links between entity structures.

The simplest kind of an *ISO-TimeML* structure is a single entity structure, which is a pair <m,a> consisting of a markable m and an annotation a, or a single link structure which relates two entity structures. More complex annotation structures consist of a set of entity structures and a set of link structures which link the entity structures together through temporal and inter-event relations.

More formally, an *ISO-TimeML* annotation structure consists of two sets M and L, where M is a set of pairs <*markable, entity structure*> and L is a set of triples <*markable, entity structure, entity structure*>, such that each element of L contains at least one entity structure that occurs in M (This latter condition ensures that the links in an annotation structure relate to the entity structures that it contains; they may, in addition, also contain links to elements outside the current annotation structure).

7.2.3.2 Entity structures

There are five types of annotations that may form an entity structure, containing information about (1) events; (2-4) temporal objects (intervals, instants, and amounts of time); (5) frequencies of events; and (6) explicit temporal relations (as, for example, expressed in English by temporal prepositions).

- An **event structure** is a 7-tuple <C, T, A, Σ, N, P_N, V> where C is a member of the set of event classes; T and A are a tense and an aspect, respectively; Σ is a set-theoretical type (such as an individual object or set of individual objects); N is a natural number (e.g. the number 2 for dealing with such examples as “John kissed Mary twice”); P_N is an amount of time (such as “two and a half hours”, for such examples as “John called Mary twice every two and a half hours”), and V is a veracity (claimed truth or falsity, corresponding to positive or negative polarity in natural language).
- An **instant structure** is either a triple <*time zone, date, clock time*>, where a date is a triple consisting of a calendar year, a calendar month, and a calendar day number; or a triple <*time amount structure, instant structure, temporal relation*> (“half an hour before midnight”).

- The following set-theoretical structures are **interval structures**:
  - a pair <t_1,t_2> of two instant structures, corresponding to the beginning and end points of the interval;
  - a triple <*time amount structure, instant structure, temporal relation*> ("three weeks before Christmas"; "two years from today");
  - a triple <t_1,t_2,R> where t_1 and t_2 are either instant structures or interval structures, and where R is a duration relation (examples: "from 1992 until 1995"; "from 1882 through 1995");
- A **time-amount structure** is a pair <n,u>, where n is a real number and u a temporal unit, or a triple <R,n,u>, where R is a numerical relation (such as greater than) and n and u are as before;
- A **temporal relation structure** is just a temporal relation.

7.2.3.3 Link structures

There are five types of link structures in *ISO-TimeML*: for temporal ***anchoring*** of events in time; for temporal ***ordering of events*** and/or ***intervals or instants*** relative to each other; for measuring the length of an interval; for ***subordination relations*** between events, and for ***aspectual relations*** between events.

- A **temporal anchoring structure** is a triple <event structure, interval structure, temporal anchoring relation>, or a triple <event structure, instant structure, temporal anchoring relation>;
- A **temporal relation structure** is a triple <event structure, event structure, temporal relation>, or a triple <interval or instant structure, interval or instant structure, temporal relation>;
- A **time measurement structure** is a pair <event structure, time-amount structure> or a pair <interval structure, time-amount structure>;
- A **subordination structure** is a triple <event structure, event structure, subordination relation>;
- An **aspectual structure** is a triple <event structure, event structure, aspectual relation>.

7.3 Concrete XML-based syntax

7.3.1 TimeML vs. ISO-TimeML: Stand-off annotation and other differences

A concrete syntax consists of the specification of names for the various sets forming the conceptual vocabulary, plus a listing of specific named elements of these sets, and a specification of how to represent the annotation structures defined by the abstract syntax.

The *ISO-TimeML* concrete syntax has been inspired by the TimeML format [Pustejovsky *et al*. (2003; 2007)]. and is reproduced below with a few amendments to do justice to the stand-off character of *ISO-TimeML* annotations and to the conceptual differences between *ISO-TimeML* and the original TimeML.

One of the differences between *ISO-TimeML* and TimeML is that *ISO-TimeML* adopts a stand-off annotation format, where TimeML has been using an in-line annotation format. The differences between the two formats are illustrated in the examples (1) and (2) below:

(1) John left on 31 December 2007.

```
John
 <EVENT  eid="e1"  eiid="ei1"  type="transition"  tense="past">left</EVENT>
 <SIGNAL sid="s1">on</SIGNAL>
 <TIMEX3 tid="t1" type="date" value="2007-12-31">31 December 2007</TIMEX3>
```

(2) John left on 31 December 2007.

```
<s>
 <w xml:id="w1">John</w>
 <w xml:id="w2">left</w>
 <w xml:id="w3">on</w>
 <w xml:id="w4">31</w>
 <w xml:id="w5">December</w>
 <w xml:id="w6">2007</w>
</s>
 <EVENT eid="e1" eiid="ei1"target="#w2"  type="transition" tense="past"/>
 <SIGNAL xml:id="s1" target="#w3"/>
 <TIMEX3 tid="t1" target="#range(w4,w6)"  type="date" value="2007-12-31"/>
```

The first example shows the TimeML representation (somewhat simplified) in in-line format; the second shows the same annotation in stand-off format, together with the segmentation of the text.

For linking annotations to primary text in stand-off format, *ISO-TimeML* makes use of the @target attribute defined in TEI (2010) *P5 Guidelines*. This attribute can be used to point to units in the primary data preprocessed by another annotation scheme, such as tokenization, morpho-syntactic annotation, or syntactic annotation. It can also be used to point to word tokens identified in an in-line form of segmentation; see examples (3) and (4). Moreover, the TEI-defined functions “range” and “group” can be used to construct references to contiguous or non-contiguous text spans, respectively.

ISO-TimeML has adopted the following additional modifications of TimeML.

- The attribute names @eid, @tid and @sid have been changed simply to @xml:id, while marking up their values as they are.

- The element <MAKEINSTANCE> has been deleted along with its associated attribute @eiid or any other attributes and move some of its attributes such as @tense and @aspect to the element <EVENT>.

- The attribute @eiid has been deleted; each <EVENT> is interpreted in *ISO-TimeML* as denoting an event instance. As a result, the attributes such as @eventInstanceID and @relatedToEventInstance that occurred in the link elements such as <TLINK> and <ALINK> are replaced by such attributes as @eventID and @relatedToEvent.

- The attribute @pred has been introduced, having a value which is understood as denoting the content related to that event.

- Durations, such as “five seconds”, have been introduced as a separate type of temporal entity, different from instants and temporal intervals.

Here are two examples. In the first one, the segmentation of the text into words is annotated in an in-line format.

(3) John left on 31 December 2007.

```
<s>
 <w xml:id="w1">John</w>
 <w xml:id="w2">left</w>
 <w xml:id="w3">on</w>
 <w xml:id="w4">31</w>
 <w xml:id="w5">December</w>
 <w xml:id="w6">2007</w>
</s>
<isoTimeML>
 <EVENT xml:id="e1" target="#w2" pred="LEAVE" class="OCCURRENCE" type="TRANSITION" tense="PAST"/>
 <SIGNAL xml:id="s1" target="#w3"/>
 <TIMEX3 xml:id="t1" target="#range(w4,w6)" type="DATE" value="2007-12-31"/>
 <TLINK eventID="#e1" relatedToTime="#t1" signalID="#s1" relType="IS_INCLUDED"/>
</isoTimeML>
```

In the second example, the given text is assumed to be preprocessed by a tokenizer.

(4) John taught on Tuesday but not on Wednesday.

```
<isoTimeML>
 <EVENT xml:id="e1" target="#token1" pred="TEACH" class="OCCURRENCE" type="PROCESS" tense="PAST"
   polarity="POS"/>
 <SIGNAL xml:id="s1" target="#token2"/>
 <TIMEX3 xml:id="t1" target="#token3" pred="TUESDAY" type="DATE" value="2007-12-31"/>
 <EVENT xml:id="e2" pred="TEACH" class="OCCURRENCE" type="PROCESS" tense="PAST" polarity="NEG"/>
 <SIGNAL xml:id="s2" target="#token6"/>
 <TIMEX3 xml:id="t2" target="#token7" pred="WEDNESDAY" type="DATE" value="2007-01-01"/>
 <TLINK eventID="#e1" relatedToTime="#t1" signalID="#s1" relType="IS_INCLUDED"/>
 <TLINK eventID="#e2" relatedToTime="#t2" signalID="#s2" relType="IS_INCLUDED"/>
</isoTimeML>
```

7.3.2 Naming conventions

Inasmuch as XML is case-sensitive, it is necessary for *ISO-TimeML* to specify exactly the case of all its elements. This document follows the convention of indicating tag names and attribute values all in upper case (e.g. EVENT, PROGRESSIVE) and attribute names in lower or mixed case (e.g. @tense, @relatedToTime). Since attribute values are typically atomic (one-word) while attribute names often consist of multiple words, this convention would seem to maximize readability of the annotation.

Attributes that range over values of XML datatype URI references to IDs typically consist of the name of the element indexed, followed by “ID” (e.g. eventID) or a descriptive name (e.g. relatedToTime).

The values of the various ID attributes are specified as beginning with one or two characters, followed by an integer. This scheme is mandated by the syntax of XML. While attribute values of type ID can consist of any sequence of letters, digits, and the hyphen, underscore, and period characters, they shall begin with either an underscore or a letter. Therefore, e23 is a valid XML ID, but 23 is not. This naming convention also helps make the examples a bit more readable, especially in the case of link tags, which can contain multiple URIs of different kinds.

7.3.3 Example annotations

Though this document describes the full *ISO-TimeML* language, many of the example annotations provided show the result of annotation only through the output of initial automatic tagging combined with human annotation/editing, but do not include elements (e.g. attributes and/or attribute values) that may be introduced by later processing components (e.g. the closure tool). In particular, TIMEX3 tags that are treated as temporal functions typically appear in the examples in an underspecified form. However, those elements that do appear are sufficient for the output of manual annotation.

Finally, note that all examples in this document have been validated against an *ISO-TimeML* schema compliant with the specification presented in Annex H.

7.3.4 Basic elements: <EVENT>, <TIMEX3>, and <SIGNAL>

7.3.4.1 <EVENT>

The <EVENT> element is used to annotate those elements in a text that describe what is conventionally referred to as an eventuality. Syntactically, events are typically expressed as inflected verbs, although event nominals and nominalizations, such as “crash” in “... killed by the crash” and “examination” in “a recent examination of evidence”, should also be annotated as <EVENT>.

The <EVENT> tag is also used to annotate a subset of the states in a document, typically expressed as adjectives. This subset of states includes those that are either transient or explicitly marked as participating in a temporal relation. See Annex A, *Core annotation guidelines*, for more details.

Table 1 provides the syntax and definition for the <EVENT> element.

Table 1 — Specification of <EVENT>

target	specifies the destination of the reference by supplying one or more URI References.
pred	denotes the content related to that event through the indication of a lexical predicate.
class	characterizes each event as occurrence, perception, reporting, aspectual, state, I-state or I-action.
type	three event types are distinguished: state, process and transition.
tense	captures standard distinctions in the grammatical category of verbal tense.
aspect	captures standard distinctions in the grammatical category of verbal aspects such as perfective, imperfective, progressive and so on.
polarity	Boolean attribute that conveys the polarity associated with the event in question.
modality	conveys the modality associated with the event: different degrees of epistemic modality, deontic modality, etc.
comment	remarks made by an annotator.

— The value STATE occurs twice as a possible value of the attribute @class and also as a possible value of the attribute @type. As a result, we may have an example such as: <EVENT xml:id="e1" pred="LOVE" class="STATE" type="STATE">. For the present, there will remain a redundancy in the attribute specification, until the attribute @class is replaced with some ontological marker.

— The range of possible values of the @modality attribute is great, much depending on various features of particular languages. For instance, Italian requires CONDITIONAL as a modal value of @modality besides SUBJUNCTIVE, and Korean requires RESTROSPECTIVE and CONJECTURAL as possible values of the optional attribute @mood, which is not a strictly semantic attribute but which has implications for the semantic modality associated with an event. This will be shown in the annexes of this document.

— The attribute @comment is a part of all the *ISO-TimeML* tags, and is there for annotators to add clarifications and other observations about the text being marked.

Each <EVENT> tag represents a unique instance of an event, identified as the event instance identification number. If additional instances of an event are needed, a non-consuming <EVENT> tag can be created with a new event ID.

The attributes @tense and @aspect of the element <EVENT> are represented by specific attribute values within this tag. In addition, if the event is modified by a negation, this is indicated by the appropriate value in the @polarity attribute. The term “mood” in traditional grammar refers to SUBJUNCTIVE or INDICATIVE: “If I were (PRESENT SUBJUNTIVE) a bird, I would fly” vs. “If I am (PRESENT INDICATIVE) a bird, I can fly”, “If I had been (PAST SUBJUNCTIVE) in the airport, I would have died (PAST COUNTERFACTUAL CONDITIONAL sentence).” The optional attribute @mood is used when syntactic mood is expressed by inflectional morphology on the verb; @modality, on the other hand, is reserved for the semantic annotation of an explicit modal auxiliary verb, such as “should” or “must”. The attributes @tense and @aspect are expected to have their values filled in by a pre-processing program, according to the following paradigm. See Tables 2 and 3.

Table 2 — Active voice

verb group	tense	aspect
teaches	PRESENT	NONE
is teaching	PRESENT	PROGRESSIVE
has taught	PRESENT	PERFECTIVE
has been teaching	PRESENT	PERFECTIVE_PROGRESSIVE
taught	PAST	NONE
was teaching	PAST	PROGRESSIVE
had taught	PAST	PERFECTIVE
had been teaching	PAST	PERFECTIVE_PROGRESSIVE
will teach	FUTURE	NONE
will be teaching	FUTURE	PROGRESSIVE
will have taught	FUTURE	PERFECTIVE
will have been teaching	FUTURE	PERFECTIVE_PROGRESSIVE

Table 3 — Passive voice

verb group	tense	aspect
is taught	PRESENT	NONE
is being taught	PRESENT	PROGRESSIVE
has been taught	PRESENT	PERFECTIVE
has been being taught	PRESENT	PERFECTIVE_PROG
was taught	PAST	NONE
was being taught	PAST	PROGRESSIVE
had been taught	PAST	PERFECTIVE
had been being taught	PAST	PERFECTIVE_PROG
will be taught	FUTURE	NONE
will be being taught	FUTURE	PROGRESSIVE
will have been taught	FUTURE	PERFECTIVE
will have been being taught	FUTURE	PERFECTIVE_PROG
being taught	NONE	PRESPART

- The attributes @tense, @aspect, and @modality require additional values to deal with languages other than English, as illustrated in Annex C (Chinese), D (Italian), and E (Korean). This is also the case for the optional attribute @mood.

The values of the attributes @polarity and @modality are determined by modifiers found near the event in the text. For languages that encode mood in the morphology of the verb, the attribute @mood is used. Here are some examples:

(5) should have bought

<**EVENT xml:id**="e1" **target**="#token2" **pred**="BUY" **class**="OCCURRENCE" **type**="TRANSITION" **tense**="PAST" **aspect**="PERFECTIVE" **modality**="SHOULD" **polarity**="POS"/>

(6) did not teach

<**EVENT xml:id**="e2" **target**="#token2" **pred**="TEACH" **class**="OCCURRENCE" **type**="PROCESS" **tense**="PAST" **aspect**="NONE" **polarity**="NEG"/>

(7) must not teach

<**EVENT xml:id**="e3" **target**="#token2" **pred**="TEACH" **class**="OCCURRENCE" **type**="PROCESS" **tense**="PRESENT" **aspect**="NONE" **modality**="MUST" **polarity**="NEG"/>

7.3.4.2 <TIMEX3>

The <TIMEX3> element is used to mark up explicit temporal expressions, such as times, dates, and durations. It is modelled on Setzer's (2001) <TIMEX> element, as well as the TIDES [Ferro, *et al*. (2003)] <TIMEX2> element. Since it differs both in attribute structure and in use, it was given a separate name, which reveals its heritage while at the same time indicating that it is different from its forebears.

Table 4 provides the syntax and definition for the <TIMEX3> element.

Table 4 — Specification of <TIMEX3>

target	specifies the destination of the reference by supplying one or more URI References.
pred	denotes the content related to that time through the indication of a lexical predicate.
type	characterizes each temporal expression as denotation date, time, duration or set.
functionInDocument	indicates the function of the <TIMEX3> in providing a temporal anchor for other temporal expressions in the document.
beginPoint	indicates the beginning point of a period of type DURATION.
endPoint	indicates the end point of a period of type DURATION.
quant	specifies temporal quantification.
freq	specifies temporal frequency.
temporalFunction	indicates whether the element <TIMEX3> is used as a temporal function.
value	has a specific value of the type of duration, dateTime, time, date, gYearMonth, gYear, gMonthDay, gDay, or gMonth.
valueFromFunction	pointer to a temporal function that determines its value.
anchorTimeID	points to time expression to which the <TIMEX3> markable is temporally anchored. That is, the time expression needed in order to compute its attribute value.
comment	remarks made by an annotator.

— The optional attribute @functionInDocument indicates the function of the <TIMEX3> in providing a temporal anchor for other temporal expressions in the document. If this attribute is not explicitly supplied, the default value is NONE. The non-empty values take their names from the temporal metadata tags in the Prism draft standard, which is available at http://www.prismstandard.org and are intended to have the same interpretations:

There are several times that mark the major milestones in the life of text: The time the story is published, the time it may be released (if not immediately), the time it is received by a customer, and the time that the story expires (if any). Dates and times should be represented using the W3C-defined profile of ISO 8601:2004 [W3C-NOTE-datetime]. The elements are shown in Table 5.

Table 5 — Elements for time and date information

element	role
prism:creationTime	Date and time the identified resource was first created
prism:expirationTime	Date and time when the right to publish material expires
prism:modificationTime	Date and time the resource was last modified
prism:publicationTime	Date and time when the resource is released to the public
prism:releaseTime	Earliest date and time when the resource may be distributed
prism:receptionTime	Date and time when the resource was received on current system

— There can be as many instances of TIMEX3s containing a @functionInDocument attribute with a non-empty value as there are <TIMEX3>s that express different functions. In practice, there will probably be no more than two, one with <CREATION_TIME> and another with <PUBLICATION_TIME>, since these are likely to be the only attributes that will appear in the text of documents to be annotated.

— <RELEASE_TIME> does not indicate when the document was actually released. It is a specification of when the document is allowed to be released. This comes up in documents that are syndicated and where the issuing organization wants to delay publication by syndicators, so as not to be scooped.

— The Prism standard, at least in its temporal indicators, is interested only in the document as an artefact, a piece of intellectual property. This means that the Prism values do not indicate the function of a <TIMEX3> relative to the internal narrative of the document. The specification of the *ISO-TimeML* language can fill this gap by adding values for the @functionInDocument attribute that capture narrative functions. At present, we leave the specification of possible values as is, and will defer the obvious extension until annotation of existing texts indicates that this is a pressing issue.

— The optional attribute @temporalFunction indicates whether the element <TIMEX3> is used as a temporal function, e.g. two weeks ago. If this attribute is not explicitly supplied, the default value is false. It is used in conjunction with the attribute @anchorTimeID, which indicates the element <TIMEX3> to which its denotation is applied. It also appears with the attribute @valueFromFunction, a pointer to a temporal function that determines its value. As was noted above, <TIMEX3> tags that behave as temporal functions are often underspecified in the example annotations below.

— The data types specified for the value attribute - Period, Date, Time, WeekDate, WeekTime, Season, PartOfYear, PaPrFu - are XML data types based on the 2002 TIDES guideline, which extends the ISO 8601:2004 standard for representing dates, times, and durations.

— The attribute @mod is an optional attribute adopted from TIDES. It is used for temporal modifiers that cannot be expressed either within value proper, or via links or temporal functions.

Examples for the use of @type, @value, and @mod are shown below:

(8) no more than 60 days

<**TIMEX3 xml:id**="t1" **target**="#range(token0,token4)" **type**="DURATION" **value**="P60D" **mod**="EQUAL_OR_LESS"/>

(9) the dawn of 2000

<**TIMEX3 xml:id**="t2" **target**="#range(token0,token3)" **type**="DATE" **value**="2000" **mod**="START"/>

— The attribute @anchorTimeID is used to point to another <TIMEX3> in the case of expressions such as "last week", which have a functional interpretation. The value of @anchorTimeID provides the reference point to which the functional interpretation applies.

— The attributes @quant and @freq are used to specify sets that denote quantified times in TIMEX3. The attribute @quant is generally a literal from the text that quantifies over the expression. The attribute @freq contains an integer value and a time granularity to represent any frequency contained in the set, just as a period of time is represented as a period.

Examples for the use of @quant and @freq are shown below:

(10) twice a month

```
<TIMEX3 xml:id="t3" target="#range(token0,token2)" type="SET" value="P1M" freq="2X"/>
```

(11) three days every month

```
<TIMEX3 xml:id="t4" target="#range(token0,token3)" type="SET" value="P1M" quant="EVERY" freq="3D"/>
```

(12) daily

```
<TIMEX3 xml:id="t5" target="#token0" type="SET" value="P1D" quant="EVERY"/>
```

— The attributes @beginPoint and @endPoint are used to anchor periods (of duration) to other time expressions in the document. If there is no explicit @xml:id to assign to one of these values, then an empty <TIMEX3> tag is created to represent the unspecified point. Conversely, if both the beginning and end points of a period are explicitly stated in the document, an empty <TIMEX3> tag is created to represent the unspecified duration.

Examples:

(13) two weeks from June 7, 2003

```
<TIMEX3 xml:id="t6" target="#token0 #token1" type="DURATION" value="P2W" beginPoint="#t61" endPoint="#t62"/>
<SIGNAL xml:id="s1" target="#token2"/>
<TIMEX3 xml:id="t61" target="#range(token3,token5)" type="DATE" value="2003-06-07"/>
<TIMEX3 xml:id="t62" type="DATE" value="2003-06-21" temporalFunction="true" anchorTimeID="#t6"/>
```

(14) 1992 through 1995

```
<TIMEX3 xml:id="t71" target="#token0" type="DATE" value="1992"/>
<SIGNAL xml:id="s1" target="#token1"/>
<TIMEX3 xml:id="t72" target="#token2" type="DATE" value="1995"/>
<TIMEX3 xml:id="t7" target="#range(token0,token2)" type="DURATION" value="P4Y" beginPoint="#t71" endPoint="#t
72"
 temporalFunction="true"/>
```

7.3.4.3 <SIGNAL>

The element <SIGNAL> is used to annotate sections of text, typically function words, that indicate how temporal expressions or eventualities are to be related to each other. The specification is shown in Table 6.

Table 6 — Specification of <SIGNAL>

target	specifies the destination of the reference by supplying one or more URI References
pred	denotes the content related to that signal through the indication of a lexical predicate
comment	remarks made by an annotator

The material marked by <SIGNAL> constitutes several types of linguistic elements: indicators of temporal relations such as temporal prepositions (e.g. “on”, “during”) and other temporal connectives (e.g. “when”) and subordinators (e.g. “if”). This functionality of the <SIGNAL> element was introduced by Setzer (2001).

7.3.5 Link elements: <TLINK>, <SLINK>, <ALINK> and <MLINK>

7.3.5.1 <TLINK>

The element <TLINK> (temporal link) is one of the four *ISO-TimeML* link elements. Link elements encode the various relations that exist between the temporal elements of a document. The motivations for having multiple types of links are the following:

- to distinguish between event types, such as those introduced by conjunction, quantification, or negation;
- to adequately handle subordinating contexts involving modality and reported speech.

The specification is shown in Table 7.

While <TLINK> establishes the ordering of an event or temporal interval relative to another event or interval, <SLINK> introduces a syntactically and semantically subordinating context, such as that introduced by modality or reporting predicates. <ALINK> establishes an aspectual relationship between two events. Finally, <MLINK> establishes a measuring relation between a temporal expression and the event it measures.

Table 7 — Specification of <TLINK>

origin	attribute supplied by closure.
eventID	points to the event involved in a temporal link.
timeID	This is the ID of the event or the time involved in the temporal link.
signalID	points, when applicable, to the explicit signal of the temporal relation.
relatedToEvent	Pointer to the entity that is being related to the event if this entity is of an event type.
relatedToTime	Pointer to the entity that is being related to the time expression if this entity is of temporal type.
relType	Temporal relation holding between the entities.
comment	remarks made by an annotator.

The value of the optional attribute @origin will be supplied by closure. This information and the link ID are primarily used by the closure algorithm. All links in *ISO-TimeML* may have these two attributes, but neither will be included in the examples presented here.

Examples:

(15) John taught on Monday and on Tuesday too.

```
<EVENT xml:id="e1" target="#token1" pred="TEACH" class="OCCURRENCE" type="PROCESS"
  tense="PAST" aspect="NONE" polarity="POS"/>
<SIGNAL xml:id="s1" target="#token2"/>
<TIMEX3 xml:id="t1" target="#token3" pred="MONDAY" type="DATE" value="xxxx-wxx-1"/>
<SIGNAL xml:id="s2" target="#token5"/>
<EVENT xml:id="e2" pred="TEACH" class="OCCURRENCE" type="PROCESS"
  tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t2" target="#token6" pred="TUESDAY" type="DATE" value="xxxx-wxx-2"/>
<TLINK eventID="#e1" signalID="#s1" relatedToTime="#t1" relType="IS_INCLUDED"/>
<TLINK eventID="#e2" signalID="#s2" relatedToTime="#t2" relType="IS_INCLUDED"/>
```

(16) John left 5 minutes after the explosion.

```
<EVENT xml:id="e1" target="#token1" pred="LEAVE" class="OCCURRENCE" type="TRANSITION" tense="PAST"
  aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#token2 #token3" type="DURATION" value="PT5M" beginPoint="#t2" endPoint="#t3"/>
<SIGNAL xml:id="s1" target="#token4" pred="AFTER"/>
<EVENT xml:id="e2" target="#token6" pred="EXPLODE" class="OCCURRENCE" type="TRANSITION"
  tense="NONE" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t2" type="TIME" value="xxxx-xx-xx" temporalFunction="true" anchorTimeID="t1"/>
<TIMEX3 tid="t3" type="TIME" value="xxxx-xx-xx" temporalFunction="true" anchorTimeID="#t1"/>
<TLINK eventID="#e2" signalID="#s1" relatedToTime="#t1" relType="BEGINS"/>
<TLINK eventID="#e2" relatedToTime="#t2" relType="IS_INCLUDED"/>
<TLINK eventID="#e1" signalID="#s1" relatedToTime="#t3" relType="ENDS"/>
<TLINK eventID="#e1" relatedToTime="#t3" relType="IS_INCLUDED"/>
```

Treatment of temporal functions:

(17) John taught from September to December last year.

```
<EVENT xml:id="e1" target="#token1" pred="TEACH" class="OCCURRENCE" type="PROCESS"
  tense="PAST" aspect="NONE" polarity="POS"/>
<SIGNAL xml:id="s1" target="#token2" pred="FROM"/>
<TIMEX3 xml:id="t1" target="#token3" pred="SEPTEMBER" type="DATE" value="xxxx-09"/>
<SIGNAL xml:id="s2" target="#token4" pred="TO"/>
<TIMEX3 xml:id="t2" target="#token5" pred="DECEMBER" type="DATE" value="xxxx-12"/>
<TIMEX3 xml:id="t5" type="DURATION" value="P4M" beginPoint="#t1" endPoint="#t2" temporalFunction="true"/>
<TIMEX3 xml:id="t3" target="#range(token6,token7)" pred="LAST_YEAR" type="DATE" value="1995"             t
temporalFunction="true" anchorTimeID="t4"/>
<TIMEX3 xml:id="t4" type="DATE" value="1996-03-27" functionInDocument="CREATION_TIME"/>
<TLINK timeID="#t1" signalID="#s1" relatedToTime="#t5" relType="BEGINS"/>
<TLINK timeID="#t2" signalID="#s2" relatedToTime="#t5" relType="ENDS"/>
<TLINK eventID="#e1" relatedToTime="#t5" relType="SIMULTANEOUS"/>
```

(18) John taught last week.

```
<EVENT xml:id="e1" target="#token1" pred="TEACH" class="OCCURRENCE" type="PROCESS"
  tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#range(token2, token3)" pred="LAST_WEEK" type="DATE" value="XXXX-WXX"
  temporalFunction="true" anchorTimeID="t2"/>
<TIMEX3 xml:id="t2" type="DATE" value="1996-03-27" functionInDocument="CREATION_TIME"/>
<TLINK eventID="#e1" relatedToTime="#t1" relType="IS_INCLUDED"/>
```

The <TLINK> may also relate to <TIMEX3> expressions. This is the only representation that will adequately express the temporal anchoring of this event.

(19) John taught last week on Monday.

```
<EVENT xml:id="e1" target="#token1" pred="TEACH" class="OCCURRENCE" type="PROCESS"  tense="PAST"
  aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#range(token2,token3)" pred="LAST_WEEK" type="DATE" value="XXXX-WXX"
 temporalFunction="true" anchorTimeID="t2"/>
<SIGNAL xml:id="s1" target="#token4" pred="on"/>
<TIMEX3 xml:id="t3" target="#token5" pred="MONDAY" type="DATE" value="XXXX-WXX-
1" temporalFunction="true"/>
<TIMEX3 xml:id="t2" type="DATE" value="1996-03-27" functionInDocument="CREATION_TIME"/>
<TLINK eventID="#e1" relatedToTime="#t1" relType="IS_INCLUDED"/>
<TLINK timeID="#t3" signalID="#s1" relatedToTime="#t2" relType="IS_INCLUDED"/>
```

7.3.5.2 <SLINK>

The element <SLINK> represents a subordination link between two events that is used for contexts involving modality, evidentials, and factives. An <SLINK> is used in cases where an event instance subordinates another event instance type. These are cases where a verb takes a complement and subordinates the event instance referred to in this complement. The specification is shown in Table 8.

Table 8 — Specification of <SLINK>

eventID	points to the event involved in a temporal link.
subordinatedEvent	points to the event subordinated to an event in a subordination link.
relType	expresses the kind of subordination relation holding between the two events.
comment	remarks made by an annotator.

The following <EVENT> classes interact with <SLINK>: REPORTING, I_STATE or I_ACTION

Some lexical notes:

— Verbs that introduce I_STATE EVENTs that induce <SLINK>: “want”, “desire”, “crave”, “lust”, “believe”, “doubt”, “suspect”, “hope”, “aspire”, “intend”, “fear”, “hate”, “love”, “enjoy”, “like”, “know”;

— Verbs that introduce I_ACTION EVENTs that induce <SLINK>: “attempt”, “try”, “persuade”, “promise”, “name”, “swear”, “vow”.

Examples:

(20) If Graham leaves today, he will not hear Sabine.

```
<SIGNAL xml:id="s1" target="#token0" pred="IF"/>
<EVENT xml:id="e1" target="#token2" pred="LEAVE" class="OCCURRENCE" type="TRANSITION"  tense="PRESENT"
  aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#token3" pred="TODAY" type="DATE" value="XXXX-XX-XX" temporalFunction="true"/>
<EVENT xml:id="e2" target="#token8" pred="HEAR" class="OCCURRENCE" type="PROCESS" tense="FUTURE"
  aspect="NONE" polarity="NEG"/>
<SLINK eventID="#e1" subordinatedEvent="#e2" signalID="#s1" relType="CONDITIONAL"/>
<TLINK eventID="#e1" relatedToEvent="#e2" relType="BEFORE"/>
```

(21) Bill denied that John taught on Monday.

```
<EVENT xml:id="e1" target="#token1" pred="DENY" class="I_ACTION" type="PROCESS" tense="PAST" aspect="NONE"
  polarity="POS"/>
<EVENT xml:id="e2" target="#token4" pred="TEACH" class="OCCURRENCE" type="PROCESS" tense="PAST"
  aspect="NONE" polarity="POS"/>
<SIGNAL xml:id="s1" target="#token5" pred="ON"/>
<TIMEX3 xml:id="t1" target="#token6" pred="MONDAY" type="DATE" value="XXXX-WXX-1"/>
<TLINK eventID="#e2" signalID="#s1" relatedToTime="#t1" relType="IS_INCLUDED"/>
<SLINK eventID="#e1" subordinatedEvent="#e2" relType="NEG_EVIDENTIAL"/>
```

(22) Bill wants to teach on Monday.

```
<EVENT xml:id="e1" target="#token1" pred="WANT" class="I_STATE" type="STATE" tense="PRESENT" aspect="NONE"
  polarity="POS"/>
<EVENT xml:id="e2" target="#token3" pred="TEACH" class="OCCURRENCE" type="PROCESS" aspect="NONE"
  tense="NONE" polarity="POS"/>
<SIGNAL xml:id="s2" target="#token4" pred="on"/>
<TIMEX3 xml:id="t1" target="#token5" pred="MONDAY" type="DATE" value="XXXX-WXX-1"/>
<TLINK eventID="#e2" signalID="#s2" relatedToTime="#t1" relType="IS_INCLUDED"/>
<SLINK eventID="#e1" subordinatedEvent="#e2" relType="INTENSIONAL"/>
```

(23) John is believed to have lived in Rome.

```
<EVENT xml:id="e1" target="#token2" pred="BELIEVE" class="I_STATE" type="STATE" tense="PRESENT" aspect="
NONE"
  polarity="POS"/>
<EVENT xml:id="e2" target="#token5" pred="LIVE" class="OCCURRENCE" type="STATE" tense="NONE"
  aspect="PERFECTIVE" polarity="POS"/>
<SLINK eventID="#e1" subordinatedEvent="#e2" relType="INTENSIONAL"/>
```

(24) Bill attempted to save her.

```
<EVENT xml:id="e1" target="#token1" pred="ATTEMPT" class="I_ACTION" type="PROCESS" tense="PAST"
  aspect="NONE" polarity="POS"/>
<EVENT xml:id="e2" target="#token3" pred="SAVE" class="OCCURRENCE" type="PROCESS"
  tense="NONE" aspect="NONE" polarity="POS"/>
<SLINK eventID="#e1" subordinatedEvent="#e2" relType="INTENSIONAL"/>
```

7.3.5.3 <ALINK>

The element <ALINK> is an aspectual link; it indicates an aspectual connection between two events. In some ways, it is like a cross between <TLINK> and <SLINK> in that it indicates both a relation between two temporal elements, as well as aspectual subordination. The specification is shown in Table 9.

