The Specification Language TimeML

James Pustejovsky, Robert Ingria, Roser Saurí, José Castaño, Jessica Littman
Brandeis University

Rob Gaizauskas, Andrea Setzer
University of Sheffield

Graham Katz
University of Osnabrück

Inderjeet Mani
Georgetown University

Abstract

In this paper we provide a description of TimeML, a rich specification language for event and temporal expressions in natural language text, developed in the context of the AQUAINT program on Question Answering Systems. Unlike most previous work on event annotation, TimeML captures three distinct phenomena in temporal markup: (1) it systematically anchors event predicates to a broad range of temporally denotating expressions; (2) it orders event expressions in text relative to one another, both intrasententially and in discourse; and (3) it allows for a delayed (underspecified) interpretation of partially determined temporal expressions. We demonstrate the expressiveness of TimeML for a broad range of syntactic and semantic contexts, including aspectual predication, modal subordination, and an initial treatment of lexical and constructional causation in text.

1 Introduction

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 paper, we report on TimeML, a specification language for events and temporal expressions, which was developed in the context of a six-month workshop, TERQAS (`www.time2002.org`), funded under the auspices of the AQUAINT program. The ARDA-funded program AQUAINT is a multi-project effort to improve the performance of question answering systems over free text, such as that encountered on the Web. An important component to this effort is the access of information from text through content rather than keywords. Named entity recognition (Chinchor et al, 1999) 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. Beyond these metadata tags (ontological types), however, there is only a limited ability at marking up text for real content. One of the major problems that has not been solved is the recognition of events and their temporal anchorings. In this paper, we report on an AQUAINT project to create a specification language for event and temporal expressions in text.

Events in articles are naturally anchored in time within the narrative of a text. For this reason, temporally grounded events are the very foundation from which we reason about how the world changes. Without a robust ability to identify and extract events and their temporal anchoring from a text, the real "aboutness" of the article 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.

For example, currently questions such as those shown below are not supported by question answering systems.

1. a. Is Gates currently CEO of Microsoft?
   b. When did Iraq finally pull out of Kuwait during the war in the 1990s?
   c. Did the Enron merger with Dynegy take place?

What characterizes these questions as beyond the scope of current systems is the following: they refer, respectively, to the temporal aspects of the properties of the entities being questioned, the relative ordering of events in the world, and events that are mentioned in news articles, but which have never occurred.

There has recently been a renewed interest in temporal and event-based reasoning in language and text, particularly as applied to information extraction and reasoning tasks (cf. Mani and Wilson, 2000, *ACL Workshop on Spatial and Temporal Reasoning*, 2001, *Annotation Standards for Temporal Information in Natural Language*, LREC 2002). Several papers from the workshop point to promising directions for time representation and identification (cf. Filatova and Hovy, 2001, Schilder and Habel, 2001, Setzer, 2001). Many issues relating to temporal and event identification remain unresolved, however, and it is these issues that TimeML was designed to address. Specifically, four basic problems in event- temporal identification are addressed:

(a) Time stamping of events (identifying an event and anchoring it in time);

(b) Ordering events with respect to one another (lexical versus discourse properties of ordering);

(c) Reasoning with contextually underspecified temporal expressions (temporal functions such as *last week* and *two weeks before*);

(d) Reasoning about the persistence of events (how long does an event or the outcome of an event last).

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

2 Introduction to TimeML

Unlike most previous attempts at event and temporal specification, TimeML separates the representation of event and temporal expressions from the anchoring or ordering dependencies that may exist in a given text. There are four major data structures that are specified in TimeML (Ingria and Pustejovsky, 2002, Pustejovsky et al., 2002): `EVENT`, `TIMEX3`, `SIGNAL`, and `LINK`. These are described in some detail below. The features distinguishing TimeML from most previous attempts at event and time annotation are summarized below:

