NRRC Summer Workshop on Temporal and Event Recognition for Question Answering Systems

Version: 0.1.0

Release Date: 29-09-02

Authors James Pustejovsky, Luc Belanger, José Castaño, Rob Gaizauskas, Patrick Hanks, Bob Ingria, Graham Katz, Dragomir Radev, Anna Rumshisky, Antonio Sanfilippo, Roser Saurí, Andrea Setzer, Beth Sundheim, Marc Verhagen

Contents

Chapter 1

Executive Summary

1.1 Introduction

From January 30, 2002, through July 22, 2002, a workshop, funded through the National Regional Research Center (NRRC), was held at MITRE Bedford and Brandeis University. The funding was fully sponsored by the Advanced Research Development Agency (ARDA). This document reports their activities and accomplishments.

The purpose of this workshop was to address the problem of how to answer temporally-based questions about the events and entities in text, specifically news articles. 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, 2002). Many issues relating to temporal and event identification have remained unresolved, however, and it was these issues that the workshop addressed. Specifically, the workshop goals were twofold: (a) to examine how to formally distinguish events and their temporal anchoring in language (text); and (b) to develop algorithms for ordering events in text relative to each other, and the operations for computing closure over an entire discourse of events. Four basic problems in event-temporal identification were 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).

SOMETHING

1.2 Overview of Activities and Accomplishments

In the context of the workshop we focused on two efforts, reflecting the major deliverables of the contract:

1. TIMEML: Definition and Design a Metadata Standard for Markup of events, their temporal anchoring, and how they are related to each other in News articles.
2. TIMEBANK: Creation of a gold standard corpus of 300 articles marked up for temporal expressions, events, and basic temporal relations, based on the specification of TimeML.

In addition to these major deliverables, several secondary milestones were achieved, including the following:

1. Creation of Algorithms for recognizing:
   - Temporal Expressions,
   - Event Expressions
   - Times associated with Events
   - Ordering between Events and Times
2. Development of a Text Segmented Closure Algorithm
3. Creation of a Semi-graphical Annotation Tool
4. Creation of Guidelines for Annotation
5. Query Database Creation Tool and Guidelines
6. Scoring and Inter-annotator Evaluation Tool
7. Standards Impact:
   - Integration with ISO
   - Adopted by European and Japanese Projects
8. Dissemination of Knowledge:
   - CSLI /U. Chicago Press Book on TimeML
   - Special Issue of Journal on TERQAS
   - AAAI Spring Symposium 2003
   - ESSLLI 2003 Course on TimeML

The remainder of this executive summary overviews the activities and accomplishments of our research effort. Chapter 2 of this report provides greater detail. Chapter 3 is a catalogue of results and technical products of our work.

Chapter 2

Description of Workshop Activities and Technical Results

2.1 Participants and Conduct of the Workshop

Luc Belanger	Université de Montréal	Email: lucbelanger@mac.com
Bran Boguraev	IBM Research	Email: bkb@watson.ibm.com
José Castaño	Brandeis, (Project Manager)	Email: jcastano@cs.brandeis.edu
Lisa Ferro	MITRE	Email: lferro@mitre.org
John Frank	MetaCarta	Email: jrf@metacarta.com
Rob Gaizauskas	U. of Sheffield	Email: r.gaizauskas@dcs.shef.ac.uk
Patrick Hanks	Brandeis University	Email: patrick.hanks@rcn.com
Jerry Hobbs	SRI International	Email: hobbs@ai.sri.com
Robert Ingria	Brandeis University	Email: ingria@world.std.com
Graham Katz	U. of Stuttgart	Email: gkatz@uos.de
Marcia Lazo	MITRE	Email: mlazo@mitre.org
Andrew Latto	MetaCarta	Email:
Inderjeet Mani	Georgetown U., MITRE	Email: imani@mitre.org
James Pustejovsky (Team Lead)	Brandeis University	Email: jamesp@cs.brandeis.edu
Dragomir Radev	U. of Michigan	Email: radev@si.umich.edu
Anna Rumshisky	Brandeis University	Email: arum@cs.brandeis.edu
Antonio Sanfilippo	SRA	Email: antonio_sanfilippo@sra.com
Roser Saurí	Brandeis University	Email: roser@cs.brandeis.edu
Andrew See	Brandeis University	Email: aragorn@cs.brandeis.edu
Beth Sundheim	SPAWAR Systems Center	Email: sundheim@spawar.navy.mil
Marc Verhagen	Brandeis University	Email: marc@cs.brandeis.edu
Harris Wu	U. of Michigan	Email: harriswu@umich.edu

The schedule of the workshop was as follows:

1. *January 30-31, 2002*: Kick-off Meeting
2. *March 11-15, 2002*: Corpus Selection, Query Studies, TimeML
3. *April 22-26, 2002*: TimeML Specification, Corpus Work
4. *May 8-15, 2002*: Annotation Fest

5. *June 10-21, 2002*: Algorithm Specification, Annotation
6. *July 15-19, 2002*: Wrap-up and Evaluation
7. *August-September, 2002*: Prepare final report

2.2 Data Acquisition and Corpus Formation

2.3 Review of Existing Standards for Time and Event

2.3.1 TIDES

2.3.2 STAG

2.3.3 Comparing TIME Ontologies

This document provides a comparison of four time ontologies: ONTOLINGUA (simple-time), SUMO (http://ontology.teknowledge.com/rsigma/arch.html#Temporal), Jerry Hobbs time ontology for DAML (Hobbs 2002) and CYC (http://www.cyc.com/cycdoc/vocab/time-vocab.html). The goal of this comparison is to capture common traits, as well as relative strengths, weaknesses and gaps to formulate a set of recommendations for a time annotation scheme.

Following Hobbs (2002), the development of a time ontology can be seen as the combined specification of 6 areas:

1. Topological temporal relations
   (a) Basic temporal entities (e.g. instants, intervals, events)
   (b) Properties of temporal entities
   (c) Relations between temporal entities
2. Measuring durations
   (a) Turning durations into reals
3. Clock and calendar
   (a) Time zones
   (b) Clock and Calendar units
4. Deictic time
   (a) Specification of temporal concepts such as now, tomorrow, last monday
5. Sequence of temporal entities
   (a) Specification of temporal sequence concepts such as quarterly, every other monday
6. Vague temporal concepts
   (a) Specification of vague temporal concepts such as soon, later

Accordingly, our first step is to chart the specification of each area across the 4 ontologies under comparison.

Topological temporal relations

Basic temporal entities and their properties *(click on hyperlinks for additional information)*			
ONTOLINGUA	**SUMO**	**Hobbs 02**	**CYC**
Time-Point A point in real, historical time (on earth). It is independent of observer and context. Neither a measurement of time, nor a specification of time. It is the point in time. The time-points at which events occur can be known with various degrees of precision and approximation, but conceptually time-points are point-like and not interval-like. That is, it doesn't make sense to talk about what happens during a time-point, or how long the time-point lasts. Time-Range denotes a certain period of time. It consists of a start time, an end time. A start time must proceed an end time. Relations between TIME-RANGEs are defined after James Allen's interval relations. Duration denotes a period of time. It consists of a value and a measure Time-Point A point in real, historical time (on earth). It is independent of observer and context. Neither a measurement of time, nor a specification of time. It is the point in time. The time-points at which events occur can be known with various degrees of precision and approximation, but conceptually time-points are point-like and not interval-like. That is, it doesn't make sense to talk about what happens during a time-point, or how long the time-point lasts. Time-Range denotes a certain period of time. It consists of a start time, an end time. A start time must proceed an end time. Relations between TIME-RANGEs are defined after James Allen's interval relations.	TimeMeasure The class of temporal durations and Time Positions TimeDuration Any measure of length of time, with or without respect to the universal timeline TimePosition Any TimePoint or TimeInterval along the universal timeline. TimeInterval An interval of time. TimeInterval has both an extent and a location on the universal timeline. TimeInterval has no gaps, i.e. this class contains only convex time intervals TimePoint An extensionless point on the universal timeline. PositiveInfinity The TimePoint that is after all other TimePoints NegativeInfinity The TimePoint that is before all other TimePoints	temporal entity An interval or an instant Instant The start an end of an interval is itself. Negative Infinity The instant which is before any other instant Positive Infinity The instant which is after any other instant Interval proper-interval Interval whose start and end are not identical.	Event events or actions, things that we say are `happening', or changes in the state of the world. Generalization of 10 event types; each may have numerous (hierarchical) extensions. • Action ... • HolidaySeason, ... • Event-Localized ... • EconomicEvent, ... • GeneralizedTransfer, ... • AtLeastPartiallyMentalEvent ... • IntrinsicStateChangeEvent ... • ImprovementEvent ... • Compounding-WordFormationProcess ... • Conversion-WordFormationProcess ... TimeInterval A temporal thing characterized fully by its temporal attributes. For example, the year A.D. 1967. On the other hand, the event of Neil Armstrong's walking on the Moon is an Event, not a TimeInterval, since it is not fully characterized by its temporal extent or other temporal attributes. TimePoint An interval of time that has no duration (or, if you prefer, an infinitely small duration). Time-Quantity A physical quantity, corresponding to a certain amount of time Always-TimeInterval The interval of time which encompasses all time. (birthDate X Y) indicates that the Entity X came into existence during Date Y (endingDate X Y) indicates that X stopped happening sometime on date Y. (dateOfDeath X Y) indicates that the Entity X ceased to exist during Date Y (dateOfEvent EVENT DATE) says that EVENT both starts and ends during DATE. (duration TEMPTHING DURATION) means that DURATION is the length of time TEMPTHING happened or existed (measure TEMPTHING MEASURE) means that MEASURE is the total elapsed time from when TEMPTHING started to happen or exist to when TEMPTHING ended or ceased to exist. (temporallyContinuous TEMP-OBJ) TEMP-OBJ occupies one continuous chunk of time (holdsIn TEMP-THING FORMULA) the formula FORMULA is true at every moment during the TemporalThing TEMP-THING . (holdsSometimeDuring TEMP-THING FORMULA) the formula FORMULA is during the TemporalThing TEMP-THING