Table 9 — Specification of <ALINK>

eventID	points to the event involved in a temporal link.
signalID	points, when applicable, to the explicit signal of the temporal relation.
relatedToEvent	points to the entity that is being related to the event if this entity is of an event type.
relType	temporal relation holding between the events.
comment	remarks made by an annotator.

Examples:

(25) The boat began to sink.

```
<EVENT xml:id="e1" target="#token2" pred="BEGIN" class="ASPECTUAL" type="PROCESS" tense="PAST"
  aspect="NONE" polarity="POS"/>
<EVENT xml:id="e2" target="#token4" pred="SINK" class="OCCURRENCE" type="TRANSITION"
  tense="NONE" aspect="NONE" polarity="POS"/>
<ALINK eventID="#e1" relatedToEvent="#e2" relType="INITIATES"/>
```

(26) The search party stopped looking for the survivors.

```
<EVENT xmlid="e1" target="#token3" pred="STOP" class="ASPECTUAL" type="TRANSITION"
  tense="PAST" aspect="NONE" polarity="POS"/>
<EVENT xml:id="e2" target="#token4 #token5" pred="LOOK_FOR" class="OCCURRENCE"
  type="PROCESS" tense="NONE" aspect="NONE" polarity="POS"/>
<ALINK eventID="#e1" relatedToEvent="#e2" relType="TERMINATES"/>
```

7.3.5.4 <MLINK>

The abstract syntax specified in *ISO-TimeML* makes a distinction between an interval and the length of the interval (an “amount of time”; see Bunt and Pustejovsky, 2010). The element <MLINK> is a link for measuring the length of a time span or the duration of an event. <MLINK> has the inherent relation type of MEASURES. A temporal expression such as “3 hours” is expressed. The example after Table 10 illustrates the use.

Table 10 —Specification of <MLINK>

eventID	points to the event involved in a temporal link.
signalID	points, when applicable, to the explicit signal of the temporal relation.
relatedToEvent	points to the entity that is being related to the event if this entity is of an event type.
relType	refers to measures.
comment	remarks made by an annotator.

EXAMPLE

(27) John taught for three hours.

```
<EVENT xml:id="e1" target="#token2" pred="TEACH" class="OCCURRENCE" type="PROCESS"
  tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#range(token4, token5)" type="DURATION" value="P3H"/>
<MLINK eventID="#e1" relatedToTime="#t1" relType="MEASURES"/>
```

7.3.6 Other tags: <CONFIDENCE>, <CERTAINTY> and <ISO-TimeML>

7.3.6.1 <CONFIDENCE> and <CERTAINTY>

It has been noted to be useful to be able to mark confidence values for parts of an annotation. In general, it is useful to allow confidence values to be assigned to any tag, and, in fact, to any attribute of any tag.

This is conveniently done with the <CONFIDENCE> tag, which consumes no input, and which has the specification shown in Table 11.

Table 11 — Specification of <CONFIDENCE>

tagType	points to any element in *ISO-TimeML*
tagID	points to an element under consideration
attributeName	(OPTIONAL) points to a particular aspect in the element under consideration
confidenceValue	ranges over the rationals between 0 and 1, indicating the degree of confidence
comment	remarks made by an annotator

The attribute @tagType would range over the names of all the tags of *ISO-TimeML*. The attribute @tagID would range over the set of actual tag IDs within the current document (XML type URI). The attribute @attributeName would range over the names of all the attributes of all the tags of *ISO-TimeML*. The attribute @confidenceValue would range over the rationals (i.e. would have a floating point value) between 0 and 1.

So, for example, given this annotation:

The TWA flight crash-landed on Easter Island two weeks ago.

```
<EVENT xml:id="e1" target="#token3" pred="CRASHLAND" class="OCCURRENCE" type="TRANSITION"
 tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#range(token7,token9)" pred="TWO_WEEKS_AGO" type="DURATION"
 value="P2W" beginPoint="#t2" endPoint="#t3"/>
<TIMEX3 xml:id="t3" type="DATE" value="1999-12-06" temporalFunction="true" anchorTimeID="#t1"/>
 .........
<TIMEX3 xml:id="t2" type="DATE" functionInDocument="CREATION_TIME" value="1999-12-20"/>
<TLINK eventID="#e1" relatedToTime="#t3" relType="IS_INCLUDED"/>
```

If we want to indicate that we are unsure to have annotated "two weeks ago" correctly, we may add:

```
<CONFIDENCE tagType="TIMEX3" tagID="#t1" confidenceValue="0.50"/>
<CERTAINTY target="#t1" degree="0.50"/>
```

Here the lack of the optional attribute @attributeName indicates that the confidence applies to the whole tag.

On the other hand, if we wanted to indicate that we weren't sure if the tense of “crash-landed” was really PAST, we could add this annotation:

```
<CONFIDENCE tagType="EVNT" tagID="#e1" attributeName="TENSE" confidenceValue="0.75"/>
<CERTAINTY target="#t1" degree="0.50"/>
```

Abstracting confidence measures as a separate tag frees the annotation from having to include a @confidenceValue attribute in every tag and eliminates the problem of uncertainty over the exact attribute of a tag the confidence value applies to.

As for how confidence values should be assigned in manual annotation, we feel that, in a large-scale annotation effort such as TIMEBANK, two conditions should be satisfied:

- Fairly high inter-annotator agreement on the tag assignment in the text;
- Ease of use and habitability of the tool from the annotator's perspective.

Therefore, the annotation of a scalar value such as confidence should have at least two features:

The choice of confidence values should be as clearly defined as possible to cover the options; this relates to the granularity and orders of magnitude as well. This would suggest a selection from a small set (e.g. low, mid, high; not_sure, sure, abolutely_sure). These could be interpreted or rescaled to a (0,1] range, if need be, for subsequent inference. There should be a default value specified [at high (=1)] so that it is not necessary to annotate all links and attributes for them with a confidence. The constraint on human annotators to a subset of the possible values should be documented in the annotation guidelines and implemented in the annotation tool. And it would probably be best if the annotation tool did not present numbers but rather natural language descriptions such as those suggested above, which would be represented in the underlying annotation numerically. For example, the annotator might pick “moderately certain”, which would enter the annotation as 0.5.

Moreover, for manual annotation, it does not seem that the 0 and 1 values will be used/useful. Presumably if the annotator does not trust an annotation at all s/he will not add it. And, as was suggested above, 1, at least for manual annotation, should be the default or unmarked value, and so need not be noted, since it would bulk up the files considerably, even if it were used only on entire tags.

7.3.6.2 The root element <isoTimeML>

Inasmuch as every well-formed XML document must have a single root node, we supply the <*ISO-TimeML*> element as this node. For example, a sample annotated *ISO-TimeML* document might look like this:

Source text:

FAMILIES SUE OVER AREOFLOT CRASH DEATHS

The Russian airline Aeroflot has been hit with a writ for loss and damages, filed in Hong Kong by the families of seven passengers killed in an air crash. All 75 people died on board the Aeroflot Airbus when it ploughed into a Siberian mountain in March 1994.

Tokenized text:

ISO 24611 *Language resource management – Morpho-syntactic annotation framework* (MAF) allows the following tokenization in an in-line format:

```
<maf>
 <token xml:id="token0">FAMILIES</token>
 <token xml:id="token1">SUE</token>
 <token xml:id="token2">OVER</token>
 <token xml:id="token3">AREOFLOT</token>
 <token xml:id="token4">CRAQSH</token>
 <token xml:id="token5">DEATHS</token>
 <token xml:id="token6">The</token>
 <token xml:id="token7">Russian</token>
 <token xml:id="token8">airline</token>
 <token xml:id="token9">Aeroflot</token>
 <token xml:id="token10">has</token>
 <token xml:id="token11">been</token>
 <token xml:id="token12">hit</token>
 <token xml:id="token13">with</token>
 <token xml:id="token14">a</token>
 <token xml:id="token15">writ</token>
 <token xml:id="token16">for</token>
 <token xml:id="token17">loss</token>
 <token xml:id="token18">and</token>
 <token xml:id="token19">damages</token>
 <token xml:id="token20">,</token>
 <token xml:id="token21">filed</token>
 <token xml:id="token22">in</token>
 <token xml:id="token23">Hong</token>
 <token xml:id="token24">Kong</token>
 <token xml:id="token25">by</token>
 <token xml:id="token26">the</token>
 <token xml:id="token27">family</token>
 <token xml:id="token28">of</token>
 <token xml:id="token29">seven</token>
 <token xml:id="token30">passengers</token>
 <token xml:id="token31">killed</token>
 <token xml:id="token32">in</token>
 <token xml:id="token33">an</token>
 <token xml:id="token34">air</token>
 <token xml:id="token35">crash</token>
 <token xml:id="token36">.</token>
 <token xml:id="token37">All</token>
 <token xml:id="token38">75</token>
 <token xml:id="token39">people</token>
 <token xml:id="token40">died</token>
 <token xml:id="token41">on</token>
 <token xml:id="token42">board</token>
 <token xml:id="token43">the</token>
 <token xml:id="token44">Aeroflot</token>
 <token xml:id="token45">Airbus</token>
 <token xml:id="token46">when</token>
 <token xml:id="token47">it</token>
 <token xml:id="token48">ploughed</token>
 <token xml:id="token49">into</token>
 <token xml:id="token50">a</token>
 <token xml:id="token51">Siberian</token>
 <token xml:id="token52">mountain</token>
 <token xml:id="token53">in</token>
 <token xml:id="token54">March </token>
 <token xml:id="token55">1994</token>
 <token xml:id="token56">.</token>
</maf>

<isoTimeML>
 <EVENT xml:id="e1" target="#token12" pred="HIT" class="OCCURRENCE" type="TRANSITION"
   tense="PRESENT" aspect="PERFECTIVE" polarity="POS"/>
 <EVENT xml:id="e2" target="#token21" pred="FILE" class="OCCURRENCE" type="PROCESS"
   tense="PAST" aspect="NONE polarity="POS"/>
 <EVENT xml:id="e3" target="#token31" pred="KILL" class="OCCURRENCE" type="TRANSITON"
   tense="PAST" aspect="NONE" polarity="POS"/>
 <SIGNAL xml:id="s1" target="#token32"/>
 <EVENT xml:id="e4" target="#token35" pred="CRASH" class="OCCURRENCE" type="TRANSITION"
   tense="NONE" aspect="NONE" polarity="POS"/>
 <EVENT xml:id="e5" target="#token40" pred="DIE" class="OCCURRENCE" type="TRANSITION"
   tense="PAST" aspect="NONE" polarity="POS"/>
 <EVENT xml:id="e6" target="#token41 #token42" pred="ON_BOARD" class="OCCURRENCE" type="STATE"
   tense="NONE" aspect="NONE" polarity="POS"/>
 <SIGNAL xml:id="s2" target="#token46" pred="WHEN"/>
 <EVENT xml:id="e7" target="#token48" pred="PLOUGH" class="OCCURRENCE" type="TRANSITION"
   tense="PAST" aspect="NONE" polarity="POS"/>
 <TLINK eventID="#e7" relatedToEvent="#e5" relType="INCLUDES"/>
 <TLINK eventID="#e5" signalID="#s2" relatedToEvent="#e7" relType="IAFTER"/>
 <TLINK eventID="#e7" signalID="#s3" relatedToTime="#t2" relType="IS_INCLUDED"/>
 <TLINK eventID="e7" relatedToEvent="#e4" relType="IDENTITY"/>
 <SIGNAL xml:id="s3" target="#token53" pred="IN"/>
 <TIMEX3 xml:id="t2"  target="range(#token55,#token56)" pred="MARCH_1994" type="DATE" value="1994-03"/>
  ...
 <TIMEX3 xml:id="t1" type="DATE" value="1996-03-27"/>
 <TLINK eventID="#e1" relatedToTime="#t1" relType="BEFORE"/>
 <TLINK eventID="#e2" relatedToEvent="#e1" relType="BEFORE"/>
 <TLINK eventID="#e3" relatedToEvent="#e2" relType="BEFORE"/>
 <TLINK eventID="#e3" signalID="#s1" relatedToEvent="#e4" relType="IS_INCLUDED"/>
 <TLINK eventID="#e7" signalID="#s3" relatedToTime="#t2" relType="IS_INCLUDED"/>
</isoTimeML>
```

Some further notes concerning changes from version TimeML 1.2 to *ISO-TimeML*:

1. The @nf_morph attribute that was part of <MAKEINSTANCE> has been changed to the attribute @pos (part of speech), and the PASTPART, PRESPART, INFINITIVE, and GERUNDIVE values of nf_morph have been redistributed to the attribute @vform. Both are syntactic, rather than semantic attributes, and therefore belong to another layer of annotation than *ISO-TimeML*. They are considered as optional attributes here. The same holds for the attribute @syntax – see next point.

2. The optional attribute @syntax was added to <SLINK>, <ALINK>, and <TLINK>. The attribute @syntax can be used to hold <CDATA>, but is generally only used by annotation programs to hold the data that led to the creation of the tag.

3. The optional attribute @comment was added to all *ISO-TimeML* elements, for the purpose of giving (human) annotators a place to put observations about annotated text.

8 Towards a semantics for ISO-TimeML

8.1 Overview

The syntactic apparatus of *ISO-TimeML* touches on several distinct areas of linguistic description. These include temporal semantics, as well as issues of modally subordinating contexts and the semantics of Aktionsarten (event classifications). In this clause, we focus on the semantics of the first of these areas, in particular: the representation of predicates as events; the representation of temporal expressions as intervals or quantification over intervals; and the relations between these interval structures. Besides the interval semantics of *ISO-TimeML* to be presented in clause 8.4, an event-based semantics will also be presented in 8.5.

8.2 Tense and aspect in language

8.2.1 Tense

Tense can be defined as "the grammaticalized expression of location in time" (Comrie, 1986). This grammaticalized expression involves marking, via change of form, of particular syntactic elements, e.g. the verb and auxiliaries. For example, in "John ran a marathon", the past tense morpheme represented as "-ed" (producing the inflected verb form "ran") is used to indicate that the event occurred at a time earlier than the speech time. In "John will run a marathon", the modal auxiliary "will" is used to locate the event as occurring at a future time, i.e. later than the speech time. While tense is mainly marked on the verb and auxiliaries associated with the verb group, in some languages, such as the North American Indian language Nootka (Comrie, 1986), tense is expressed on the noun phrase.

Tense is not the only mechanism for expressing location in time. In languages such as Mandarin Chinese, which lacks tense morphemes, aspectual markers can be used to express location in time, though sometimes even these may be absent (Lin, 2003). There are also non-grammaticalized expressions of location in time given by temporal adverbials, e.g. "tomorrow", "yesterday", "two hours later", etc. In the case of "tomorrow" or "yesterday", the temporal location is with respect to the speech time. Temporal locations can also of course be expressed relative to a coordinate system given by a calendar, e.g. "1991 (C.E.)", or a cyclically occurring event, e.g. "morning", "spring", or an arbitrary event, e.g. "the day after he married her".

The few languages that lack tense altogether are not able to distinguish past from present or future. However, they all have a "realis/irrealis" distinction. In Burmese, for example (Comrie 1986[34]), events that are ongoing or that were observed in the past are expressed by sentence-final realis particles "–te", "-th'a", "-ta", and "–hta". In other cases, i.e. for unreal or hypothetical events (including future events, present events, and hypothetical past events), the sentence-final irrealis particles "–m'e", "-ma", and "–hma" are used.

ISO-TimeML expresses four values for the attribute @tense plus the value NONE. These are: FUTURE, PAST, PRESENT, and IMPERFECT.

8.2.2 Aspect

ISO-TimeML makes the traditional linguistic distinction between tense and aspect. The tense attribute values cover all the languages thus far examined by the *ISO-TimeML* Specification Working Group, but obviously there will be languages not covered by the current set.

While tense allows the speaker to relate the time of an eventuality to a deictic centre or some other reference point, grammatical aspect allows the speaker to represent the structure of an eventuality. Here there is a distinction between ***perfective aspect***, where an entire eventuality is presented without its internal temporal structure, e.g. "John built a house", and ***imperfective aspect***, where the speaker represents internal phases of the eventuality, e.g. "John is building a house". Perfective aspect can express termination or completion of an eventuality, while imperfective aspect can express the ongoing nature of an activity. It is important to realize that many of the traditional tenses, e.g. Spanish imperfective, as in "Juan leia cuando entre" ("John was reading when I entered"), may combine both tense and aspect, e.g. past and imperfective. The same is true of the 'complex tenses' in English, such as present progressive, present perfect, etc. Grammatical aspect is expressed in systematic ways across languages, depending on the lexical aspect of the eventuality. The following cross-linguistic account, derived from arguments by Smith (1991), summarizes some of this systematicity.

In English and French, ***perfective aspect*** is signalled by verbal tense and aspect morphemes. Termination is expressed in activities, completion is expressed in accomplishments and achievements, and statives can either express termination (e.g. French ***passé composé*** tense morpheme) or not (English, e.g. "I have lived in Paris"). In Mandarin Chinese, which lacks tense markers but which does have the semantic notion of tense (Lin, 2003), the perfective is signalled by morphemes "–le" and "–guo", usually indicating termination for activities, accomplishments and achievements; completion is indicated by a separate resultative morpheme "–wan". In Russian, the perfective does not apply to statives, but is signalled by prefixes "po-" (short duration) and "pro-" (unexpected interval) in activities.

The ***imperfective aspect*** is signalled in English by the progressive morpheme "-ing". It occurs in activities, accomplishments and achievements. In French, as in Russian, it is signalled by tense morphemes (e.g. the French ***imparfait***). In Mandarin, it is signalled by the progressive morpheme "–zai" and resultative morpheme "–zhe". The particle "-le" can also have an imperfective use with atelic predicates (Lin, 2003).

ISO-TimeML expresses five values for the attribute @aspect, plus the value NONE. These are: PROGRESSIVE, PERFECTIVE, IMPERFECTIVE, PERFECTIVE_PROG, and IMPERFECTIVE_PROG. The values not mentioned here are described in the annexes for those languages introducing them.

8.3 Temporal relations

ISO-TimeML assumes the general framework of Allen's (1984) interval algebra. In Allen's interval algebra, there are 13 basic (binary) interval relations, where six are inverses of the other six, excluding equality.

a. before (b), after (bi); (1)

b. overlap (o), overlappedBy (oi);

c. start (s), startedBy (si);

d. finish (f), finishedBy (fi);

e. during (d), contains (di);

f. meet (m), metBy (mi);

g. equality (eq).

These are shown schematically in Figure 2.

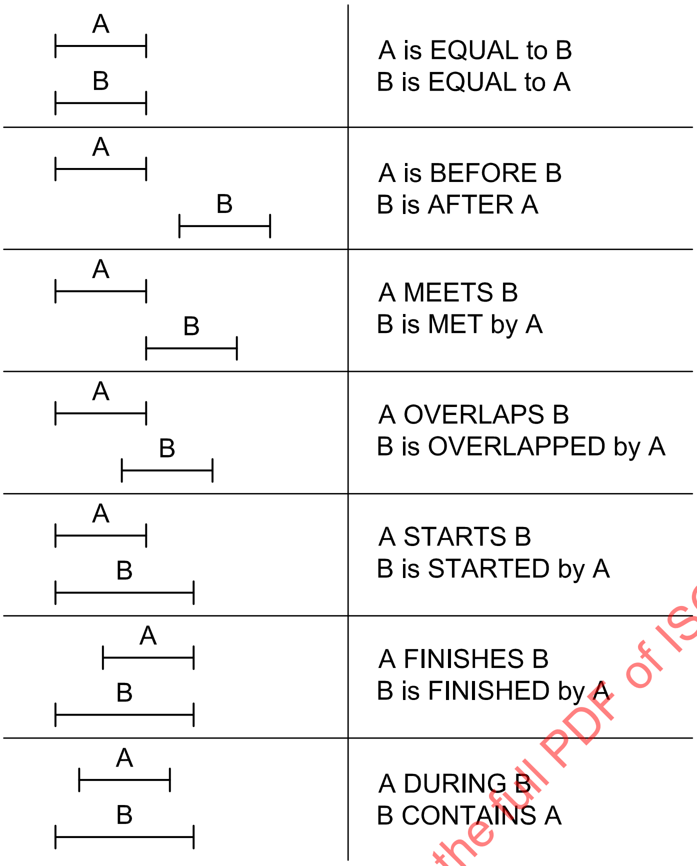

Figure 2 — Allen's interval relations

In the clauses that follow, we will follow the syntax for *ISO-TimeML* by assuming a single type of event class where there is no distinction made in Aktionsarten. Subsequent work within the *ISO-TimeML* community will likely extend the specification to account for differing event types in language, but this is not a necessary component to the current scope of the specification.

8.4 An interval-based semantics for ISO-TimeML

8.4.1 Technical preliminaries for interval temporal logic

We assume the usual apparatus and notation of the Simply Typed Lambda Calculus with primitive types i and t. The type i is identified with the set of non-empty compact, connected subsets of the reals (hereinafter: intervals), while the type t is identified with the set $\{\top , \bot\}$ (hereinafter: Booleans). The primitive types are assumed to be mutually disjoint. We further assume a variety of temporal and logical constants. For example, the constant $\wedge$ of type $(t(tt))$ will be used to represent conjunction in the usual way. Likewise, the constant Q_{EVERY} of type $((it)(it)t)$ will map two properties of intervals to $\top$ just in case every interval satisfying the first satisfies the second. Such constants will be introduced as and when required.

The semantics presented here concerns only a subset of *ISO-TimeML*. This is partly because some of the information recorded in *ISO-TimeML* is essentially syntactic rather than semantic in character, and partly because much of the semantic content of natural language texts has, for one reason or another, no clear rendition within the usual apparatus of formal semantics. Specifically, only three varieties of *ISO-TimeML* elements: <EVENT>, <TIMEX3> and <TLINK> are considered. In addition, only certain slots within these elements are taken to contribute to the semantics.

8.4.2 Basic event-structure

8.4.2.1 Sample annotation

Consider the following text:

After his talk with Mary, John drove to Boston. (2)

The *ISO-TimeML* annotation of this text contains a pair of <EVENT> elements and a <TLINK> element:

(3)

<EVENT xml:id="e2" **target=**"#token2" **pred=**"TALK"/>
<EVENT xml:id="e1" **target=**"#token7" **pred=**"DRIVE"/>
<TLINK eventID="#e1" **relatedToEvent=**"#e2" **relType=**"AFTER"/>

8.4.2.2 Representation in a first-order formula

At the same time, the event-structure of (2) can be captured by the first-order formula

$$p_{\mathrm{DRIVE}}\,(I_{e1}) \wedge p_{\mathrm{TALK}}\,(I_{e2}\,) \wedge R_{\mathrm{AFTER}}\,(I_{e1,}\,I_{e2}), \tag{4}$$

where the variables I_{e1} and I_{e2} range over intervals, the unary predicates p_{DRIVE} and p_{TALK} are interpreted as the sets of such intervals over which John drove to Boston and John talked to Mary, respectively, and the binary predicate R_{AFTER} is interpreted as the relation which holds between intervals $[a,b]$ and $[c,d]$ just in case $d < a$. The variables I_{e1} and I_{e2} may be assumed by default to be bound by existential quantifiers that take scope over the whole text; however, this default assumption may be overridden by further *ISO-TimeML* elements, as explained below. Making these global existential quantifiers explicit would add no information, and we do not do so.

Comparing the sample annotation of the two events and their temporal link (3) and its semantic form (4), we see that each attribute pred-value π gives rise to a unary predicate p_π, each xml:id-value ei gives rise to a variable I_{ei}, and each attribute relType-value r gives rise to a binary predicate R_{r}.

For compatibility with the more complicated cases treated below, we reformulate the semantics proposed in (4) using the syntax of higher-order logic. Under this regime, all constituents — logical constants, illogical constants and variables — are regarded as functions of (at most) one argument, and the application of a function f to an argument a is denoted $(f\ a)$. In addition, where convenient, we split a conjunction $\phi \wedge \psi$ in higher-order-logic notation: $((\wedge\ \phi)\ \psi)$ into the pair of formulas ϕ and ψ. Making use of this facility for one of the conjuncts in (4) yields the pair of higher-order logic formulas

$$(p_{\pi 1}\,I_{e1}),\ ((\ \wedge\ (p_{\pi 2}\,I_{e2}\ ))(R_{\mathrm{AFTER}}I_{e1})I_{e2})). \tag{5}$$

Representations in the style of (5) are easier to generate from *ISO-TimeML* annotations than those in the style of (4); for the sake of readability, however, we continue to give both forms.

8.4.2.3 Interpretation rules

We can generate the formulas (5) as follows. Consider any <EVENT> element with xml:id-value ei and pred-value π. We first create a new variable I_{ei}, of type i, and a new constant p_π, of type $(i\ t)$. In addition, we assign to ei a semantic value $\sigma_{ei,}$ which is an expression of higher-order logic featuring I_{ei} and p_π. (In the cases encountered in (3), all semantics values have Boolean type, so we can think of them as ordinary formulas.) The interpretation rule for simple <EVENT>-tags such as those encountered in (3) may be given as

<EVENT xml:id="ei" **pred=**"π"/> $\rightarrow \sigma_{ei} := p_\pi\, I_{ei}$ (6)

Since any xml:id-value ei is allowed to occur in at most one EVENT-tag, these assignments cannot clash.

Suppose, then, that ei is the value of an attribute @xml:id in some element <EVENT>, so that σ_{ei} has been assigned. We take any <TLINK> element with an attribute @relatedToEvent with value ej to modify the expression assigned to σ_{ei}. In the case encountered in (3), this modification results simply in the addition of a conjunct expressing a temporal relation between two events, thus:

<**TLINK eventID**="#ei" **relatedToEvent**="#ej" **relType**="r"/> $\rightarrow \sigma_{ei} := ((\wedge\ \sigma_{ei})((R_r I_{ei})I_{ej})),$ (7)

where the R_r is the constant of type $(i(it))$ corresponding to $r.$

By applying rule (6) and subsequently rule (7) to example (3), we obtain the assignments

$\sigma_{e1} := p_{\pi 1} I_{e1}$
$\sigma_{e2} := ((\wedge\ (p_{\pi 2} I_{e2}))((R_{\text{AFTER}} I_{e1})I_{e2})),$

Collecting together these assignments yields the two formulas (5), as required.

In general, ei may be the relatedToEvent-value of many <TLINK>-tags such as the one encountered in (3). In that case, the corresponding instances of rule (7) may be applied in any order; the chosen order will not affect the representation of event-structure, modulo logical equivalence.

8.4.2.4 Negated events

We turn now to negated events. Consider

John did not drive to Boston, (8)

which has *ISO-TimeML* mark-up

<**EVENT xml:id**="e1" **target**="#token3" **pred**="DRIVE" **polarity**="NEG"/> (9)

We take (8) to assert that, within some contextually determined interval, no event of John's driving to Boston took place. If we represent our contextually determined interval using the variable I_{e1} (corresponding to the id-value of the relevant <EVENT>), then we can express these truth-conditions using the formula

$\neg\exists I_1(R_{\text{DURING}}(I_{1,}\ I_{e1}) \wedge p_{\text{DRIVE}}(I_1)),$ (10)

where the binary predicate R_{DURING} is interpreted as the relation which holds between intervals $[a,b]$ and $[c,d]$ just in case $c < a$ and $b < d$. Using the notation of higher-order logic, (10) may be re-written

$(Q_{=0\text{X}}\lambda I_{1.}((\wedge\ ((R_{\text{DURING}}\ I_1)\ I_{e1}))(p_\pi(I_1))),$ (11)

where the unary quantifier $Q_{=0\text{X}}$, of type $((it)t)$, has the obvious interpretation: it maps a property of intervals to $\top$ just in case that property has no instances. The formula (11) can be generated from (9) by the interpretation rule

<EVENT xml:id="ei" pred="π" polarity=NEG> (12)
$\rightarrow \sigma_{\text{ei}} := (Q_{=0\text{X}}\lambda I_1((I_1),((\wedge((R_{\text{DURING}}\ I_{1)}\ I_{ei}))(p_\pi(I_1))),$

where I_1 is a variable of type i, and p_π is the constant of type (it) corresponding to e.

8.4.3 The interpretation of <TIMEX3>

8.4.3.1 Three types of <TIMEX3> elements

The use of the element <TIMEX3> comes in many varieties:

- Unquantified use: <**TIMEX3 xml:id**="t61" **type**="DATE" **value**="2003-06-07"/>
- Frequency: <**TIMEX3 xml:id**="t1" **pred**="EVERY_MONDAY" **type**="SET" **value**="XXXX-WXX-1" **quant**="EVERY"/>
- Quantified use: <**TIMEX3 xml:id**="t3" **pred**="TWICE_A_MONTH" **type**="SET" **value**="P1M" **freq**="2X"/>

Clearly, we cannot expect a uniform semantic treatment of these different uses of the element <TIMEX3>.

8.4.3.2 Unquantified <TIMEX3> element

Let us begin with the simplest case. Sometimes, <TIMEX3> can be treated analogously to <EVENT>. Consider, for example,

John drove to Boston on Saturday, 31st January, 2004. (13)

This text may be marked up as

(14)

<**EVENT xml:id**="e1" **target**="#token1" **pred**="DRIVE"/>
<**TIMEX3 xml:id**="t1" **target**="#range(token5,token10)" **value**="2004-01-31"/>
<**TLINK eventID**="#e1" **relatedToTime**="#t1" **relType**="DURING"/>

and its event-structure may be represented by the formula

$$p_{\text{2004-01-31}}(I_{t1}) \wedge p_{\text{DRIVE}}(I_{e1}) \wedge R_{\text{DURING}}(I_{e1},I_{t1}), \qquad (15)$$

where the variable I_{t1} ranges over intervals, and the unary predicate $p_{\text{2004-01-31}}$ is interpreted as the (singleton) set of time-intervals coinciding with the 31st January, 2004. Re-writing (15) in the style of (5), we obtain

$$(p_{\text{2004-01-31}}(I_{t1})),((\wedge\ (p_{\text{DRIVE}}\ I_{e1}))((R_{\text{DURING}}\ I_{e1})I_{t1})). \qquad (16)$$

This example suggests the following interpretation process. Given a <TIMEX3> element with @xml:id-value t, we create a new variable I_t of type t, and we assign the semantic value σ_t according to the rule

<**TIMEX3 xml:id**="t" **value**="v"/> (17)
$\rightarrow \sigma_t := (p_v I_t)$
where p_v is the constant, of type (it), corresponding to v.

In addition, we interpret the <EVENT> element in (14) using the rule (6), and the <TLINK> element using the following generalization of rule (7):

<TLINK (18)
{eventID|timeID}=x
{relatedToEvent|relatedToTime}=y
relType=r />
$\rightarrow \sigma_x := ((\wedge\ \sigma_x))((R_r I_x)I_y)).$

These rules generate the representation of event-structure (16) from (14), exactly as for example (2).