1. Extends the TIMEX2 annotation attributes;
2. Introduces **Temporal Functions** to allow intensionally specified expressions: *three years ago*, *last month*;
3. Identifies signals determining interpretation of temporal expressions;
   (a) Temporal Prepositions: *for*, *during*, *on*, *at*;
   (b) Temporal Connectives: *before*, *after*, *while*.
4. Identifies all classes of event expressions;
   (a) Tensed verbs; *has left*, *was captured*, *will resign*;
   (b) stative adjectives and other modifiers; *sunken*, *stalled*, *on board*;
   (c) event nominals; *merger*, *Military Operation*, *Gulf War*;
5. Creates dependencies between events and times:
   (a) Anchoring; *John left on Monday.*
   (b) Orderings; *The party happened after midnight.*
   (c) Embedding; *John said Mary left.*

In the design of TimeML, we began with the core of the TIDES TIMEX2 annotation effort (Ferro, et al, 2001)[1] and the temporal annotation language presented in Andrea Setzer's thesis (Setzer, 2001). Consideration of the details of this representation, however, in conjunction with problems raised in trying to apply it to actual texts, resulted in several changes and extensions to Setzer's original framework. The most significant extension is the logical separation of event

[1] TIMEX2 introduces a value attribute whose value is an ISO time representation in the ISO 8601 standard.

descriptions and the relations they enter into, defined relative to temporal expressions or other events. This resulted in a natural reification of these relations as LINK tags. Details on motivations for introducing the class of LINK tags can be found in Ingria and Pustejovsky (2002). Briefly, Setzer (2001) defines events as having the following attribute structure:

```
attributes ::=

   eid class [argEvent] [tense] [aspect]
   [([signalID] relatedToEvent eventRelType)
   | ([signalID] relatedToTime timeRelType)] ...
```

That is, an `EVENT` has as attributes a unique id, a class (see below), optionally another event which is its argument, a tense and an aspect, and then, importantly, but also optionally, either another event to which it is temporally related or a time to which it is temporally related, perhaps in each case, by a signal. One thing that is striking looking at this specification is that for both the case where the event is related to another event and the case where it is related to a time, we are dealing not with three unrelated attributes (`eid relatedtoEvent eventReltype`), but with three attributes that only make sense as a unit. The same triad also appears in the attribute structure of Setzer's definition of complex time expressions, time denoting expressions which inherently involve reference to an event, such as *five seconds after the explosion*: her specification for such complex `TIMEX`'s includes the attributes `tid relatedtoEvent relType`. Moreover, as the specification of the values for the eventRelType and timeRelType attributes of EVENT and the relType attribute of TIMEX indicates, we are really dealing with one property, whose values are specified three times. This is forced in the case of eventRelType and timeRelType for EVENT by virtue of the fact that only the name of the attribute can link it to relatedToEvent or relatedToTime, respectively. And, of course, since relType is defined on TIMEX, not EVENT, it must repeat the specification of permissible values.

All these considerations suggest that these triplets of attributes should be factored out into the form of a new abstract tag (i.e. one which consumes no input text). This would formally express the fact that these attributes are linked, allow eventRelType, timeRelType and relType to be collapsed into a single attribute, and allow the specification of the possible values of this single attribute to be stated only once.

TimeML considers "events" (and the corresponding tag `<EVENT>`) a cover term for situations that *happen* or *occur*. Events can be punctual or last for a period of time. We also consider as events those predicates describing *states* or *circumstances* in which something obtains or holds true. Not all stative predicates are marked up, however. Only those states that are directly related to a temporal expression are annotated, including those states that identifiably change over the course of a document. Events are generally expressed by means of tensed or untensed verbs, nominalizations, adjectives, predicative clauses, or prepositional phrases. The specification of EVENT is shown below:

```
attributes ::= eid class
eid ::= EventID
%EventID ::= e<integer>
class ::= ’OCCURRENCE’|’PERCEPTION’|’REPORTING’|’ASPECTUAL’
         |’STATE’|’I_STATE’|’I_ACTION’
```

Examples of each of these event types are given below:

1. **Occurrence**: *die*, *crash*, *build*, *merge*, *sell*
2. **State**: *on board*, *kidnapped*, *love*, ..
3. **Reporting**: *Say*, *report*, *announce*,
4. **I-Action**: *Attempt*, *try*, *promise*, *offer*
5. **I-State**: *Believe*, *intend*, *want*
6. **Aspectual**: *begin*, *finish*, *stop*, *continue.*
7. **Perception**: *See*, *hear*, *watch*, *feel.*

MAKEINSTANCE is a realization tag that is used to annotate information about a given event. One can create as many instances as are motivated by the text, for example in dealing with quantified event references such as *John taught three times.* All relations indicated by links (section 3) are stated over these instances. Because of this, every EVENT introduces at least one corresponding MAKEINSTANCE. The specification of MAKEINSTANCE is shown below:

```
attributes ::= eiid eventID tense aspect [polarity][modality][signalID]
[cardinality]
%eiid ::= ID
eiid ::= EventInstanceID
%EventInstanceID ::= ei<integer>}
eventID ::= IDREF
%{eventID ::= EventID}
tense ::= ’PAST’ | ’PRESENT’ | ’FUTURE’ | ’NONE’
aspect ::= ’PROGRESSIVE’ | ’PERFECTIVE’ | ’PERFECTIVE_PROGRESSIVE’ |
’NONE’
polarity ::= ’NEG’ | ’POS’
modality ::= CDATA
signalID ::= IDREF
%{signalID ::= SignalID}
cardinality ::= CDATA
```

The TIMEX3 tag is used to mark up explicit temporal expressions, such as times, dates, durations, etc. It is modeled on both Setzer’s (2001) TIMEX tag, as well as the TIDES (Ferro, et al. (2002)) TIMEX2 tag. There are four major types of TIMEX3 expressions: (a) Fully Specified Temporal Expressions, *June 11, 1989*, *Summer, 2002*; (b) Underspecified Temporal Expressions, *Monday*, *Next month*, *Last year*, *Two days ago*; (c) Durations, *Three months*, *Two years*; (d) Sets, *Twice a month*, *Daily.*

```
attributes ::= tid type [functionInDocument][beginPoint][endPoint]
                [quant][freq][temporalFunction]
                (value | valueFromFunction)
                [mod] [anchorTimeID]
tid ::= TimeID
%TimeID ::= t<integer>
type ::= ’DATE’|’TIME’|’DURATION’|’SET’
functionInDocument ::= ’CREATION_TIME’|’EXPIRATION_TIME’ |
                        ’MODIFICATION_TIME’|’PUBLICATION_TIME’|
                        ’RELEASE_TIME’|’RECEPTION_TIME’|’NONE’
temporalFunction ::= ’true’ | ’false’
%{temporalFunction ::= boolean}
beginPoint ::= IDREF
%{beginPoint ::= TimeID}
endPoint ::= IDREF
%{endPoint ::= TimeID}
quant ::= CDATA
freq ::= CDATA
%value ::= CDATA
value ::= duration | dateTime | time | date |  gYearMonth |
            gYear | gMonthDay |  gDay | gMonth
valueFromFunction ::= IDREF
%{valueFromFunction ::= TemporalFunctionID
%TemporalFunctionID ::= tf<integer>}
mod ::= ’BEFORE’|’AFTER’|’ON_OR_BEFORE’|’ON_OR_AFTER’|
        ’LESS_THAN’|’MORE_THAN’|’EQUAL_OR_LESS’|
        ’EQUAL_OR_MORE’|’START’|’MID’|’END’|’APPROX’
anchorTimeID ::= TimeID
```

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 (available at www.prismstandard.org/).

The treatment of temporal functions in TimeML allows any time-value dependent algorithms to delay the computation of the actual (ISO) value of the expression. The following informal paraphrase of some examples illustrates this point, where DCT is the Document Creation Time of the article.