Relations between temporal entities *(click on hyperlinks for additional information)*

ONTOLINGUA	SUMO	JDAML	CYC
< a time point ?time-point-1 preceeds a time point ?time-point-2. Also, the sentence {tt (< tau_1 tau_2)} is true if and only if the number denoted by tau_1 is less than the number denoted by tau_2. > a time point ?time-point-1 preceeds a time point ?time-point-2. After a time range ?time-range-1 succeeds a time range ?time-range-2. After= After or Meets. Before a time range ?time-range-1 preceeds a time ranage ?time-range-2. Before= Before or Meets. Disjoint-Time-Ranges time ranges ?time-range-1 and ?time-range-2 do not overlap During time ranges ?time-range-1 and ?time-range-2 do not overlap During= During, Starts, Finishes or Equals. Equals A time point ?time-point-1 is equal to a time point ?time-point-2. A time range ?time-range-1 is identical to a time range ?time-range-2. Finishes a time range ?time-range-1 and a time range ?time-range-2 ends at the same time and a duration of ?time-range-1 is	starts (starts ?INTERVAL1 ?INTERVAL2) means that ?INTERVAL1 and ?INTERVAL2 are both TimeIntervals that have the same initial TimePoint and that ?INTERVAL1 ends before ?INTERVAL2.") finishes finishes ?INTERVAL1 ?INTERVAL2) means that ?INTERVAL1 and ?INTERVAL2 are both TimeIntervals that have the same ending TimePoint and that ?INTERVAL2 begins before ?INTERVAL1." before (before ?POINT1 ?POINT2) means that ?POINT1 precedes ?POINT2 on the universal timeline beforeOrEqual (beforeOrEqual ?POINT1 ?POINT2) means that ?POINT1 is identical with ?POINT2 or occurs before it on the universal timeline. temporallyBetweenOrEqual (temporallyBetweenOrEqual ?POINT1 ?POINT2 ?POINT3) means that the TimePoint ?POINT1 is before or equal to the TimePoint ?POINT2 and ?POINT2 is before or equal to the TimePoint ?POINT3 temporallyBetween (temporallyBetween ?POINT1 ?POINT2 ?POINT3) means that the TimePoint ?POINT2 is between the TimePoints ?POINT1 and ?POINT3, i.e. ?POINT1 is before ?POINT2 and ?POINT2 is before ?POINT3. overlapsTemporally (overlapsTemporally ?INTERVAL1 ?INTERVAL2) means that the two TimeIntervals ?INTERVAL1 and ?INTERVAL2 have a TimeInterval in common. Note that this is consistent with ?INTERVAL1 and ?INTERVAL2 being the same TimeInterval. meetsTemporally (meetsTemporally ?INTERVAL1 ?INTERVAL2) means that the terminal point of the TimeInterval ?INTERVAL1 is the initial point of the TimeInterval ?INTERVAL2.	Inside Relation between an instant and an interval In-interval Relation between instants and intervals that says that the instant is inside or the start of the interval interval-between relation between instants and intervals that says that the instant is inside or the start of the interval. The two instants are the start and end points of the temporal entity. Before If temporal-entity T1 is before temporal-entity T2, then the end of T1 is before the start of T2. After If temporal-entity T1 is after temporal-entity T2, then T2 is before the start of T1. RELATIONS BETWEEN INTERVALS (after Allen's temporal interval calculus) int-equals start-of(T1) = start-of(T2) & end-of(T1) = end-of(T2) int-before int-before(T1,T2) <-> before(T1,T2) int-after int-after(T1,T2) <-> after(T1,T2) int-meets int-meets(T1,T2) <-> end-of(T1) = start-of(T2) int-met int-met-by(T1,T2) <-> int-meets(T2,T1) int-overlaps int-overlaps(T1,T2) <-> before(start-of(T1),start-of(T2)) & before(start-of(T2),end-of(T1)) & before(end-of(T1),end-of(T2)) int-overlapped-by int-overlapped-by(T1,T2) <-> int-overlaps(T2,T1) int-starts int-starts(T1,T2) <->	(simultaneousWith T1 T2) means that TimePoints T1 and T2 occur at exactly the same time. (after LATER EARLIER) means TimePoint LATER occurs later in time than TimePoint EARLIER (startingPoint THING POINT) means that THING begins at POINT, which is the earliest moment of its temporal extent. (endingPoint THING POINT) means that THING ends at POINT, which is the last moment of its temporal extent. (startingDate TEMPORALTHING DATE) means that TEMPORALTHING started to happen or came into existence sometime on DATE (cotemporal X Y) means that X and Y have the exact same temporal extent (temporallySubsumes LONG SHORT) means that all time points of SHORT are contained in LONG. (startsDuring X Y) means that the startingPoint of X is properly contained within Y (endsDuring X Y) means that the endingPoint of X is properly contained in Y (temporallyIntersects OBJ1 OBJ2) means that there is some TimeInterval temporally subsumed by both OBJ1 and OBJ2. (startsAfterEndingOf AFTER BEFORE) the startingPoint of AFTER is later than the endingPoint of BEFORE (Allen's AFTER relation) (endsAfterEndingOf LATER EARLY) The endingPoint of LATER is later than the endingPoint of EARLY. (startsAfterStartingOf LATER-START EARLIER-START) The startingPoint of LATER-START is later than the startingPoint of EARLIER-START. (endsAfterStartingOf ENDER STARTER) means that the endingPoint of ENDER is later than the startingPoint of STARTER (temporallyCooriginating X Y) means that the startingPoint of X is simultaneousWith the startingPoint of Y. (temporallyCoterminal X Y) means that the endingPoint of X is the same as the endingPoint of Y.

shorter than that of ?time-range-2. Finishes= Finishes or Equals Meets a time range ?time-range-1 ends at the same time a time range ?time-range-2 starts. Overlaps A time range ?time-range-1 and a time range ?time-range-2 overlaps. Overlaps= Overlaps or Meets. Starts A time range ?time-range-1 and a time range ?time-range-2 starts at the same time and a duration of ?time-range-1 is shorter than that of ?time-range-2. Start= Starts or Equals.	during (during ?INTERVAL1 ?INTERVAL2) means that ?INTERVAL1 starts after and ends before ?INTERVAL2. earlier (earlier INTERVAL1 INTERVAL2) means that INTERVAL1 ends before INTERVAL2 begins. temporalPart (temporalPart ?POS1 ?POS2) means that TimePosition ?POS1 is part of TimePosition ?POS2." holdsDuring (holdsDuring ?TIME ?FORMULA) means that the proposition denoted by ?FORMULA is true in the time frame ?TIME. Note that this implies that ?FORMULA is true at every TimePoint which is a temporalPart of ?TIME." duration (duration ?POS ?TIME) means that the duration of the TimePosition ?POS is ?TIME. Note that this Predicate can be used in conjunction with the Function WhenFn to specify the duration of any instance of Physical frequency (frequency ?PROC ?TIME) means that the Process type of ?PROC recurs after every interval of ?TIME. cooccur (cooccur ?THING1 ?THING2) means that the Object or Process ?THING1 occurs at the same time as, together with, or jointly with the Object or Process ?THING2. This covers the following temporal relations: is co-incident with, is concurrent with, is contemporaneous with, and is concomitant with. time This relation holds between an instance of Physical and an instance of TimePosition just in case the temporal lifespan of the former includes the latter. The constants located and time are the basic spatial and temporal predicates, respectively	start-of(T1) = start-of(T2) & before(end-of(T1),end-of(T2) int-started-by int-started-by(T1,T2) <-> int-starts(T2,T1) int-during int-during(T1,T2) <-> (before(start-of(T2),start-of(T1)) & before(end-of(T1),end-of(T2) int-contains int-contains(T1,T2) <-> int-during(T2,T1) int-finishes int-finishes(T1,T2) <-> before(start-of(T2),start-of(T1)) & end-of(T1) = end-of(T2) int-finished-by int-finished-by(T1,T2) <-> int-finishes(T2,T1) int-in int-starts(T1,T2) v int-during(T1,T2) int-disjoint int-before(T1,T2) v int-after(T1,T2) v int-meets(T1,T2) v int-met-by(T1,T2) LINKING TIME AND EVENTS at-time relates an eventuality to an instant, and is intended to say that the eventuality holds, obtains, or is taking place at that time during relates an eventuality to an interval, and is intended to say that the eventuality holds, obtains, or is takingplace during that interval time-span-of relates eventualities to instants or intervals. For contiguous states and processes, it tells the entire instant or interval for which the state or process obtains or takes place.	(contiguousAfter AFTER BEFORE) means that the TemporalThing AFTER starts immediately following the TemporalThing BEFORE. (temporalBoundsIdentical X Y) X and Y have the same startingPoints and also have the same endingPoints. (overlapsStart FIRST SECOND) means that FIRST starts before SECOND and ends during SECOND. (temporalBoundsContain LONGER SHORTER) means that LONGER strictly contains SHORTER (temporallyStartedBy PERIOD START) means that, the startingPoint of PERIOD and START are the same and the endingPoint of START is before the endingPoint of PERIOD (temporallyFinishedBy PERIOD FINISH) means that the endingPoint of PERIOD and FINISH are the and the startingPoint of FINISH is later than the startingPoint of PERIOD. (temporalBoundsIntersect TEMP1 TEMP2) means that the continuous time interval between the start and end of TEMP1 temporally intersects the continuous time interval between the start and end of TEMP2. (temporalUnionOf X Y) indicates that Y is one of the TemporalThings which -- taken together -- define the temporal extent of X. (temporallyDisjoint X Y) means that there are no time points in common between X and Y. (startsRelativeToStartOf AFTER TIME BEFORE) means that AFTER starts duration TIME after BEFORE starts. (startsRelativeToEndOf AFTER TIME BEFORE) means that AFTER starts duration TIME after BEFORE ends. (subsumedByIntervalType FredsBirth Wednesday) means that Fred was born on a Wednesday (followingIntervalType Saturday Sunday) Sunday is contiguousAfter the Saturday. (subsumesIntervalType X Y) indicates that every instance of X temporallySubsumes some instance of Y. (intersectsIntervalType X Y) indicates that every instance of X temporallyIntersects some instance Y.