8.4.3.3 Quantifying <TIMEX3> element

Other <TIMEX3> constructs make a more complex semantic contribution. Consider, for example,

John drove to Boston twice. (19)

which has *ISO-TimeML* mark-up

(20)

<EVENT xml:id="e1" **target**="#token1" **pred**="DRIVE"/>
<TIMEX3 xml:id="t1" **target**="#token4" **pred**="TWICE" **freq**="2X"/>
<TLINK eventID="#e1" **relatedToTime**="#t1" **relType**="DURING"/>

Assuming that “twice” here has the reading “at least twice”, we take (19) to assert that, within some contextually determined interval, at least two events of John's driving to Boston took place. If we represent this interval using the variable I_{t1} (corresponding to the xml:id-value of the relevant <TIMEX3> element), then we can express these truth-conditions using the formula

$$\exists_{e2} I_{e1}(R_{\mathrm{DURING}}(I_{e1,}\ I_{t1}) \wedge p_{\pi}(I_{e1})), \tag{21}$$

where, as before, the predicate p_{DRIVE} is true of precisely those intervals over which John drives to Boston. Note that the variable I_{e1} arising from the EVENT-tag is bound by a quantifier in (21).

In the notation of higher-order logic, (21) becomes

$$(Q_{2\mathrm{X}} \lambda I_{e1.}((\ \wedge\ ((R_{\mathrm{DURING}}\ I_{e1})\ I_{t1}))(p_{\pi}(I_{e1}))), \tag{22}$$

where $Q_{2\mathrm{X}}$ is the function mapping a property of intervals to $\top$ just in case it has at least two instances.

To generate (22) from the *ISO-TimeML*-tags in (20), we adopt the rule

<TIMEX3 xml:id=t freq=q /> $\rightarrow \lambda R \lambda p.(Q_{\mathrm{q}} \lambda I_{1.}((\ \wedge\ ((RI_1)\ I_t))(p\ (I_1))),$ (23)

where R is a variable of type $(i(it))$, p a variable of type (it), I_1 a variable of type i, Q_{q} the constant of type $((it)t)$ corresponding to q, and I_t the variable of type i corresponding to t. We also adopt the rule

<TLINK (24)
{eventID|timeID}=x
{relatedToEvent|relatedToTime}=y
relType=r />
$\rightarrow \sigma_y := ((\sigma_y R_{\mathrm{r}})\lambda I_x.\ \sigma_x\ )$
$\sigma_x :=$ undefined

where R_r is the constant of type $(i(it))$ corresponding to r, and I_x the variable of type i corresponding to x.

Let us work through the example (20) to see how these rules produce the event-structure representation (21). First,

<EVENT xml:id="e1" **pred**="DRIVE"/> (25)

is processed by rule (6), yielding the assignment

$$\sigma_{e1} := (p_{\text{DRIVE}}\, I_{e1}), \tag{26}$$

while

```
<TIMEX3 xml:id="t1" freq="2X"/>                                              (27)
```

is processed by rule (23), yielding the assignment

$$\sigma_{t1} := \lambda R \lambda p.(Q_{2\text{X}} \lambda I_1.((\wedge\ ((RI_1)\ I_{t1}))(p\ (I_1))). \tag{28}$$

Note that (28) is a function which takes a relation between intervals as argument, and returns a function which itself takes a property of intervals as argument. These assignments having been made,

```
<TLINK eventID="#e1" relatedToTime="#t1" relType="DURING"/>                  (29)
```

is processed by rule (24) to yield the re-assignment

$$\sigma_{t1} := ((\lambda R \lambda p.(Q_{2\text{X}} \lambda I_1.((\wedge\ ((RI_1)\ I_{t1}))(p\ I_1)))R_{\text{DURING}})\lambda I_{e1}.(p_{\text{DRIVE}}\, I_{e1})), \tag{30}$$

and the de-assignment of σ_{e1}. Thus, rule (24) supplies arguments for the function in (28), and gets rid of the assignment in (26) altogether. Routine calculation shows that (30) normalizes to (22); and since, following the de-assignment of σ_{e1}, there are no other semantic values to consider, this is the final semantics for (20), as required.

There is one further matter to discuss in relation to this example. As part of the process of de-assigning σ_{e1}, we assume that $e1$ is made inaccessible to any other TLINK-tags than the one which caused rule (24) to be applied. That is, no other TLINK-tags may have $e1$ as the value of any of the slots @eventID, @timeID, @relatedToEvent or @relatedToTime. This inaccessibility reflects the fact that we cannot meaningfully use a TLINK-tag to relate $e1$ to any other xml:id-value. After all, if such a TLINK-tag did involve $e1$, which of the two (or more) events of John's driving to Boston would the relation involve? *ISO-TimeML*-marked-up text violating this accessibility constraint is considered uninterpretable.

8.4.3.4 Quantifying <TIMEX3> element with unquantified complements

Moving on to a different kind of <TIMEX3> construct, consider

John drove to Boston every Monday. (31)

This text may be marked up as

```
                                                                             (32)
<EVENT xml:id="e1" target="#token1" pred="DRIVE"/>
<TIMEX3 xml:id="t1"  pred="EVERY_MONDAY"  type="SET"  value="XXXX-WXX-1"  quant="EVERY"/>
<TLINK eventID="#e1" relatedToTime="#t1" relType="DURING"/>
```

and its event-structure may be represented by the formula

$$\forall I_1((p_{\text{XXXX-WXX-1}}(I_1) \wedge R_{\text{DURING}}(I_1, I_{t1})) \rightarrow \exists I_{e1}(p_{\text{DRIVE}}(I_{e1}) \wedge R_{\text{DURING}}(I_{e1}, I_1))), \tag{33}$$

where the free variable I_{t1} represents some contextually given interval to which the universal quantification in (31) is confined, the unary predicate $p_{\text{XXXX-WXX-1}}$ is satisfied by precisely those intervals which coincide with Mondays, and the predicates p_{DRIVE} and R_{DURING} are as above. Thus, we take (31) to exhibit two levels of quantification: explicit universal quantification over Mondays, and implicit existential quantification over events within each of those Mondays.

In the notation of higher-order logic, (33) may be written

$$((Q_{\mathrm{EVERY}}\lambda I_1.((\wedge\ (p_{\mathrm{XXXX\text{-}WXX\text{-}1}}\ I_1))((R_{\mathrm{DURING}}\ I_1)\ I_{t1})))\lambda I_2.(Q_{1X}\lambda I_{e1}.((\wedge\ (p_{\mathrm{DRIVE}}\ I_{e1}))((R_{\mathrm{DURING}}\ I_{e1})I_2)))). \qquad (34)$$

Here, Q_{EVERY} is the constant of type $((it)((it)t))$ expressing the binary universal quantifier, and Q_{1X} the constant of type $((it)t)$ expressing the unary existential quantifier. That is, for all p, Q of type (it): $((Q_{\mathrm{EVERY}}\ p)Q) = T$ if and only if, for every interval I, $(pI) = T$ implies $(QI) = T$; and $(Q_{1X}p) = T$ if and only if, for some interval I, $(pI) = T$.

The question now is how to recover (34) from the *ISO-TimeML* tags in (32). We propose the rule

<**TIMEX3** **xml:id**="t" **type**="SET" **value**="Rv" **quant**="q"/> (35)
$\rightarrow \sigma_t := \lambda R\lambda p.((Q_{\mathrm{q}}\lambda I_1.((\wedge\ ((R_{\mathrm{DURING}}I_1)I_t))(vI_1)))\lambda I_2.(Q_{1\mathrm{X}}\lambda I_3.((\wedge\ ((RI_3)I_2))(pI_3))))$.

With the above rules at our disposal,

<EVENT xml:id="e1" target="#token1" pred="DRIVE"/> (36)

is processed by rule (6), yielding the assignment

$$\sigma_{e1} := (p_{\mathrm{DRIVE}}\ I_{e1}), \qquad (37)$$

while

<**TIMEX3** **xml:id**="t1" **pred**="EVERY_MONDAY" **type**="SET" **value**="XXXX-WXX-1" **quant**="EVERY"/> (38)

is processed by rule (35), yielding the assignment

$$\sigma_{t1} := \lambda R\lambda p.((Q_{\mathrm{EVERY}}\lambda I_1.((\wedge\ ((R_{\mathrm{DURING}}\ I_1)I_{t1}))(p_{\mathrm{XXXX\text{-}WXX\text{-}1}}\ I_1)))\lambda I_2.(Q_{1\mathrm{X}}\lambda I_3.((\wedge\ ((RI_3)I_2))(pI_3)))). \qquad (39)$$

These assignments having been made,

<**TLINK** **eventID**="#e1" **relatedToTime**="#t1" **relType**="DURING"/> (40)

is processed by rule (24), yielding the re-assignment

$$\sigma_{t1} := ((\lambda R\lambda p.((Q_{\mathrm{EVERY}}\lambda I_1.((\wedge\ ((R_{\mathrm{DURING}}\ I_1)I_{t1}))(p_{\mathrm{XXXX\text{-}WXX\text{-}1}}\ I_1))) \\ \lambda I_2.(Q_{1\mathrm{X}}\lambda I_3.((\wedge\ ((RI_3)I_2))(pI_3))))R_{\mathrm{DURING}})\lambda I_{e1}.(p_{\mathrm{DRIVE}}\ I_{e1})), \qquad (41)$$

and the de-assignment of σ_{e1}. But (41) normalizes to (34); and since there are no other semantic values to consider, this is the final semantics for (32), as required.

In some texts, the implicit existential quantification over events of the kind exhibited in (31) is made explicit. Consider, for example,

John drove to Boston twice a month. (42)

(Again, we assume that “twice” here has the reading “at least twice”.) This text may be marked up as

(43)

<**EVENT** **xml:id**="e1" **target**="#token1" **pred**="DRIVE"/>
<**TIMEX3** **xml:id**="t1" **target**="#range(token4,token6)" **type**="SET" **value**="month"
quant="EVERY" **freq**="2X"/>
<**TLINK** **eventID**="#e1" **relatedToTime**="#t1" **relType**="DURING"/>

and its event-structure may be represented by the formula

$$\forall J((p_{\mathrm{MONTH}}(J) \wedge R_{\mathrm{DURING}}(J, I_{t1})) \rightarrow \exists_{\geq 2} I_{e1}(p_{\mathrm{DRIVE}}(I_{e1}) \wedge R_{\mathrm{DURING}}(I_{e1}, J))), \tag{44}$$

where the variable I_{t1} represents the interval to which universal quantification over months is restricted, and the unary predicate p_{MONTH} is true of exactly those intervals which are (calendar) months. Here we have two levels of quantification: universal quantification restricted to I_{t1}, and “at-least-twice” quantification restricted to calendar months included in I_{t1}. The universal quantification is somewhat obliquely signalled by the English “a month”, but is encoded directly in the quant-value of the TIMEX3-tag.

We may re-write (44) using the notation of higher-order logic as

$$((Q_{\mathrm{EVERY}}\lambda I_1.((\wedge\ (p_{\mathrm{MONTH}}\ I_1))((R_{\mathrm{DURING}}\ I_1)I_{t1})))\lambda I_2.(Q_{2\mathrm{X}}\lambda I_{e1}.((\wedge\ (p_{\mathrm{DRIVE}}\ I_{e1}))((R_{\mathrm{DURING}}\ I_{e1})I_2)))). \tag{45}$$

This suggests the interpretation rule

<TIMEX3 xml:id="t" **type**="SET" **value**="Rv" **quant**="q" **freq**="q′ "/> (46)

$\rightarrow \sigma_t := \lambda R \lambda p.((Q_q \lambda I_1.((\wedge\ ((R_{\mathrm{DURING}}\ I_1)I_t))(vI_1)))\lambda I_2.(Q_{q'}\lambda I_3.((\wedge\ ((RI_3)I_2))(pI_3))))$.

with other tags interpreted as before.

It may be verified that, after applying these rules and normalizing, σ_{t1} is assigned the expression (45), while σ_{e1} is de-assigned. Since there are no other semantic values to consider, (45) — or its more readable equivalent, (44) — is the meaning of the *ISO-TimeML* tags in (43).

8.4.3.5 Quantifying <TIMEX3> elements with quantified complements

In example (20), the EVENT-tag, corresponding to “John drove to Boston” contributes an unquantified expression ($p_{\mathrm{DRIVE}}\ I_{e1}$), while the TIMEX3-tag, corresponding to the adverbial “twice”, contributes a quantifier Q_{2x}, binding the variable I_{e1}. Similarly, in example (32), the EVENT-tag again contributes the unquantified expression ($p_{\mathrm{DRIVE}}\ I_{e1}$), while the TIMEX3-tag, corresponding to the adverbial “every Monday”, contributes an (implicit) existential quantifier Q_{1x}, again binding the variable I_{e1}. In both these cases, a single temporal adverbial provides a quantifier binding the variable in an unquantified event-reporting expression.

It can also happen, however, that a temporal adverbial modifies an expression which has already been explicitly quantified by another adverbial. Since both adverbials will be marked with TIMEX3-tags, additional rules are required to interpret these tags. Consider, for example,

One Monday, John drove to Boston twice, (47)

where the adverbial “One Monday” modifies the explicitly quantified John drove to Boston twice. This text may be marked up as

(48)

```
<TIMEX3  xml:id="t2" target="#token0 #token1" pred="ONE_MONDAY"
 type="SET" value="XXXX-WXX-1" quant="SOME"/>
<EVENT xml:id="e1" target="token4" pred="DRIVE"/>
<TIMEX3  xml:id="t1" target="token7" pred="TWICE" freq="2X"/>
<TLINK eventID="#e1" relatedToTime="#t1" relType="DURING"/>
<TLINK xml:id="t1" relatedToTime="#t2" relType="DURING"/>
```

with the two TLINK-tags forming, so to speak, a cascade: e1→ t1→ t2. The event-structure of (47) may be represented by the formula

$$\exists I_1(p_{\mathrm{XXXX\text{-}WXX\text{-}1}}(I_1) \wedge R_{\mathrm{DURING}}(I_1, I_{t2})) \rightarrow \exists_{\geq 2} I_{e1}(p_{\mathrm{DRIVE}}(I_{e1}) \wedge R_{\mathrm{DURING}}(I_{e1}, I_1))), \tag{49}$$

or, in higher-order logic notation,

$$((Q_{\text{SOME}}\lambda I_1.((\wedge\ (p_{\text{XXXX-WXX-1}}I_1))((R_{\text{DURING}}\,I_1)I_{t2})))\lambda I_2.(Q_{2\text{X}}\lambda I_{e1}.((\wedge\ (p_{\text{DRIVE}}\,I_{e1}))((R_{\text{DURING}}\,I_{e1})I_2)))), \tag{50}$$

with the binary quantifier Q_{SOME} interpreted in the obvious way.

To see how (50) may be generated from (48), let us split the latter into

(51)

```
<EVENT xml:id="e1" target="#token4" pred="DRIVE"/>
<TIMEX3 xml:id="t1" target="#token7" pred="TWICE" freq="2X"/>
<TLINK eventID="#e1" relatedToTime="#t1" relType="DURING"/>
```

and

(52)

```
<TIMEX3 xml:id="t2" target="#token0 #token1" pred="ONE_MONDAY"
 type="SET" value="XXXX-WXX-1" quant="SOME"/>
<TLINK xml:id="t1" relatedToTime="#t2" relType="DURING"/>
```

Now, (51) is identical to (20), and will therefore be given the same semantics. Crucially, the expression (22) thereby assigned to σ_{t1} already quantifies over events; therefore, we should not proceed by analogy with (31), where the TIMEX3-tag is mapped to an expression providing implicit existential quantification. Instead, we propose the additional interpretation rules

```
<TIMEX3  xml:id="t" type="SET"  value="Rv" quant="q"/>
```

$$\rightarrow \sigma_t := (Q_q\lambda I_1.((\wedge\ ((R_{\text{DURING}}\,I_1)I_t))(vI_1))) \tag{53}$$

and

(54)

```
<TLINK
{eventID|timeID}=x
{relatedToEvent| relatedToTime}=y
relType="DURING"/>
```

$\rightarrow \sigma_t := (\sigma_y\lambda I_x.\sigma_x)$ where σ_x := undefined.

Let us work through the example (48) to see how these rules produce the event-structure representation (50). We have already seen how the tags in (51) generate the assignments:

$$\sigma_{t1} := (Q_{2X}\lambda I_3.((\wedge\ ((R_{\text{DURING}}I_{e1})I_{t1}))(p_{\text{DRIVE}}\,I_{e1})))\ \text{where}\ \sigma_{e1} = \text{undefined.} \tag{55}$$

Turning therefore to the tags in (52), rule (53) results in the assignment

$$\sigma_{t2} := (Q_{\text{SOME}}\lambda I_1.((\wedge\ ((R_{\text{DURING}}I_1)I_{t2}))(p_{\text{XXXX-WXX-1}}I_1))), \tag{56}$$

and rule (54) results in the re-assignment

$$\sigma_{t2} := (Q_{\text{SOME}}\lambda I_1.((\wedge\ ((R_{\text{DURING}}I_1)I_{t2}))(p_{\text{XXXX-WXX-1}}I_1)))\lambda I_{t1}.(Q_{2\text{X}}\lambda I_{e1}.((\wedge\ ((R_{\text{DURING}}I_{e1})I_{t1}))(p_{\text{DRIVE}}\,I_{e1})))), \tag{57}$$

and the de-assignment of σ_{t2}. But (57) is identical to (50) up to variable renaming. And since there are no other semantic values to consider, this is the final semantics for (48), as required.

8.4.4 Interpretive rule summary

We summarize the interpretation rules employed above.

Rule 1: <EVENT xml:id="ei" pred="π"/>
$\rightarrow \sigma_{ei} := (p_\pi\, I_{ei})$

Rule 2: <EVENT xml:id="ei" pred="π" polarity="NEG" />
$\rightarrow \sigma_{ei} := (Q_{=0\text{X}}\lambda I_1.((\wedge((R_{\text{DURING}}\,I_1)I_{\text{ei}}))(p_\pi(I_1)))$

Rule 3: <TIMEX3 xml:id=t freq=q />
$\rightarrow \sigma_t := \lambda R \lambda p.((Q_{\mathrm{q}} \lambda I_1.((\wedge\ ((R\, I_1)I_t))(pI_1)))$

Rule 4a: <TIMEX3 xml:id=t type="SET" value=R$_{\mathrm{v}}$ quant=q />
$\rightarrow \sigma_t := \lambda R \lambda p.((Q_{\mathrm{q}} \lambda I_1.((\wedge\ ((R_{\mathrm{DURING}}\, I_1)I_t))(vI_1)))\lambda I_2.(Q_{1\mathrm{x}} \lambda I_3.((\wedge\ ((RI_3)I_2))(pI_3))))_{ei}$
Rule 4b: <TIMEX3 xml:id=t type="SET" value=R$_{\mathrm{v}}$ quant=q />
$\rightarrow (Q_{\mathrm{q}} \lambda I_1.((\wedge\ ((R_{\mathrm{DURING}}\, I_1)I_t))(vI_1)))$

Rule 5: <TLINK
{eventID | timeID}=x
{relatedToEvent | relatedToTime}=y
relType=r />
$\rightarrow \sigma_x := ((\wedge\ \sigma_x)((R_r I_x)I_y))$

Rule 6: <TLINK
{eventID|timeID}=x
{relatedToEvent| relatedToTime}=y
relType=r/>
$\rightarrow \sigma_y := ((\sigma_y R_r)\lambda I_x.\sigma_x)$ where σ_x := undefined.

Rule 7: <TLINK
{eventID | timeID}=x
{relatedToEvent | relatedToTime}=y
relType="DURING">
$\rightarrow \sigma_y := (\sigma_y \lambda I_x.\sigma_x)$ where σ_x := undefined.

There is a general assumption that rules may only apply if the relevant types match.

8.5 An event-based semantics for ISO-TimeML

8.5.1 Introduction

Besides the interval-based semantics of the XML representation of annotation structures, described in 8.4, the *ISO-TimeML* language also has an event-based semantics associated with its abstract syntax. Just like the interval-based semantics provides an interpretation of annotation representations by translating them into the language of Interval Temporal Logic (ITL), the event-based semantics defines a mapping from the annotation structures defined by the abstract syntax into a first-order logic which has a richer ontology than ITL, including events and relations between them, and interval lengths (durations). Given that first-order logic has a well-established compositional formal semantics, this approach defines a compositional semantics for *ISO-TimeML* at the level of annotations. This semantic specification can be instantiated for each representation format, as illustrated for the XML representation format described in Clause 7, by Bunt & Overbeeke (2008). It includes a semantics for tenses, aspect, durations, subordination ('SLINK') structures and aspectual ('ALINK') structures, which are not covered by the interval-based semantics.

Before presenting the formal details of these semantics, we illustrate the way it works with a simple example. Consider the sentence "John started to read at half past ten". The *ISO-TimeML* annotation structure for this sentence would be as follows, where we identify markables by the corresponding text segments in the sentence, and use self-explanatory names for particular elements of the conceptual inventory.

Entity structures:

— event structure for the "start" event: <"started", <process,past,inchoative,individual,positive>>

— event structure for the "read" event: <"read", <process, --, --, individual, positive>>

— instant structure for "half past ten": <"10:30", <CET, <2008, --, -->, 1030>>>

Link structures:

- subordination structure relating the two events:

  <<“started”, <process,past,inchoative,individual,positive>>,

  <“read”, <process, --, --, individual, positive>>, INITIATES>

- temporal anchoring structure relating the start event to the temporal instant:

  <<“started”, <process,past,inchoative,individual,positive>>,

  <“10:30”, <CET, <2008, --, -->, 1030>>>, AT>

The semantics of this annotation structure can be computed by combining the first-order logic representations of the subordination structure and the temporal anchoring structure (see Bunt & Overbeeke, 2008 for how to do this), with the result:

$\exists e_1.\ \mathit{PROCESS}(e_1) \wedge \exists t_1.\ \mathit{calyear}(\mathit{CET},\ t_1)\mathit{=2008} \wedge \mathit{clocktime}(\mathit{CET}(t_1)\mathit{=1030} \wedge \mathit{EV\text{-}TIME}(\mathit{CET}(e_1))\mathit{=}\ t_1 \wedge$
$\exists e_2.\ \mathit{INITIATES}(e_1,e_2)$

8.5.2 Defining an event-based semantics

8.5.2.1 General

The following subclauses define an interpretation function *Infl* as a mapping from annotations to first-order logic with lambda abstraction (and real numbers). In the left hand side of the definition clauses, capital letters are used to designate elements from the conceptual inventory, and small letters to designate annotation structures.

8.5.2.2 Elements from the conceptual inventory

- To every event class C_i, tense T_j, and aspect A_m the interpretation function *Infl* assigns a predicate constant which is indicated by a quote ', e.g. *Infl*(T_j) = T_j'.

- Polarities and set-theoretic types as such are not represented in the first-order logic representations, but are interpreted through the stipulation of different interpretations of event annotations, depending on the values of these elements.

- To every temporal relation, temporal measurement function, duration relation, numerical relation, event subordination relation, and aspectual relation the interpretation function assigns a predicate constant which is indicated by a single quote (').

- To every time zone Z a function constant Z' is assigned; (functions that map the time line onto pairs consisting of a date and a clock time).

- To all calendar years, calendar months, calendar day numbers, and clock times the interpretation function assigns their usual numerical string name (such as “20090229” and “1245”).

- For every real number N, *Infl*(N) will be its usual string name (such as “5”).

- To every temporal unit, the interpretation function assigns an individual constant which is indicated by a quote ', *Infl*(u) = u'.

8.5.2.3 Event annotations

— $\mathit{Infl}(<C, T, A, \mathit{indiv}, \mathit{pos}>) = \lambda P.\ \exists e.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

— $\mathit{Infl}(<C, T, A, \mathit{indiv}, \mathit{neg}>) = \lambda P.\ \neg\exists e.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

— $\mathit{Infl}(<C, T, A, \mathit{set}, \mathit{pos}>) = \lambda P.\ \exists E.\ \forall e \in E.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

— $\mathit{Infl}(<C, T, A, \mathit{set}, \mathit{neg}>) = \lambda P.\ \neg\exists E.\ \forall e \in E.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

— $\mathit{Infl}(<C, T, A, \mathit{set}, N, \mathit{pos}>) = \lambda P.\ \exists_N e.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

— $\mathit{Infl}(<C, T, A, \mathit{indiv}, N, P_N, \mathit{pos}>) = \lambda P.\ \forall I.\ \mathit{Infl}(P_N)(I) \rightarrow \exists_N e.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

— $\mathit{Infl}(<C, T, A, \mathit{set}, N, \mathit{neg}>) = \lambda P.\ \neg\exists_N e.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

— $\mathit{Infl}(<C, T, A, \mathit{set}, N, P_N, \mathit{neg}>) = \lambda P.\ \neg\forall I.\ \mathit{Infl}(P_N)(I) \rightarrow \exists_N e.\ C'(e) \wedge T'(e) \wedge A'(e) \wedge P(e)$

8.5.2.4 Interval annotations

— $\mathit{Infl}(<t_1,t_2>) = \lambda P.\ \exists T.\ \mathit{Infl}(t_1)(\lambda x.\ \mathit{start}(T) = x) \wedge \mathit{Infl}(t_1)(\lambda y.\ \mathit{end}(T) = y) \wedge P(T)$

— $\mathit{Infl}(<<N,U>, T_1, R>) = \lambda P.\ \exists T.\ \exists T'.\ \mathit{end}(T)=\mathit{start}(T') \wedge \mathit{end}(T)=\mathit{start}\ (T_1) \wedge \mathit{length}(T',U')=N' \wedge P(T)$

— $\mathit{Infl}(<<t_1,t_2>,R>) = \lambda P.\ \exists T.\ R'(T, t_1,t_2) \wedge P(T)$

8.5.2.5 Instant annotations

— $\mathit{Infl}(<Z,d,T>) = \lambda P.\ \exists t.\ \mathit{date}(Z'(t)) = \mathit{Infl}(d) \wedge \mathit{clocktime}(Z'(t)) = T' \wedge P(t)$

— $\mathit{Infl}(<<N,U>,t_1>) = \lambda P.\ \exists t.\ \exists T.\ t=\mathit{start}(T) \wedge t_1=\mathit{end}(T) \wedge \mathit{length}(T,U')=N' \wedge P(t)$

8.5.2.6 Time-amount annotations

— $\mathit{Infl}(<N,U>) = \lambda x.\ \mathit{length}(x,U')=N'$

— $\mathit{Infl}(<R,N,U>) = \lambda x.\ R'(\mathit{length}(x,U'),N')$

8.5.2.7 Temporal relation annotations

— $\mathit{Infl}(R) = \lambda R'$

8.5.2.8 Temporal anchoring structures

— $\mathit{Infl}(<e,t,R>) = \lambda x.\ \lambda T.\ R'(\mathit{EV\text{-}TIME}(e),T)$

8.5.2.9 Temporal relation structures

— $\mathit{Infl}(<e_1,e_2,R_t>) = \lambda e.\ \lambda e'.\ R_t'(e, e')$

— $\mathit{Infl}(<t_1,t_2,R>) = \lambda t.\ \lambda t'.\ R'(t, t')$

8.5.2.10 Time measurement structures

- *Infl*(<t,<N,U>) = λT. *length*(T,U')=N'
- *Infl*(<e,<N,U>) = λT. *length*(*EV-TIME*(e),U')=N'

8.5.2.11 Subordination structures

- *Infl*(<e_1,e_2,R_s>) = $\lambda e.\ \lambda e'.\ R_s'(e, e')$

8.5.2.12 Aspectual structures

- *Infl*(<e_1,e_2,A>) = $\lambda e.\ \lambda e'.\ A'(e, e')$

8.5.2.13 Formal semantics of these representations

These first-order logic representations have a formal semantics which makes use of a model

M = <D,F>

where

D is the model structure, which is an octet D = <C_z, T, F_T, U, C_U, Z_T, *date, clocktime*>;

where

C_z is a set of event classes (such as *Process*, *Transition*, *State*);

T is an 8-tuple <Π, $\leq_T$, *start, end, cY, cM, cD, cT*> where Π is an infinite set of time points with a total ordering $\leq_T$; *start* and *end* are functions from temporal intervals to the time points defining their beginning and end; *cY, cM,* and *cD* are sets of intervals in T (calendar years, calendar months, calendar days, and clock times respectively); *cT* is a structure *cT* = < *clH, clM,..* > (of clock hours and minutes, extended, if necessary, with seconds, milliseconds, etc.);

U is a finite set of units for temporal measurement;

F_T is a set of temporal functions;

C_U is a pair consisting of a function from pairs of temporal units to real numbers, i.e. a conversion function between temporal units, and a function *length* that computes the length of a temporal interval, given a temporal unit (i.e. a function from pairs consisting of a temporal interval and a temporal unit to real numbers);

length is a function assigning a numerical value to a pair, consisting of a temporal interval and a temporal unit;

Z_T (time zones) is a set of functions from T to pairs of dates and times; event occurs at some place, the notion of time zones can also be applied to events); day numbers, and clock times, respectively;

date and *clocktime* are functions projecting each temporal value, defined by a time zone, onto its date and its clock time, respectively;

F is the interpretation function, which assigns elements of the model structure to the predicates in the first-order representations of annotation structures. The interpretation rules for first-order logic expressions are as usual.

Annex A
(normative)

Core annotation guidelines

A.1 Introduction

This annex specifies the annotations to be used for marking up text according to the *ISO-TimeML* language. These guidelines are focused on language-independent consideration, although inspired in the first place by the ways in which temporal information is expressed in English. This annex is organized as follows. The first clause explains what the *ISO-TimeML* tags (XML elements) are and how to annotate them. It also specifies, for each tag, what its attributes are. While this exposition contains many examples illustrating what and how to tag, the examples focus, for clarity's sake, on the tag under discussion at any given point. The following Annex B provides a set of fully annotated examples, illustrating all of the interactions between the various entity and relational tags.

A.2 ISO-TimeML elements and their attributes

A.2.1 The <EVENT> element

A.2.1.1 Overview

The term "event" is a cover term for expressions denoting:

- situations that happen or occur, which can both be punctual (1) or last for a period of time (2).

(1) a. Ferdinand Magellan, a Portuguese explorer, first **reached** the islands in search of spices.

b. A fresh flow of lava, gas and debris **erupted** there Saturday.

(2) a. 11,024 people, including local Aeta aborigines, **were evacuated** to 18 disaster relief centres.

b. "We're **expecting** a major eruption" he said in a telephone interview early today.

- states or circumstances in which something obtains or holds true (3).

(3) Israel has been scrambling to buy more masks abroad, after a *shortage* of several hundred thousand gas masks.

A.2.1.2 How to annotate EVENTs

A.2.1.2.1 Overview

The types of expressions denoting events vary cross-linguistically. Chinese, for instance, does not use tensed forms of verbs, whereas this is the most common way of conveying events in Germanic and Romance languages. In general, however, events can be expressed by means of (at least some of) the following phrase types: VPs headed by either tensed or untensed verbs (4-5), NPs (6), APs (7), or PPs (8).

Note that in the above sentences not all markable elements are tagged –only those that are relevant for the discussion. In example (1)b, for instance, "flow" was not marked. The following examples illustrate the variety of ways in which events can be expressed in English by words of different syntactic categories.

(4) A fresh flow of lava, gas and debris ***erupted*** there Saturday.

(5) Prime Minister Benjamin Netanyahu called the prime minister of the Netherlands *to* ***thank*** him for thousands of gas masks his country has already contributed.

(6) Israel will ask the United States to delay a military ***strike*** against Iraq until the Jewish state is fully prepared for a possible Iraqi ***attack***.

(7) A Philippine volcano, ***dormant*** for six centuries, began exploding with searing gases, thick ash and deadly debris.

(8) All 75 people ***on board*** the Aeroflot Airbus died.

A.2.1.2.2 Event identification

Event identification is based on the notion of minimal chunk, because higher constituents (i.e. phrases) may contain more than one event-denoting expression. For example: VPs headed by an aspectual verb (9), light verb constructions (10), causative constructions (11) or other VPs whose complement expresses an additional event. In *ISO-TimeML*, both event expressions will be annotated with independent tags because both verbal and nominal heads are relevant to different kinds of event information.

NOTE 1 The two tagged events will be related by means of a temporal or aspectual link (TLINK and ALINK, respectively). See A.3.

In the examples below, the phrase and minimal chunk constituency levels are marked in square and regular brackets, respectively.

(9) They [probably (would have began) (responding) to President Reagan's 600 ships plan with new construction].

NOTE 2 The VP contains two additional event expressions not signalled here: “plan” and “construction”.

(10) They [(will definitely take) it into (consideration)].

(11) a. The rains [(caused) (the flooding)].

b. John [(caused) (the fire)].

NOTE 3 The subject “the rains” denotes here an additional event.

If the event is denoted by means of a predicative construction, only the predicative element (the adjective, nominal, or prepositional complement) will be marked, disregarding the copular element (expressed by copulas equivalent to English form “be”).