1. *last week* = `(predecessor (week DCT))`: That is, we start with a temporal anchor, in this case, the DCT, coerce it to a week, then find the week preceding it.
2. *last Thursday* = `(thursday (predecessor (week DCT))`: Similar to the preceding expression, except that we pick out the day named ’thursday’ in the predecessor week.
3. *the week before last* = `(predecessor (predecessor (week DCT)))`: Also similar to the first expression, except that we go back two weeks.
4. *next week* = `(successor (week DCT))`: The dual of the first expression: we start with the same coercion, but go forward instead of back.

SIGNAL is used to annotate sections of text, typically function words, that indicate how temporal objects are to be related to each other. 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*). The basic functionality of the SIGNAL tag was introduced by Setzer (2001). In TimeML it has been expanded to also mark indicators of temporal quantification such as *twice*, *three times*, and so forth. The specification for SIGNAL is given below:

```
attributes ::= sid
%sid ::= ID
sid ::= SignalID
%SignalID ::= s<integer>}
```

To illustrate the application of these tags, consider the example annotation shown below.

> *John left 2 days before the attack.*

```
John
<EVENT eid="e1" class="OCCURRENCE">
left
</EVENT>
<MAKEINSTANCE  eiid="ei1"
eventID="e1" tense="PAST" aspect="PERFECTIVE"/>
<TIMEX3 tid="t1" type="DURATION" value="P2D"
temporalFunction="false">
2 days
</TIMEX3>
<SIGNAL sid="s1">
before
<SIGNAL>
the
<EVENT
eid="e2" class="OCCURRENCE">
attack
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"
tense="NONE" aspect="NONE"/>.
```

3 LINKS

One of the major innovations introduced in TimeML is the LINK tag. As mentioned above, the set of LINK tags encodes the various relations that exist between the temporal elements of a document, as well as establishing ordering between events directly. There are three types of link tags.

1. **TLINK:** a Temporal Link representing the temporal relationship holding between events or between an event and a time;

2. **SLINK:** a Subordination Link used for contexts introducing relations between two events, or an event and a signal;

3. **ALINK:** an Aspectual Link representing the relationship between an aspectual event and its argument event.

3.1 TLINK

TLINK represents the temporal relationship holding between events or between an event and a time, and establishes a link between the involved entities, making explicit if they are:[2]

1. Simultaneous:
2. Identical: (referring to the same event)
   John drove to Boston. During his drive he ate a donut.
3. One before the other:
   John left before Mary arrived.
4. One after the other: (cf. 3)
5. One immediately before the other:
   All passengers died when the plane crashed into the mountain.[3]
6. One immediately after the other: (cf. 5)
7. One including the other:
   John arrived in Boston last Thursday.
8. One being included in the other: (cf. 7)
9. One holding during the duration of the other:
   John was on vacation for two months.
10. One being the beginning of the other:
    John has lived in Boston since 1998.
11. One being begun by the other: (cf. 10)
12. One being the ending of the other:
    John stayed in Boston till 1999.
13. One being ended by the other: (cf. 12)

The specification for TLINK is given below.

```
attributes ::= [lid][origin](eventInstanceID | timeID) [signalID]
               (relatedtoEventInstance | relatedtoTime) relType
%lid ::= ID
lid ::= LinkID
%LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
timeID ::= IDREF
signalID ::= IDREF
relatedToEventInstance ::= IDREF
```

[2] See Allen (1984) , Allen and Kautz, (1985) for motivation.

[3] In terms of causal reasoning, these two events must be ordered rather than simultaneous.

```
relatedToTime ::= IDREF
relType ::= 'BEFORE'|'AFTER'|'INCLUDES'|'IS_INCLUDED'
            |'DURING'|'SIMULTANEOUS'|'IAFTER' |'IBEFORE' |
            'IDENTITY'|'BEGINS' |'ENDS' | 'BEGUN_BY' |
            'ENDED_BY'
```

To illustrate the function of this link, let us return to the sentence above, now adding the annotation of the TLINK, which orders the two events mentioned in the sentence.