Measuring Durations

(*click on hyperlinks for additional information*)

ONTOLINGUA	SUMO	JDAML	CYC
Seconds-Of function from time points to integers representing the seconds component of the time specification. This is not the internal representation of the universal time, (e.g., number seconds since some historical date). seconds is the number of seconds past the minute, hour, day, etc. specified in the other components of the Year-Of function from time points to integers representing the year component of the time specification. The integer represents the number of years A.D., e.g., 1992. Minutes-Of function from time points to integers representing the minutes component of the time + denotes a time range ?time-range-2 whose length is	TimeIntervalFn A BinaryFunction that takes two TimePoints as arguments and returns the TimeInterval defined by these two TimePoints. BeginFn UnaryFunction that maps a TimeInterval to the TimePoint at which the interval begins EndFn A UnaryFunction that maps a TimeInterval to the TimePoint at which the interval end WhenFn A UnaryFunction that maps an Object or Process to the exact TimeInterval during which it exists. Note that, for every TimePoint ?TIME outside of the TimeInterval (WhenFn ?THING), (time ?THING ?TIME) does not hold PastFn A UnaryFunction that maps a TimePosition to the TimeInterval that meets it and that begins at NegativeInfinity ImmediatePastFn A UnaryFunction that maps a TimePosition to a short, indeterminate TimeInterval that immediately precedes the TimePosition. FutureFn A UnaryFunction that maps a TimePosition to the TimeInterval which it meets and which ends at PositiveInfinity. ImmediateFutureFn A UnaryFunction that maps a TimePosition to a short, indeterminate TimeInterval that immediately follows the TimePosition YearFn A UnaryFunction that maps a number to the corresponding calendar Year. For example, (YearFn 1912) returns the Class containing just one instance, the year of 1912. As might be expected, positive integers return years in the Common Era, while negative integers return years in B.C.E. Note that this function returns a Class as a value. The reason for this is that the related functions, viz. MonthFn, DayFn, HourFn, MinuteFn, and SecondFn, are used to generate both specific TimeIntervals and recurrent intervals, and the only way to do this is to make the domains and ranges of these functions classes rather than individuals MonthFn A BinaryFunction that maps a subclass of Month and a subclass of Year to the class containing the Months corresponding to thos Years. For example (MonthFn January (YearFn 1912)) is the class containing the eighth Month, i.e. August, of the Year 1912. For another example, (MonthFn August Year) is equal to August, the class of all months of August. Note that this function returns a Class as a value. The reason for this is that the related functions, viz. DayFn, HourFn, MinuteFn, and SecondFn, are used to generate both specific TimeIntervals and recurrent intervals, and the only way to do this is to make the domains and ranges of these functions classes rather than individuals DayFn A BinaryFunction that assigns a PositiveRealNumber and a subclass of Months to the Days within each Month corresponding to that PositiveRealNumber. For example, (DayFn 16 August) is the Class of all sixteenth days of August. For another example, (DayFn 9 Month) would return the class of all ninth days of any	Hath Hath(S,N,u,x): A set S of N calendar intervals of type u hath the calendar interval x. For example, if x is some month of September and S is the set of the successive days of that September, then "Hath(S,30,"Day",x)" would be true. Start-of From temporal entities to instants End-of From temporal entities to instants year, month, week, day, hour, minute, second Defined as functions from temporal entities to reals.	SecondsDuration Function from one or two real numbers to a Time-Quantity measured in seconds. (SecondsDuration MIN MAX) denotes the Time-Quantity of being at least MIN and at most MAX seconds in duration X-Duration Function from one or two real numbers to a Time-Quantity measured in X, for X= {minutes | hours | days | months | quarters | years | aFewMinutes | aFewDecades | aFewHours | aFewSeconds } StartFn applied to an instance THING of TemporalThing returns the TimePoint at which THING began EndFn a function that takes a TemporalThing and returns the TimePoint it ends (STIF X) returns the `Short Time Interval Following' X (STIB THING) returns the `Short Time Interval Before' THING. (TimeIntervalBetweenFn BEFORE AFTER) denotes the time interval between, but not including, BEFORE and AFTER. (IntervalBeforeFn X D) returns the time interval, of duration D, immediately preceding X. (IntervalAfterFn T-OBJ DUR) denotes the TimeInterval which immediately follows T-OBJ, lasting for duration DUR. (IntervalStartedByFn TEMP-OBJ) denotes the time interval that begins when TEMP-OBJ ends, and continues until the end of all time (Always-TimeInterval), if time has an end. (IntervalEndedByFn TEMP-OBJ) denotes the time interval which ends when TEMP-OBJ starts. The beginning of this interval coincides with the beginning of all time (Always-TimeInterval), if it has a beginning. (SecondFn S MINUTE) denotes second number

longer than ?time-range-1 by a duration ?duration. A difference between two time points ?time-point-1 and ?time-point-2 is a duration ?duration. Also: If tau_1, ..., tau_n are numerical constants, then the term {tt (+ $tau_1 ... tau_n$)} denotes the sum tau of the numbers corresponding to those constants. Duration-Of denotes a duration of a time range End-Time-Of denotes an end time of a time range Start-Time-Of denotes a start time of a time range Unit-Of denotes a unit of a time point Value-Of returns a length of a duration in a cetain measure.	month. For still another example, (DayFn 18 (MonthFn 8 (YearFn 1912))) denotes the 18th day of August 1912 HourFn A BinaryFunction that assigns a PositiveRealNumber and a subclass of Days to the Hours within each Day corresponding to that PositiveRealNumber. For example, (HourFn 12 Thursday) is the Class of all instances of noon Thursday. For another example, (HourFn 24 Day) would return the class of all instances of midnight. For still another example, (HourFn 14 (DayFn 18 (MonthFn 8 (YearFn 1912)))) denotes 2 PM on the 18th day of August 1912. MinuteFn A BinaryFunction that assigns a PositiveRealNumber and a subclass of Hours to the Minutes within each Hour corresponding to that PositiveRealNumber. For example, (MinuteFn 30 (HourFn 17 Day)) is the Class of all 5:30's in the afternoon. For another example, (MinuteFn 15 Hour) would return the class of all instances of quarter past the hour. For still another example, (MinuteFn 15 (HourFn 14 (DayFn 18 (MonthFn 8 (YearFn 1912))))) denotes 15 minutes after 2 PM on the 18th day of August 1912 SecondFn A BinaryFunction that assigns a PositiveRealNumber and a subclass of Minutes to the Seconds within each Minute corresponding to that PositiveRealNumber. For example, (SecondFn 4 (MinuteFn 5 Hour)) is the Class of all fourth Seconds of every fifth Minute of every hour. For another example, (SecondFn 8 Minute) would return the eighth second of every minute. For still another example, (SecondFn 9 (MinuteFn 15 (HourFn 14 (DayFn 18 (MonthFn 8 (YearFn 1912)))))) denotes 9 seconds and 15 minutes after 2 PM on the 18th day of August 1912 TemporalCompositionFn The basic Function for expressing the composition of larger TimeIntervals out of smaller TimeIntervals. For example, if ThisSeptember is an instance of September, (TemporalCompositionFn ThisSeptember Day 30) denotes the Class of 30 consecutive days that make up ThisSeptember.		S of minute MINUTE. (MinuteFn MINUTE HOUR) is a CalendarMinute MINUTE of the CalendarHour HOUR: (MinuteFn 12 (HourFn 18 (DayFn 14 (MonthFn February (YearFn 1966))))) is 6:12pm, Feb. 14th, 1966. (HourFn H D) denotes a CalendarHour. (DayFn DAY MONTH) denotes a CalendarDay (MonthFn MONTH YEAR) is an instance of CalendarMonth. (QuarterFn N YEAR) is the Nth CalendarQuarter (q.v.) of the CalendarYear YEAR (YearFn NUMBER) is an instance of CalendarYear. (DateAfterFn DATE DUR) returns the instance of Date which is DUR amount of time after DATE (DateBeforeFn DATE DUR) returns the instance of Date which is DUR amount of time before DATE. (DayOfMonthFn ?N) denotes the collection of CalendarDays which are the Nth day of the calendar month (DayOfYearFn ?MNTH ?N) denotes the collection of CalendarDays which are the Nth day of the month ?MNTH (MonthOfYearFn ?N) denotes the collection of CalendarMonths which are the Nth month of some year

Clock and Calendar

(click on hyperlinks for additional information)