(12) a. There is no reason why we [would not be (fully prepared)].

b. If, in spite of everything, we [will not be (ready)], we will ask the United States to delay the operation.

c. James Pustejovsky [was (CTO of LingoMotors)] for several years.

A.2.1.2.3 Event tag span

The EVENT tag will span over minimal chunks, whether contiguous or not. Instead of marking just a head, as in (13)b. As is illustrated in Annex B, non-contiguous chunks such as “**has** he **arrived**?” or “Mary **looked** me **up**” are all marked in *ISO-TimeML*. The basic principle of *ISO-TimeML* is to make the element <EVENT> match a minimal chunk, excluding complements and modifiers. In the examples below, the event-denoting chunk is in bold face

(13) a. A fresh flow of lava, gas and debris **erupted** there Saturday.

b. Israel **has been scrambling** to **buy** more masks abroad.

c. The private sector **could establish** a private agency.

d. Kaufman **did not disclose** details of the deal.

e. Additional distribution centres **would be set up** next week.

(14) a. **A fresh flow of lava, gas and debris** erupted there Saturday.

b. Israel will ask the United States to delay **a military strike** against Iraq until the Jewish state is fully prepared for **a possible Iraqi attack**.

(15) a. A Philippine volcano, **dormant** for six centuries, began exploding with searing gases, thick ash and deadly debris.

b. There is no reason why we **would not be fully prepared**.

If the event is expressed by a prepositional chunk, we will annotate the noun head of the embedded NP in case it expresses an event (16)a. Otherwise, we will annotate the whole PP (16)b.

(16) a. Prof. Abramovitz, on **sabbatical** in Heidelberg University, declared the discovery is of remarkable relevance.

b. All 75 people **on board** the Aeroflot Airbus died

A.2.1.2.4 What *NOT* to tag:

Do not tag events in the following two situations:

- when they express states that are not temporally relevant; that is, states that
  - are not directly related to a temporal expression, or
  - are not identifiably changed over the course of the text being marked up. See A.2.1.3 on events belonging to the class STATE;
- when the event reading of a logically polysemous nominalization is not exploited in the predication.

  For example, in (17), “reports” is not tagged as an event because the sense of the nominalization relevant here is the “information” or “content” interpretation of the noun.

(17) Newspaper reports have said Amir was infatuated with Har-Shefi.

A.2.1.3 Specification of the <EVENT> element

A.2.1.3.1 A Summary

The <EVENT> tag is used to annotate those elements in a text that describe what is conventionally referred to as an ***eventuality***. The specification is reiterated in Table A.1.

Table A.1 — Specification of <EVENT> reiterated

target	specifies the destination of the reference by supplying one or more URI References.
pred	denotes the content related to that event through the indication of a lexical predicate.
class	characterizes each event as occurrence, perception, reporting, aspectual, state, I-state or I-action.
type	refines the category of the corresponding element.
tense	captures standard distinctions in the grammatical category of verbal tense.
aspect	captures standard distinctions in the grammatical category of verbal aspects such as perfective, imperfective, progressive and so on.
pos	(part of speech) captures distinctions among the grammatical categories of phrases which are marked as events, as not all such phrases contain finite verbs.
polarity	Boolean attribute that conveys the polarity of the event in question.
mood	captures the mood of the event.
modality	conveys the modality nature of the event: different degrees of epistemic modality, deontic modality, etc.

Some of the attributes are described below.

A.2.1.3.2 The attribute @class

A.2.1.3.2.1 The attribute @class is required to be specified with one of the following values: “REPORTING”, “PERCEPTION”, “ASPECTUAL”, “I_ACTION”, “I_STATE”, “OCCURRENCE” and “STATE”. Some of these values are described below.

A.2.1.3.2.2 REPORTING: Reporting events describe the action of a person or an organization declaring something, narrating an event, informing about an event, etc. Some examples in English: “say”, “report”, “tell”, “explain”, “state”:

(18) a. Punongbayan **said** that the 4,795-foot-high volcano was spewing gases up to 1,800 degrees.

b. No injuries were **reported** over the weekend.

c. **Citing** an example, ...

A.2.1.3.2.3 PERCEPTION: Events involving the physical perception of another event. In English, such events are typically expressed by verbs such as: “see”, “watch”, “glimpse”, “behold”, “view”, “hear”, “listen”, “overhear”.

(19) a. Witnesses tell Birmingham police they **saw** a man running.

b. “You can hear the thousands of small explosions down there”, a witness **said.**

A.2.1.3.2.4 ASPECTUAL: In languages such as English and French, there is a grammatical device of aspectual predication, which focuses on different facets of event history:

a. Initiation: “begin”, “start”, “commence”, “set out”, “originate”, “initiate”.

b. Reinitiation: “restart”, “reinitiate”, “reignite” (metaphorically).

c. Termination: “stop”, “terminate”, “cease”, “discontinue”, “interrupt”, “quit”.

d. Culmination: “finish”, “complete”.

e. Continuation: “continue”, “keep”, “go on”.

Below are a couple of examples:

(20) a. The volcano began showing signs of activity in April for the first time in 600 years,...

b. All non-essential personnel should begin evacuating the sprawling base.

A.2.1.3.2.5 I_ACTION: I_ACTION stands for intensional action. I_ACTIONs describe an action or situation which introduces another event as its argument, which must be in the text explicitly. Explicit performative predicates (like those in (e)-(i), below) are also included here.

NOTE 1 The I_ACTION class does not cover states (but see the description of I_STATES below).

NOTE 2 Note the distinction between “intensional” and “intentional” (purposeful). This class includes but is broader than actions with intended consequences.

The following list of English predicates is representative (not exhaustive) of the types of events included in this class. In the examples, I_ACTIONs are in bold face and their event arguments, underlined.

a) **attempt, try, scramble**:

(21) Companies such as Microsoft are **trying** to monopolize Internet access.

b) **investigate, investigation, look at, delve**:

(22) A new Essex County task force began **delving** Thursday into the slayings of 14 black women.

c) **delay, postpone, defer, hinder, set back**:

(23) Israel will ask the United States to **delay** a military strike against Iraq.

d) **avoid, prevent, cancel**:

(24) Palestinian police **prevented** a planned pro-Iraq rally by the Palestinian Professionals' Union.

e) **ask, order, persuade, request, beg, command, urge, authorize**:

(25) Iraqi military authorities **ordered** all Americans and Britons in Kuwait to assemble at a hotel.

f) **promise, offer, assure, propose, agree, decide**:

(26) Germany has **agreed** to lend Israel 180,000 protective kits against chemical and biological weapons, and Switzerland **offered** to lend Israel another 25,000 masks.

g) **swear, vow**.

h) **name, nominate, appoint, declare, proclaim**.

i) **claim, allege, suggest**.

A.2.1.3.2.6 I_STATE: They are similar to the events in the previous class. I_STATEs also select another event as their argument, but contrary to I_ACTIONs, they denote stative situations. As above, the I_STATE is in bold face, whereas the embedded argument is underlined. The following list is not exhaustive, but only representative.

a) **believe, think, suspect, imagine, doubt, feel, be conceivable, be sure**:

(27) We believe that his words cannot distract the world from the facts of Iraqi aggression.

b) **want, love, like, desire, crave, lust**:

NOTE The verb “love” as in “John loves Paul's cousin” is not considered an I_STATE. Similarly for the verb “like”. An I_STATE must govern another event.

(28) We aim at triggering associations that will generate **lust** for change.

c) **hope, expect, aspire, plan**:

(29) We **aim** at triggering associations that will generate lust for change.

d) **fear, hate, dread, worry, be afraid**:

(30) The agencies **fear** they will be unable to crack those codes to eavesdrop on spies and crooks.

e) **need, require, demand**

f) **be ready, be eager, be prepared**

(31) The young industry's rapid growth also is attracting regulators **eager** to <u>police</u> its many facets.

g) **be able, be unable**

(32) The agencies fear they will be **unable** to <u>crack</u> those codes to eavesdrop on spies and crooks.

A.2.1.3.2.7 OCCURRENCE: This class includes all the many other kinds of events describing situations that happen or occur in the world.

(33) a. The Defense Ministry said 16 planes have **landed** so far with protective equipment against biological and chemical warfare.
b. Two moderate **eruptions** shortly before 3 p.m. Sunday appeared to signal a larger **explosion**.
c. RMS said it had a **loss** of $158,666, or 10 cents a share, in the third quarter, compared with a year-earlier **loss** of $29,956, or two cents a share.
d. Ralston said its restructuring **costs** include the **phase-out** of a battery facility in Greenville, N.C.

A.2.1.3.2.8 The value "STATE" is also treated as a value of the attribute @type as well as the value of the attribute @class, as described in detail in the following subclause. The redundancy of this value is ignored for the time being.

A.2.1.3.3 The attribute @type

The attribute @type is a required attribute, providing a finer-grained classification of events into STATE, PROCESS and TRANSITION. States describe circumstances in which something obtains or holds true. However, we will only annotate temporally relevant states; that is:

a) States that are identifiably changed over the course of the document being marked up. In these and the following examples the markable state is in bold face.

(34) a. All 75 people **on board** the Aeroflot Airbus died.
b. Israel has been scrambling to buy more masks abroad, after a **shortage** of several hundred thousand gas masks was discovered.
c. No **injuries** were reported over the weekend.

b) States that are directly related to a temporal expression. This criterion includes all states that are linked to a TIMEX3 markable by means of a TLINK (see A.2.2 and A.3.1). Two examples are given here, where the state is in bold face and the temporal expression associated with it is underlined.

(35) a. James Pustejovsky was **CTO** for <u>several years</u>.
b. They **lived** in U.N.-run refugee camps for <u>2 1/2 years</u>.

c) States that are introduced by an I_ACTION, an I_STATE, or a REPORTING event. States are in bold face, the introducing event underlined.

(36) a. He <u>mediated</u> the **crisis**.
b. Saddam Hussein <u>sought</u> **peace** on another front.
c. Har-Shefi <u>told</u> police that Rabin was a **traitor**.

d) Predicative states the validity of which is dependent on the document creation time. In spite of not being explicitly related to any TIMEX3 expression, the states underlined in the examples in bold face below will be tagged because their validity is relative to the point in time they have been asserted (the DCT).

(37) a. A total of about 3,000 Americans, 3,000 Britons and more than 450 Japanese are **in Iraq**.
b. Overall, more than 2 million foreigners are **in both countries**.

This criterion also includes quantitative statements such as those from financial journals:

(38) Gas prices fell from a **twenty-two dollar barrel level** down to the **fourteen dollars** we're seeing today.

NOTE The current class, STATE, does not contain states that have been tagged as I_STATEs.

A.2.1.3.4 The attribute @pos

The attribute @pos is an optional attribute. It captures distinctions among the grammatical categories of elements which are marked as events. This attribute can have the following values: ADJECTIVE, NOUN, VERB, PREPOSITION and OTHER. Note that these values vary from language to language. Sentences (4, 6, 7, 8), repeated below, illustrate the use of attribute values VERB (39)a, NOUN (39) b), ADJECTIVE (39)c and PREPOSITION (39)d.

(39) a. A fresh flow of lava, gas and debris **erupted** there Saturday.
b. Israel will ask the United States to delay a military **strike** against Iraq until the Jewish state is fully prepared for a possible Iraqi **attack**.
c. A Philippine volcano, **dormant** for six centuries, began exploding with searing gases, thick ash and deadly debris.
d. All 75 people **on board** the Aeroflot Airbus died.

A.2.1.3.5 The attribute @tense

The attribute @tense is a required attribute, capturing standard distinctions in the grammatical category of verbal tense. For English, it can have the values: PRESENT, PAST, FUTURE, IMPERFECT or NONE. For languages in which tense distinctions do not apply, the value NONE will be used as default. This value can, however, be overwritten if there is a non-verbal element (e.g. adverbs of time) in the sentence conveying a value equivalent to tense for a particular event.

Among languages with tense distinctions, tenses are not easily mapped. The annotators should develop a specific *ISO-TimeML* spec for annotating tense (and also aspect) in the particular language they annotate. Basic guides for deciding among the different values are:

— PRESENT: It is the tense generally used to express action at the present time and states, but also, possibly: habitual events, occurrences in the near future, or actions that started in the past and still hold in the present;

— PAST: Expressing action and states of being in a past time;

— FUTURE: Used for describing events as not having happened yet, but expected to in the future;

— IMPERFECT: Assigned to finite forms, the imperfect is a descriptive past tense which indicates an ongoing state of being or a repeated or incomplete action. The beginning and end of the state of being or action are not indicated, and the ***imparfait*** is very often translated in English as “was” or “was __-ing”;

— NONE: No tense value is found relevant for non-verbal event-denoting expressions: nouns, adjectives, and prepositions.

A.2.1.3.6 The attribute @aspect

The attribute @aspect is a required attribute, similarly to @tense, it captures standard distinctions in the grammatical category of verbal aspect. It can have values PROGRESSIVE, IMPERFECTIVE, PERFECTIVE, IMPERFECTIVE_PROGRESSIVE, PERFECTIVE_PROGRESSIVE or NONE.

As with @tense, languages without aspect distinctions in the verbal system will assign the value NONE as default. This value can, however, be overwritten if there is a non-verbal element (e.g. adverbs of time, phrases, etc.) in the sentence conveying a value equivalent to aspect for a particular event.

For languages in which aspect distinctions apply, the basic guides for deciding among different aspect values are:

— PROGRESSIVE: Expressing, among other possibilities: actions in progress and outgoing activities; durative activities and continuous states; activities posing the background for other activities; simultaneous activities; etc. (e.g. “Prof. Abramovitz was teaching that day/could be teaching on Friday.”);

— PERFECTIVE: Generally expressing states and activities which were ended (e.g. “Prof. Abramovitz has conducted experiments in different countries around the world.”);

— IMPERFECTIVE: Generally expressing states and activities that are seen from a particular viewpoint as ongoing, habitual, repeated, or generally containing internal structure. This is distinct from the progressive. English does not have a proper imperfective aspect;

— PERFECTIVE_PROGRESSIVE: Combining the meanings of progressive and perfective (e.g. “Prof. Abramovitz has been teaching for his whole life.”);

— IMPERFECTIVE_PROGRESSIVE: Combining the meanings of progressive and imperfective in languages that have imperfective aspect;

— NONE: No aspect value is found. Relevant for non-verbal event-denoting expressions (nouns, adjectives, and prepositions).

A.2.1.3.7 The attribute @polarity

The attribute @polarity is a required attribute, Boolean attribute that conveys the polarity of the event in question. If it is set to NEG, the event instance is negated. If it is set to POS, the event instance is not negated.

A.2.1.3.8 The attribute @mood

The attribute @mood is an optional attribute, capturing the mood associated with the event. Presently, it can have either SUBJUNCTIVE or NONE. If no inflectional morphology is present to indicate mood, then the default value is NONE.

A.2.1.3.9 The attribute @modality

The attribute @modality is an optional attribute, conveying the modality nature associated with the event: different degrees of epistemic modality, deontic modality, etc. The particular values for this attribute will be language specific.

A.2.1.3.10 The attribute @vform

The attribute @vform is an optional attribute, capturing standard distinctions in the grammatical category of non-tensed verbal forms . It can have values INFINITIVE, PRESPART, PASTPART, GERUNDIVE or NONE.

A.2.2 The <TIMEX3> element

A.2.2.1 Overview

The <TIMEX3> tag annotates any temporal expression (a.k.a. timex) referring to:

— day times (“noon”, “3p.m.”, “the evening”, ...);

— dates of different granularity: days (“yesterday”, “Jan 8 2001”, “last Friday”, etc.), weeks (“next week”, “the second week of July”, etc.), months (“in two months”, “August 1971"), seasons or business quarters (“last spring”, “the third quarter”, etc.), years (“1978”, “the previous year”), centuries, etc.;

— durations (“two months”, “five hours”, “0.2 seconds”);

— sets of temporal entities (“every Thursday”, “the first Sunday of the month”).

Previous to the *ISO-TimeML* initiative, the Sheffield Temporal Annotation Guidelines introduced the tag TIMEX for annotating temporal expressions in the context of newswire articles, and the TIDES project, introduced the TIMEX2 tag. The specifics of the *ISO-TimeML* tagset differ in detail from both of these; because of that, the tag name TIMEX3 is adopted here.

A.2.2.2 How to annotate <TIMEX3>

A.2.2.2.1 Timex tag span

The TIMEX3 span will be based on the constituent structure of each particular language, but it will also make use of the classification of temporal units shown in Table A.2, and the type of relations holding among two different temporal expressions (40).

Table A.2 — Classification of temporal intervals

t < day	*t = day*	*t < year*	*t = year*	*t > year*
twelve o'clock midnight morning ten minutes to two	Wednesday tomorrow Jan. 2 the 8th Christmas Day Worker's Day	first week month semester January season Fall	1984 next year	last century

(40)

a. **Specification relation**: Involving two temporal expressions, one of which is helping to further specify the other (e.g. “[twelve o'clock] [midnight]”, “[four] in [the afternoon]”, “[Tuesday] [Jan. the 18th]”, “[this year's] [summer]”, “[some Thursdays] in [1984]”).

b. **Anchoring relation**: Involving two temporal expressions, one of which is ordered, or anchored, relative to the other. In English, they generally involve the use of temporal prepositions and conjunctions such as “from”, “before”, “after”, “following”, “prior to”, etc. For instance, “[two weeks] from [next Tuesday]”, “[2 days] before [yesterday]”, “[ten minutes] to [four]”, “[three years ago] [today]”. These expressions are also known as anchored durations.

c. **Conjunction relation**: Involving two temporal expressions related by a coordination conjunction – mainly, “and” and “or” (e.g. “[six months] or [a year]”).

A.2.2.2.2 General rules

The TIMEX3 span need be compliant with the following general rules.

a) **The full extent of the tag** shall correspond to one of the following categories:

— noun phrase (“the afternoon”, “last summer”, “yesterday”, “Sunday”);

— adjective phrase (“half an hour long”, “half-hour” as in “a half-hour trip”);

— adverbial phrase (“fairly recently”).

Therefore, any preposition preceding a temporal expression (as in “in the afternoon”, “before yesterday”, “in half an hour”,...) will not be included as part of the tag:

Temporally relevant prepositions will be annotated as signals. See A.2.3.

(41) in the afternoon

```
<TIMEX3 xml:id=“t1" target=“#token1 #token2"/>
```

On the other hand, adverbial postmodifiers (“ago”, “ever”) will be considered part of the TIMEX3 markable expression (42-43), but not postmodifiers that express an event (44-46).

(42) the best second quarter *ever*

<TIMEX3 xml:id="t1" **target**="#range(token0,token3)"/>

(43) three years *ago*

<TIMEX3 xml:id="t1" **target**="#range(token0,token2)"/>

(44) five days after he came back

<TIMEX3 xml:id="t1" **target**="#token0 #token1"/>
<EVENT xml:id="e1" **target**="#token3 #token4" **pred**="COME" **class**="OCCURRENCE"/>

(45) nearly four decades of experience

<TIMEX3 xml:id="t1" **target**="#range(token0,token2)"/>
<EVENT xml:id="e2" **target**="#token4" **class**="STATE"/>

(46) months of renewed hostility

<TIMEX3 xml:id="t1" **target**="#token0"/>
<EVENT xml:id="e2" **target**="#token3" **class**="STATE"/>

b) **Two temporal expressions in a specification relation will be marked up with a single tag if**:

- the two expressions belong to the same class in Table A.2 (e.g. “12 o'clock midnight”, “Tuesday”, “Jan 18th”, “eleven in the morning”);

- the two expressions belong to the same syntactic constituent. Syntactic constituency can be checked using the fronting or clefting tests (examples 47-48 below). Different constituents allow fronting and clefting, but not two parts of the same constituent. For example, the two temporal expressions in (47) are part of the same constituent, while those in (48) are not.

(47) The different groups will meet ***at 11 a.m. Jan. 3, 2005***.

(48) **On Jan. 3, 2005**, the different groups will meet **at 11 a.m**.

c) **Temporal expressions in an anchoring relation will be generally marked up with two independent tags**.

For instance: “[two weeks] from [next Tuesday]”, “[2 days] before [yesterday]”, “[three years ago] [today]” (49-51).

Links will be used to express the relative ordering of the two temporal expressions – see A.3. In addition, these types of expression are considered anchored durations and can be annotated as such in the newest version of *ISO-TimeML* – see A.2.2.3. The only exception will be those temporal expressions denoting day time, such as “[ten minutes] to [four]", which will be annotated with one single tag (52).

(49) I'm leaving on vacation ***two weeks from next Tuesday***.

<TIMEX3 xml:id="t1" **target**="#token4 #token5"/>
<TIMEX3 xml:id="t2" **target**="#token7 #token8"/>

(50) John left ***2 days before yesterday***.

<TIMEX3 xml:id="t1" **target**="#token2 #token3"/>
<TIMEX3 xml:id="t2" **target**="#token5"/>

(51) A major earthquake struck Los Angeles ***three years ago today***.

<TIMEX3 xml:id="t1" **target**="#token6 #token7"/>
<TIMEX3 xml:id="t2" **target**="#token9"/>

(52) I'm leaving **at *ten minutes to four*.**

<TIMEX3 xml:id="t1" **target**="#range(token3,token6)"/>

d) **Two temporal expressions in a conjunction relation will be marked up as two different tags**.

(53) Saddam might play the whole game again ***six months or a year from now***.

<TIMEX3 xml:id="t1" **target**="#token7 #token8"/>
<TIMEX3 xml:id="t2" **target**="#token10 #token11"/>
<TIMEX3 xml:id="t3" **target**="#token13"/>

In the example above, the expression (from) now is in an anchoring relation –with both six months and a year. Hence, it needs to be marked up with an independent tag.

A.2.2.3 Attributes for TIMEX3

A.2.2.3.1 The attribute @xml:id (<TIMEX3> ID number)

Required attribute. Each <TIMEX3> expression has to be identified by a unique ID number.

A.2.2.3.2 The attribute @target

Required attribute. Each markable shall be anchored to a segment (contiguous or non-contiguous) or a token at some preprocessed level of primary data.

A.2.2.3.3 The attribute @type

A.2.2.3.3.1 Required attribute. Each <TIMEX3> is assigned one of the following types: DATE, TIME, DURATION or SET.

A.2.2.3.3.2 DATE: The expression describes a calendar time.

(54) Mr. Smith left **Friday, October 1, 1999**.
the second of December
yesterday
in **October of 1963**
in **the summer of 1964**
on **Tuesday 18th**
in **November 1943**
last week

DATE can also be the value for the @type attribute of each of the two <TIMEX3> markable expressions constituting a range, as long as they describe a calendar time.

(55) a. John left between **Monday** and **Wednesday**.
b. <TIMEX3 id="t1" target="#token3" type="DATE"/>
<TIMEX3 id="t2" target="#token5" type="DATE"/>

A.2.2.3.3.3 TIME: The expression refers to a time of the day, even if in a very indefinite way (as in the two examples below):

(56) Mr. Smith left at **ten minutes to three**.
at **five to eight**
at **twenty after twelve**
at **half past noon**
at **eleven in the morning**
at **9 a.m. Friday, October 1, 1999**
the morning of January 31
late last night
last night

As before, TIME can also be the @type value for each of two <TIMEX3> markable expressions that together refer to a temporal range (e.g. “Mr. Smith left between **8:00 a.m.** and **10:00 a.m.**”).

NOTE The last example also, i.e. (56), depends upon the context to receive the value TIME (instead of DURATION or AMBIGUOUS).

A.2.2.3.3.4 DURATION: The expression describes a contiguous duration. This value is assigned only to explicit durations such as the following:

(57) Mr. Smith stayed **3 hours** last Monday in Boston.
48 hours
three weeks
all last night
20 days in July
3 hours last Monday.

As a rule, if any specific calendar information is supplied in the temporal expression, then the @type of the element <TIMEX3> shall be either DATE or TIME. A phrase such as “1985” might be marked as a DURATION if the context suggests that an event holds throughout that year. However, temporal expression such as the one described here shall be of type DATE, since they refer to a particular area in the temporal axis — even though that area spans over a period of time. By contrast, durations are periods of time not pointing at any specific area in the temporal axis.

A.2.2.3.3.5 SET: The expression describes a set of times. This value is assigned to expressions such as those in section 3.5 of TIDES(2002). For example:

(58) John swims **twice a week.**
every 2 days

A.2.2.3.4 The attribute @value

A.2.2.3.4.1 The attribute @value (equivalent to VAL in TIMEX2) will be annotated exactly as specified in TIDES(2002) sections 3.2.and 3.3. Note, however that, these sections also include the use of two additional attributes, ANCHOR_VAL and ANCHOR_DIR, which are not used in *ISO-TimeML*. The format of this attribute @value is determined by the @type attribute. For instance, a DURATION shall have a value that begins with the letter “P” (standing for period of time) and a TIME with the letter “T” (standing for a time) that includes times of the day. The following examples, from previous clauses, partially illustrate the use of the value attribute for times of the day, dates, durations, and sets:

A.2.2.3.4.2 Times of the day, and dates:

(59) a. 4 p.m.
b. <TIMEX3 xml:id="t1" target="#range(token0,token4)" type="TIME" value="T16:00"/>

The annotator will introduce as much information as is available both in the time expression and from the context. In some cases, the text includes some reference to the specific date in which the time is anchored. For instance, given the sentence “Last Friday's meeting didn't start until 4 p.m.”, and assuming that the document creation time is Friday, July 12, 2002, then the @value attribute shall specify the full date that can be computed from the document creation time, and be:

(60) a. 4 p.m.
b. <TIMEX3 xml:id="t1" target="range(#token0,#token4)" type="TIME" value="2002-07-05T16:00"/>

Unknown information is left underspecified by means of the placeholder “X”. In the next example, for instance, the year is unknown.

(61) a. the second of December
b. <TIMEX3 xml:id="t5" target="#range(token0,token3)" type="DATE" value="XXXX-12-02"/>

A.2.2.3.4.3 Durations:

(62) a. 4 months
b. <TIMEX3 xml:id="t1" target="#range(token0,#token1)" type="DURATION" value="P4M"/>

(63) a. during two entire days in the summer of 1999
b. <TIMEX3 xml:id="t1" target="#range(token2,token4)" type="DURATION" value="P2D"/>
<TIMEX3 xml:id="t2" target="#range(token6,token8)" "type="DATE" value="1999-SU">

A.2.2.3.4.4 Sets:

To fully annotate sets, the TIMEX3 shall also include either the @quant or @freq attributes, if not both. The following examples begin the annotation of a TIMEX3 set as pertains to the @value attribute:

(64) a. twice a week
b. <TIMEX3 xml:id="t1" target="#range(token0,token2)" type="SET" value="P1W"/>

(65) a. every 2 days
b. <TIMEX3 xml:id="t1" target="#range(token0,token2)" type="SET" value="P2D"/>

(66) a. 3 days each week
b. <TIMEX3 xml:id="t1" target="#range(token0,token3)" type="SET" value="P1W">

(67) a. every October
b. <TIMEX3 xml:id="t1" target="#range(token0,token1)" type="SET" value="XXXX-10">

This is so we can capture the calendar information that is present in the temporal expression. Some annotators may find it confusing as to when a DATE-like annotation is used and when a DURATION-like format is preferred. In general, if there is no specified calendar date (for example, “October” or “Tuesday”), then the value for the SET will be like that of a DUARTION.

A.2.2.3.5 The attribute @mod

This is an optional attribute, inherited directly from the TIMEX2 MOD attribute. Its value is as specified in TIDES(2002), section 3.4.

A.2.2.3.6 The attribute @temporalFunction

Binary attribute which expresses whether the value of the temporal expression needs to be determined via evaluation of a temporal function. Temporal functions will be applied as a postprocess. The value for this attribute will be positive for those cases that do not contain all the information necessary to fill the higher-order (left-hand) positions in the value attribute (68). This will apply even if value can be completely filled, given additional information provided by the context.

(68) a. **eleven in the morning**: missing the particular day.
b. **January, 31**: missing the year.
c. **last week**: missing the month and year.

On the other hand, for cases in which the higher-order position of @value are filled from the information provided by the tagged temporal expression, @temporalFunction should be assigned a negative value. Such cases include:

(69) a. **twelve o'clock January 3, 1984**
b. **summer of 1964**
c. **Friday, October 1, 1999**
d. **the morning of January 31, 1999**

Durations whose length is underspecified will receive true as the value of @temporalFunction. Compare (70) with (71), which indicates a specific length.

(70) a. in **recent months**
b.<TIMEX3 xml:id="t1" target="#range(token1,token2)" type="DURATION" value="PXM" temporalFunction="true"/>

(71) a. for **two months**
b.<TIMEX3 xml:id="t1" target="#range(token1,token2)" type="DURATION" value="P2M" temporalFunction="false"/>

A.2.2.3.7 The attribute @valueFromFunction

This attribute is not relevant for the purposes of manual annotation, but only for postprocessing. The human annotator should ignore it.

A.2.2.3.8 The attribute @functionInDocument

This attribute is optional. It indicates the function of a <TIMEX3> in providing a temporal anchor for other temporal expressions in the document. There are several times that mark the major milestones in the life of a textual document:

- the time the text is created;
- the time the text is modified;
- the time the text is published;
- the time it may be released (if not immediately);
- the time it is received by a reader;
- the time that the text expires (if any).

The possible values for this attribute are then:

“CREATION_TIME”, “MODIFICATION_TIME”, “PUBLICATION_TIME”, “RELEASE_TIME”, “RECEPTION_TIME”, “EXPIRATION_TIME”, “NONE”.

If this attribute is not explicitly supplied, the default value is “NONE”.

The following attributes are used to strengthen the annotation of durations and sets in *ISO-TimeML*. Each is optional and can be used at the annotator's discretion. It is important to note, however, that @beginPoint and @endPoint should only be used when the type of the expression is DURATION, and that @quant and @freq should only be used when the expression is a set.

A.2.2.3.9 The attributes @beginPoint and @endPoint

This is used when a duration is anchored by one or two time expressions indicating its beginning and/or end points. If only one of these points is provided, the annotator can create an empty TIMEX3 to represent the missing point.

NOTE The values stored in these attributes can be used by temporal functions to compute the missing points and create a tag for them.

In some ways, the @beginPoint and @endPoint attributes are similar to @anchorTimeID. In (72), for instance, the new @xml:id introduced by the empty <TIMEX3> can be used to link the teaching event directly to the time at which it takes place.

(72) a. John begins teaching one week from September 15.
b. <TIMEX3 xml:id="t1" target="#range(token4,token5)" type="DURATION" value="P1W" beginPoint="t2" endPoint="t3"/>
<TIMEX3 xml:id="t2" target="#range(token7,token8)" type="DATE" value="XXXX-9-15"/>
<TIMEX3 xml:id="t3" target=nil type="DATE" value="XXXX-9-22" temporalFunction="TRUE" anchorTimeID="t1" />

A.2.2.3.10 The attributes @quant and @freq

This is used when a temporal expression is of the type SET. The attribute @quant is generally a literal from the text that quantifies over the expression. The attribute @freq contains an integer value and a time granularity that represent the frequency at which the temporal expression regularly reoccurs. These attributes are only used if their values are supplied by the temporal expression (or by a temporal anchor). Though it seems on occasion that values for these attributes can be inferred, they will not be for purposes of manual annotation. Although, if there is no specified @quant, one imagines that the set is universally quantified. The following examples complete the annotations of the sets listed earlier in this clause:

(73) a. twice a week
b. <TIMEX3 xml:id="t1" target="#range(token0,token2)" type="SET" value="P1W" freq="2X"/>

(74) a. every 2 days
b. <TIMEX3 xml:id="t1" target="#range(token0,token2)" type="SET" value="P2D" quant="EVERY"/>

(75) a. 3 days each week
b. <TIMEX3 xml:id="t1" target="#range(token0,token3)" type="SET" value="P1W" quant="EACH" freq="3d"/>

(76) a. every October
b. <TIMEX3 tid="t1" target="#range(token0,token1)" type="SET" value="XXXX-10" quant="EVERY"/>

A.2.3 The <SIGNAL> element

A.2.3.1 Overview

A signal is a textual element that makes explicit the relation holding between two entities (timex and event, timex and timex, or event and event). Signals are generally:

- temporal prepositions: “on”, “in”, “at”, “from”, “to”, “before”, “after”, “during”, etc.;
- temporal conjunctions: “before”, “after”, “while”, “when”, etc.;
- special characters: “-” and “/”, in temporal expressions denoting ranges (“September 4-6”, “April 1999/July 1999”, etc.).

A.2.3.2 How to annotate SIGNALs

Generally, the SIGNAL tag spans over one word or element:

(77) a. John taught on Monday
<SIGNAL xml:id="s1" target="#token2"/>

b. All passengers died when the plane crashed into the mountain.
<SIGNAL xml:id="s1" target="#token3"/>

When two distinct signals appear side by side, they can be annotated separately, if they belong to different signal classes as listed above. However, some situations require that they shall be annotated as a single <SIGNAL>. For example, in (78) the three temporal prepositions need to be collapsed into a single <SIGNAL> in order to properly recover the IS_INCLUDED relation of the TLINK between the genocide and the role events (see A.3).