> *John left 2 days before the attack.*

```
<TLINK  eventInstanceID="ei1" signalID="s1"
relatedToEventInstance="ei2" relType="BEFORE"/>
```

3.2 SLINK

SLINK or Subordination Link is used for contexts introducing relations between two events, or an event and a signal, of the following sort:

1. Modal: Events that introduce a reference to a possible world; these are mainly I_STATEs:
   a. *Mary wanted John to buy some wine.*

2. Factive: Certain verbs introduce an entailment (or presupposition) of the argument's veracity. They include *forget* in the tensed complement, *regret*, *manage*:
   a. *John forgot that he was in Boston last year.*
   b. *Mary regrets that she didn't marry John.*
   c. *John managed to leave the party.*

3. Counterfactive: The event introduces a presupposition about the non-veracity of its argument: *forget* (to), *unable* to (in past tense), *prevent, cancel, avoid, decline*, etc.
   a. *John forgot to buy some wine.*
   b. *Mary was unable to marry John.*
   c. *John prevented the divorce.*

4. Evidential: Evidential relations are introduced by REPORTING or PERCEPTION:
   a. *John said he bought some wine.*
   b. *Mary saw John carrying only beer.*

5. Negative evidential: Introduced by REPORTING and some PERCEPTION events conveying negative polarity:
   a. *John denied he bought only beer.*

The specification for the SLINK relation is given below:

```
attributes ::= [lid][origin][eventInstanceID]
                 subordinatedEventInstance [signalID]
                 relType
%lid ::= ID
lid ::= LinkID
%LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
subordinatedEventInstance ::= IDREF
signalID ::= IDREF
relType ::= 'MODAL'|'EVIDENTIAL'|'NEG_EVIDENTIAL'
            |'FACTIVE'|'COUNTER_FACTIVE'
```

A modally subordinating predicate such as *want* is typed as introducing a SLINK, as shown below.

> *Bill wants to teach on Monday.*

```
Bill
<EVENT eid="e1" class="I_STATE">
wants
</EVENT>
<MAKEINSTANCE eiid="ei1"
eventID="e1" tense="PRESENT" aspect="NONE"/>
<SIGNAL sid="s1">
to
</SIGNAL>
<EVENT eid="e2"
class="OCCURRENCE">
teach
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense="NONE"
aspect="NONE"/>
<SIGNAL sid="s2">
on
</SIGNAL>
<TIMEX3 tid="t1" type="DATE"
temporalFunction="true"
value="XXXX-WXX-1">
Monday
</TIMEX3>
<TLINK
eventInstanceID="ei2" relatedToTime="t1"
relType="IS_INCLUDED"/>
<SLINK
eventInstanceID="ei1" signalID="s1"
subordinatedEventInstance="ei2"
relType="MODAL"/>
```

3.3 ALINK

The ALINK or Aspectual Link represents the relationship between an aspectual event and its argument event. Examples of the possible aspectual relations that are encoded are shown below:

1. Initiation:
   John started to read.
2. Culmination:
   John finished assembling the table.
3. Termination:
   John stopped talking.
4. Continuation:
   John kept talking.

```
attributes ::= [lid][origin] eventInstanceID [signalID]
     relatedToEventInstance relType
%lid ::= ID
lid ::= LinkID
%LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
signalID ::= IDREF
relatedToEventInstance ::= IDREF
relType ::= 'INITIATES'|'CULMINATES'|'TERMINATES'|'CONTINUES'
```

To illustrate the behavior of ALINKs, notice how the aspectual predicate *begin* is treated as a separate event, independent of the logically modified event; the "phase" is introduced as the relation within the ALINK.

The boat began to sink.

```
The boat
<EVENT eid="e1" class="ASPECTUAL">
began
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1 "tense="PAST" aspect="NONE"/>
<SIGNAL sid="s1">
to
</SIGNAL>
<EVENT eid="e2" class="OCCURRENCE">
sink
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"  "tense="NONE" aspect="NONE"/>
<ALINK eventInstanceID="ei1" signalID="s1" relatedToEventInstance="ei2"
relType="INITIATES"/>
```

4 Events and Causation in TimeML

Event causation involves more than proximate (or related) temporal precedence of events. However, for a significant number of cases in text, the axioms associated with temporal ordering together with information linked to specific lexical items is sufficient for deriving causal-like inferences between events.