ONTOLINGUA	SUMO	JDAML	CYC
Day-Name denotes a name of a day of a week. Day-Number denotes a day of a month. Hour-Number denotes an hour of a day Minute-Number denotes a minute of a hour Month-Name The months of the year, specified as an extensionally-defined (i.e., enumerated) set of objects, in English. Instances of this class of months are not symbols, they are months that may be denoted by object constants. Month-Number denotes a month of a year Second-Number denotes a second of a minute. Calendar-Date a specification of a point in absolute calendar time, at the resolution of one day (inherits from Time-Point) Calendar-Year a specification of a point in absolute calendar time, at the resolution of one year (inherits from Time-Point) Universal-Time-Spec a specification of a point in real-world, historical, wall-clock time, independent of timezone and with one second resolution (inherits from Time-Point) Year-Number A year expressed as the number of years A.D. Day-Name-Of denotes a name of a day of a time point specification Day-Of denotes a day of a time point Hour-Of denotes an hour of a time point Minute-Of denotes a minute of a time point Month-Name-Of denotes a name of a month of a time point Month-Of function from time points to months, representing the month component of the time specification. Months are not integers, but named objects Second-Of denotes a second of a time point	Year The Class of all calendar Years, e.g. LeapYear Month includes specification of how many days are in it. Day includes specification of day's place in a week. Hour Minute Second date A BinaryPredicate that specifies a TimePosition in absolute calendar time, at the resolution of one day, for a particular Object or Process. birthTime A BinaryPredicate that specifies, at any level of resolution, the TimePosition at which a particular Organism was born. deathTime A BinaryPredicate that specifies, at any level of resolution, the TimePosition at which a particular Organism died	year, month, week, day, hour, minute, second defined as entities CE Common era, defined as a function from time zones to intervals	• QualitativeTimeOfDay Sunrise Sunset Night Evening Midday Twilight Morning Afternoon • AcademicYear • AcademicQuarter • AcademicSemester • AcademicTrimester • Date CalendarHalfCentury CalendarQuarter CalendarSeason FiscalQuarter CalendarMinute CalendarCentury CalendarSecond CalendarDecade CalendarHour CalendarYear CalendarMonth CalendarWeek CalendarDay FiscalYear • TimeOfDay DaytimeWorkingHours TimeOfDayPM TimeOfDayAM WakingHours TheStartOfTheCommonEra the instant of time between the years BC and AD Holiday • ReligiousHoliday • LegalHoliday SeasonOfYear

Deictic time

Deictic time (*click on hyperlinks for additional information*)			
ONTOLINGUA	**SUMO**	**Hobbs 2002**	**CYC**
			Now a special TimePoint which denotes the current moment Always-TimeInterval The interval of time which encompasses all time.

Sequence of temporal entities

Sequence of temporal entities (*click on hyperlinks for additional information*)			
ONTOLINGUA	**SUMO**	**Hobbs 2002**	**CYC**

Vague temporal concepts

Vague Temporal concepts (*click on hyperlinks for additional information*)			
ONTOLINGUA	**SUMO**	**Hobbs 2002**	**CYC**
			Time-Quantity A physical quantity, corresponding to a certain amount of time • AfewMinutesDuration 2-10 • AfewDecadesDuration 2-10 • AfewHoursDuration 2-10 • AfewSecondsDuration 2-30

2.4 Creation of TimeML

2.4.1 Introduction

This document represents the current specification of TimeML. This revision specifies the syntax of TimeML, i.e. essentially its tags and their attributes, with examples illustrating their basic use. Since the pure syntax of TimeML will often leave open how a particular phenomenon should be annotated (e.g. should modals in English be marked up as SIGNALs or EVENTs), this document leaves a number of issues underspecified. Fuller discussion of the conventions by which TimeML should be applied can be found in the accompanying annotation guidelines (Pustejovsky, et al. (2002)).

The document begins with the leaf nodes of TimeML: the tags that include texts that describe the basic temporal elements within a document. The next section introduces SIGNAL, the tag that wraps expressions that specify how temporal elements should be related. The third section deals with links, empty tags that explicitly annotate the temporal relations either marked by signals or indicated purely syntactically, The next section deals with miscellaneous other tags. The last section deals with open questions.

2.4.2 Naming Conventions

Inasmuch as XML is case-sensitive, it is necessary for TimeML to specify exactly the case of all its elements. This document follows the convention of indicating tag names and attribute values in all 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. (Multi-word attribute values use the underscore character to separate their component parts.)

This document also follows the attribute naming convention introduced in Setzer (2001). Attributes that range over values of XML datatype ID—a unique index—are short, consisting of one or two characters indicating the name of the element, followed by id (e.g. tid, eiid). Attributes that range over values of XML datatype IDREF—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 must 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 IDREFs of different kinds.

Finally, in the descriptions of the values of attributes, where XML DTD and XML schema definitions would differ, the schema definition is indicated between $\{\}$.

Though this document describes the full 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 vaues) 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 a TimeML DTD corresponding to the BNF given here, using the oXygen XML editor, version 1.1.

Temporal Entities

<**EVENT**> The EVENT tag is used to annotate those elements in a text that mark the semantic events described by it. Syntactically, EVENTs are typically verbs, although event nominals, such as crash in ...killed by the crash, will also be annotated as EVENTs.

The EVENT tag is also used to annotate a subset of the states in a document. This subset of states includes those that are either transient or explicitly marked as participating in a temporal relation. See the TimeML annotation guidelines for more details.

```
attributes ::= eid class tense aspect

eid ::= ID
{eid ::= EventID
EventID ::= e<integer>}
class ::= 'OCCURRENCE' | 'PERCEPTION' | 'REPORTING' | 'ASPECTUAL'
         | 'STATE' | 'I_STATE' | 'I_ACTION' | 'MODAL'
tense ::= 'PAST' | 'PRESENT' | 'FUTURE' | 'NONE'
aspect ::= 'PROGRESSIVE' | 'PERFECTIVE' | 'PERFECTIVE_PROGRESSIVE' | 'NONE'
```

<**TIMEX3**> The TIMEX3 tag is used to mark up explicit temporal expressions, such as times, dates, durations, etc. It is modelled on Setzer's (2001) TIMEX tag, as well as the TIDES (Ferro, et al. (2002)) TIMEX2 tag. Since it differs both in attribute structure and in use, it seemed best to give it a separate name, which reveals its heritage while at the same time indicating that it is different from its forebears.

```
attributes ::= tid type [functionInDocument] [temporalFunction]
  (value | valueFromFunction) [mod] [anchorTimeID | anchorEventID]

tid ::= ID
{tid ::= TimeID
TimeID ::= t<integer>}
type ::= 'DATE' | 'TIME' | 'DURATION'
functionInDocument ::= 'CREATION_TIME' | 'EXPIRATION_TIME' |
     'MODIFICATION_TIME' | 'PUBLICATION_TIME' |
 'RELEASE_TIME'| 'RECEPTION_TIME' | 'NONE'
  {default, if absent, is 'NONE'}
temporalFunction ::= 'true' | 'false' {default, if absent, is 'false'}
{temporalFunction ::= boolean}
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 ::= IDREF
{anchorTimeID ::= TimeID}
```

```
anchorEventID ::= IDREF
{anchorEventID ::= EventID}
```

<**SIGNAL**>

```
attributes ::= sid

sid ::= ID
{sid ::= SignalID
SignalID ::= s<integer>}
```

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). This functionality of the SIGNAL tag was introduced by Setzer (2001). polarity indicators such as not, no, none, etc. indicators of temporal quantification such as twice, three times, etc.

Links

Link tags encode the various relations that exist between the temporal elements of a document. The motivations for having multiple types of links are the following:

1. To distinguish between event types and event instances, such as those introduced by conjunction, quantification, or negation.

2. To adequately handle subordinating contexts involving modality and reported speech.

<**MAKEINSTANCE**> MAKEINSTANCE is a realization link; it indicates different instances of a given event. One can create as many instances as are motivated by the text. All relations indicated by the other links are stated over these instances. Because of this, every EVENT introduces at least one corresponding MAKEINSTANCE.

```
attributes ::= eiid eventID [signalID] [cardinality]

eiid ::= ID
{eiid ::= EventInstanceID
EventInstanceID ::= ei<integer>}
eventID ::= IDREF
{eventID ::= EventID}
signalID ::= IDREF
{signalID ::= SignalID}
cardinality ::= CDATA
```

A MAKEINSTANCE can be considered to be a functional object that takes an EventID as its input and produces an EventInstanceID as its output.

signalID indicates a SIGNAL that either motivates the existence of the MAKEINSTANCE, or which indicates the value of the cardinality attribute (see annotation of John taught 20 minutes every Monday below for an example of this).

The possible value of cardinality is given as CDATA, i.e. any ASCII text. In reality, its values are most likely to range over the integers, along with a limited number of quantificational elements such as ”EVERY”, ”MOST”, etc. It may be possible to create a more constraining datatype (e.g. Cardinality), based on the string datatype, which constrains it to a fixed set of word tokens, and any sequence of digits, but we have not yet done this.

<**TLINK**> TLINK is a temporal link. It represents the relation between two temporal elements.

```
attributes ::= (eventInstanceID | timeID) [signalID]
         (relatedtoEvent | relatedtoTime) relType [magnitude]

eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
timeID ::= IDREF
{timeID ::= TimeID}
signalID ::= IDREF
{signalID ::= SignalID}
relatedToEvent ::= IDREF
{relatedToEvent ::= EventInstanceID}
relatedToTime ::= IDREF
{relatedToTime ::= TimeID}
relType ::= 'BEFORE' | 'AFTER' | 'INCLUDES' | 'IS_INCLUDED'
            | 'HOLDS'
 'SIMULTANEOUS' | 'IAFTER' | 'IBEFORE' | 'IDENTITY'
  |'BEGINS' | 'ENDS' | 'BEGUN_BY' | 'ENDED_BY'
magnitude ::= IDREF
{magnitude ::= TimeID}
```

Note that the optional magnitude attribute ranges over the IDs of TIMEX3 expressions that represent the magnitude of the relation (e.g. 5 minutes before, 10 days after, etc.).

<**SLINK**> This is a subordination link that is used for contexts involving negation, modality, evidentials, and factives. There are two cases of SLINK.

1. Cases where an event instance subordinates an event type; these are cases where a verb takes a complement and subordinates the event referred to in this complement. In these cases, the subordinated event is an event type, i.e., subordinatedEvent.

2. Cases where a subordination is introduced by a modal or negation. In these cases, the subordinated event is an instance of an event type, i.e., subordinatedEventInstance.

This link is between a matrix event instance and a subordinate event type (though in some cases, such as negation, the subordinated element is an event instance). SLINK is also used to indicate (negative) polarity, via the ”NEGATIVE” value of the relType attribute.

```
attributes ::= [eventInstanceID] (subordinatedEvent |
              subordinatedEventInstance) [signalID] relType

eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
subordinatedEvent ::= IDREF
{subordinatedEvent ::= EventID}
```

```
subordinatedEventInstance ::= IDREF
{subordinatedEventInstance ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relType ::= 'MODAL' | 'NEGATIVE' | 'EVIDENTIAL' | 'NEG_EVIDENTIAL'
          | 'FACTIVE' | 'COUNTER_FACTIVE'
```

Note that eventInstanceID is optional because an event can be subordinated (e.g. in a conditional) without being subordinated to a particular event.