(78) a. They will investigate the role of the US before, during and after the genocide.
b. <SIGNAL xml:id="s1" target="#token9 #token10 #token12"/>

A.3 The link elements: <TLINK>, <SLINK>, <ALINK> and <MLINK>

A.3.1 Overview

There are four types of link tags: <TLINK>, <SLINK>, <ALINK> and <MLINK>. The function of each will be introduced here, before we move on to explaining in detail how links are annotated.

A.3.2 The <TLINK> element

A.3.2.1 The attribute @relType values

A.3.2.1.1 A <TLINK> (temporal link) element represents the temporal relationship holding between two events, two times or between an event and a time, and indicates how they are related the possible @relType values for the <TLINK> are as follows.

A.3.2.1.2 SIMULTANEOUS: Two events are judged simultaneous if they happen at the same time, or are temporally indistinguishable in context, i.e. they occur close enough so that further distinguishing their times makes no difference to the temporal interpretation of the text. This is also used for expressing the duration of an ongoing event, as in

(79) Mary **taught** from **2 to 4**

A.3.2.1.3 One before the other (BEFORE): As in the following example between the events “slayings” and “arrested”:

(80) The police looked into the **slayings** of 14 women. In six of the cases, suspects have already been **arrested**.

A.3.2.1.4 One after the other (AFTER): This is just the inverse of the preceding relation. So the two events of the previous example can alternatively be annotated as expressing an after relation, if the directionality is changed.

A.3.2.1.5 One immediately before the other (IBEFORE): As in the following sentence between “crash” and “died”.

(81) All passengers died when the plane crashed into the mountain

A.3.2.1.6 One immediately after than the other (IAFTER): This is the inverse of the preceding relation.

A.3.2.1.7 One including the other (INCLUDES): As is the case between the temporal expression and the event in the following example:

(82) John arrived in Boston last Thursday.

A.3.2.1.8 One being included in the other (IS_INCLUDED): The inverse relation to the preceding one.

A.3.2.1.9 One holds during the other (DURING): Similar to INCLUDES, but used to relate one event included within another event.

(83) Mary sneezed while running.

A.3.2.1.10 One being the beginning of the other (BEGINS): As it holds between the first of the temporal expressions and the event in the following example:

(84) John was in the gym between 6:00 p.m. and 7:00 p.m.

A.3.2.1.11 One being begun by the other (BEGUN_BY): The inverse relation to the one just introduced.

A.3.2.1.12 One being the ending of the other (ENDS): As in:

(85) John was in the gym between 6:00 p.m. and 7:00 p.m..

A.3.2.1.13 One being ended by the other (ENDED_BY): The inverse relation to the one just introduced.

A.3.2.1.14 In addition, <TLINK>s are also used in the following situations:

A.3.2.1.14.1 Event identity (IDENTITY): Event identity is annotated via the <TLINK> (e.g. "John drove to Boston. During his drive he ate a donut."). Furthermore, <TLINK> will be used in order to relate the events in:

— **Causative constructions**. Triggered by verbs like the following, in their causative sense: "cause", "stem from", "lead to", "breed", "engender", "hatch", "induce", "occasion", "produce", "bring about", "produce", "secure". Two cases can be distinguished

Case 1 EVENT cause EVENT: The [rains] [caused] the [flooding].

Case 2 ENTITY cause EVENT: John [caused] the [fire].

The event introduced by the subject in Case 1 is related to the verbal predicate by a <TLINK> expressing event identity (we will see later that this means setting the <TLINK> @relType attribute as IDENTITY); on the other hand, the relation between the verbal event and that expressed by the object is represented by a <TLINK> of type BEFORE. Case 2 exhibits what is called event metonymy. Such constructions will not be annotated in the current specification.

— **Light verb constructions**: Similar to the case above, the verbal and nominal events will be related by means of a <TLINK> expressing event identity.

Event identity is a very important relationship, which will not be picked up during the closure part of the annotation. So it is extremely important to make sure that all identity links are annotated.

A.3.2.1.14.2 When a set/subset relationship occurs in the text:

An example is:

(86) The police looked into the **slayings** of 14 women. In six of the cases suspects have already been **arrested.**

Two <EVENT> tags are created, for each of the event sets: a first one marking up slayings, with cardinality 14, and a second one for cases, with cardinality 6. The two events will be related via a <TLINK> with the temporal relation IS_INCLUDED (or INCLUDES, depending on the directionality).

A.3.2.2 How to annotate <TLINK>s

Here and in the following clauses for <SLINK>s and <ALINK>s, we use examples to demonstrate how to create each of the link types. In these examples we do not give detailed annotation of events, times and signals – please refer to the appropriate clauses for instructions on annotating these. Also, we only show the mark-up for those entities which are relevant to the examples. A <TLINK> has to be created each time a temporal relationship holding between events or an event and a time needs to be annotated. This includes the important relationship of event identity. Examples:

(87) John taught on Monday

The temporal relationship holding between the event and the time expression, as indicated by the signal "on", is marked up by introducing the following <TLINK>:

(88) <TLINK eventID="#*e*1" relatedToTime="#*t*1" signalID="#*s*1" relType="IS_INCLUDED"/>

Here is another example:

(89) John taught every Monday

The TIMEX3 representing the multiple instances of "Monday" looks as follows:

(90) <**TIMEX3 xml:id**="*t*1" **target**="#range(token3, token4)" **type**="SET"**value**="XXXX-WXX-1"**quant**="EVERY"/>

The <TLINK> representing the temporal relation holding between the event and the temporal expression looks like this:

(91) <**TLINK eventID**="#*e*9" **relatedToTime**="#*t*1" **relType**="IS_INCLUDED"/>

(92) John taught from 2 to 4 on Monday.

The <EVENT> tag representing the event "taught" looks as follows:

(93) <**EVENT xml:id**="*e*4" **target**="#token1" **pred**="TEACH"/>

Two <TLINK>s have to be introduced.

One <TLINK> captures the fact that the "taught" event holds throughout the period from 2 to 4, and one <TLINK> captures the fact that this interval is included in Monday.

(94) a. <TLINK eventID="#*e*4" relatedToTime="#*t*2" signalID="#*s*5" relType="SIMULTANEOUS"/>
b. <TLINK timeID="#*t*2" relatedToTime="#*t*3" signalID="#*s*6" relType="IS_INCLUDED"/>

(95) John drove to Boston. During his drive he ate a donut.

The <EVENT> tags presenting the events drove and drive look as follows:

(96) a. <EVENT xml:id="e1" target="#token2" pred="DRIVE"/>
b. <EVENT xml:id="e2" target="#token7" pred="DRIVE"/>

A <TLINK> can be added to represent the identity of these two events:

<TLINK eventID="#*e*2" relatedToEvent="#*e*1" relType="IDENTITY"/>

A.3.3 The <SLINK> element

A.3.3.1 The attribute @relType values of <SLINK>

A.3.3.1.1 An <SLINK> (or subordination link) is used for contexts introducing relations between two events. <SLINK>s are of one of the following sorts:

- INTENSIONAL: This relation is brought up by events introducing a reference to a possible world – mainly I_ACTIONs and I_STATEs:

  (97) John **promised** Mary to buy some beer. Mary **wanted** John to buy some wine. The police **attempted** to arrest the robber.

- FACTIVE: Certain verbs presuppose or entail the veradicity (or factuality) of their event argument. They include “forget” (with a tensed complement), “regret” or “manage” (in positive contexts):

  (98) John **forgot** that he was in Boston last year. Mary **regrets** that she didn't marry John. John **managed** to leave the party

- COUNTER_FACTIVE: Contrary to the previous relation, in this case the event presupposes the non-veracity of its argument, e.g. “forget (to)”, “unable to” (in past tense), “prevent”, “cancel”, “avoid”, “decline”, etc.

  (99) John **forgot to** buy some wine. Mary was **unable to** marry John. John **prevented** the divorce.

- EVIDENTIAL: Evidential relations are typically introduced by REPORTING or PERCEPTION events:

  (100) John **said** he bought some wine. Mary **saw** John carrying only beer.

- NEG_EVIDENTIAL: Introduced by REPORTING and PERCEPTION events conveying negative polarity:

  (101) John **denied** he bought only beer.

A.3.3.1.2 <SLINK>s can be of the following nature:

a) **Lexically-based**:

They are triggered by an event of class I_ACTION, I_STATE, PERCEPTION, or REPORTING, which are events that generally take a clausal complement or an NP headed by an event-denoting nominal. The <SLINK> is established between those events and the one denoted by the complement.

For each REPORTING or PERCEPTION event, an <SLINK> has to be introduced. In the following example, the REPORTING and PERCEPTION events are in bold face, whereas the subordinated events are underlined:

(102) 15 minutes later **I saw** the other plane just slam into the World Trade Center. “It sounded like a jet or rocket," **said** Eddie Gonzalez.

Similarly, for each I_ACTION or I_STATE, an <SLINK> is introduced, which expresses the relation between the intensional event (in bold face) and its subordinated event (underlined):

(103) Subcomandante Marcos **attempted** to explain this difference in a letter in 1995. We **want** to participate directly in the decisions which concern us, to control those who govern us.

The subordinating event class constrains the <SLINK> relation type in the following way:

i. **PERCEPTION events**:

They will always introduce <SLINK>s of @relType EVIDENTIAL or NEG_EVIDENTIAL.

ii. **I_ACTION, I_STATE events**:

They can introduce <SLINK>s of @relType INTENSIONAL, FACTIVE, and COUNTER_FACTIVE.

iii. **REPORTING events**:

They can introduce <SLINK>s of any @relType.

b) **Structurally-based**:

1) **Purpose clauses**: In a sentence involving a purpose clause, an <SLINK> relates the event in the main clause (bold face) and the one in the purpose clause modifying it (underlined).

(104) The environmental commission must **adopt** regulations to ensure people are not exposed to radioactive waste.

2) **Conditional contructions**: In a conditional construction, an <SLINK> relates the event in the antecedent section and the one in the consequent section.

(105) On Dec. 2 Marcos promised to return to the negotiating table if the conflict zone was **demilitarized**.

A.3.3.2 How to annotate <SLINK>s

A.3.3.2.1 Lexically-based SLINKS

(106) John **said** that he taught on Monday.

To express the fact that the “taught" event is reported by the “said” event, the following <SLINK> is created:

(107) <SLINK eventID="#e2" subordinatedEvent="#e3" relType="EVIDENTIAL"/>

(108) John **denied** that he taught on Monday.

To express the fact that the “taught" event is being reported by the “denied” event, the following <SLINK> is created:

(109) <SLINK eventID="#e1" subordinatedEvent="#e2" relType="NEG_EVIDENTIAL"/>

In some cases the same subordinating event will introduce more than one <SLINK>. For instance, in the example below the event “said” is SLINK-ed to two events: “listed” and “gave”.

(110) Rita **said** they correctly listed his name but gave a false address for him.

(111) <SLINK eventID="#e1" subordinatedEvent="#e2" relType="EVIDENTIAL"/>
<SLINK eventID="#e1" subordinatedEvent="#e3" relType="EVIDENTIAL"/>

A.3.3.2.2 Structurally-based <SLINK>s

A.3.3.2.2.1 Purpose clauses: The event in the main clause will correspond to the value of the attribute @eventID. The event in the purpose clause will be taken as the @subordinatedEvent value. These <SLINK>s will always receive @relType="INTENSIONAL". The preposition "to" will be taken as the value of the @signalID attribute.

(112) a. The environmental commission must **adopt** regulations to ensure people are not exposed to radioactive waste.
b. <SLINK eventIID="#e1" subordinatedEvent="#e2" signalID="#s1" relType="INTENSIONAL"/>

A.3.3.2.2.2 Conditional contructions: The event in the antecedent clause corresponds to the value in the @eventID attribute. The one in the consequent, to the value of the attribute @subordinatedEvent. The conditional conjunction (“if", “when") will be taken as the value of the @signalID attribute. These <SLINK>s will always receive @relType="CONDITIONAL".

(113) a. Mexico **pledged**_e2 to support an inquiry into Guantanamo **if**_s1 it is **put**_e1 to the vote at the UN Human Rights Commission.
b. <SLINK eventID="#e1" subordinatedEvent="#e2" signalID="#s1" relType="CONDITIONAL"/>

The presence of the same event in several <SLINK>s is also possible in structurally-based <SLINK>s, as for instance in the following conditional construction, where the antecent is a coordination. In this case, the repeated event is the subordinated one (return), since it is the event in the consequent section.

(114) a. On Dec. 2 Marcos **promised_e1** to return to the negotiating table **if_s1** the conflict zone was **demilitarized_e2**, Congress **passed_e3** a bill on indigenous rights and culture, and around 100 Zapatista prisoners were **released_e4**.
b. <SLINK eventID="#e2" subordinatedEvent="#e1" signalID="#s1" relType="CONDITIONAL"/>
<SLINK eventID="#e3" subordinatedEvent="#e1" signalID="#s1" relType="CONDITIONAL"/>
<SLINK eventID="#e4" subordinatedEvent="#e1" signalID="#s1" relType="CONDITIONAL"/>

A.3.4 The <ALINK> element

A.3.4.1 The attribute @relType values of <ALINK>

An <ALINK> or (aspectual link) represents relations between aspectual events and their event arguments. Types of aspectual relations to be encoded are:

— INITIATION:

(115) John **started** to read.

— CULMINATION:

(116) John **finished** assembling the table.

— TERMINATION:

(117) John **stopped** talking.

— CONTINUATION:

(118) John **kept** talking.

A.3.4.2 How to annotate <ALINK>s

Some annotation examples are given below.

(119) John started to read.

The two <EVENT> tags for the two events are the following:

(120) <EVENT xml:id="e5" target="#token1" pred="START"/>
<EVENT xml:id="e6" target="#token3" pred="READ"/>

The <ALINK> that has to be created between the aspectual verb “started” and the event “read” is the following:

(121) <ALINK eventID="#e5" relatedToEvent="#e6" relType="INITIATES"/>

(122) John finished reading.

The two <EVENT> tags for the two events are the following:

(123) <EVENT xml:id="e7" target="#token1" pred="FINISH"/>
<EVENT xml:id="e8" target="#token2" pred="READ"/>

The <ALINK> that has to be created between the aspectual verb and its argument is the following:

(124) <ALINK eventID="#e7" relatedToEvent="#e7" relType="TERMINATES"/>

A.3.5 The <MLINK> element

<MLINK> is a link for measuring the duration of an event. Like the other linking elements, the element <MLINK> has three required attrributes: @eventID, @relatedToTime and @relType. The value of @relType is MEASURES. Optional attributes are: @signalID, @comment, @syntax, etc. Below is an example:

(125) John taught for three hours

<**EVENT xm**l:id="e1"target="#token2" pred="TEACH" class="OCCURRENCE" type="PROCESS"
tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#range(token4,token5)" type="DURATION" value="P3H"/>
<MLINK eventID="e1" relatedToTime="#t1" relType="MEASURES"/>

Annex B
(informative)

Completely annotated examples

B.1 Complex TIMEX3 examples

a) John left 2 days before yesterday.

```
<EVENT xml:id="e1" target="#token1" pred="LEAVE" type="TRANSITION" class="OCCURRENCE"
 pos ="VERB" tense="PAST" aspect="NONE" polarity="POS" />
<TIMEX3 xml:id="t1" target="#token2 #token3" type="DURATION" value="P2D" beginPoint="#t2"
 endPoint="#t3" />
<SIGNAL xml:id="s1" target="#token4" pred="BEFORE"/>
<TIMEX3 xml:id="t2" target="#token5" pred="YESTERDAY" type="DATE" value="2002-07-10"
 temporalFunction="true" anchorTimeID="#t0" />
<TIMEX3 xml:id="t3" type="DATE" value="2002-07-08" temporalFuntion="true" anchorTimeID="#t1"/>
<TLINK timeID="#t1" relatedToTime="#t2" signalID="#s1" relType="BEFORE"/>
<TLINK eventID="#e1" relatedToTime="#t3" relType="IS_INCLUDED"/>
```

The TIMEX3 that is annotated as a DURATION includes @begin and @endpoint information. The annotator also could have included additional <TLINK>s with this information.

The @type attribute of the <TIMEX3> for “yesterday" denotes a DATE, which can be computed by a temporal function relative to the temporal offset “*t*0" (the DCT, data creation time). Similarly, the final <TIMEX3> can be computed by a @temporalFunction relative to the initial PERIOD, annotated as “*t*1".

b) I'm leaving on vacation two weeks from next Tuesday.

```
<TIMEX3 xml:id="t1" target="#token4 #token5" pred="TWO_WEEKS" type="DURATION" value="P2W"
  beginPoint="#t2" endPoint="#t3"/>
<SIGNAL xml:id="s1" target="#token6" pred="FROM"/>
<TIMEX3 xml:id="t2" target="#token7 #token8" pred="NEXT_TUESDAY" type="DATE" value="2002-07-02"
  temporalFunction="true" anchorTimeID="#t0"/>
<TIMEX3 xml:id="t3" type="DATE" value="2002-07-23" temporalFunction="true" anchorTimeID="#t1"/>
<TLINK timeID="#t1" relatedToTime="#t2" signalID="#s1" relType="AFTER"/>
<TLINK eventID="ei1" relatedToTime="t3" relType="IS_INCLUDED"/>
```

c) A major earthquake struck Los Angeles three years ago today.

```
<EVENT xml:id="e1" target="#token2" pred="EARTHQUAKE" type="PROCESS" class="OCCURRENCE"
  pos="NOUN" tense="NONE" aspect="NONE"/>
<EVENT xml:id="e2" target="#token3" pred="STRIKE" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#token6 #token7 #token8" pred="THREE_YEARS_AGO" type="DURATION"
  value="P3Y" beginPoint="#t2" endPoint="#t3"/>
<TIMEX3 xml:id="t2" target="#token9" pred="TODAY" type="DATE" value="2002-07-12"
  temporalFunction="true" anchorTimeID="#t0"/>
<TIMEX3 xml:id="t3" type="DATE" value="1999-07-12" temporalFunction="true" anchorTimeID="#t1"/>
<TLINK eventID="#e1" relatedToEvent="#e2" relType="IBEFORE"/>
<TLINK eventID="#e1" relatedToTime="#t3" relType="IS_INCLUDED"/>
```

d) John left 2 days ago.

```
<EVENT xml:id="e1" target="#token1" pred="LEAVE" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#token2 #token3 #token4" pred="TWO_DAYS_AGO" type="DATE"
  value="2002-07-08" temporalFunction="true" anchorTimeID="#t0"/>
```

NOTE “ago” (token4) is NOT a signal but a part of the <TIMEX3> expression.

The <TIMEX3> expression returns a DATE (not a DURATION), which needs to be computed by a temporal function relative to the DCT or the speech time. “2 days ago” is ALWAYS a DATE computed relative to the DCT, in contrast to expressions like “2 days before".

e) John left 2 days before the attack.

```
<EVENT xml:id="e1" target="#token1" pred="LEAVE" type="TRANSITION" class="OCCURRENCE"
 pos ="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#token2 #token3" pred="TWO_DAYS" type="DURATION" value="P2D"
  temporalFunction="false"/>
<SIGNAL xml:id="s1" target="#token4" pred="BEFORE"/>
<EVENT xml:id="e2" target="token6" pred="ATTACK" type="TRANSITION" class="OCCURRENCE"
pos="NOUN" tense="NONE" aspect="NONE"/>
<TLINK eventID="#e1" signalID="#s1" relatedToEvent="#e2" relType="BEFORE"/>
```

f) 5 days after he came back Mary got sick.

```
<TIMEX3 xml:id="t1" target="#token0 #token1" pred="FIVE_DAYS" type="DURATION" value="P5D"
  temporalFunction="false"/>
<SIGNAL xml:id="s1" target="#token2" pred="AFTER"/>
<EVENT xml:id="e1" target="#token4" pred="COME" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS" />
<EVENT xml:id="e2" target="#token7" pred="GET" type="TRANSITION" class="OCCURRENCE"
 pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<TLINK eventID="#e1" signalID="#s1" relatedToEvent="#e2" relType="BEFORE"/>
```

g) Two months before the attack, a report was sent.

```
<TIMEX3 xml:id="t1" target="#token0 #token1" pred="TWO_MONTHS" type="DURATION" value="P2M"
  temporalFunction="false"/>
<SIGNAL xml:id="s1" target="#token2" pred="BEFORE"/>
```

The <TIMEX3> expression here is considered here to be of @type="DURATION", since it establishes the length of the interval separating the 2 events. As such, the value for the @value attribute is already known (P2D, P5M, etc.) and therefore the @temporalFunction attribute returns false as its value.

There is only one <TLINK> relating the two events, which introduces both the @magnitude attribute (pointing to the ID of the <TIMEX3> expression) and the @signalID attribute.

B.2 Complex TLINK and SLINK examples

a) The attack was not expected at all, although a report had been sent 2 months before.

```
  <EVENT xml:id="e1" target="#token1" pred="ATTACK" type="TRANSITION" class="OCCURRENCE"
    pos="NOUN" tense="NONE" aspect="NONE" polarity="POS" />
  <EVENT xml:id="e2" target="#token4" pred="EXPECT" type="TRANSITION" class="I_STATE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="NEG"/>
  <EVENT xml:id="e3" target="#token13" pred="SEND" type="TRANSITION" class="OCCURRENCE"
    pos="VERB" tense="PAST" aspect="PERFECTIVE" polarity="POS"/>
```

```
<TIMEX3 xml:id="t1" target="#token14 #token15" pred=TWO_MONTHS" type="DURATION" value="P2M"
  beginPoint="#e2" endPoint="#e1" />
<SIGNAL xml:id=s2 target="#token15" pred="BEFORE"/>
<SLINK eventID="#e2" subordinatedEvent="#e1" relType="INTENSIONAL"/>
<TLINK eventID="#e1" relatedToEvent="#e3" relType="AFTER" signalID="#s2"/>
```

b) Mary arrived yesterday but John left 2 days before.

```
<EVENT xml:id="e1" target="#token1" pred="ARRIVE" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t1" target="#token2" pred="YESTERDAY" type="DATE" value="2002-07-09"
 temporalFunction="true" anchorTimeID="#t0"/>
<EVENT xml:id="e2" target="#token5" pred="LEAVE" type = "TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<TIMEX3 xml:id="t2" target="#token6 #token7" pred="TWO_DAYS" type="DURATION" value="P2D"
  temporalFunction="false" beginPoint="#e2" endPoint="#e1"/>
```

The two events are related by means of a <TLINK>. In addition, there is a second <TLINK> relating the event (“arrived”) linked to the date and this date (“yesterday”).

c) She was sick after the play.

```
<EVENT xml:id="e1" target="#token2" pred="SICK" type= "STATE" class="STATE" pos="ADJECTIVE"
  tense="PAST" aspect="NONE" polarity="POS" />
<SIGNAL xml:id="s1" target="#token3" pred="AFTER"/>
<EVENT xml:id="e2" target="#token5" pred="PLAY" type="TRANSITION" class="OCCURRENCE"
  pos="NOUN" tense="NONE" aspect="NONE"/>
<TLINK eventID="#e1" signalID="#s1" relatedToEvent="#e2" relType="AFTER"/>
```

d) She was sick for 2 hours after the play.

```
<EVENT xml:id="e1" target="#token2" pred="SICK" type= "STATE" class="STATE" pos="ADJECTIVE"
  tense="PAST" aspect="NONE" polarity="POS"/>
<SIGNAL xml:id="s1" target="#token3" pred="FOR"/>
<TIMEX3 xml:id="t1" target="#token4 #token5" pred="TWO_HOURS" type="DURATION" value="P2H"
  temporalFunction="false"/>
<SIGNAL xml:id="s2" target="#token6" pred="AFTER"/>
<EVENT xml:id="e2" target="#token8" pred="PLAY" type="TRANSITION" class="OCCURRENCE"
  pos="NOUN" tense="NONE" aspect="NONE"/>
<TLINK eventID="#e1" signalID="#s1" relatedToTime="#t1" relType="SIMULTANEOUS"/>
<TLINK eventID="#e1" signalID="#s2" relatedToEvent="#e2" relType="AFTER"/>
```

There are two TLINKs. The first one introduces the holding relation between the state of being sick and the time it took (2 hours). The second one states the ordering of the two events.

e) John taught for 20 minutes every Monday.

```
<EVENT xml:id="e1" target="#token1" pred="TEACH" type= "PROCESS" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<SIGNAL xml:id="s1" target="#token2" pred="FOR"/>
<TIMEX3 xml:id="t1" target="#token3 #token4" pred="20_MINUTES" type="DURATION" value="P2M"
  temporalFunction="false"/>
<TIMEX3 xml:id="t2" target="#token5 #token6" pred="EVERY_MONDAY" type="SET"
 value="XXXX-WXX-1"/>
<TLINK eventD="#e1" relatedToTime="#t2" relType="SIMULTANEOUS"/>
<TLINK timeID="#t1" relatedToTime="#t2" relType="IS_INCLUDED"/>
```

f) John left between Monday and Wednesday

```
<EVENT xml:id="e1" target="#token1" pred="LEAVE" type="TRANSITION" class="OCCURRENCE"
  tense="PAST" aspect="NONE" />
<SIGNAL xml:id="s1" target="#token2" pred="BETWEEN"/>
<TIMEX3 xml:id="t1" target="#token3" pred="MONDAY" type="DATE" value="2002-07-15"
 temporalFunction="true" anchorTimeID="t0"/>
<TIMEX3 xml:id="t2" target="token4" pred="WEDNESDAY" type="DATE" value="2002-07-17"
   temporalFunction="true" anchorTimeID="#t0" valueFromFunction="tf3"/>
<TLINK eventID="#e1" relatedToTime="#t1" signalID="#s1" relType="IAFTER"/>
<TLINK eventID="#e1" relatedToTime="#t2" signalID="#s1" relType="IBEFORE"/>
```

NOTE This current solution is not completely adequate, but we will keep it temporarily.

g) John taught from 1994 through 1999.

In this case, one <EVENT> and the two <TIMEX3>s need to be created. In addition, the following tags are needed.

1) One <TLINK> to capture the fact that the event started in 1994.

2) One <TLINK> to capture the fact that the event finished in 1999.

This then should lead to a duration, which is automatically created by the closure part of the tool.

```
<EVENT xml:id="e4" target="#token1" pred="TEACH" type="TRANSITION" class="OCCURRENCE"
  tense="PAST" aspect="NONE" polarity="POS"/>
<SIGNAL xml:id="s5" target="#token2" pred="FROM"/>
<TIMEX3 xml:id="t2" target="#token3" pred="1994" type="DATE" value="1994"/>
<SIGNAL xml:id="s6" target="#token4" pred="THROUGH"/>
<TIMEX3 xml:id="t3" target="#token5" pred="1999" type="DATE" value="1999"/>
<TIMEX3 xml:id="t1" type="DURATION" value="P2Y" beginPoint="#t2" endPoint="#t3"/>
<TLINK eventID="#e4" relatedToTime="#t2" signalID="#s5" relType="BEGUN_BY"/>
<TLINK eventID="#e4" relatedToTime="#t3" signalID="#s6" relType="ENDED_BY"/>
```

h) John did not leave on Monday but on Tuesday.

One <EVENT>, another related <EVENT> with @target="NIL", and two <TLINK>s need to be created.

```
<EVENT xml:id="e1" target="#token3" pred="LEAVE" type="TRANSITION" class="OCCURRENCE"
  tense="PAST" aspect="NONE" polarity="NEG"/>
<SIGNAL xml:id="s6" target="#token4" pred="ON"/>
<TIMEX3 xml:id="t3" target="#token5" pred="MONDAY" type="DATE" value="XXXX-WXX-1"/>
<EVENT xml:id="e2" target="NIL" pred="LEAVE" type="TRANSITION" class="OCCURRENCE"
  tense="PAST" aspect="NONE" polarity="POS"/>
<SIGNAL xml:id="s7" target="#token7" pred="ON"/>
<TIMEX3 xml:id="t4" target="#token8" pred="TUESDAY" type="DATE" value="XXXX-WXX-2"/>
<TLINK eventID="#e1" relatedToTime="#t3" signalID="#s6" relType="IS_INCLUDED"/>
<TLINK eventID="#e2" relatedToTime="#t4" signalID="#s7" relType="IS_INCLUDED"/>
```

B.3 Causative examples

a) The rains caused the flooding.

```
<EVENT xml:id="e1" target="#token1" pred="RAIN" type="PROCESS" class="OCCURRENCE"
  pos="NOUN"/>
<EVENT xml:id="e2" target="#token2" pred="CAUSE" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<EVENT xml:id="e3" target="#token4" pred="FLOOD" type="PROCESS" class="OCCURRENCE"
  pos="NOUN"/>
<TLINK eventID="#e1" relatedtoEvent="#e3# relType="BEFORE"/>
<TLINK eventID="#e1" relatedtoEvent="#e2# relType="IDENTITY" />
```

b) John caused the fire.

```
<EVENT xml:id="e1" target="#token1" pred="CAUSE" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<EVENT xml:id="e2" target="#token3" pred="FIRE" type="PROCESS" class="OCCURRENCE"
  pos="NOUN"/>
<TLINK eventID="#e1" relatedtoEvent="#e2# relType="BEFORE"/>
```

c) Kissinger secured the peace at great cost.

```
<EVENT xml:id="e1" target="#token1" pred="SECURE" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<EVENT xml:id="e2" target="#token3" pred="PEACE" type="STATE" class="OCCURRENCE"
pos="NOUN"/>
<TLINK eventID="#e1" relatedtoEvent="#e2" relType="BEFORE"/>
```

d) He kicked the ball, and it rose into the air.

```
<EVENT xml:id="e1" target="#token1" pred="KICK" type="TRANSITION" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"/>
<EVENT xml:id="e2" target="#token7" pred="RISE" type="PROCESS" class="OCCURRENCE"
  pos="VERB" tense="PAST" aspect="NONE" polarity="POS"//>
<TLINK eventID="#e1" relatedtoEvent="#e2" relType="BEFORE"/>
```

Discourse relations acting as a causative will be handled in the next release as causatives in a Causative Link (CLINK), but this is out of the scope of our current discussion.

Annex C
(informative)

Event and temporal annotations for Chinese

(1)

他是中国人。

He is a Chinese.

(2)

花是红色的。

Flowers are red.

Examples (1) and (2) are not annotated in *ISO-TimeML*, as stated in the ISO document, if a STATE is deemed persistent throughout the event line of the document, it is factored out and not annotated.