Causative predicates raise issues as to whether the event signaled by the causative is genuinely distinct from the event which may be the causative's logical subject. For example, in

The rains caused the flooding.

is the *cause* event distinct from the *rain* event for annotation purposes? We have identified three distinct cases of event causal relations that must be identified in texts:

1. EVENT cause EVENT

   The [**rains**] [**caused**] *the* [**flooding**].

2. ENTITY cause EVENT

   John [**caused**] *the* [**fire**].

3. EVENT. Discourse marker EVENT

   He [**kicked**] *the ball, and it* [**rose**] *into the air.*

In the current specification, we adopt the following treatment for explicit causative predicates in TimeML. For Case (1) above, we treat the causal predicate as denoting a separate event, which is identified as identical to the initial event in the logical subject position. A second TLINK establishes the precedence relation between this event and the "caused" event in object position. This is illustrated below.

The rains caused the flooding.

```
The
<EVENT eid="e1" class="OCCURRENCE">
rains
</EVENT>
<MAKEINSTANCE  eiid="ei1"
eventID="e1" tense="NONE" aspect="NONE"/>
<EVENT  eid="e2" class="OCCURRENCE">
caused
</EVENT>
<MAKEINSTANCE
eiid="ei2" eventID="e2" tense="PAST" aspect="NONE"/>
the
<EVENT eid="e3"
class="OCCURRENCE">
flooding
</EVENT>
<MAKEINSTANCE  eiid="ei3" eventID="e3"
tense="NONE" aspect="NONE"/>
<TLINK eventInstanceID="ei1" relatedToEventInstance="ei2"

relType="IDENTITY"/>
<TLINK eventInstanceID="ei2"
relatedToEventInstance="ei3"
relType="BEFORE"/>
```

For Case (2) above, there is no explicit event in subject position, hence the causal predicate alone will be temporally ordered relative to the object event, thereby obviating an “event metonymy” interpretation of the sentence (Pustejovsky, 1993).

Kissinger secured the peace at great cost.

```
Kissinger
<EVENT eid="e1" class="OCCURRENCE">
secured
</EVENT>
<MAKEINSTANCE
eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"/>
the
<EVENT  eid="e2"
class="OCCURRENCE">
peace
</EVENT>
<MAKEINSTANCE  eiid="ei2" eventID="e2"
tense="NONE" aspect="NONE"/>
at great cost.
<TLINK  eventInstanceID="ei1"
relatedToEventInstance="ei2"
relType="BEFORE"/>
```

Both solutions are adopted for verbs such as the following, in their causative senses: *cause, stem from, lead to, breed, engender, hatch, induce, occasion, produce, bring about, produce, secure.*

For Case (3) above, the annotation can optionally identify the discourse marker *and* as a signal for a `TLINK` introducing the `relType BEFORE` (and hence the reading of causation).

5 Conclusion and Future Developments

In this paper, we have reported on work done towards establishing a broad and open standard metadata markup language for natural language texts, examining events and temporal expressions. What is novel in this language, TimeML, we believe, is the integration of three efforts in the semantic annotation of text: TimeML systematically anchors event predicates to a broad range of temporally denotating expressions; it provides a language for ordering event expressions in text relative to one another, both intrasententially and in discourse; and it provides a semantics for underspecified temporal expressions, thereby allowing for a delayed interpretation. Most of the details of this last component of TimeML have, unfortunately, not been discussed in this paper.

Significant efforts have been launched to annotate the temporal information in large textual corpora, according to the specification of TimeML described above. The result is a gold standard corpus of 300 articles, known as TIMEBANK, which has been completed and will be released early in 2004 for general use. We are also working towards integrating TimeML with the DAML-TIme language (Hobbs, 2002), for providing an explicit interpretation of the markup described in this paper. It is hoped that this effort will provide a platform on which to build a multi-lingual, multi- domain standard for the representation of events and temporal expressions. We are currently working on a semantics for TimeML expressions and their compositional properties as seen in the LINK relations. This will be reported in Pustejovsky and Gaizauskas (2004). Further information may be found at `www.timeml.org`.