<**ALINK**> 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.

```
attributes ::= eventInstanceID [signalID] relatedToEvent relType

eventInstanceID ::= ID
{eventInstanceID ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
eventID ::= IDREF
{relatedToEvent ::= EventID}
relType ::= 'INITIATES' | 'CULMINATES' | 'TERMINATES' |
                    'CONTINUES' | 'REINITIATES'
```

<**CONFIDENCE**> In various discussions of the full TERQAS groups, the utility of being able to mark confidence values for various aspects of the annotation was pointed out. In general, it would be useful to allow confidence values to be assigned to any tag, and, in fact, to any attribute of any tag.

A convenient way to do this would be to create a confidence tag, which would consume no input, and which would have the following attributes:

```
attributes ::= tagType tagID [attributeName] confidenceValue

tagType ::= CDATA
tagID ::= IDREF
attributeName ::= CDATA
confidenceValue ::= CDATA
{confidenceValue ::= 0 < x < 1}
```

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

So, for example, given this annotation:

```
(19) The TWA flight
<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="NONE">
crashlanded
</EVENT>
```

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<TLINK eventInstanceID="ei1" signalID="s1" relatedToTime="t2"
relType="BEFORE" magnitude="t1"/>
on Easter Island
<TIMEX3 tid="t1" type="DURATION" value="P2W">
two weeks
</TIMEX3>
<SIGNAL sid="s1">
ago
</SIGNAL>.

...

<TIMEX3 tid="t2" type="DATE" functionInDocument="CREATION_TIME"
value="1999-12-20">
12-20-1999
</TIMEX3>
```

if we wanted to indicate that we were unsure that we had annotated two weeks correctly, we could add this annotation:

```
<CONFIDENCE tagType="TIMEX3" tagID="t1" confidenceValue="0.50"/>
```

where 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 crashlanded was really "PAST", we could add this annotation:

```
<CONFIDENCE tagType="EVENT" tagID="e1" attributeName="TENSE"
confidenceValue="0.75"/>
```

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

Note: currently TLINKs, SLINKs, and ALINKs do not have IDs. If we want to apply confidence measures to links and/or their attributes, we will need to give each link a unique ID under this proposal.

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:

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

2.5 TimeBank Corpus

2.5.1 Creation of TIMEBANK

This is general information about the document corpora being used for TERQAS. Information on the question corpora will be found separately. Corpora to be Annotated Annotation corpora are divided into training, devtest and evaltest. Training sets are on NRRC host computer. Devtest and Evaltest data are in password-protected zip files on the NRRC host computer.

Two subsets of training sets have been identified for early annotation:

1. A set of 6 articles, 3 from ACE and 3 from DUC, were annotated by multiple TERQAS participants, with an aim to revealing significant issues with the annotation guidelines. The annotation of those 6 articles is being perfected through an adjudication process to serve as an initial TimeML gold standard.

2. A set of 44 relatively short articles are being annotated by "serious" TERQAS annotators, led by Roser Sauri and Andy See at Brandeis. This set includes 22 ACE articles (newswire and broadcast news) and 22 Propbank articles. The goal is to complete single annotation on as many of these documents as possible by the end of the workshop, preferably with some adjudication and correction done by Roser and Andy.

In addition to the total of 50 articles described above, another 250 articles have been identified for annotation by the end of September. These articles include ones from ACE, DUC and Propbank. A subset of 24 relatively short articles will be annotated by multiple TERQAS annotators, and an assessment made of interannotator agreement, using comparison and scoring software from Mitre. This set includes 5 ACE newswire, 5 ACE broadcast news, 2 DUC "Sununu", 3 DUC "Iraq", and 9 Propbank.

1. DUC (TIPSTER)
   (a) 3 Training clusters (all docs from each cluster)
      i. d03 - biography (Sununu) – total of 11 docs
      ii. d09 - single event (Iraq) – total of 16 docs
      iii. d16 - sequence of events (earthquakes) – total of 8 docs
   (b) 5 Devtest clusters (5 docs from each cluster)
      i. d25 - biography
      ii. d40 - biography
      iii. d21 - opinion
      iv. d20 - sequence of events
      v. d33 - single event
   (c) 5 Evaltest clusters (5 docs from each cluster)
      i. d42 - biography
      ii. d47 - biography
      iii. d26 - opinion
      iv. d36 - sequence of events
      v. d60 - single event

2. ACE (TDT2) : Jan-Jun 1998. These sets are available both with and without manually generated TIMEX2 tags.
   (a) Broadcast news (ABC, CNN, PRI, VOA). 100 articles (50 training, 25 devtest, 25 evaltest) Method of selection: Removed sports and pop culture articles from ACE data set, plus most of the shortest (1K) files, leaving 100 docs. Identified general topic of each doc. Each of the three TERQAS data sets includes proportional number of docs from each source (ABC, CNN, PRI, VOA). The first n docs (in alpha order by filename) were selected as training. The rest were arbitrarily assigned to devtest or evaltest, but with some conscious balancing by topic (U.S. news vs. foreign news, for example).
   (b) Newswire (AP and NYT). 99 articles (49 training, 25 devtest, 25 evaltest) Method of selection: Removed all sports stories, leaving approximately 100 docs. Every second one was assigned to training; the remaining were alternately assigned to devtest and evaltest.

3. Propbank (from Treebank2) 216 WSJ articles (166 training, 25 devtest, 25 evaltest) The Propbank/Treebank2 documents lack header information, including any reference time for the document. We tracked down the original documents from a Tipster CD, and assigned them file names that match the Propbank/Treebank2 file names, but with ".orig" (original) appended. These original files will be used for TERQAS annotation. Due to time constraints, a rigorous procedure was not used to select the articles. Most of the first ones to be annotated are taken from the shortest of the first 100 that were matched. The rest also come from the top of the list of files that were matched, but with less regard for length.

Reference Corpora

1. TIPSTER : These are the DUC clusters that are not included in the corpora to be annotated.
2. TDT2 : This is the whole corpus, including the small portions that are in the corpora to be annotated.
3. TREEBANK : Right now, all docs are included, including the Propbank articles that are in the corpora to be annotated.
4. PROPBANK : This excludes the portions that are in the corpora to be annotated.
5. REUTERS-21578
6. AP (HANKS)
7. NAMTC (LDC) – The North American News Text Corpus is a collection of journalistic text in English from newswire and newspaper sources in the United States. The sources and time periods covered by this collection are as follows: LA Times and Washington Post (May 1994 - August 1997), NY Times News Syndicate (July 1994 - December 1996), Reuters News Service, general and financial (April 1994 - December 1996), WSJ (July 1994 : December 1996)
8. ProMed
9. ENTHUSIAST dialogue corpus (appointment scheduling dialogues), with TIMEX2 tags
10. BNC (British National Corpus)

2.5.2 Features of TIMEBANK

The TERQAS collection of annotated texts (TIMEBANK consists of 300 texts with careful, detailed annotations of terms denoting events (including states and changes of state as well as actions, occurrences, and processes), temporal expressions, and signals (modals, negatives, temporal connectives, etc.) and, most importantly, links between them. TERQAS links are of three kinds: temporal (actual sequences of events), modal (possible events) and aspectual (e.g. phased predicators expressing concepts such as start, continuation, and completion).

This collection of annotated texts provides a solid empirical basis for future research into the way in which texts actually express and connect series of events, which differs in interesting respects from a priori expections. For example, annotators found that ellipsis of actual event words commonly occurs, so that for example, through metonymy, a price may represent an event, insofar as it is a change from an earlier prices. Striking differences were observed by the annotators between the typical event structure of business reports, for example in the Wall Street Journal, and the typical event structure of political, social, and military events.

Future research developments building on this foundation may be expected to included:

1. continuing development of empirically well founded algorithms for computational identification of the sequence of events in newspaper reports and other narratives. (We are exploring possibilities for refining existing algorithms in the light of empirical findings, so as to improve analysis of unseen texts, using TIMEBANK as a benchmark.)

2. relationship between event structures (as expressed in texts), arguments structures, and discourse structures. (Not part of TERQAS, but a natural and potentially fruitful corollary.)

3. relationship between causal and non-causal sequences of events.
4. relationship among multiple texts expressing different segments of a developing sequence of events.
5. processing ellipsis.

We would expect much future research to focus on what actually happens, rather than on what might possibly happen. The distinction between the probable and the possible is fundamental to empirical text analysis and will have a profound effect on the character of algorithms for practical text processing.

2.6 Creation of Query Corpus and Classification

Classes of Questions (values for temporal complexity type attribute)

This classification is based on the ways questions signal their time or event(s) dependence and on how straightforward it is to determine what time information needs to be understood from a question in order to be able to provide a suitable answer. Almost all examples are from the Excite question log, with some minor editing. Time expressions that are TIMEX2 markables are enclosed in curly braces. The classes are ordered according to a notion of simplicity in terms of presumed TimeML elements and links.

- **NO-LINK**: No TimeML link
- **SE-ST**: State-ST or Event-ST TML link
- **E-E**: Event-event TML link
- **S-T** State-Timexp link
- **T-T**: Timexp-Timexp link
- **E-T**: Event-Timexp link

2.6.1 Classes of Questions - values for temporal complexity type attribute

NO-LINK: No TML link

1. Question focus is an expression of time

   Do you have anything on {the decade from {1940} to {1950}}?

   Do you know anything about {the years 1720}-{1729}}?

   Tell me about 1950.

2. Question focus is a reference to a historical period, personal biography, etc.

   What do you know about the history of Castle Garden?

   Tell me more about Roman History.

3. Question focus is an entity (not event or state) that is located in time by a time expression

   Where can I find {40's} music I can download on my computer free?

   Who starred in the 1930 musical "Girl Crazy" ?

SE-ST: State-ST or Event-ST TML link

1. Indexical questions with events/states as question focus. In some cases, the head is a noun, like ”population”, ”price”, ”fares”, ”president”, ”weather”, etc. In sme others, the head is a verb.

   What is the population of Griffin, Georgia?

   What is the weather in Ellicotville, NY? (First reading of this seems to be time dependent.)

   WHO IS THE PRESIDENT OF VENEZUELA?

   How much francium is there in the world? (Does this Q belong in a different class?) subitem What date is the AFC Championship Game on?

   When is {Ramadan}? (this year) [note that this question is ambiguous]

   How much do dental assistants make?

   How much is the Death of Superman comic book worth?

   What company is ranked number 1 on the fortune 500 list of companies?

2. Indexical questions whose temporal dependency is signaled by Qword other than ”when”?

   How old is the bristlecone pine tree which was found in California? ii. How many people are there in the world?

   How many people live in the world?

3. The question asks for information about who is/was a person. The tense may signal whether the person is still alive or not, which could help in disambiguating name homonyms.

   Who was Johann Carl Friedrich Gauss? ii. Who is Britney Spears?

4. A superlative adjective or expressions like ”the top X” modify (a part of) the question focus, signaling the time dependency.

   Where can I find the cheapest minidisc player/recorder?

   What is the fastest car you can buy?

   Where can I find information on the newest cellular phones?

   What are the most popular hair styles?

E-E: Event-event TML link

1. In some cases, both events are expressed as full clauses.

   How old was Jackie Robinson when he died?

   Who was the ruler of Egypt when Jesus Christ was born?

   How was abortion performed before it was legalized?

   What must happen before the {Christmas} feast can begin in Poland?

   Does cider vinegar have to be refrigerated after it has been opened?

   Why do people have to stand when court starts?

   Where can I find a keychain that beeps or chirps when I clap my hands?

   Do all saints’ bodies not decompose after they die?

   How do I disable my modem speaker so it won’t make that awful noise when I connect to an online site?

   Where do i find information on john paul II’s life before he was a pope?

2. One or both of the events may be expressed as something other than a tensed clause.

   What kinds of weapons were used during the American revolutionary war?

   Is it better to eat before or after a workout?

   Is there life after death?

   What are my payment options when shopping online?

   What happened to the teacher that went ot jail after getting pregnant with a student?

   It is safe to eat salmon while pregnant?

   What president had two vice-presidents die while in office?

   How is baseball {today} different from baseball {years ago}?

   Where do i find information on john paul II's life before he was a pope?

S-T: State-Timexp link

1. Note: We view the "when" question-phrase to be a timexp.

   Who was the president of the U.S. in 1958?

   Who is the {current} leader of China?

   Who was onboard the Aeroflot plane yesterday?

   When was Eisenhower president of the U.S.?

T-T: Timexp-Timexp link

Note: We view the "when" question-phrase to be a timexp.

1. "When" question-phrase in relation to just one other timexp

   When is {Ramadan}? (in general)

   When was {the first day of winter 1999}?

   When is the next full moon?

   When does {the new millennium} begin?

2. 2. "When" question phrase in relation to two or more other timexp's.

   When is {Ramadan} {this year}?

   What time does the {winter} solstice occur for {1999}?

   What is {the last day to contribute to a Roth IRA for {1999}}?

E-T: Event-Timexp link

Note: We view the "when" question-phrase to be a timexp.

1. The timexp is a "when" question phrase

   ON WHAT DATE DID THE CHALLENGER SPACE SHUTTLE EXPLODE?

   When did the Challenger explosion happen?

   What date is the AFC Championship Game on?

   What year was the toilet invented in?

How often do you feed a pet gerbil?

How long do I cook a sweet potato?

What is the life of four AA batteries?

WHAT YEAR WAS IT WHEN THE KANSAS-NEBRASKA ACT ALLOWED FOR THE SPREAD OF SLAVERY? (Note: ”What year was it when...?” is treated here as synonym for ”in what year did...?”)

2. The timexp is not a ”when” question phrase

   What are some of the expected technological advancements of {the next century}?

   Where can I find the major events of the world during {the 1990s}?

   What are some of the greatest achievements from {the years 1000}to {2000}?

   Where can I find information on {medieval} life in a castle? (Note: This is phrased in the log as Where can I find information on life in a {medieval} castle?)

   What are some of the expected technological advancements of {the next century}?

   Where can I find the major events of the world during {the 1990s}?

   What are some of the greatest achievements from {the years 1000} to {2000}?

   How many servings of Coca-Cola were consumed in {1994}?

   What happened on {Easter}?

   Who was born on {the 18th of December}?

   What was food like in {the 1400s}?

   How can I find how many seconds have elapsed since {1970}?

   What are some activities adult can play at {Christmas}?

   Where can I find what the prime rate was for {April 1, 1999}?

   {1876} executions by hanging

   Which days in the year 2000 will banks be closed?

3. The timexp is expressed by a sequencing term, such as ”first”, ”initial”, ”latest”, etc.

   Who won the first Rose Bowl game?

   Where can I find information about the first man on the moon?

   Who was the second president of the United States?

   Where can I find the latest United Mine Workers contract?

   What was Pierre de Fermat’s last theorem?

2.7 Creation of Annotation Tools

The Alembic Workbench is a generic tool for adding SGML tags to a text. It allows annotators to define and edit tags and has a mechanism to define relations between text strings which can be used to annotate links. The Workbench was adapted to the requirements of TimeML in two ways. Firstly, a tag preferences file was added that defined the TimeML tags EVENT, TIMEX3 and SIGNAL (illustrated in figure 2.1). Secondly, a new relation TML-LINK was added.

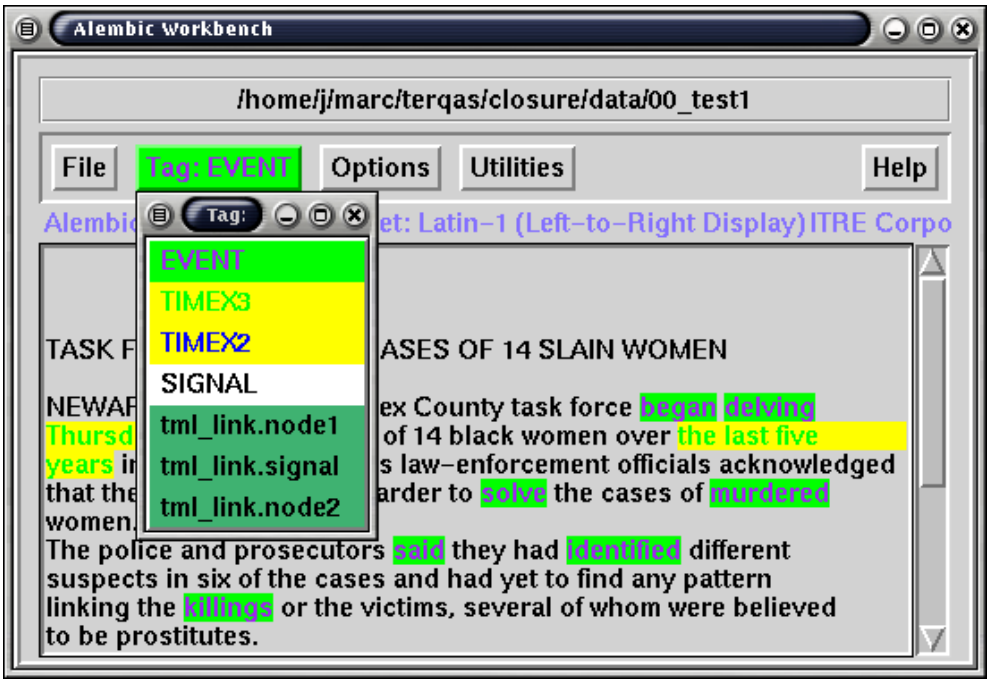


Figure 2.1: Modifications in Alembic: Annotation Options

Alembic works well for annotation of events and time expressions, but the mechanism to deal with large amounts of temporal relations is not ideal. Temporal links are displayed in a table and are added by a rather click-intensive sequence of mouse actions.

The table display does not display the temporal relation intuitively. Conceptually, TimeML annotation places events and time expressions on a time line. A graphical representation would more intuitively display temporal relations between events. Moreover, implicit relations are made visible too. Compare the difference between the table and the timeline below:

```
TABLE:     [sleep]        BEFORE  [wake_up]
           [brush_teeth]  BEFORE  [sleep]