(3)

他在跑步。

He is running.

```
<EVENT
 xml:id="e1"
 target="#token2 #token3"
 pred="RUN"
 class="OCCURRENCE"
 type="PROCESS"
 pos="VERB"
 tense="NONE"
 aspect="PROGRESSIVE"
 polarity="POS"/>
```

(4)

很多人在公园里跳健美操。

Many people are practicing aerobics in the park.

```
<EVENT
 xml:id="e1"
 target="#token7"
 pred="PRACTISE"
 class="OCCURRENCE"
 type="PROCESS"
 pos="VERB"
 tense="NONE"
 aspect="PROGRESSIVE"
 polarity="POS"/>
<CONFIDENCE
 tagType="EVENT"
 tagID="#e1"
 attributeName="ASPECT"
 confidenceValue="0.50"/>
```

(5)

这部电影拍了三年的时间终告竣工。

The film took three years to shoot.

```
<EVENT
 xml:id="e1"
 target="#token4 #token5"
 pred="SHOOT"
 class="OCCURRENCE"
 type="TRANSITION"
 pos="VERB"
 tense="NONE"
 aspect="PERFECTIVE"
 polarity="POS"/>
<TIMEX3
 xml:id="t1"
 target="#token6 #token7"
 pred="THREE_YEARS"
 type="DURATION"
 value="P3Y"/>
<EVENT
 xml:id="e2"
 target="#token13 #token14"
 pred="COMPLETE"
 class="ASPECTUAL"
 type="TRANSITION"
 pos="VERB"
 tense="NONE"
 aspect="PERFECTIVE"
 polarity="POS"/>
<TLINK eventD="#e1" relatedToTime="#t1" relType="IS_INCLUDED"/>
<TLINK eventID="#e2" relatedToEvent="#e1" relType="IAFTER"/>
```

(6)

我花了一周时间才读完这本书。

I spent one week and finished reading this book.

```
<EVENT
 xml:id="e1"
 target="#token1 #token2"
 pred="SPEND"
 class="OCCURRENCE"
 type="PROCESS"
 pos="VERB"
 tense="NONE"
 aspect="PERFECTIVE"
 polarity="POS"/>
<TIMEX3
 xml:id="t1"
 target="#token3 #token4 #token5 #token6"
 type="DURATION"
 value="P1W"/>
<EVENT
 xml:id="e2"
 target="#token8 #token9"
 pred="READ"
 class="OCCURRENCE"
```

```
  type="PROCESS"
  pos="VERB"
  tense="NONE"
  aspect="PERFECTIVE"
  polarity="POS"/>
<TLINK eventD="#e1" relatedToTime="#t1" relType="IS_INCLUDED"/>
<TLINK
  eventID="#e2"
  signaledID="#s1"
  relatedToEvent="#e1"
  relType="IAFTER"/>
```

(7)

钥匙掉到地上去了。

The key dropped to the floor.

```
<EVENT
  xml:id="e1"
  target="#token2"
  pred="DROP"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
```

(8)

门突然被打开。

The door was suddenly pushed open.

```
<EVENT
  xml:id="e1"
  target="#token2"
  pred="PUSH"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
```

(9)

他在做作业。

He is doing his homework.

```
<EVENT
  xml:id="e1"
  target="#token2"
  pred="DO"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="NONE"
  aspect="PROGRESSIVE"
  polarity="POS"/>
```

(10)

他做了三个小时的作业才做完。

He spent three hours and finished doing his homework.

```
<EVENT
  xml:id="e1"
  target="#token1 #token2"
  pred="DO"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="NONE"
  aspect="PERFECTIVE"
  polarity="POS"/>
<TIMEX3
  xml:id="t1"
  target="#timrange(token3,token6)"
  type="DURATION"
  value="P3H"/>
<SIGNAL xml:id="s1" target="#token10"/>
<EVENT
  xml:id="e2"
  target="#token11 #token12"
  pred="DO"
  class="OCCURRENCE"
  type="PROCESS"
  pos="VERB"
  tense="NONE"
  aspect="PERFECTIVE"
  polarity="POS"/>
<TLINK eventID="#e1" relatedToTime="#t1" relType="IS_INCLUDED"/>
<TLINK
  eventID="#e2"
  signalID="#s1"
  relatedToEvent="#e1"
  relType="IAFTER"/>
```

(11)

他做了作业。

He has done/did his homework.

```
<EVENT
  xml:id="e1"
  target="#token1 #token2"
  pred="DO"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="NONE"
  aspect="PERFECTIVE"
  polarity="POS"/>
```

(12)

9月11日，这是一个让全美国人民陷于悲痛的日子。

9/11 is a day that made all the Americans saddened.

```
<TIMEX3
  xml:id="t1"
  target="#range(token0,token3)"
  type="DATE"
  value="XXXX-09-11"/>
<EVENT
  xml:id="e1"
  target="#token6"
  pred="DO"
  class="OCCURRENCE"
  type="PROCESS"
  pos="VERB"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
<EVENT
  xml:id="e2"
  target="#token15"
  pred="DO"
  class="OCCURRENCE"
  type="PROCESS"
  pos="VERB"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
<EVENT
  xml:id="e3"
  target="#token17 #token18"
  pred="SAD"
  class="STATE"
  pos="ADJECTIVE"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
<TLINK eventID="#e1" relatedToTime="#t1" relType="IS_INCLUDED"/>
<TLINK eventID="#e2" relatedToEvent="#e3" relType="IDENTITY"/>
```

(13)

他一直都被蒙在鼓里。

He has been kept away from the truth.

```
<EVENT
  xml:id="e1"
  target="#token5"
  pred="KEEP"
  class="OCCURRENCE"
  type="PROCESS"
  pos="VERB"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
```

(14)

自他从北京回来后，脸上的笑容逐渐多了起来。

His face has shown more and more smiles since he got back from Beijing.

```
<EVENT
  xml:id="e1"
  target="#token5 #token6"
  pred="GET_BACK"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
<SIGNAL xml:id="s1" target="#token7"/>
<EVENT
  xml:id="e2"
  target="#token16"
  pred="MORE"
  class="OCCURRENCE"
  type="PROCESS"
  pos="ADJECTIVE"
  tense="NONE"
  aspect="NONE"
  polarity="POS"/>
<TLINK
  eventID="#e2"
  signalID="#s1"
  relatedToEvent="#ei1"
  relType="AFTER"/>
```

(15)

我吃过饭了。

I have had/had my dinner.

```
<EVENT
  xml:id="e1"
  target="#token1 #token2"
  pred="EAT"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="NONE"
  aspect="PERFECTIVE"
  polarity="POS"/>
```

Annex D
(informative)

Annotation for Italian fragment

D.1 Introduction

This annex describes the annotation guidelines for marking up Italian text according to the *ISO-TimeML* language. It is organized as follows. The first part explains how the *ISO-TimeML* tags are realized in Italian and how to annotate them. The second part is more informative and contains a fully annotated example, illustrating all of the interactions between the various entities and relational tags. For the sake of convenience, Annex A will be referred to throughout the annex.

D.2 Basic references

This annex relies on the following references:

BERTNETTO, P.M. (2001) *Sulle proprietà tempo-aspettuali dell'Infinito italiano*, available at http://linguistica.sns.it/QLL/QLL01/PMB/Infinito.pdf

BERTNETTO, P.M. (1991) Il Verbo. In L. Renzi and G. Salvi (a cura di), *Grande Grammatica Italiana di Consultazione*, Il Mulino pp. 13-163

BERTINETTO, P.M. (1986) *Tempo, Aspetto e Azione nel verbo Italiano*, Accademia della Crusca, Firenze.

LAVELLI, A., MAGNINI, B., NEGRI, M., PIANTA, E., SPERANZA, M. and SPRUGNOLI, R.(2005) Italian Content Annotation Bank (I-CAB): Temporal Expressions (V. 1.0), ITC-irst Technical Report.

D.3 ISO-TimeML elements and their attributes

D.3.1 How to annotate EVENTs

The types of expressions detonting an event in Italian are much the same as those in A.2.1.1. For clarity's sake the following phrase types are some examples. Event-denoting expressions are emphasized.

D.3.1.1 Verbs (finite or non-finite form)

- I pompieri *hanno isolato* la sala.

  [The firefighters isolated the room.]

- Fim-Cisl e Uilm-Uil *hanno annunciato* oggi una conferenza stampa.

  [Fim-Cisl and Uilm-Uil have announced a press meeting today.]

- La città mostra i segni della battaglia: cassonetti *incendiati* o *rivoltati*.

  [The city shows the signs of the battle: garbage bins burned or turned out.]

D.3.1.2 Nominalizations

— La *caduta* della base aerea di Ubdina allontana il fronte di 120 km.

[The fall of the air base of Ubdina has taken the front away of 120 km.]

D.3.1.3 Adjectives

— Si ritiene *furbo*

[He thinks he is clever.]

D.3.1.4 Predicative sections

— Al Sayed *è il nuovo presidente* della Fermenta.

[Al Sayed is the new president of Fermenta.]

D.3.1.5 Prepositional phrases

— Una giovane turista *in vacanza* nel villaggio “Katibubbo” è morta.

[A young tourist on vacation at the “Kartitubbo” resort died.]

— Un centinaio di giovani è tuttora *agli arresti*.

[About a hundred young men and women are still under arrest.]

D.3.1.6 Nouns with an event-like reference

— Queste le principali indicazioni strategiche fatte da Cabassi al termine dell'*assemblea*.

[These are the main strategic directions made by cabassi at the end of the meeting.]

— *Allarme* inconsueto alla Tate Gallery.

[Unusual alarm at the Tate Gallery.]

— Rota auspica una *pace* rapida.

[Rota foretells a quick peace.]

— Questo consentirebbe di discutere con serenità e fermezza i *problemi* della minoranza italiana.

[This will allow to fairly and firmly discuss the problems of the Italian minority.]

D.3.2 Event identification and tag span

D.3.2.1 General

D.3.2.1.1 As proposed in Annex A, the annotation of Italian texts is based on the notion of minimal chunk. This means that only the head of the chunk will be covered by the tag and thus auxiliaries, clitics, polarity markers, particles, modifiers, complements and specifiers will be disregarded. In the following examples, the event-denoting chunk is in bold face and the tagged head is underlined.

— I pompieri ***hanno isolato*** la sala.

[The firefighters isolated the room.]

- ***Accusandoli*** di omicidio...

  [Accusing them of murder...]

- Il PIL Italiano ***non è cresciuto*** nell'ultimo trimestre.

  [The Italian GDP did not grow in the last quarter.]

- ***La caduta della base aerea di Ubdina*** allontana il fronte di 120 km.

  [The fall of the air base of Ubdina has taken the front away of 120 km.]

- Al Sayed ***è il nuovo presidente*** della Fermenta.

  [Al Sayed is the new president of Fermenta.]

- Si ritiene *furbo*

  [He thinks he is clever.]

D.3.2.1.2 As far as prepositions are concerned, if the prepositional chunk denotes an event, then the entire prepositional phrase shall be included into the tag; otherwise, only the noun head of the embedded NP shall be annotated.

- Le strade mostrano ancora i segni ***della battaglia***.

  The streets still show the signs of the battle.

- Un centinaio di giovani è tuttora ***agli arresti.***

  About a hundred young men and women are still under arrest.

D.3.2.1.3 Most event tags will span over only one word, i.e. the minimal chunk head. However, in case of light verb constructions, copular constructions and causative constructions require an event tag for all the items involved. In the following examples the event tags are represented in square brakets, e.g.

- I guardiani hanno [*fatto*] [*scattare*] l'allarme.

  [The guards made the alarm go on.]

- Gli Usa hanno [*fatto*] [*sapere*] che non sono disponibili.

  [The USA said that they are not available.]

- L'assemblea ha [*preso*] [*visione*] del bilancio consolidato.

  [The assembly took vision of the consolidated financial statements.]

- Marco ha [*fatto*] una [*passeggiata*].

  [Marco had a walk.]

D.3.2.2 Modal verbs

Modal verbs in Romance languages are very different from the English ones. In Italian, modal verbs are to be considered similar to other lexical verbs in that it is possible to assign them values for tense and aspect. Consequently, each instance of Italian modal verbs (dovere, potere, volere, sapere) will be annotated with the tag <EVENT>, e.g.

— L'assemblea [*deve*] [*prendere*] una decisione....

— Non ho [*potuto*] [*chiamare*] l'ufficio cambi.

D.3.2.3 Verbal periphrases

In Italian, it is possible to identify different instances of verbal periphrases. We accept here the proposal of Bertinetto (1991) to identify a hierarchy of verbal periphrases:

a) aspectual periphrases which code progressive or habitual aspect, e.g.

   1) Sta mangiando. [He is eating.],

   2) È solito riposare dopo pranzo. [He is used to rest after lunch.].

b) modal periphrases which code modality are not realized by proper modal verbs, e.g.

   1) essere in grado di + INFINITIVE,

   2) c'è da + INFINITIVE,

   3) andare + INIFINITIVE,

   4) avere da + INFINITIVE.

c) phasal (aspectual) periphrases which code information on a particular phase (or aspect) in the description of a particular event, corresponding to aspectual verbs or super-lexical aspectual morphemes, e.g. iniziare/finire/continuare/terminare. . . + V.

NOTE Phasal verbs are called aspectual verbs in *ISO-TimeML*. The *ISO-TimeML* term will be used to refer to these kinds of verbs.

Following Bertinetto (1991), we propose that in the presence of modal periphrases and phasal/aspectual periphrases both elements involved be annotated, while in the case of the aspectual periphrasis only the main verb be marked. In the examples below, the event denoting chunk is in italics, whereas the tagged head(s) is underlined.

— La borsa *stava perdendo l'1,1 %* in prima mattinata.(Progessive periphrasis)

[The market was down 1 % in the early morning.]

— A oggi siamo in grado di dire che l'accordo non è stato raggiunto. (Modal periphrasis)

[As for now, we can say that the agreement has not been reached.]

— *C'è da dire* che questo trattamento non è soddisfacente. (Modal periphrasis)

[We have to say that this treatement is not satisfying.]

— Il magistrato *ha iniziato a condurre* le indagini sulla morte di Calipari. (Aspectual periphrasis)

[The prosecutor has begun to investigate on Calipari's death.]

D.3.3 What NOT to tag

Events are not to be tagged in the situations described in A.2.1.2.4.

D.3.4 Introductory note

A.2.1.3 is referred to for attribute definitions. In this clause, we illustrate the attribute @class for informative purposes, and the language specific values of the other attributes.

D.3.4.1 The attribute @class

D.3.4.1.1 Each event belongs to one of the following classes, as definied in A.2.1.3.2.

D.3.4.1.2 REPORTING: dire [to say/tell], spiegare [to explain], raccontare [to tell], affermare [to claim/state]…

(1) a. Il governatore ha **ribadito** che l'accordo può essere raggiunto. [The Governor has claimed that the agreement may be reached.]

b. **Citando** l'esempio di... [Citing the example of...]

D.3.4.1.3 PERCEPTION: vedere [to see], guardare [to look at], osservare [to gaze/to observe], ascoltare [to listen to], sentire [to hear]…

(2) a. Dei testimoni hanno dichiarato alla polizia di aver **visto** delle persone fuggire. [Two witnesses have declared to the police that they saw some people running away.]

b. “Puoi **sentire** le esplosioni da qui", ha detto un testimone. [“You can hear the explosions from here” a witness has reported.]

D.3.4.1.4 ASPECTUAL: iniziare [to begin/set out], incominciare [to start/lead off], rincominciare [to restart], smettere [to stop/cease], terminare [to terminate], cessare [to finish], interrompere [to interrupt], finire [to finish/conclude], completare [to complete], continuare [to go on/carry on], andare avanti [to proceed], persistere [to persist]…

(3) a. Il vulcano ha **iniziato** a mostrare segni di attività in Aprile. [The volcano began showing signs of activity in April.]

b. Ho **continuato** a leggere quell'articolo per tutto il giorno. I went on reading that article for the whole day.

D.3.4.1.5 I_ACTION. In the examples, we report the I_ACTION events in bold face and the event arguments, underlined.

a) cercare [to try], provare [to demonstate]:

(4) Compagnie coma la Microsft stanno **cercando** di monopolizzare il mercato. [Companies like Microsoft are trying to monopolize the market.]

b) investigare, indagare [to investigate] ricercare [to search]:

(5) Una nuova task force ha inizato a **indagare** sull'omicidio di Castellari. [A new task force has begun to investigate on the murder of Castellari.]

c) ritardare [to delay], postporre [to postpone], ostacolare [to hinder]:

(6) Israele chiederà agli Stai Uniti di **ritardare** l'attacco contro l'Iraq. [Irael will ask the United States to delay the attack against Iraq.]

d) evitare [to avoid], impedire, prevenire [to prevent], cancellare [to cancel]:

(7) La Questura di Livorno ha **impedito** lo svolgimento della manifestazione di Forza Nuova indetta per il 10 Febbraio. [The police prevented the manifestation of Forza Nuova for February, 10th.]

e) chiedere [to ask], ordinare [to order], persuadere [to persuade], comandare [to command], richiedere [to require]:

(8) Le autorità hanno **richiesto** la massima collaborazione da parte dei mezzi di informazione. The authorities required the maximun collabortion by the media.

f) promettere [to promise], offrire [to offer], assicurare [to assure], proporre [to propose], accordarsi [to agree].

g) giurare [to swear].

h) nominare [to appoint], eleggere [to elect], dichiarare [to declare], proclamare [to proclaim].

D.3.4.1.6 I_STATE. The I_STATE events are events in bold face and the event arguments, underlined.

a) credere [to believe], pensare [to think], immaginare [to imagine], essere sicuro [to be sure], sospettare [to suspect], autorizzare [to allow].

(9) **Crediamo** che le sue parole non abbiano distratto il pubblico da quello che è accaduto. [We believe that his words have not distracted the public from what is happening].

b) sembrare [to seem], desiderare [to desire], bramare [to crave], auspicare [to foretell].

(10) Il governo italiano ha **auspicato** un'intesa in tempi rapidi. [The Italian government foretells a quick agreement.]

c) sperare [to hope], aspirare [to aspire], decidere [to decide].

(11) **Sperano** che i residenti rientreranno nelle loro case una volta cessato l'allarme. [They hope the inhabitants will go back to their houses once the alarm has ceased.]

d) temere [to fear], odiare [to hate], essere preoccupato [to be worried], aver paura [to be afraid], spaventarsi [to get frightened].

(12) **Temevano** per la loro incolumità. [They were afraid for their safety.]

e) aver bisogno, necessitare [to need].

f) dovere [to have to / must], potere [can/may/might], volere [to want], sapere [to know], essere in grado di, riuscire [to be able to].

D.3.4.1.7 STATE: this value is also treated as a value for the attribute @type. Only a susbset of states are annotated, namely:

a) states that are identifiably changed over the course of the document being marked up. In these and the following examples the markable state is in bold face;

(13) a. Numerosi punk sono tutt'ora **agli arresti**. [Lots of punks are still under arrest.]

b. Si deve guardare agli andamenti economici, in modo da portere correzzioni dove **necessario**. [It is necessary to observe the economic developement, in order to bring corrections where necessary.]

b) states that are directly related to a temporal expression;

(14) Silvio Berlusconi è stato il **Presidente** del Consiglio negli ultimi 5 anni. [Silvio Berlusconi was the Italian Prime Minister during the last 5 years.]

c) states that are introduced by an I_ACTION, an I_STATE, or a REPORTING event. States are in square brackets while the introductory event is underlined;

(15) a. Una partecipazione garantita dalla **presenza** dei nostri ministri. [A participation guranteed by the presence of our ministers.]

b. Ha dichiarato che è un **bugiardo**. [He declared that he is a liar.]

d) predicative states the validity of which is dependent on the document creation time (DCT) even if it is not explicitly stated.

(16) Più di 2.000 soldati italiani sono **in Afghanistan**. [More that 2 000 Italian soldiers are in Afghanistan.]

D.3.4.1.8 OCCURRENCE:

(17) Il patrimonio dell'Assofondi è **cresciuto**. [The estate of Assofondi grew.]

(18) I ministri dei 150 Paesi se ne **tornano** in patria. [The ministers of the 150 countries are going back home.]

D.3.4.2 The attribute @tense

This attribute can have values PRESENT, PAST, FUTURE, or NONE. The values assigned to this attribute mirror the highly-surface based character of *ISO-TimeML*. The values presented are based on classical tense distinctions in Italian. It is important to stress the fact that on the level of general temporal reference there are no major differences between Italian and English and also among other Indo-European languages. In Table D.1, correspondences between the classical grammatical tense classification system and the *ISO-TimeML* values are presented:

Table D.1 —Tense classification

Classical Grammatical Tense Classification	ISO-TimeML values
Presente Semplice	PRESENT
Passato Composto	PRESENT
Imperfetto	PAST
Passato Semplice	PAST
Trapassato	PAST
Piucchepperfetto (or Trapassato Prossimo)	PAST
Futuro Semplice	FUTURE
Futuro Composto	FUTURE

D.3.4.3 The attribute @aspect

Similar to tense, it captures standard distinctions in the grammatical category of semantic aspect or viewpoint, i.e. the way in which an event is presented. It is a surface attribute. It can have values PROGRESSIVE, PERFECTIVE, IMPERFECTIVE, or NONE. With respect to English, Italian does not have a clearcut morphological distinction to code semantic aspect though some default viewpoint values can be identified for the various tense forms. In section D.6 some examples on aspect annotation for Italian are presented.

D.3.4.4 The attribute @mood

D.3.4.4.1 This attribute captures the mood of the event. This attributes applies only for events realized by verbs. It can have the following values:

D.3.4.4.2 CONDITIONAL: it signals the conditional mood which in Italian is realized by the morphological inflection on the verb. It is used to speak of an event whose realization is dependent on a certain condition, particularly, but not exclusively, in conditional clauses.

(19) **Mangerei** del pesce. [I would like to eat some fish.]

D.3.4.4.3 SUBJUNCTIVE: it has several uses in independent clauses. This mood is required for certain types of dependent clauses.

(20) Voglio che tu te ne **vada**. [I want you to go away.]

D.3.4.4.4 IMPERATIVE: it expresses direct commands or requests. It is also used to signal a prohibition, permission or any other kind of exhortation.

(21) **Taci**. [Shut up.]

D.3.4.4.5 NONE: it is the deafult value and corresponds to the indicatve mood. In addition, this value is used when no inflectional morphology is present.

D.3.4.5 The attribute @vForm

It encodes information for non-finite/non-tensed verb forms. Its values are INFINITIVE, PARTICIPLE, GERUNDIVE and NONE.

D.3.4.6 The attribute @modality

It is used to convey the different degrees of modality nature of an event, mainly epistemic and deontic. Due to the fact that it is not an easy task to recognize the correct modality value of an event, it has been decided that this attribute is to be fulfilled in presence of the modal verbs "*dovere*", "*potere*", "*volere*" and modal periphrases. Its values will be represented by the modal verb (or modal periphrasis) as in the following example.

(22) I profughi **devono** abbandonare le loro case.

```
<EVENT
 xml:id="e1"
 target="token2"
 pred="DOVERE"
 type="STATE"
 class="I_STATE"
 tense="PRESENT"
 aspect="NONE"
 pos="VERB"
 mood="NONE"
 vform="NONE"
 modality="DOVERE"/>
```

D.4 The <SIGNAL> element

Readers are referred to A.2.3 for definitions and instructions on annotation. However, for Italian, it is necessary to consider the proper annotation of those SIGNALs which are realized by complex prepositions of the kind "alle", "dalle", "dal", "del", "sul", "al"…, where a definite article is merged with a preposition ("al=a+il"). In such cases, the annotation must be conducted as illustrated below:

dalle 3 di oggi. [from today at 3 o'clock]

```
<SIGNAL
  xml:id="s1"
  target="token0"
  pred="DAL"/>

<TIMEX3
  xml:id="t1"
  target="token1, token2, token3"
  pred="3_DI_OGGI"
  type="DATE"
  class="2006-12-20T15:00"/>
```

D.5 The link tags

See A.3 for definitions and instructions on the annotation of the four link tags.

D.6 Informative: Examples of tense, aspect and mood annotation in Italian

In this clause, some rules for annotating tense, aspect and mood in Italian are presented.

a) Events realized by finite verb forms:

1) tense= “PRESENT”

   i) gioca aspect= IMPERFECTIVE mood= NONE

   ii) sta giocando aspect= PROGRESSIVE mood= NONE

   iii) ha mangiato aspect= PERFECTIVE mood= NONE

   iv) é mangiato aspect = IMPERFECTIVE mood= NONE

   v) (che) mangi aspect= IMPERFECTIVE mood = SUBJUNCTIVE

2) tense= “PAST”

   i) giocò aspect= PERFECTIVE mood= NONE

   ii) ha giocato aspect= PERFECTIVE mood= NONE

   iii) ebbe l'abitudine di giocare aspect= PERFECTIVE mood= NONE

   iv) fu mangiato aspect= PERFECTIVE mood= NONE

   v) è stato mangiato aspect= PERFECTIVE mood= NONE

   vi) (che) abbia mangiato aspect= PERFECTIVE mood = SUBJUNCTIVE

   vii) aveva giocato aspect= PERFECTIVE mood= NONE

   viii) ebbe giocato aspect= PERFECTIVE mood= NONE

   ix) era stata mangiata aspect= PERFECTIVE mood= NONE

   x) (che) avesse mangiato aspect= PERFECTIVE mood = SUBJUNCTIVE

   xi) giocava aspect= IMPERFECTIVE mood= NONE

xii) stava giocando aspect= PROGRESSIVE mood= NONE

xiii) aveva l'abitudine di giocare aspect= IMPERFECTIVE mood= NONE

xiv) era mangiata aspect= IMPERFECTIVE mood= NONE

xv) (che) mangiasse aspect= IMPERFECTIVE mood = SUBJUNCTIVE

3) tense= “FUTURE”

i) giocherà aspect= PERFECTIVE mood= NONE

ii) avrà giocato aspect= PERFECTIVE mood= NONE

iii) sarà mangiata aspect= PERFECTIVE mood= NONE

iv) starà mangiado aspect= PROGRESSIVE mood= NONE

4) tense= “NONE”

i) mangerebbe aspect= IMPERFECTIVE mood= CONDITIONAL

ii) sarebbe mangiato aspect= IMPERFECTIVE mood= CONDITIONAL

iii) avrebbe mangiato aspect= PERFECTIVE mood= CONDITIONAL

iv) sarebbe stato mangiato aspect= PERFECTIVE mood= CONDITIONAL

v) starebbe mangiando aspect= PROGRESSIVE mood= CONDITIONAL

vi) mangia aspect= NONE mood= IMPERATIVE

b) Events realized by non-finite verb forms:

5) tense= “NONE” vform=“INFINITIVE” mood= NONE

i) giocare aspect= NONE - aver giocato aspect= PERFECTIVE

ii) stare giocando aspect= IMPERFECTIVE_PROGRESSIVE

6) tense= “NONE” vform=“GERUND” mood= NONE

i) giocando aspect= NONE - avendo giocato aspect= PERFECTIVE

7) tense= “NONE” vform=“PARTICIPLE” mood= NONE

i) giocante aspect= NONE

ii) giocato aspect= PERFECTIVE

NOTE Non-finite verb forms do not have automonous temporal reference.

c) Events realized by adjectives, nouns or prepositional phrases:

8) tense= “NONE” aspect=“NONE” mood= “NONE” vForm = “NONE”

D.7 Sample of Italian annotation

La Repubblica 30/01/1985

La Fiom contesta le scelte dell'Flm. I DELEGATI RESPINGONO L'ACCORDO CORNIGLIANO.

GENOVA - L'assemblea dei lavoratori Italsider di Cornigliano (erano presenti duemila operai) ha sostanzialmente contestato l'accordo raggiunto venerdì scorso tra la Finsider e la Flm nazionale e regionale, in base al quale lo stabilimento genovese riprenderà a produrre dal primo maggio con 1600 addetti e sarà gestito da una società pubblica (Nuova Italsider, Dalmine e Acciaierie di Piombino), in attesa dei privati. I delegati della lega Fiom di Cornigliano e dell' “Oscar Senigallia”, in particolare, come già il consiglio di fabbrica, hanno attaccato la Flm nazionale e regionale accusandole di averli esclusi dalle trattative, e non hanno firmato la bozza di intesa.

```
<ISO-TimeML>
 <TIMEX3
   xml:id="t2"
   target="#range(token2, token4)"
   pred="30/01/1985"
   functionInDocument="PUBLICATION_TIME"
   temporalFunction="false"
   type="DATE"
   value="1985-01-30"/>
 <EVENT
   xml:id="e1"
   target="#token27"
   pred="PRESENTE"
   type="STATE"
   class="STATE"
   tense="PAST"
   aspect="PERFECTIVE"
   pos="ADJECTIVE"
   vform="NONE"
   mood="NONE"
   polarity="POS"/>
 <EVENT
   xml:id="e2"
   target="#token32"
   pred="CONTESTARE"
   type="PROCESS"
   class="I_ACTION"
   tense="PAST"
   aspect="PERFECTIVE"
   pos="VERB"
   vform="NONE"
   mood="NONE"
   polarity="POS"/>
 <EVENT
   xml:id="e3"
   target="#token34"
   pred="ACCORDO"
   type="TRANSITION"
   class="OCCURRENCE"
   tense="NONE"
   aspect="NONE"
   pos="NOUN"
   vform="NONE"
   mood="NONE"
   polarity="POS"/>
 <EVENT
```

```
  xml:id="e4"
  target="#token35"
  pred="RAGGIUNGERE"
  type="TRANSITION"
  class="I_ACTION"
  tense="NONE"
  aspect="PERFECTIVE"
  pos="VERB"
  vform="PARTICIPLE"
  mood="NONE"
  polarity="POS"/>
<TIMEX3
  xml:id="t3"
  target="#token36 #token37"
  pred="VENERDI'_SCORSO"
  anchorTimeID="t2"
  functionInDocument="NONE"
  temporalFunction="false"
  type="DATE"
  value="1985-01-25"/>
<EVENT
  xml:id="e5"
  target="#token54"
  pred="RIPRENDERE"
  type="TRANSITION"
  class="ASPECTUAL"
  tense="FUTURE"
  aspect="PERFECTIVE"
  pos="VERB"
  vform="NONE"
  mood="NONE"
  polarity="POS"/>
<EVENT
  xml:id="e6"
  target="#token56"
  pred="PRODURRE"
  type="PROCESS"
  class="OCCURRENCE"
  tense="NONE"
  aspect="NONE"
  pos="VERB"
  vform="INFINITIVE"
  mood="NONE"
  polarity="POS"/>
<SIGNAL xml:id="s1" target="#token57" pred="DA"/>
<TIMEX3
  xml:id="t4"
  target="#token58 #token59"
  pred="PRIMO_MAGGIO"
  anchorTimeID="t2"
  functionInDocument="NONE"
  temporalFunction="false"
  type="DATE"
  value="1985-05-01"/>
<EVENT
  xml:id="e7"
  target="#token65"
  pred="GESTIRE"
  type="PROCESS"
  class="OCCURRENCE"
  tense="FUTURE"
  aspect="PERFECTIVE"
  pos="VERB"
  vform="NONE"
  mood="NONE"
  polarity="POS"/>
<EVENT
  xml:id="e29"
  target="#token77 #token78"
  pred="IN_ATTESA"
  type="STATE"
  class="STATE"
  tense="NONE"
  aspect="NONE"
  pos="PREPOSITION"
  vform="NONE"
  mood="NONE"
  polarity="POS"/>
<SIGNAL xml:id="s2" target="#token84" pred="GIA'"/>
<EVENT
  xml:id="e8"
  target="#token90"
  pred="ATTACCARE"
  type="PROCESS"
  class="OCCURRENCE"
  tense="PAST"
  aspect="PERFECTIVE"
  pos="VERB"
  vform="NONE"
  mood="NONE"
  polarity="POS"/>
<EVENT
  xml:id="e9"
  target="#token96"
  pred="ACCUSARE"
  type="PROCESS"
  class="I_ACTION"
  tense="NONE"
  aspect="IMPERFECTIVE"
  vform="GERUND"
  pos="VERB"
  mood="NONE"
  polarity="POS"/>
<EVENT
  xml:id="e10"
  target="#token99"
  pred="ESCLUDERE"
  type="TRANSITION"
  class="I_ACTION"
  tense="NONE"
  aspect="PERFECTIVE"
  vform="INFINITIVE"
  pos="VERB"
  mood="NONE"
  polarity="POS"/>
<EVENT
  xml:id="e11"
  target="#token101"
  pred="TRATTATIVA"
  type="TRANSITION"
```

```
 class="OCCURRENCE"
 tense="NONE"
 aspect="NONE"
 pos="NOUN"
 mood="NONE"
 polarity="POS"/>
<EVENT
 xml:id="e12"
 target="#token105"
 pred="FIRMARE"
 type="TRANSITION"
 class="OCCURRENCE"
 tense="PAST"
 aspect="PERFECTIVE"
 pos="VERB"
 vform="NONE"
 mood="NONE"
 polarity="NEG"/>
<TLINK
 xml:id="l11"
 origin="USER"
 relType="BEFORE"
 relatedToTime="#t2"
 timeID="#t3"/>
<TLINK
 xml:id="l12"
 origin="USER"
 relType="INCLUDES"
 relatedToTime="#t2"
 timeID="#t5"/>
<TLINK
 xml:id="l13"
 origin="USER"
 relType="BEFORE"
 relatedToTime="#t4"
 timeID="#t5"/>
<TLINK
 xml:id="l14"
 eventID="#e4"
 origin="USER"
 relType="IS_INCLUDED"
 relatedToTime="#t3"/>
<TLINK
 xml:id="l15"
 eventID="#e2"
 origin="USER"
 relType="AFTER"
 relatedToEvent="#e4"/>
<TLINK
 xml:id="l19"
 eventID="#e6"
 origin="USER"
 relType="BEGUN_BY"
 relatedToTime="t4"
 signalID="s1"/>
<TLINK
 xml:id="l20"
 eventID="#e7"
 origin="USER"
 relType="BEGUN_BY"
```

```
  relatedToTime="#t4"
  signalID="s1"/>
 <TLINK
  xml:id="l21"
  eventID="#e29"
  origin="USER"
  relType="DURING"
  relatedToEvent="#e7"/>
 <TLINK
  xml:id="l27"
  eventID="#e11"
  origin="USER"
  relType="BEFORE"
  relatedToTime="#t3"/>
 <TLINK
  xml:id="l30"
  eventID="#e2"
  origin="USER"
  relType="BEFORE"
  relatedToTime="#t2"/>
 <TLINK
  xml:id="l31"
  eventID="#ei8"
  origin="USER"
  relType="BEFORE"
  relatedToTime="#t2"/>
 <SLINK
  xml:id="l1"
  eventID="#e2"
  origin="USER"
  relType="FACTIVE"
  subordinatedEvent="e3"/>
 <SLINK
  xml:id="l3"
  eventID="#e9"
  origin="USER"
  relType="FACTIVE"
  subordinatedEvent="e10"/>
 <SLINK
  xml:id="l4"
  eventID="#e10"
  origin="USER"
  relType="FACTIVE"
  subordinatedEvent="#e11"/>
</ISO-TimeML>
```

Annex E
(informative)

Temporal annotation of predicates in Korean

E.1 Introduction

In Korean, grammatical morphemes that are agglutinated to predicate stems carry information on tense, aspect, modality, mood, and sentence type. This annex focuses on ways of annotating temporal and event-related information conveyed by such grammatical morphemes as well as predicate stems themselves.