Acknowledgements The authors would like to thank the other members of the TERQAS and TANGO Working Groups on TimeML for their contribution to the specification language presented here: Antonio Sanfilippo, Jerry Hobbs, Beth Sundheim, and Andy Latto, as well as Andrew See, Patrick Hanks, and Bob Knippen. This work was performed in support of the Northeast Regional Reseach Center (NRRC) which is sponsored by the Advanced Research and Development Activity in Information Technology (ARDA), a U.S. Government entity which sponsors and promotes research of import to the Intelligence Community which includes but is not limited to the CIA, DIA, NSA, NIMA, and NRO. It was also funded in part by the Defense Advanced Research Projects Agency as part of the DAML program under Air Force Research Laboratory contract F30602-00-C-0168.

References

Allen, J.F. (1984). Towards a general theory of action and time. Artificial Intelligence 23, pp. 123-154.

Allen, James F., and Henry A. Kautz. 1985. “A Model of Naive Temporal Reasoning”, *Formal Theories of the Commonsense World*, ed. by Jerry R. Hobbs and Robert C. Moore, Ablex Publishing Corp., pp. 251-268.

Allen, J.F. and P.J. Hayes (1989). Moments and points in an interval-based temporal logic. Computational Intelligence 5, pp. 225-238.

Allen, J.F. and G. Ferguson (1997). Actions and events in interval temporal logic. In Oliveiro Stock (ed.), Spatial and Temporal Reasoning, Kluwer Academic Publishers, pp. 205-245.

Chinchor, Nancy, Erica Brown, Lisa Ferro and Patty Robinson, 1999 Named Entity Recognition Task Definition, MITRE, 1999.

Ferro, Lisa, Inderjeet Mani, Beth Sundheim and George Wilson, 2001, TIDES Temporal Annotation Guidelines. Version 1.0.2 MITRE Technical Report, MTR 01W0000041.

Filatova, Elena and Eduard Hovy, 2001. “Assigning Time-Stamps To Event-Clauses” in *Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing*, ACL-2001. Toulose, France, 6-11 July. pp. 88-95.

Gaizauskas, Robert and Andrea Setzer, 2002, editors, *Annotation Standards for Temporal Information in Natural Language*, LREC 2002.

Hobbs, Jerry, 2002. “An Ontology of Time”, available at DAML Website.

Ingria, Robert and James Pustejovsky, 2002. “TimeML Specification 1.0”, available at `http://time2002.org`.

McDermott, Drew, Jonathan Borden, Mark Burstein, Doug Smith, and Richard Waldinger, 2001. “A Proposal for Encoding Logic in RDF/DAML”, available at `http://www.cs.yale.edu/homes/dvm/daml/`.

Mani, Inderjeet and George Wilson. 2000, “Robust Temporal Processing of News”, *Proceedings of the ACL’2000 Conference*, 3-6 October 2000, Hong Kong.

Pustejovsky, J. 1993. “Type Coercion and Lexical Selection,” in J. Pustejovsky (ed.), *Semantics and the Lexicon*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Pustejovsky, James, Robert Gaizauskas, Roser Sauri, Andrea Setzer, Robert Ingria, 2002. “Annotation Guideline to TimeML 1.0”, available at `http://time2002.org`.

Pustejovsky, James and Robert Gaizauskas, 2004. *Time and Event Recognition in Natural Language*, John Benjamins Publishers.

Schilder, Frank and Christopher Habel, 2001. “From Temporal Expressions To Temporal Information: Semantic Tagging Of News Messages” in *Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing*, ACL-2001. Toulose, France, 6-11 July. pp. 65-72.

Setzer, Andrea, 2001. “Temporal Information in Newswire Articles: an Annotation Scheme and Corpus Study”, PhD dissertation, University of Sheffield.