TIMELINE:  [brush_teeth]  ------>  [sleep]  ------> [wake_up]
                          BEFORE            BEFORE
```

It requires scanning and interpreting the table to figure out that *brush_teeth* is before *wake_up*. The timeline on the other hand immediately clarifies the relation.

2.7.1 Event Diagram

The Event Diagram is a first step towards a fully editable timeline annotation tool. It is a semi-graphical annotation tool that displays spatially how a focus event is temporally related to all other events and time expressions in a text. Figure 2.2 shows how *murdered*, the event in focus, relates to all other events and time expressions. The box below the focus holds all events that are identical or simultaneous with the focus event. The boxes to the left and right hold events that precede or follow the focus event. So, in this case, *slayings* and *murdered* are identical events and the time expression *Thursday* is temporally situated after *murdered*.

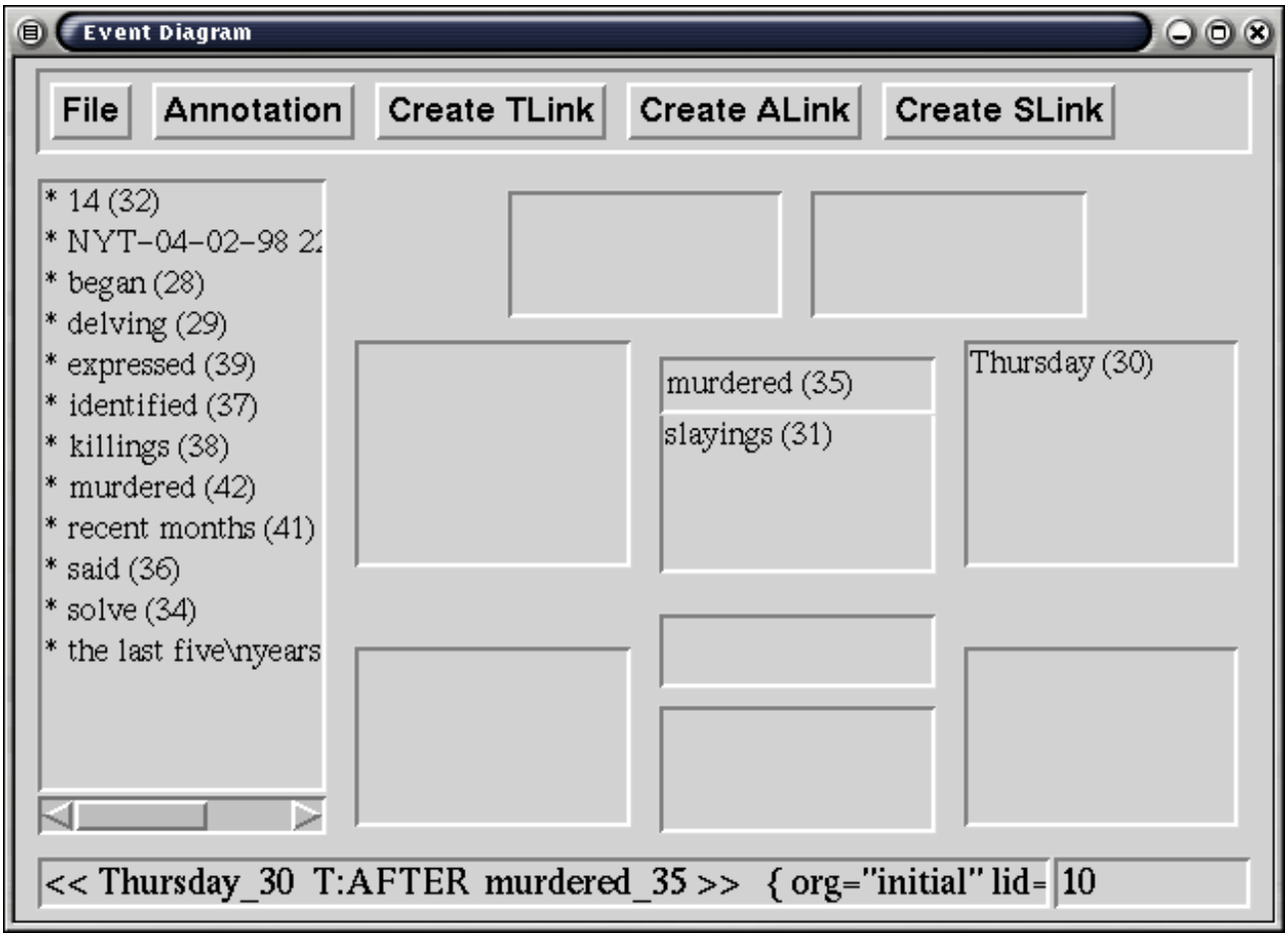


Figure 2.2: Visual Annotation: Link Tool

Adding links is much simpler and about twice as fast than in Alembic. It is also easy to change focus and follow a chain of links. In addition, it is possible to zoom in on a segment of the text, ignoring all events and time expressions outside the event.

The Event Diagram comes with an extra window that displays the text and highlights the events selected in the diagram. The text window has tag editing functionalities similar to Alembic, but adds many short cuts that make editing of events and time expressions easier.

This new tool was designed to initially complement and eventually replace Alembic. It was written in Tcl/Tk and functions as an extension to Alembic.

2.8 Algorithms for Recognizing Events and Times

2.8.1 Goals

- Extend Timex2 both in terms of coverage and functionality in Temporal Expressions and Event Recognition

- Delay Computation of Temporal Math
  Uses temporal functions and it is forseen to be integrated with Hobb's Semantic Web temporal system.

- Serve preprocessing of human annotation as well as explore automatic annotation alternatives

2.8.2 General Architecture

Modular development: It is organized in a set of cascaded processes.

- Part of Speech and Chunking Preprocessing is required: Alembic POS tagger/chunker was used.
- TIMEX and Signal Recognizer
- Event Predicates Recognizer
- Link Builder Transducer

2.8.3 Development of Temporal Functions and Temporal Expressions identification

The input consisted Alembic preprocessed XML files with POS tags, Named Entity Tags and Chunk tags. The required output was an XML file that preserved input tags (except for TIMEX2 tags that must be removed) and added the following tags:

1. TIMEX3 tags
2. Signal tags related to TIMEX3 tags
3. Temporal Function tags

TIMEX3 tags must be annotated according to the TimeML specifications resolving absolute ISO time values and bookeeping the necessary information to resolve later the ISO time of temporal expressions that are contextual dependent, which include relative references and incomplete time specifications that do not include a year, will be tagged using temporal functions. The output is a TIMEX3 tag, followed by a chain of temporal function tags.

TIMEX3 tags that can only have an actual temporal value once an anchor is determined contain the additional attributes **temporalFunction**, **valueFromFunction**, and **temporalFunctionID**.

The value of the temporalFunction attribute is a boolean, 1 for a temporal function, 0 for TIMEX3 with absolute ISO values.

The value of **valueFromFunction** is the ID of one of the corresponding **temporalFunction** tags, indicating that the time referred to by the TIMEX3 is the time that is the output of that temporal function.

The value of the anchorTimeID or anchoreEventID attributes corresponds to the id of the TIMEX3 or EVEMT that provides the input to the first of the temporal functions in the chain, that is, the anchoring temporal expression or event relative to which the time reference is resolved.

Since the temporal anchor cannot be resolved at an initial stage the output produced has a value for temporalAnchorID that is a string starting with ”unknown”, which will not be the ID of any tag in the document. A subsequent processing module would replace ”unknownxxx” by the ID of an appropriate event or TIMEX3 tag.

SIGNAL Tags related to TIMEX3 expressions have to be identified in this module.

Arguments to temporal functions are specified by specifying the ID of the argument in the argumentID attribute (or for 2-argument functions, the argumentID1 and argumentID2 attributes). For those temporal functions whose argument is an unknown anchor, the value specified will be an id starting with the string ”unknown”. In those cases where it can be determined that multiple temporal functions, or a temporal function and a TIMEX, refer to the same anchor, the same ”unknownxx” ID string will be used. In particular,

the first temporal function in the chain will have the same ”unknownxx” value for its argumentID as the TIMEX has for its temporalAnchorID value.

The following is a list of temporal functions, together with possible values for attributes, and some notes on intended semantics.

In all cases:

tfid is the attribute indicating the unique ID for this temporalFunctiontag instance.

argumentID, **argumentID1**, and **argumentID2** are used as pointers to other tags. The value specified will be either a string beginning with ”unknown”, followed by a number, or the ID of a TIMEX3, event, or another temporal function. SignalID, an optional attribute, will, when provided, give the ID of a signal tag containing the text that indicated that this temporal function should be called, in the case where this text is not itself part of the text of the TIMEX3 expression itself. In most cases, however, there will be no signalID provided, since the words signalling the temporal function to be used will be contained within the TIMEX3 tag.

2.8.4 A possible catalog of Temporal Functions

In this section we enumerate a possible catalog of temporal functions which need not be exhaustive nor complete, but representative of the kind of temporal functions that might be required or used. Given the development of possible temporal functions was performed independently from the *core* TimeML language, there might be some overlapping, or choice alternatives on how the same expressions might be tagged or how these resource can be exploited by *client* programs. Such an example are the *begin* and *end* functions.

Zero argument functions

```
<indefinitefuture tfid=/>
<indefinitepast tfid=/>
```

These are needed in the annotation of text that indicates a time period with a specified start (typically specified relative to an anchor) , but an unspecified ending, or vice versa, such as ”to date” or ”from now on”.

One argument functions

```
<CoerceTo tfid= argumentID= scale= />
```

Scale is the name of a type of time period, such as **”hour, minute,day,week,year”**. This function takes a time or time period (or an event, considered for the purpose of these functions as equivalent to the moment or if time period over which the event occurs) as argument, and returns the enclosing time period of the specified type.

```
<Predecessor tfid= argumentID= count= />
<Successor tfid= argumentID= count= />
```

The functions successor and predecessor, are analogous to each other. One moves forward in the time line and the other backwards. For example, given a time period of a standard type, such as *hour*, *day* ,etc. this function returns a time period of the same type, prior to the specified time period, preceding it my a number of those time periods specified in the numeric attribute ”value”.

If the value of the temporal function with ID tf2 is a week, the following tag represents the week located 4 weeks prior to the week denoted by tf2.

```
<Predecessor tfid=tf1 argumentID=tf2 count=4 />
```

The identity function returns the same time or time period as its argument. Needed for annotating expressions such as *now*.

```
<Identity tfid= argumentID= value= />
```

Given a time period as argument, the function **GetNamedElementOf** chooses one or more smaller time periods of the type given by ”type” from this time period. One sub-time-period is selected unless quantity is specified. When quantity is specified, it is either numeric or the special value ”every”.

```
<GetNamedElementOf tfid= argumentID= value= type= signalID= quantity=>
```

The following functions select the Monday out of its argument (typically a week) i.e., the specified time interval (would be used in annotating ”every Monday in January”, or ”Mondays in 1997”, or just ”Mondays”, with an unknown argument. :

```
<GetNamedElementOf tfid=tf1 argumentID=tf2 value=Monday type=day>

<GetNamedElementOf tfid=tf1 argumentID=tf2 value=Monday type=day quantity=every>
```

An unspecified day within the time period (used to annotate ”a day in June”:

```
<GetNamedElementOf tfid=tf1 argumentID=tf2 type=day quantity=1>
```

Three unspecified days within the time period (”three days in June”):

```
<GetNamedElementOf tfid=tf1 argumentID=tf2 type=day quantity=3>
```

Every day during the specified period (”every day in July”):

```
<GetNamedElementOf tfid=tf1 argumentID=tf2 type=day quantity=every>
```

The function **adjust** is used to indicate a modification of the argument time or time period, a sort of aproximation function.