NOTE These morphemes are affixes, often called ***endings***, as opposed to ***particles*** that are agglutinated to nouns and some other parts of speech such as adverbs, conveying information on grammatical functions or semantic roles, but treated as a categorized part of speech called ***josa*** literally meaning *auxiliary part of speech*.

As in other languages, adverbials and nouns also carry time and event-related information. But the annotation of these expressions is not treated here, for they can be annotated according to the general annotation guidelines as specified in *ISO-TimeML*.

E.2 Basic references

This annex heavily relies on the following references:

- CHANG, SUK-JIN (1996), *Korean,* John Benjamins Publishing Co., Amsterdam.
- LEE, IKSOP, and RAMSEY, S. (2000), *The Korean Language*, State University of New York Press, Albany.
- SOHN, HO-MIN (1999), *The Korean Language*, Cambridge University Press, Cambridge.

These are general introductory books on the Korean language and its grammar for foreign readers, published most recently. The Englsih names of grammatical terms follow these references in general.

E.3 Morpholoy of Korean predicates

There are three types of predicates in Korean:

a) verbs (e.g. 가-ㅆ-다 ka-ss-ta 'go+PAST'),

b) adjectives (e.g. 예뻐-ㅆ-다 yeppe-ss-ta 'be+PAST pretty'), and

c) nominal predicates (e.g. 선생-이-었-다 sensayngn-i-ess-ta 'be+PAST a mountain').

Adjectives in most western languages require a copular verb to function as predicates. In Korean, however, adjectives as in (b) above morpho-syntactically behave like verbs, each concatenating with a sequence of agglutinative grammar morphemes. Likewise, nominal predicate stems like 선생－이 sensayng-i- 'teacher + Copula' concatenate with grammatical bound morphemes or so-called endings just like genuine verbal or adjectival stems. Here are examples:

(1)

a.
미아가 잤다 (verb)
mia-ka ca-ss-ta
Mia-NOM sleep-PAST-DECL
“Mia slept”

b.
미아가 예뻤다 (adjective)
mia-ka yey.pp-ess-ta
Mia-NOM pretty-PAST-DECL
“Mia was pretty”

c.
미아가 선생이었다 (nominal predicate)
mia-ka sensayng-i-ess-ta
Mia-NOM teacher-COP-PAST-DECL
“Mia was a teacher”

There are several different ways of romanizing the Korean characters in written text. The Yale romanization is adopted here mainly because it is considered as being able to transcribe the morpho-syntactic properties of Korean words and phrases in an unambiguous manner, thus being accepted in most of the linguistic journals that are published in English. In the Yale romanization system adopted here, the dot indicates a syllable break and the hyphen a morphological break. The verb 갔다 kass.ta 'go', for instance, consists of two syllables, while it consists of three morphemes, ka-ss-ta, STEM-PAST-DECL. Here is the first list of abbreviations for grammatical categories.

List of grammtical categories 1:

- COP stands for COPULAR bound morpheme,
- PAST stands for PAST tense morpheme, and
- DECL stands for DECLARATIVE sentence type morpheme.

Being an agglutinative language, Korean concatenates noun or predicate stems with a sequence of nominal particles or predicate endings, respectively, each of which is treated as a grammatical bound morpheme, while such a sequence can consist of several morphemes. Here is an example of verbal agglutination:

(2)

잡히시었겠습니다요
cap-hi-si-ess-keyss-swup-ni-ta-yo
STEM-PASS-SH-PAST-CONJEC-AH-IND-DECL-JOSA
“must have been caught"

This is a verbal stem 잡 cap- 'catch' followed by eight grammatical bound morphemes agglutianted as in the following order:

List of grammatical categories 2:

- PASS: passive suffix -hi
- SH: subject honorific suffix -si
- PAST: past tense morpheme -ess

- CONJEC: CONJECTURAL modal morpheme -keyss
- AH: addressee honorific morpheme -swup
- IND: INDICATIVE mood morpheme -ni
- DECL: declarative sentence type morpheme -ta
- JOSA: particular part of speech agglutinated to nouns, adverbs and other parts of speech. The JOSA 요 yo here is agglutinated to the whole verbal construction 'must have been caught', conveying pragmatic information that the speech type is addressee-honorific and polite, but informal and colloquial.

Some of these bound morphemes express tense, aspect, modality, mood or sentence type. The following examples illustrate their agglutinative ordering that express such temporal information.

(3)

만나-고 있 -었 -겠 -더 -라
stem-aspect-tense-modality-mood-sType
manna-ko iss-ess -keyss -te -ra
meet-BR PROG PAST CONJEC RETRO DECL
“must/may have been meeting"

(4)

예쁘 -더 -라
yeyppu -te -ra
pretty RETRO DECL
“I found her pretty"

There are several differences in whitespacing between the North and South Korean orthographies. One concerns the PROGRESSIVE morpheme. There is white space between 고 -ko and 있 -iss in the South Korean orthography, whereas there is no such white space between them in the North Korean orthography. With the orginal draft being written by a South Korean project leader of the document, the South Korean orthography is mainly followed in the current version of the annex. When these two differing orthographies are normalized in the future, it is expected to make proper modifications as needs arise.

List of grammatical categories 3:

- sType stands for the type of sentence-final sentence type morpheme.
- BR again stands for a bridge between a main verb and its auxiliary verb.
- PROG stands for PROGRESSIVE aspect.
- RETRO stands for RETROSPECTIVE mood.

Each of these agglutinative aspect and mood as well as tense morphemes will be discussed in the following clauses.

E.4 Temporal structure: informative

Time and tense are entities of two different types. Time is part of the world with a certain structure, while tense is a linguistic feature related to time. But the use of terms such as ***present***, ***future*** and ***past*** can be ambiguous: they can either be used in an ontological sense or in a grammatical sense.

To make this distinction clear, the ontological ***present***, ***future***, and ***past*** are written in the lower case, while PRESENT, FUTURE, and PAST, written in the upper case, are understood to be grammatical entities that serve as values of the attribute tense.

The temporal structure assumed here is a quadruple

$<T, I, n, R>$

where

T is a set of points of time;

I is a set of intervals of time;

n is the uniquely designated point of time in T, known as the present moment of time;

R is a set of temporal relations over T or I that include:

- the partial (precedence) relation,
- the overlap relation over intervals, and
- the neighborhood N of a point of time t in T such that $N(t)$ is an open interval that includes the point of time t.

On the basis of this structure, the following intervals are defined.

- The present (time) refers to an open interval $N(n)$, the neighborhood of the designated point of time n.
- The past (time) refers to an interval preceding the designated point of time n.
- The future (time) refers to an interval which is not preceded by nor identical with the designated point of time n.

According to this definition, the past and the future time interval each may overlap with the present time interval, $N(n)$. The end point of the past may be included in $N(n)$ and the beginning point of the future may also be in $N(n)$.

By introducing the notion of neighborhood, as discussed in Lee, K. (1998), the present moment of time can be understood as a temporal construct to define other temporal structures, whereas the linguistic expression like 지금 cikum "now" in Korean can be understood as referring to its neighborhood, namely $N(n)$. Such an interpretation allows the co-occurrence of 지금 cikum "now" with the PAST tensed verb as in 지금 도착하였다 cikum tochakha-ye−ss-ta (now arrive-PAST-DECL) "have arrived now".

E.5 Temporal annotation of non-Latin texts

The tense and other temporal features of an event are annotated according to *ISO-TimeML*. For texts written in characters other than Latin characters, as in English or French texts, however, the attribute @romanization may be introduced. This attribute specifies how non-Latin character texts like Korean texts are romanized. Here is an example:

(5)

미아가 지금 (now) 도착했다

```
<EVENT
  xml:id="e1"
  target="#token2"
  pred="ARRIVE"
  romanization="to.chak.hayss.ta"
  tense="PAST"
  vForm="sFINAL"/>
```

The datatype of the attribute @pred is CDATA, any non-empty sequence of characters, representing the information content of a string that it is anchored to. If desirable for some other purposes, this content may also be represented by a logical form like ARRIVE(mia).

ISO-TimeML also provides guidelines for annotating temporal adverbs and nominal expressions. But here, to focus on the annotation of temporal features associated with events, other temporal expressions like temporal adverbs may sometimes be annotated simply with their English equivalents in parentheses in the text.

As will be discussed presently, the interpretation of tense in Korean depends on the type of the attribute @vForm which completes a verb or adjective stem as a word form.

The value of @vForm -다 -ta is a DECLARATIVE verbal ending. But, for the purpose of interpreting tense, it is sufficient to specify it with a more general value sFinal, standing for sentence-final verbal endings.

E.6 Tense

E.6.1 Tense markers

E.6.1.1 Specification of the two PAST tense markers

Despite its allomorphic variants, the attribute @tense in Korean has a single value, namely PAST, for a single tense morpheme –ㅆ –ss. This form has, however, three other variants: -ess, -ass,and -yess, the choice of which depends on the syllable structure of a verbal stem to which each tense morpheme is agglutinated (See Lee, K. 1999). The PAST tense marker has a doubled form -ess.ess, being treated as PAST-PAST, PAST-PERFECTIVE, PLUPERFECT or REMOTE PAST. In this annex, it is specified with @tense="PAST" and @aspect="PERFECTIVE".

E.6.1.2 List of PAST tense endings

— PAST: -ㅆ(-ss), -었(-ess), -았(-ass), -였(-yess)

— PAST-PAST: -ㅆ었(-ss-ess), -었었(-ess-ess), -았었(-ass-ess), -였었(-yess-ess)

E.6.1.3 Examples

(6)

어제 나는 미아를 만났다
ecey na-nun mia-rul manna-ss-ta yesterday
I-TOP Mia-ACC meet-PAST-DECL
“Yesterday I met Mia”

(7)

전에 나는 미아를 만났었다
ceney na-nun mia-rul manna-ss.ess-ta
before I-TOP Mia-ACC meet-PASTperfective-DECL
“I had met Mia before”

(8)

지금 나는 물을 마신다
cikum na-nun mwul-ul masi-n-ta
now I-TOP water-ACC drink-IND-DECL
“Now I drink (am drinking) water"

(9)

모레 나는 미아를 만난다
morey na-nun Mia-rul manna-n-ta the day after tomorrw
I-TOP Mia-ACC meet-IND-DECL
“The day after tomorrow I'll meet Mia”

List of grammatical categories 4:

- TOP stands for TOPICALIZER;

- ACC stands for ACCUSATIVE. Both TOP and ACC are called GYEOK JOSA.b Both Subject and Object can be marked by TOP, while Object is marked by ACC;

- IND is here treated as an INDICATIVE mood morpheme [see Sohn (1999)].

E.6.1.4 Adnominal forms

Besides the finite past marker -ㅆ -ss and its variants, there are a past tense morpheme -ㄴ -n and its variant - 은 -un that are agglutinated to verbal stems, thus forming ADNOMINAL verbal forms. Here are examples:

(10)

a.
미아가 마신 물
mia-ka masi-n muwl
Mia-NOM drink-pastADN water
“water that Mia drank”

b.
미아가 먹은 사과
mia-ka mek-un sakwa
Mia-NOM eat-pastADN apple
“apple that Mia ate”

List of grammatical terms 5:

- pastADN indicates a PAST ADNOMINAL verbal ending, that is, a bound morpheme agglutinated to a verbal stem forming a PAST ADNOMINAL form.

If an adjective or a nominal predicate ends in an adnoun-forming grammatical morpheme -ㄴ -n or its variant -은 -un, then it does not refer to a PAST, but to a PRESENT state: for example, 예쁜 소녀 yeppu-n sonye “pretty girl”. Besides these adnoun-forming morphemes, there are three other bound morphemes that create adnominal forms.

- The adnominalizer -는 -nun is agglutinated to a verbal stem: e.g. 마시-는 masi-nun “drinking" or 먹-는 mek-nun “eating”.

- The adnominalizer -ㄹ -l and its variants -를 -rul and -을 -ul are agglutinated to both a verbal and an adjectival stem, as illustrated by: 마시-ㄹ masi-l “to drink", 흐리-ㄹ huri-l “to be cloudy”.

- The adnominalizer -던 -ten are also agglutinated to both a verbal and an adjectival stem: e.g. 마시-던 masi-ten “used to drink", 예쁘-던 yeppu-ten “used to be pretty”, 마시-었-던 masi-ess-ten “used to have drunk” or 예뻤-던 yepp.ess-ten “used to have been pretty”.

E.6.2 Annotation guidelines for the attribute @tense

E.6.2.1 Verbal forms

Annotation depends on the surface information of markables only. At least for implementation purposes, the task of annotation should be carried out purely on surface information through routine manner. The attribute @tense is thus annotated independently of its contextual information, although its interpretation varies contextually:

- PAST: If a verb or an adjective word form contains a tense marker -었 -ess or its morphological variants, then it is annotated with @tense="PAST".

- PAST PERFECTIVE: If a verb or an adjective word form contains a tense marker -었었 -ess.ess or its morphological variants, then it is annotated with @tense="PAST" with the additional specification of @aspect="PERFECTIVE".

- NONE: If a verb or an adjective contains no tense marker, then it is annotated with @tense="NONE".

The so-called INDICATIVE mood markers -는 -nun or -ㄴ n occur with verbs which are interpreted as referring to an event in the present or future, as in 다음 주 떠난다 (next week leave-IND-DECL). But even in such a case, the value of the attribute @tense is assigned NONE.

The verbal sentence-pre-final morpheme -n/-nun/-nu is often treated as PRESENT tense marker. See Chang (1996: pp. 118). But Sohn (1999) treats it as the (NON- PAST) INDICATIVE mood marker, for it occurs only with the sentence-final DECLARATIVE ending -ta or the sentence-final APPERCEPTIVE DECLARATIVE sentence ending –kuna, when the stem is a type of genuine (action) VERB which is not an adjectival stem.

The ending -keyss is treated as a CONJECTURAL modal marker. It can be used with the PAST tense marker, as in manna-ss-keyss-ta "must/might have met".

E.6.2.2 Adnominal forms

Adnominal forms are annotated as follows.

- If a verb ends in an adnoun-forming grammatical morpheme -ㄴ -n or its variant -은 -un, then it is annotated with @tense="PAST".

- If an adjective or a nominal predicate ends in an adnoun-forming morpheme -ㄴ -n or its variant -은 -un, then it is annotated with @tense="NONE".

- If a predicate, namely a verb, adjective or nominal predicate, ends in -ㄹ -l or -을 -ul, then it is annotated with @tense="NONE" and @modal="CONJECTURAL", as will be discussed presently.

- If a predicate, namely a verb, adjective or nominal predicate, ends in -던 - ten, then it is annotated with @tense="NONE" and @mood="RETROSPECTIVE", as will be discussed presently.

- If a predicate ends in －었-던 -ess-ten, then it is annotated with @tense="PAST" @mood="RETROSPECTIVE".

E.6.3 Contextual interpretation of tense

E.6.3.1 Four types of tense

Each specification of the attribute @tense is interpreted differently depending on its context. Consider the following <EVENT>-annotated complex example:

(11)

미아가 사온 사과를 먹고 잤다고 용이 말했다
mia-ka sa.o-n sakwa-rul mek-ko ca-ss-ta-ko yong-i mal.hayss-ta
Mia-NOM buy-ADNOM apple-ACC eat-CONJ sleep-PAST-DECL-COMP
Yong-NOM say-PAST-DECL
“Yong said that he ate an apple which Mia had bought and slept”

```
<s>
 <seg type="eojeol" xml:id="t1">미아가</seg>
 <seg type="eojeol" xml:id="t2">사온</seg>
 <seg type="eojeol" xml:id="t3">사과를</seg>
 <seg type="eojeol" xml:id="t4">먹고</seg>
 <seg type="eojeol" xml:id="t5">잤다고</seg>
 <seg type="eojeol" xml:id="t6">용이</seg>
 <seg type="eojeol" xml:id="t7">말했다</seg>
</s>
<EVENT
  xml:id="e1"
  target="#t2"
  pred="BUY"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  vForm="ADNOMINAL"
  tense="PAST"/>
<EVENT
  xml:id="e2"
  target="#t4"
  pred="EAT"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  vForm="CONJUNCTIVE"
  tense="NONE"/>
<EVENT
  xml:id="e3"
  target="#t5"
  pred="SLEEP"
  class="OCCURRENCE"
  type="PROCESS"
  pos="VERB"
  vForm="COMPLEMENT"
  tense="PAST"/>
<EVENT
  xml:id="e4"
  target="#t7"
  pred="SAY"
  class="OCCURRENCE"
  type="PROCESS"
  pos="VERB"
  vForm="sFINAL"
  tense="PAST"/>
```

Each of the four <EVENT>s here is specfied with @tense and its value. But each occurrence of the @tense values should be interpreted differently. There are, for instance, three <EVENT> elements identified as e1, e3 and e4 that are all specified with the same type of @tense specification @tense="PAST". But they are interpreted as having a different temporal relation with each other: the event e1 is understood to have occurred BEFORE e2 and e2 BEFORE e3.

To capture such a difference in temporal ordering, the interpretation of @tense in Korean may be described by a sub-featuring tense into:

— Absolute tense or simply tense,

— Embedded tense,

— Relative tense, and

— Inherited tense.

E.6.3.2 Absolute interpretation of tense

Consider the following:

(12)

어제 (yesterday) 나는 미아를 만났다
"Yesterday I met Mia"

```
<EVENT
    xml:id="e12"
    target="#token3"
    pred="MEET"
    class="OCCURRENCE"
    type="TRANSITION"
    pos="VERB"
    tense="PAST"
    vForm="sFINAL" />
```

(13)

전에 (before) 나는 미아를 만났다
"I had met Mia before"

```
<EVENT
    xml:id="e13"
    target="#token3"
    pred="MEET"
    class="OCCURRENCE"
    type="TRANSITION"
    pos="VERB"
    tense="PAST"
    aspect="PERFECTIVE"
    vForm="sFINAL" />
```

(14)

모레 (the day after tomorrow) 나는 미아를 또 만난다
“I'll meet Mia again the day after tomorrow”

```
<EVENT
      xml:id="e14"
      target="#token4"
      pred="MEET"
      class="OCCURRENCE"
      type="TRANSITION"
      pos="VERB"
      tense="NONE"
      vForm="sFINAL" />
```

(15)

미아는 예쁘다
“Mia is pretty”

```
<EVENT
      xml:id="e15"
      target="#token1"
      pred="PRETTY"
      class="STATE"
      type="STATE"
      pos="ADJECTIVE"
      tense="NONE"
      vForm="sFINAL" />
```

As will be discussed presently, the interpretation of @tense in Korean depends on the type of @vForm which completes a verb or adjective stem as a word form. Each occurrence of the tensed events referred to in the above examples is interpreted independently from any other event, although some of them are temporally related to the time period referred to by the temporal expressions like 어제 ecey "yesterday", 전에 ceney "before" or 모레 morey "the day after tomorrow". It should, however, be noted that the interpretation of a verb with the specificaiton @tense="NONE" may depend on a temporal expression, for it can be interpreted as being ambiguous, either referring to an event in the present or to an event in the future. Consider the following:

(16) 모레 (the day after tomorrow) 나는 미아를 만난다

“I'll meet Mia tomorrow”

```
<EVENT
      xml:id="e16"
      target="#token3"
      pred="MEET"
      class="OCCURRENCE"
      type="TRANSITION"
      pos="VERB"
      tense="NONE"
      vForm="sFINAL" />
```

Interpretation conditions for absolute tense:

Given an <EVENT> element specified with @vForm="sFINAL",

— PAST: If the tense attribute in that <EVENT> element is specified with the value PAST, then that event is interpreted as having occurred in the past.

— PASTperfective: If the @tense attribute in that <EVENT> element is specified with the value PAST, and if that <EVENT> element contains the specification @aspect="PERFECTIVE", then that event is interpreted as having occurred and also been completed in the past.

— NONE: If the value of the @tense attribute in that <EVENT> element is NONE, then that event is interpreted as occurring in the present.

But the present time interval can be extended to the future time, if the event is contextualized (particularly by a temporal expression referrring to the future).

In Korean, the present and future interpretations of verbal expressions are differentiated not by a tense value, but are specified by a temporal adverb like 모레 morey "the day after tomorrow" or a modal operator like the CONJECUTARAL –겠 -keyss. This particular modal morpheme has thus been treated as FUTURE tense marker or FUTURITY modal marker [Lee, C. (1987)]. Subordinating morphemes like the CONCESSIVE ending -지만 -ciman may also allow the absolute interpretation of tense, which will be discussed on another occasion.

E.6.3.3 Embedded Tense

E.6.3.3.1 Verbs of saying or asking and their complements

Verbs of saying or asking have tensed sentences as complements. Here are some examples:

(17)

a.
미국에 언제 갔냐고 미아가 물었다
mikuk-ey encey ka-ss-nya-ko mia-ka mwul-ess-ta
US-to when go-PAST-INTERRO-COMP mia-NOM ask-PAST-DECL
"Mia asked when Yong went to US"

b.
곧 간다고 용이 대답하였다
kot ka-n-ta-ko taytapha-yess-ta
soon go-IND-DECL-COMP answer-PAST-DECL
"Yong answered that (he) would go soon"

c.
그리곤 어제 떠났다
kuriko-n ecey ttena-ss-ta
and yesterday leave-PAST-DECL
"And then (he) left yesterday"

(18)=(17a)

```
<EVENT
    xml:id="e1"
    target="#token2"
    pred="GO"
    class="OCCURRENCE"
    type="TRANSITION"
    pos="VERB"
    vForm="COMP"
    tense="PAST" />
```

```
<EVENT
      xml:id="e2"
      target="#token4"
      pred="ASK"
      class="OCCURRENCE"
      type="TRANSITION"
      pos="VERB"
      vForm="sFINAL"
      tense="PAST" />

<TLINK
      eventID="#e1"
      signalID="#s1"
      relatedToEvent="#e2"
      relType="BEFORE"/>
```

(19)=(17b,c)

```
<EVENT
      xml:id="e3"
      target="#token1"
      pred="GO"
      class="OCCURRENCE"
      type="TRANSITION"
      pos="VERB"
      vForm="COMP"
      tense="NONE" />

<EVENT
      xml:id="e4"
      target="#token3"
      pred="ANSWER"
      class="OCCURRENCE"
      type="TRANSITION"
      pos="VERB"
      vForm="sFINAL"
      tense="PAST" />

<TLINK
      eventID="#e3"
      signalID="#s2"
      relatedToEvent="#e4"
      relType="AFTER"/>

<TIMEX3
      xml:id="t1"
      target="#token6"
      pred="YESTERDAY"
      type="DATE"
      value="2007-05-06/>

<EVENT
      xml:id="e5"
      target="#7"
      pred="LEAVE"
      class="OCCURRENCE"
      type="TRANSITION"
      pos="VERB"
      vForm="sFINAL"
      tense="PAST" />
```

```
<TLINK
    eventID="#e5"
    relatedToTime="#t1"
    relType="IS_INCLUDED"/>
```

- α is an <EVENT> element specified with @xml:id="ei" for some integer i and also with @ <vForm="COMP"/>,

- σ is a <SIGNAL> element specified with @xml:id="sk" for some integer k for @vForm="COMP" in α, and

- β is another <EVENT> element specified with @xml:id="ej" for some integer j that immediately follows both of the elements, α and σ,

  introduce:

  ```
  <TLINK eventID="#ei" relatedToEventID="#ej"/>
      specified with @relType="BEFORE", if α is specified with @tense="PAST",
      specified with @relType="AFTER" otherwise.
  ```

E.6.3.3.2 Interpretation of embedded tense

The interpretation of embedded tense is governed by its related <TLINK> and is almost obvious by the attribute @relType between the given event instances.

Interpretaiton conditions:

- If an event ei is related to another event ej and the type of their relation is AFTER, then ei is understood to occur after ej.

- But if this relation is BEFORE, ei is understood to have occurred before ej.

E.6.3.4 Relative tense

E.6.3.4.1 Adnominal endings

Adnominal verb endings and adnominal adjective endings both carry temporal information. Verbs and adjectives, however, have different sets of adnominal affixes, called predicate endings.

E.6.3.4.2 Adnominal endings for verbs

The following list shows how each of the adnominal endings for verbs is annotated:

- @tense="NONE" : -는 -nun

- @tense="PAST" : - ㄴ-n, -은 -un

- @modal="COJECTURAL" : -ㄹ -l, -을 -ul

- @tense="NONE" and @mood="RETROSPECTIVE" : -던 (-ten)

- @tense="PAST" and @aspect="RETROSPECTIVE" : -었던 (-ess.ten)

Here are examples:

(20)

미아가 만나는 남자를 순이가 사랑하였다
mia-ka manna-nun nam.ca-rul swuni-ka sa.rang.ha-yess-ta
mia-NOM meet-ADNOM male-ACC swuni-NOM love-PAST-DECL
“Swuni loved a man Mia meets/met”

(21)

미아가 만날 남자를 순이가 사랑하였다
mia-ka manna-l nam.ca-rul swuni-ka sa.rang.ha-yess-ta
mia-NOM meet-conjADNOM male-ACC swuni-NOM love-PAST-DECL
“Swuni loved a man Mia will/would meet”

(22)

미아가 만난 남자를 순이가 사랑하였다
mia-ka manna-n nam.ca-rul swuni-ka sa.rang.ha-yess-ta
mia-NOM meet-pastADNOM male-ACC swuni-NOM love-PAST-DECL
“Swuni loved a man Mia met”

(23)

미아가 만나던 남자를 순이가 사랑하였다
mia-ka manna-ten nam.ca-rul swuni-ka sa.rang.ha-yess-ta
mia-NOM meet-pastDurADNOM male-ACC swuni-NOM love-PAST-DECL
“Swuni loved a man Mia used to meet”

(24)

미아가 만났던 남자를 순이가 사랑하였다
mia-ka manna-ss-ten nam.ca-rul swuni-ka sa.rang.ha-yess-ta
mia-NOM meet-pastPerfADNOM male-ACC swuni-NOM love-PAST-DECL
“Swuni loved a man Mia had met”

E.6.3.4.3 Adnominal endings for adjectives and the nominal predicate

Adjectives have a different set of adnominal endings:

— tense=“NONE”: -ㄴ-n, -은 -un

— modal=“CONJECTURAL”: -ㄹ -l, -을 -ul

Examples are as follows:

(25)

미아가 돈많은 남자를 사랑하였다
mia-ka tonmanh-un namca-rul sarangha-yess-ta
mia-NOM money-much-ADNOM man-ACC love-PAST-DECL
“Mia loved a man who had a lot of money”

(26)

미아가 돈많을 남자를 사랑하였다
mia-ka tonmanh-ul namca-rul sarangha-yess-ta
mia-NOM money-much-conjecADNOM man-ACC love-PAST-DECL
“Mia loved a man who will/would have a lot of money”

(27)

미아가 돈많던 남자를 사랑하였다
mia-ka tonmanh-ten namca-rul sarangha-yess-ta
mia-NOM money-much-pastDurADNOM man-ACC love-PAST-DECL
“Mia loved a man who used to have a lot of money”

(28)

미아가 돈많았던 남자를 사랑하였다
mia-ka tonmanh-ass-ten namca-rul sarangha-yess-ta
mia-NOM money-much-pastPerfADNOM man-ACC love-PAST-DECL
“Mia loved a man who had had a lot of money”

E.6.3.4.4 Annotation of relative tense

(29)

미아가 사랑하는 남자를 순이가 사랑하였다
“Suni loved the man Mia loves/loved”

```
<EVENT
 xml:id="e1"
 target="#token1"
 pred="LOVE"
 class="STATE"
 type="STATE"
 pos="VERB"
 tense="NONE"
 vForm="ADNOMINAL"/>
<EVENT
 xml:id="e2"
 target="#token4"
 pred="LOVE"
 class="STATE"
 type="STATE"
 pos="VERB"
 tense="PAST"
 vForm="sFINAL"/>
<TLINK
  eventID="#e1"
  relatedToEvent="#e2"
  relType="SIMULTANEOUS"/>
```

(30)

미아가 사랑할 남자를 순이가 사랑하였다
“Suni loved the man Mia will/would love”

```
<EVENT
 xml:id="e1"
 target="#token1"
 pred="LOVE"
 class="STATE"
 type="STATE"
 pos="VERB"
 tense="FUTURE"
 vForm="ADNOMINAL"/>
<EVENT
 xml:id="e2"
 target="#token4"
 pred="LOVE"
 class="STATE"
 type="STATE"
 pos="VERB"
 tense="PAST"
 vForm="sFINAL"/>
<TLINK
  eventID="#e1"
  relatedToEvent="#e2"
  relType="AFTER"/>
```

(31)

미아가 사랑한 남자를 순이가 사랑하였다
“Suni loved the man Mia loved/had loved”

```
<EVENT
     xml:id="e1"
     target="#token1"
     pred="LOVE"
     class="STATE"
     type="STATE"
     pos="VERB"
     tense="PAST"
     vForm="ADNOMINAL"/>
<EVENT
     xml:id="e2"
     target="#token4"
     pred="LOVE"
     class="STATE"
     type="STATE"
     pos="VERB"
     tense="PAST"
     vForm="sFINAL"/>
<TLINK
     eventID="#e1"
     relatedToEvent="#e2"
     relType="BEFORE"/>
```

(32)

미아가 사랑하던 남자를 순이가 사랑하였다
“Suni loved the man Mia used to love”

```
<EVENT
      xml:id="e1"
      target="#token1"
      pred="LOVE(mia,x)"
      class="STATE"
      type="STATE"
      pos="VERB"
      tense="PAST"
      aspect="PERFECTIVE"
      vForm="ADNOMINAL"/>
<EVENT
      xml:id="e2"
      target="#token4"
      pred="LOVE(swuni,x)"
      class="STATE"
      type="STATE"
      pos="VERB"
      tense="PAST"
      vForm="sFINAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="BEFORE"/>
```

As stated earlier, the attribute @pred may have a predicate-logic-type logical form like "LOVE(mia,yong)" as in (32).

Sentences with adnominal adjectives can also be annotated in a similar manner.

(33)=(25)

미아가 돈많은 남자를 사랑하였다
“Mia love a man who has/had a lot of money”

```
<EVENT
      xml:id="e1"
      target="#token1"
      pred="RICH"
      class="STATE"
      type="STATE"
      pos="ADJECTIVE"
      tense="NONE"
      vForm="ADNOMINAL"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="LOVE"
      class="STATE"
      type="STATE"
      pos="VERB"
      tense="PAST"
      vForm="DECLARATIVE"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="SIMULTANEOUS"/>
```

(34)=(26)

미아가 돈많을 남자를 사랑하였다
“Mia loved a man who will/would have a lot of money”

```
<EVENT
      xml:id="e1"
      target="#token1"
      pred="RICH"
      class="STATE"
      type="STATE"
      pos="ADJECTIVE"
      tense="NONE"
      modal="CONJECTURAL"
      vForm="ADNOMINAL"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="LOVE"
      class="STATE"
      type="STATE"
      pos="VERB"
      tense="PAST"
      vForm="sFINAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="SIMULTANEOUS"/>
```

(35)=(27)

미아가 돈많던 남자를 사랑하였다
“Mia loved a man who used to have a lot of money”

```
<EVENT
      xml:id="e1"
      target="#token1"
      pred="RICH"
      class="STATE"
      type="STATE"
      pos="ADJECTIVE"
      tense="PAST"
      aspect="DURATIVE"
      vForm="ADNOMINAL"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="LOVE"
      class="STATE"
      type="STATE"
      pos="VERB"
      tense="PAST"
      vForm="sFINAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="BEOFRE"/>
```

(36)=(28)

미아가 돈많았던 남자를 사랑하였다

“Mia loved a man who had had a lot of money”

```
<EVENT
      xml:id="e1"
      target="#token1"
      pred="RICH"
      class="STATE"
      type="STATE"
      pos="ADJECTIVE"
      tense="PAST"
      aspect="PERFECTIVE"
      vForm="ADNOMINAL"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="LOVE"
      class="STATE"
      type="STATE"
      pos="VERB"
      tense="PAST"
      vForm="sFINAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="BEFORE"/>
```

If a predicate is adnominal, then its tense may also be relativized with respect to the tense of the main clause or the clause in which the adnominoal or relativized section is embedded. For this case, a relevant <TLINK> is introduced. Otherwise, the attribute @tense of an adnominal clause carries its given value, being interpreted in an absolute sense, such that it is anchored to the designated time of origin or utterance. Consider sentence (20), which is repeated here:

(37=20)

미아가 만나는 남자를 순이가 사랑하였다
mia-ka manna-nun nam.ca-lul swuni-ka sa.rang.ha-yess-ta
mia-NOM meet-ADNOM male-ACC swuni-NOM love-PAST-DECL
“Swuni loved a man Mia meets/met”

Here, the time of Mia's meeting a man can be anchored to the present time, instead of taking place in the past. This becomes clearer, if the temporal adverb 지금 cikum 'now' is added to it or make the main verb carry PAST PERFECTIVE.

(38)

미아가 지금 만나는 남자를 순이가 사랑하였다
mia-ka cikum manna-nun nam.ca-rul swuni-ka sa.rang.ha-yess-ta
mia-NOM now meet-ADNOM male-ACC swuni-NOM love-PAST-DECL
“Swuni loved a man Mia meets now”

(39)

미아가 만나는 남자를 순이가 사랑하였다
mia-ka meet-nun nam.ca-rul swuni-ka sa.rang.ha-yess.ess-ta
mia-NOM meet-ADNOM male-ACC swuni-NOM love-pastPerfective-DECL
“Swuni loved a man Mia meets/met”

Because of such an ambiguity of adnominal tense, a <TLINK> is optional.

E.6.3.4.5 Annotation guidelines

Given two events, e1 and e2: if an <EVENT> element associated with e1 contains @vForm="ADNOMINAL", then e1 may be related to e2 by <TLINK> such that the @tense of e1 is relativized with respect to that of e2 as below:

- if the @tense of e1 is NONE, then @relType="SIMULATANEOUS" or @relType="AFTER", and
- if the @tense of e1 is PAST, then @relType="BEFORE".

Interpretation conditions:

The conditions for interpreting the above annotations need not be stated here, for they are provided by the general guidelines.

E.6.3.5 Inherited tense

E.6.3.5.1 Conjunctive endings

Verbal expressions with a CONJUNCTIVE ending like -고 -ko often lack a tense marker, but inherit the attribute @tense value from the main verbal expression at the end of a sentence. Here is an example:

(40)

어제 밤 사과를 먹고 차를 마시고 잤다
ecey pam sakwa-rul mek-ko cha-rul masi-ko ca-ss-ta
apple-ACC eat-CONJ tea-ACC drink-CONJ sleep-PAST-DECL
“ate an apple, drank tea, and slept”

Here, neither the verb 먹고 mek-ko “eat-CONJ” nor the verb 마시고 masi-ko “drink-CONJ” is marked with a tense, but inherits the tense information from the sentence final verb 잤다 ca-ss-ta, marked with PAST tense. Through this inheritance process, these CONJUNCTIVE tenseless verbs can refer to the PAST events of eating and drinking that occurred before the event of sleeping in the past.

Consider anothe case of tense inheritance. The sentence-non-final verbal ending -다가 -taka may occurr with the PASTperfective marker -었었 -ess.ess. Here are examples:

(41)

미아가 미국에 가다가 서울로 왔다
mia-ka mikwuk-ey ka-taka seowul-ro wa-ss-ta
mia-NOM US-GOAL go-CONJ Seoul-DIR come-PAST-DECL
“While going to the States Mia came back to Seoul”

(42)

미아가 미국에 갔다가 서울로 왔다
mia-NOM mikwuk-ey ka-ss-taka seowul-ro wa-ss-ta
US-GOAL go-PERFECTIVE-CONJ Seoul-DIR come-PAST-DECL
“Having been to the States Mia came back to Seoul”

Both of the events of one's going to the States took place before his coming back to Seoul. But in (41) his going to the States was incomplete, while it was completed in (42).

E.6.3.5.2 Annotation of inherited tense information

(43)

어제 밤 미아가 사과를 먹고 차를 마시고 잤다
“Last night Mia ate an apple, drank tea, and slept”

```
<TIMEX3
    xml:id="t1"
    target="#token0 #token1"
    pred="LAST_NIGHT"
    type="TIME"
    value="2007-03-31TNT"
    temporalFunction="TRUE"
    anchorTime="t0"
    comment="TNT stands for 'night time'"/>
<EVENT
    xml:id="e1"
    target="#token4"
    pred="EAT"
    class="OCCURRENCE"
    type="PROCESS"
    pos="VERB"
    tense="NONE"
    vForm="CONJ"/>
<EVENT
    xml:id="e2"
    target="#token6"
    pred="DRINK"
    class="OCCURRENCE"
    type="PROCESS"
    pos="VERB"
    tense="NONE"
    vForm="CONJ"/>
<EVENT
    xml:id="e3"
    target="#token7"
    pred="SLEEP"
    class="OCCURRENCE"
    type="PROCESS"
    pos="VERB"
    tense="PAST"
    vForm="sFINAL"/>
```

```
<TLINK
        eventID="#e1"
        relatedToEvent="#e3"
        relType="IBEFORE"/>
<TLINK
        eventID="#e2"
        relatedToEvent="#e3"
        relType="IBEFORE"/>
<TLINK
        eventID="#e3"
        relatedToTime="#t1"
        relType="INCLUDED"/>
```

(44)

미아가 어제 미국에 가다가 서울로 왔다
“Mia came to Seoul on his way to America yesterday”

```
<TIMEX3
        xml:id="t1"
        target="#token1"
        pred="YESTERDAY"
        type="TIME"
        value="2007-05-05"
        temporalFunction="TRUE"
        anchorTime="t0"/>
<EVENT
        xml:id="e1"
        target="#token3"
        pred="GO"
        class="OCCURRENCE"
        type="TRNANSITION"
        pos="VERB"
        tense="NONE"
        vForm="CONJ"/>
<EVENT
        xml:id="e2"
        target="#token5"
        pred="COME"
        class="OCCURRENCE"
        type="TRNANSITION"
        pos="VERB"
        tense="PAST" \
        vForm="sFINAL"/>
<TLINK
        eventID="#e1"
        relatedToEvent="#e2"
        relType="IBEFORE"/>
<TLINK
        eventID="#e2"
        relatedToTime="#t1"
        relType="INCLUDED"/>
```

(45)

미아가 어제 미국에 갔다가 서울로 왔다
“Yesterday Mia went to Seoul and then came to Seoul”

```
<TIMEX3
       xml:id="t1"
       target="#token1"
       pred="YESTERDAY"
       type="TIME"
       value="2007-05-05"
       temporalFunction="TRUE"
       anchorTime="t0"/>
<EVENT
        xml:id="e1"
       target="#token3"
       pred="GO"
       class="OCCURRENCE"
       type="TRNANSITION"
       pos="VERB"
       tense="PAST"
       vForm="-taka,WHILE"/>
<EVENT
       xml:id="e2"
       target="#token5"
       pred="COME"
       class="OCCURRENCE"
       type="TRNANSITION"
       pos="VERB"
       tense="PAST"
       vForm="sFINAL"/>
<TLINK
       eventID="#e1"
       relatedToEvent="#e2"
       relType="IBEFORE"/>
<TLINK
       eventID="#e2"
       relatedToTime="#t1"
       relType="INCLUDED"/>
```

Interpretation of PAST in @vForm="-taka, WHILE":

— If an EVENT element contains @vForm="-taka, WHILE" and also PAST, then it is interpreted as referring to an event that has been completed or accomplished.

E.6.3.5.3 Annotation of tense in conditional clauses

The @tense PAST in a conditional clause may not refer to an event in the past. Consider:

(46)

모레 미아가 오면 좋겠다
morey mia-ka o-meyn coh-keyss-ta
the day after tomorrow Mia-NOM come-COND nice-CONJEC-DECL
“It would be nice if Mia comes the day after tomorrow”

(47)

모레 미아가 왔으면 좋겠다
morey mia-ka o-ass-umyen coh-keyss-ta
the day after tomorrow Mia-NOM come-PAST-COND nice-CONJEC-DECL
“It would be nice if Mia comes/would come the day after tomorrow”

(48)

어제 미아가 왔으면 좋았겠다
ecey mia-ka o-ass-umyen coh-ass-keyss-ta yesterday
Mia-NOM come-PAST-COND nice-PAST-CONJEC-DECL
“It would have been nice if Mia had come yesterday”

(49)

어제 미아가 왔더라면 좋았겠다
ecey mia-ka o-ass-te-ra-myen coh-ass-keyss-ta
yesterday Mia-NOM come-PAST-RETRO-IND-COND nice-PAST-CONJEC-DECL
“It would have been nice if Mia had come yesterday”

(50)=(46)

모레 (the day after tomorrow) 미아가 오면 좋겠다
“It would be nice if Mia comes the day after tomorrow”

```
<EVENT
      xml:id="e1"
      target="#token2"
      pred="COME"
      class="OCCURRENCE"
      type="TRNANSITION"
      pos="VERB"
      vForm="CONDITIONAL"
      tense="NONE"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="NICE"
      class="I_STATE"
      type="STATE"
      pos="ADJECtIVE"
      vForm="sFIANAL"
      tense="NONE"
      modality="CONJECTURAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="SIMULATANEOUS"/>
```

(51)=(47)

모레 (the day after tomorrow) 미아가 왔으면 좋겠다
“It would be nice if Mia comes/would come the day after tomorrow”

```
<EVENT
      xml:id="e1"
      target="#token2"
      pred="COME"
      class="OCCURRENCE"
      type="TRNANSITION"
      pos="VERB"
      vForm="CONDITIONAL"
      tense="PAST"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="NICE"
      class="I_STATE"
      type="STATE"
      pos="ADJECtIVE"
      vForm="sFIANAL"
      tense="PAST"
      modality="CONJECTURAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="SIMULATANEOUS"/>
```

(52)=(48)

어제 (yesterday) 미아가 왔으면 좋았겠다
“It would have been nice if Mia had come yesterday”

```
<EVENT
      xml:id="e1"
      target="#token2"
      pred="COME"
      class="OCCURRENCE"
      type="TRNANSITION"
      pos="VERB"
      vForm="CONDITIONAL"
      tense="PAST"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="NICE"
      class="I_STATE"
      type="STATE"
      pos="ADJECtIVE"
      vForm="sFIANAL"
      tense="PAST"
      modality="CONJECTURAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="SIMULATANEOUS"/>
```

(53)=(49)

어제 (yesterday) 미아가 왔더라면 좋았겠다

“It would have been nice if Mia had come yesterday”

```
<EVENT
      xml:id="e1"
      target="#token2"
      pred="COME"
      class="OCCURRENCE"
      pos="VERB"
      vForm="CONDITIONAL"
      tense="PAST"
      mood="RETROSPECTIVE"/>
<EVENT
      xml:id="e2"
      target="#token3"
      pred="NICE"
      class="I_STATE"
      type="STATE"
      pos="ADJECtIVE"
      vForm="sFIANAL"
      tense="PAST"
      modality="CONJECTURAL"/>
<TLINK
      eventID="#e1"
      relatedToEvent="#e2"
      relType="SIMULATANEOUS"/>
```

E.7 Aspect

E.7.1 Aspect markers

In Korean, aspect PROGRESSIVE is expressed by complex verbal structures such as follows:

— -고 있 (STEM-ko iss) mostly for OCCURRENCE class verbs

— -어 있 (STEM-e iss) for STATE class verbs

Here are examples:

(54)

사과를 먹고 있었다
sakwa-rul mek-ko iss-ess-ta
apple-ACC eat-BR auxVerbPROG-PAST-DECL
“was/were eating an apple”

(55)

하루 종일 앉아 있었다
haru cong-il anc-a iss-ess-ta
all day sit-BR auxVerbDURA-PAST-DECL
“was/were sitting all day”

Unlike English, stative verbs like -가지 kaci- “have/own” and -알 al- “know” may form the PROGRESSIVE construction: 가지고 있다 kaci-ko iss-ta and 알고 있다 al-ko iss-ta. Unlike the -ko iss-ta construction, the –어 있다 -e iss-ta construction is very restricted. Only few intransitive verbs like 눕-/누우- /nup-/nuw- “lie”, -앉 anc- “sit”, and -서 se- “stand” or passivized verbs with the ending -지 -ci like 알려- allyeci- “be known” take up this form.

E.7.2 Annotation of aspect markers

The annotation of aspect features is straighforward. The complex aspectual constructions are treated as single chunks without separating the main STEM part from the auxiliary part. Here are examples:

(56)=(54)

사과를 먹고 있었다

```
<EVENT
    xml:id="e56"
    target="#token1 #token2"
    pred="EAT"
    class="OCCURRENCE"
    type="PROCESS"
    pos="VERB"
    tense="PAST"
    aspect="PROGRESSIVE"
    vForm="sFINAL"/>
```

(57)=(55)

어제 (yesterday) 하루 종일 (all day) 앉아 있었다

```
<EVENT
    xml:id="e57"
    target="#token3 #token4"
    pred="SIT"
    class="STATE"
    type="STATE"
    pos="VERB"
    tense="PAST"
    aspect="PROGRESSIVE"
    vForm="sFINAL"/>
```

(58)

미아는 돈을 많이 가지고 있다

```
<EVENT
    xml:id="e58"
    target="#token3 #token4"
    pred="HAVE/OWN"
    class="STATE"
    type="STATE"
    pos="VERB"
    tense="NONE"
    aspect="PROGRESSIVE"
    vForm="sFINAL"/>
```

(59)

지금 (now) 미아가 미국에 가 있다

```
<EVENT
    xml:id="e59"
    target="#token3 #token4"
    pred="GO"
    class="STATE"
    type="STATE"
    pos="VERB"
    tense="NONE"
    aspect="PROGRESSIVE"/>
```

Sentence (58) means “Mia has a lot of money” and sentence (59) means “Mia has gone to US, and she is there now”.

E.7.3 Interpretation of aspectual features

The concept of aspect involves the event ontology, the types and structure of events. *ISO-TimeML* introduces seven classes of verbs and three types of events:

class ::= 'OCCURRENCE' | 'PERCEPTION' | 'REPORTING' | 'ASPECTUAL' | 'STATE' | 'I_STATE' | 'I_ACTION'
type ::= 'STATE' | 'PROCESS' | 'TRANSITION'

The structure of an event may be segmented into the initial, medial, and terminal part. Depending on the perspective of a language user or interpreter, one's focus shifts from the initial to the medial (on-going) or to the terminal (completed) part. Suppose someone says “Mia is eating an apple”. This statement is understood as focusing on the medial or on-going part of the event of Mia's eating an apple. On the other hand, if one says “Mia is wearing a beautiful hat”, its focus is on the terminal or resulting part of the event of Mia's putting on a hat. This statement is interpreted as restulting STATE rather than as an OCCURRENCE of process.

Sentence (60) is ambiguous:

(60)

미아가 옷을 입고 있다 (PROCESS or STATE)
mia-ka os-ul ip-ko iss-ta
Mia-NOM clothes-ACC wear-COMP auxVerbPROG-DECL
“Mia is wearing clothes”

(61)

미아가 지금 목욕을 하고 옷을 입고 있다 (PROCESS)
mia-ka cikum mokyok-ul ha-ko os-ul ip-ko iss-ta
Mia-NOM now bath-ACC do-AND clothes wear-COMP auxVerbPROG-DECL
“Mia took a bath and is now putting on clothes”

(62)

미아가 오늘은 빨간 옷을 입고 있다 (STATE)
mia-ka onul-un ppalkan os-ul ip-ko iss-ta
Mia-NOM today-TOPIC red clothes-ACC wear-COMP auxVerbPROG-DECL
“Today Mia is dressed in red”

The PROGRESSIVE form of the verbal expression in sentence (60) can be interpreted in two ways: either in the on-going OCCURRENCE/PROCESS sense as in (61) or in the resulting STATE sense as in (62). Sentence (60) is thus annotated differently as shown below:

(63)=(60)

미아가 옷을 입고 있다

```
<EVENT
    xml:id="e1"
    target="#token2 #token3"
    pred="WEAR"
    class="OCCURRENCE"
    type="PROCESS"
    pos="VERB"
    tense="NONE"
    aspect="PROGRESSIVE"
    vForm="sFINAL"/>
```

(64)

미아가 옷을 입고 있다

```
<EVENT
      xml:id="e2"
      target="#token2 #token3"
      pred="WEAR"
      class="STATE"
      type="STATE"
      pos="VERB"
      tense="NONE"
      aspect="PROGRESSIVE"
      vForm="sFINAL"/>
```

The basic difference between these two is that (63) refers to an event of class OCCURRENCE/PROCESS, while (64) refers to an event of class STATE. In order to make this distinction clear, PROGRESSIVE STATE may be annotated as @class="STATE" and @aspect="DURATIVE". The verbalSTEM-e iss-ta construction constrasts with a particular set of verbs with the PAST verbal form like ip- “wear” and cuk- “ie”. Consider:

(65)

미아가 예쁜 옷을 입었다
mia-ka yeyppun os-ul ip-ess-ta
Mia-NOM pretty clothes-ACC wear-PAST-DECL
“Mia wore/is wearing pretty clothes”

(66)

새가 죽었다
say-ka cuwk-ess-ta
bird-NOM die-PAST-DECL
“(the) bird died/is dead”

(67)

미아는 돈을 많이 가졌다
mia-nun ton-ul manhi kaci-ess-ta
Mia-TOP money-ACC much have-PAST-DECL
“Mia had/has much money”

(68)

봄이 왔다
pom-i wa-ss-ta
spring-NOM come-PAST-DECL
“Spring came/has come”

Each of the verbs in these examples is marked with the PAST tense ending. But the examples above each refer to a STATE at the present, the terminal phase of some event that may have occurred or started in the past. Sentence (65), for instance, means that Mia has put on pretty clothes and is still wearing them. Sentence (66) means that a bird died and is dead now, sentence (67) that Mia (has earned money and) owns a lot of money now, and sentence (68) that spring came and that it is spring now. By marking up the verbal ending -ㅆ -ss not just as PAST, but as PERFECTIVE, we can obtain appropriate interpretations for these exceptional cases. Here is an example:

(69)

새가 죽었다
(The) bird died
<**EVENT**
 xml:id="e1"
 target="#token1"
 pred="DEAD/DIE"
 class="STATE"
 type="STATE"
 pos="VERB"
 tense="NONE"
 aspect="PERFECTIVE"/>

E.7.4 Interpretation conditions of aspect

— PROGRESSIVE process: If an <EVENT> element is specified with @aspect="PROGRESSIVE" and @class ="OCCURRENCE/PROCESS, then that event is interpreted as an on-going process being continued at some interval of time.

— PROGRESSIVE/DURATIVE state: If an <EVENT> element is specified with @aspect="PROGRESSIVE" and @class= "STATE", then that event is interpreted as a state being uniformly retained at some interval of time. Note that this state may be annotated as @aspect="DURATIVE".

— PRESENT PROGRESSIVE: If an <EVENT> element is specified with @aspect="PROGRESSIVE" and @tense="NONE", then that event is interpreted as taking place in the neighborhood of the present moment of time, namelyN(n).

— PAST PROGRESSIVE: If an <EVENT> element is specified with aspect="PROGRESSIVE" and tense="PAST", then that event is interpreted as taking place in the neighborhood of some moment of time t in the past interval of time, N(t) which is a subinterval of the past time interval.

— PRESENT RESULTATIVE If an EVENT element is specified with @aspect="PERFECTIVE" and @tense="NONE", then an event associated with that element is interpreted as an event the initial OCCURRENCE of which might have completed in the past, but with the resulting state retained in the neighborhood of the present moment of time, namely N(n).

E.8 Modality

E.8.1 Conjectural modal markers

The CONJECTURAL modality is expressed by the verbal ending -keyss or -/ -l/ul kes. It is claimed by Lee, K. (1998) that these two differ in the degree of certainty expressed by each: the former expresses certainty, while the latter expresses probability. But here this difference is ignored and only the ending -keyss is illustrated. Here are examples:

(70)

지금 부산에 비가 오고 있겠다
cikum pusan-ey rain-ka o-ko iss-keyss-ta
now Busan-LOC rain-NOM come-COMP PROG-CONJEC-DECL
“It must be raining in Busan now”

(71)

미아는 어제 떠났겠다
mia-nun ecey ttena-ss-keyss-ta
Mia-TOP yesterday leave-PAST-CONJEC-DECL
“Mia must have left yesterday”

(72)

어릴 때에 미아는 예뻤겠다
eri-l ttay-ey mia-ka yeyppe-ss-keyss-ta
young-ADNOM time-LOC Mia-NOM pretty-PAST-CONJEC-DECL
“When she was young, Mia must have been pretty”

When used with the first person Subject, the CONJECTURAL -keyss often expresses her or his intention, as illustrated by:

(73)

나는 내일 떠나겠다
na-nun nayil ttena-keyss-ta
I-TOP tomorrow leave-CONJEC-DECL
“I will leave tomorrow”

But note that the following examples do not express the speaker's intention:

(74)

나는 지금 아파 죽겠다
na-nun cikum aph-a cwuk-keyss-ta
I-TOP now sick-CONJUNCTIVE die-CONJEC-DECL
"I'm now sick and so must die"

(75)

이번에는 내가 복권에 당첨되겠다
ipen-ey-nun nay-ka pokkwon-ey tangchemtoy-keyss-ta
time-LOC-TOP I-NOM lottery win-CONJEC-DECL
“This time I must win/be winning a lottery”

E.8.2 Annotation of modality CONJECTURAL

The annotation of the CONJECTURAL modality is again very simple.

(76)=(71)

미아는 어제 (yesterday) 떠났겠다

```
<EVENT
  xml:id="e76"
  target="token2"
  pred="LEAVE"
  class="OCCURRENCE"
  type="TRANSITION"
  pos="VERB"
  tense="PAST"
  modality="CONJECTURAL"
  vForm="sFINAL"/>
```

(77)

부산에는 지금 (now) 비가 오고 있을 것이다

```
<EVENT
  xml:id="e77"
  target="token3 token4 token5"
  pred="COME"
  class="OCCURRENCE"
  type="PROCESS"
  pos="VERB"
  tense="NONE"
  modality="CONJECTURAL"
  vForm="sFINAL"/>
```

E.8.3 Interpretation of modality CONJECTURAL

If the modality of an <EVENT> element is specified with the value CONJECTURAL, then it is understood as expressing the speaker's conjectural certainty of the occurrence of that event referred to.

Furthermore, if the event referred to is controllable by the speaker himself, then it is understood as expressing the speaker's intention to make that event realized.

E.9 Mood

E.9.1 Mood markers

The verbal ending –더 -te is treated as the RETROSPECTIVE mood marker. Here are examples:

(78)

베트남은 참 덥더라
peytunam-un cham tep-te-la
Vietnam-TOP really hot-RETRO-DECL
“Vietnam was really hot, I recall”

(79)

전쟁 때 베트남 사람들의 삶이 비참하였겠더라
cencayng ttay peytunam saramtul-uy salm-i pichamha-yess-keyss-te-la
war time Vienam people-GEN life-NOM terrible-PAST-CONJEC-RETOR-DECL
“In the war Vietnamese people's life must have been terrible, I think”

(80)

십년 후엔 베트남이 잘 살겠더라
sipnyen hwu-ey-n peytunam-i cal sal-keyss-te-la
10 years after Vietnam-NOM well live-CONJEC-RETRO-DECL
“Ten years later (the people of) Vietnam must live well, I think.”

(81)

점을 쳤더니 미아가 다음 달에 결혼을 하더라
cem-ul chy-ess-te-ni mia-ka taum tal-ey kyelhon-ul ha-te-la
fortune tell-RETRO-CONN Mia-NOM next month marriage do-RETRO-DECL
“As I saw the fortune telling, Mia will be marrying next month”

(82)

일기 예보를 들으니 모레 비가 오겠던데
ilki yeypo-rul tul-u-ni moreyil pi-ka o-keyss-te-nte
weather forecast-ACC hear the day after tomorrow rain-NOM come-CONJEC-RETRO-DECL
“As I listened to the weather forecast, it must be raining the day after tomorrow, I think”

E.9.2 Annotation of mood RETROSPECTIVE

Verbal expressions with the verbal ending -te can be automatically marked with mood RETROSPECTIVE.

(83)=(78)

베트남은 참 덥더라

```
<EVENT
  xml:id="e83"
  target="#token2"
  pred="HOT"
  class="STATE"
  type="STATE"
  pos="ADJECTIVE"
  tense="NONE"
  mood="RETROSPECTIVE"/>
```

(84)=(79)

전쟁 때 (at the time of the war) 베트남 사람들의 삶이 비참하였겠더라

```
<EVENT
  xml:id="e84"
  target="#token5"
  pred="TERRIBLE"
  class="STATE"
  type="STATE"
  pos="ADJECTIVE"
  tense="PAST"
  modality="CONJECTURAL"
  mood="RETROSPECTIVE"/>
```

(85)=(80)

10 년 후엔 (10 years later) 베트남이 잘 살겠더라

```
<EVENT
  xml:id="e85"
  target="#token4"
  pred="LIVE"
  class="STATE"
  pos="VERB"
  tense="NONE"
  modality="CONJECTURAL"
  mood="RETROSPECTIVE"/>
```

Chang (1996) treats the verbal ending –겠 -keyss as one of the mood markers, namely the VOLITIONAL mood. But, since it can co-occur with the RETROSPECTIVE marker –더 -te as shown above, thus illicitly duplicating the value of the attribute @mood, it is treated here in this annex as a @modality marker. The adnominal-forming –던 -ten is also annotated with @mood="RETROSPECTIVE".

E.9.3 Interpretation of RETROSPECTIVE mood

Chang (1996: 131) Sohn (1999: 359) states: "The retrospective mood denotes the speaker's experience or observation in retrospect." also expresses a similar view: "Retrospective mood denotes a situation in which someone recalls a fact or an event he witnessed or experienced, and thus has meanings such as “I saw, observed, experienced” in declaratives and “did you see, oberserve, experience” in interrogatives. To formalize this notion of experience or observation in situation-theoretic terms, Lee, K. (1993) introduced the notion of observational accessibility. The retrospective mood is used when the event referred to is observationally accessible to the speaker-observer.

If the attribute @mood of an <EVENT> element is specified with the value RETROSPECTIVE, then the event referred to by that <EVENT> element is interpreted as implying that the event is/was within the (spatio-temporal) location observationally accessible to the speaker.

E.10 Specific values for <EVENT> attributes in Korean

The specific values for each of the nine attributes for <EVENT> in Korean is given in BNF, as below. Some of the values vary from the list given in Annex A, but they are allowed according to the general principle of multilingual applicaiton of *ISO-TimeML*.

@class ::=
'REPORTING' | 'PERCEPTION' | 'ASPECTUAL' | 'I_ACTION' | 'I_STATE' | 'OCCURRENCE' | 'STATE'

@type :: 'STATE' | 'PROCESS' | 'TRANSITION'

@pos ::= 'ADJECTIVE' | 'NOUN' | 'VERB' | 'NONE'

@tense ::= 'FUTURE' | 'PAST' | 'PRESENT' | 'NONE' {default, if absent, is 'NONE'}

@aspect ::= '(DURATIVE)' | 'PROGRESSIVE' | 'PERFECTIVE' | 'NONE' {default, if absent, is 'NONE'}

@polarity ::= 'NEG' | 'POS' {default, if absent, is 'POS'}

@mood ::= 'RETROSPECTIVE' | 'NONE' {default, if absent, is 'NONE'}

@modality ::= 'CONJECTURAL' | 'INDICATIVE' {default, if absent, is 'INDICATIVE'}

@vForm ::= 'COMP' | 'CONJUNCTIVE' | 'CONDITIONAL' | 'sFINAL' | CDATA {default, if absent, is 'sFINAL'}

E.11 Summary

The temporal annotation of verbal endings in Korean is summarized in Table E.1.

Table E.1 — Verbal temporal endings

ending	class	tense	aspect	modality	mood	example
nun	OCCUR				IND	mek-nun-ta (eats)
ess-sup-ni	OCCUR	PAST			IND	mek-ess-sup-ni-ta (ate, polite)
ess	OCCUR	PAST				mek-ess-ta (ate)
ess	STATE		PERF			kaci-ess-ta (has)
ess.ess	OCCUR	PAST	PERF			mek-ess-ess-ta (had eaten)
keyss	OCCUR			CONJEC		mek-keyss-ta (may eat)
ess-keyss	OCCUR	PAST		CONJEC		mek-ess-keyss-ta (might have eaten)
te	OCCUR				RETRO	mek-te-la (I recall, ate)
ess-te	OCCUR	PAST			RETRO	mek-ess-te-la (I recall, had eaten)
keyss-te	OCCUR			CONJEC	RETRO	mek-keyss-te-la (I recall, may have eaten)
ko iss	OCCUR		PROG			mek-ko iss-ta (is eating)
ko iss	STATE		PROG			al-ko iss-ta (know)
ko iss-ess-keyss	OCCUR	PAST	PROG	CONJEC		mek-ko iss-ess-keyss-ta (must have been eating)
e iss	STATE		PROG(DURA)			nwu-e iss-ta (is lying)
e iss-keyss-te	STATE		PROG(DURA)	CONJEC	RETRO	nwu-e iss-keyss-te-la (must be lying, I recall)
-nun	OCCUR					mek-nu eat (ADNOMINAL)
-un	OCCUR	PAST				mek-un ate (ADNOMINAL)
-un	STATE					coh-un good (ADNOMINAL)
-ten	OCCUR					mek-ten used to eat (ADNOMINAL)

Annex F
(informative)

Past and current activities on temporal and event annotation

F.1 Introductory remarks

Much of the discussion here is adapted from a reference by one of the editors “An Introduction to Part IV Temporal Annotation” in Mani, Pustejovsky, and Gaizauska (2005). The automatic recognition of temporal and event expressions in natural language text has recently become an active area of research in computational linguistics and semantics. In this annex, the work done on temporal and event annotation is reviewed.

F.2 Annotating temporal expressions

The most obvious temporal feature to annotate in texts, and the one which historically was addressed first, is temporal referring expressions (as found in temporal adverbials, for example); that is, expressions which refer to times (“July 1, 1867”), durations (“three months”) or frequencies (“weekly”). Being able to identify and distinguish these types of expression is crucial to being able to situate the events described in text either absolutely in terms of some conventional calendrical time frame or relatively with respect to other events. The examples just given perhaps understate the complexity of the phenomena to be addressed.

When devising an annotation scheme to capture temporal referring expressions, a variety of complications must be dealt with:

- indexicals: expressions such as “now”, “yesterday” – and other contextually dependent expressions such as partially specified calendrical times (e.g. “Wednesday– which Wednesday?”) or relatives such as “next week”, “three weeks ago”, all of which depend for their interpretation on knowledge of a deictic centre;

- relational expressions: expressions which explicitly specify times in relation to other times (“two weeks after Christmas”) or to events (“5 seconds after the first explosion”); and

- vagueness: expressions referring to times whose boundaries are inherently vague (“spring”, “evening”) or which contain modifiers which blur the time reference (“several days ago”, “sometime after 7 p.m.”).

Work to devise annotation schemes for temporal referring expressions appears to have begun as part of the Named Entity (NE) tagging subtask within the DARPA Message Understanding Conference (MUC) series of evaluations, specifically in MUC-6 [MUC (1995)]. In this task, participants' systems were to tag (by inserting SGML tags into running text) expressions which named persons, organizations, locations, dates, times, monetary amounts, and percentages. A key part of this exercise was that a set of texts was manually tagged by human annotators to provide a “gold standard” measure of correctness.

Metrics, principally the recall and precision metrics adapted from information retrieval research, were used to compare system-supplied annotations (or responses) against human-supplied annotations (or answer keys). Recall that the proportion of the answer keys for which a correct response is supplied is a measure of coverage or completeness of a system; precision, the proportion of responses which are correct, i.e. match the answer key, is a measure of correctness or soundness of a system. In MUC-6, date and time (of day) expressions were labeled using a <TIMEX> tag. Only absolute time expressions were to be annotated, i.e. expressions which indicated a specific minute, hour, day, month, season, year, etc. Relative time expressions (e.g. “last July”) were excluded, though subexpressions within them (e.g. “July” in this example) were to be tagged.

A set of thirty manually annotated newswire texts were used for a blind evaluation. The top scoring automated system scored 0,97 recall and 0,96 precision on the <TIMEX> tagging task. In MUC-7 [MUC(1998)] the principal change was to capture relative as well as absolute date and time expressions, though the two did not need to be distinguished in the tagging. Thus indexicals, such as “yesterday”, and “last July”, were to be marked, as were so-called “time-relative-to-event” phrases such as “the morning after the July 17 disaster”.