```
<adjust tfid= argumentID= signalID= direction= quantity= value=>
```

The attribute DIRECTION has the following possible values:

```
 later|earlier|larger|smaller|unspecified
```

the first two for points in time, the last two for time periods. Value is a time unit (*day,month,hour* etc.) or unspecified, and quantity can be either a number, or ”unspecified”, or ”small”. ”Unspecified” in the direction field is for tagging things like ”about two years”, where the adjustment could be in either direction.

Examples:

“for just over two years”:

```
<Timex3 tid=t1 type=duration value=P2Y/>
<adjust tfid=tf1 argumentID=t1 direction=larger quantity=small>
```

”just after January 4, 1998”:

```
<Timex3 tid=t1 value="01041998"/>
<adjust tfid=tf1 argumentID=t1 direction=forward quantity=small>
```

Functions that extract endpoints: **BeginPoint** and **EndPoint** take an interval as an argument and return the corresponding boundary of that interval, e.g. ”The start of January” and ”The end of January” could be tagged as:

```
<TIMEX3 tid=1 temporalFunction="true" valuefromfunction=tf2 temporalAnchorID=unknow
The start of January
</TIMEX3>
<getnamedelementof tfid=tf1 argumentID=unknown1 value=January/>


<TIMEX3 tid=1 temporalFunction="true" valuefromfunction=tf2 temporalAnchorID=unknow
The end of January
</TIMEX3>
<getnamedelementof tfid=tf1 argumentID=unknown1 value=January/>
<EndPoint tfid=tf2 argumentID=tf1>
```

Two argument functions

This function takes two arguments, the start and end time of an interval, and returns the interval.

```
<maketimeperiod tfid= argumentID1= argumentID2= signal= />
```

”between 2 and 3 O’clock” could be tagged as:

```
<TIMEX3 tid=1 temporalFunction="true" valuefromfunction=tf3 temporalAnchorID=unknow
between 2 and 3 O’clock
</TIMEX3>

<getnamedelementof tfid=tf1 argumentID=unknown1 type=hour value=2/>
<getnamedelementof tfid=tf2 argumentID=unknown1 type=hour value=3/>
<maketimeperiod tfid=tf3 argumentID1=tf1 argumentID2=tf2/>
```

Samples of annotated text

Each of the following examples is considered separately. If these occured in the same document, ID fields and unknown fields would be adjusted so as to be unique throughout the document.

this week

```
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
chorID="unknown1">
This week
</TIMEX3>
<CoerceTo tfid="tf1" argumentID="unknown1" scale="WEEK"/>
```

every week

```
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
chorID="unknown1">
"every week"
</TIMEX3>
<getnamedelementof argumentID="unknown1" quantity="all" type="week"
tfid="tf1"/>
```

every week in July

```
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf2" temporalAn-
chorID="unknown1">
"every week in July"
</TIMEX3>
<getnamedelementof argumentID="unknown1" name="july" tfid="tf1"/>
<getnamedelementof argumentID="tf1" quantity="all" type="week" tfid="tf2"/>
```

every day except Monday

```
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
chorID="unknown1">
every day
</TIMEX3>
except
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf2" temporalAn-
chorID="unknown1">
Monday
</TIMEX3>
<GetNamedElementOf argumentID="unknown1" quantity="every" type="day" tfid="tf1"/>
<GetNamedElementOf argumentID="unknown1" name="monday" type="day" tfid="tf2"/>
```

three days a week

```
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf2" temporalAn-
chorID="unknown1">
three days a week
</TIMEX3>
<CoerceTo tfid="tf1" scale="WEEK" argumentID="unknown1"/>
<GetNamedElementOf argumentID="tf1" type="day" quantity="3" tfid="tf2"/>
```

between 2pm and 4pm on the last thursday of july

```
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf7" temporalAn-
chorID="unknown1">
"between 2pm and 4pm on the last thursday of july"
</TIMEX3>
<GetNamedElementOf tfid="tf1" argumentID="unknown1" type="month" value="JULY"/>
<EndPoint tfid="tf2" argumentID="tf1"/>
<CoerceTo tfid="tf3" scale="WEEK" argumentID="tf2"/>
<GetNamedElementOf tfid="tf4" argumentID="tf3" type="day" value="thursday"/>
<GetNamedElementOf tfid="tf5" argumentID="tf4" type="hour" value="2PM"/>
<GetNamedElementOf tfid="tf6" argumentID="tf4" type="hour" value="4PM"/>
<MakeTimePeriod tfid="tf7" argumentID1="tf5" argumentID2="tf6"/>
```

in the next ten years

```
<signal sid="1">
in
<signal/>
```

```
the
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf3" temporalAn-
chorID="unknown1">
next ten years
</TIMEX3>
<CoerceTo tfid="tf1" scale="YEAR" argumentID="unknown1"/>
<Successor tfid="tf2" count="10" argumentID="tf1"/>
<MakeTimePeriod tfid="tf3" argumentID1="unknown1" argumentID2="tf2"/>


<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf3" temporalAn-
chorID="unknown1">
next Monday
</TIMEX3>
<CoerceTo scale="week" tfid="tf1" argumentID="unknown1"/>
<Successor tfid="tf2" argumentID="tf1" count="1" />
<GetNamedElementOf tfid="tf3" argumentID="tf2" value="monday" type="DAY"/>

<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf2" temporalAn-
chorID="unknown1">
every Monday in January
</TIMEX3>
<GetNamedElementOf tfid="tf1" argumentID="unknown1" value="January" type="month"/>
<GetNamedElementOf tfid="tf2" argumentID="tf1" value="Monday" type="day" quan-
tity="every"/>

<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf2" temporalAn-
chorID="unknown1">
a day in June
</TIMEX3>
<GetNamedElementOf tfid="tf1" argumentID="unknown1" value="June" type="month"/>
<GetNamedElementOf tfid="tf2" argumentID="tf1" type="day" quantity="1"/>

<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf2" temporalAn-
chorID="unknown1">
three weeks in August
</TIMEX3>
<GetNamedElementOf tfid="tf1" argumentID="unknown1" value="August" type="month"/>
<GetNamedElementOf tfid="tf2" argumentID="tf1" type="week" quantity="3"/>

<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf2" temporalAn-
chorID="unknown1">
every day in July
</TIMEX3>
<GetNamedElementOf tfid="tf1" argumentID="unknown1" value="July" type="month"/>
<GetNamedElementOf tfid="tf2" argumentID="tf1" type="day" quantity="every"/>

<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
```

```
chorID="unknown1">
for just over two years
</TIMEX3>
<TIMEX3 id="2" temporalFunction="false" type="duration" value="P2Y"/>
<Adjust tfid="tf1" argumentID="2" direction="larger" quantity="small"/>
```

Note that in this and the following examples, since we cannot nest Timexes, the internal time reference is instead annotated as a TIMEX that does not consume text.

```
<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
chorID="unknown1">
for over three months
</TIMEX3>
<TIMEX3 id="2" temporalFunction="false" type="duration" value="P3M"/>
<Adjust tfid="tf1" argumentID="2" direction="larger" quantity="unspecified"/>

<TIMEX3 id="1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
chorID="unknown1">
for about two years
</TIMEX3>
<TIMEX3 id="2" temporalFunction="false" type="duration" value="P2Y"/>
<Adjust tfid="tf1" argumentID="2" direction="unspecified" quantity="small"/>

<TIMEX3 id="t1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
chorID="unknown1">
on or about January 4, 1998
</TIMEX3>
<TIMEX3 tid="t2" temporalFunction="false" value="01041998"/>
<Adjust tfid="tf1" argumentID="t2" direction="unspecified" quantity="small">

<TIMEX3 id="t1" temporalFunction="true" valueFromFunction="tf1" temporalAn-
chorID="unknown1">
just after January 4, 1998
</TIMEX3>
<TIMEX3 tid="t2" temporalFunction="false" value="01041998"/>
<Adjust tfid="tf1" argumentID="t2" direction="forward" quantity="small">

<TIMEX3 tid="1" temporalFunction="true" valuefromfunction="tf2" temporalAn-
chorID="unknown1">
The start of January
</TIMEX3>
<GetNamedElementOf tfid="tf1" argumentID="unknown1" value="January"/>
<BeginPoint tfid="tf2" argumentID="tf1">

<TIMEX3 tid="1" temporalFunction="true" valuefromfunction="tf3" temporalAn-
chorID="unknown1">
The last day of February
</TIMEX3>
```

```
<GetNamedElementOf tfid="tf1" argumentID="unknown1" value="February"/>
<EndPoint tfid="tf2" argumentID="tf1">
<CoerceTo scale="week" tfid="tf3" scale="day" argumentID="tf2"/>

<TIMEX3 tid="1" temporalFunction="true" valuefromfunction="tf3" temporalAn-
chorID="unknown1">
between 2 and 3 O'clock
</TIMEX3>
<GetNamedElementOf tfid="tf1" argumentID="unknown1" type="hour" value="2"/>
<GetNamedElementOf tfid="tf2" argumentID="unknown1" type="hour" value="3"/>
<MakeTimePeriod tfid="tf3" argumentID1="tf1" argumentID2="tf2"/>

for
<TIMEX3 tid="1" temporalFunction="true" valuefromfunction="tf1" temporalAn-
chorID="unknown1">
approximately 14 hours per day
</TIMEX3>
<Adjust tfid="tf1" argumentID2="tf2" direction="unspecified" quantity="small"/>
<GetNamedElementOf tfid="tf2" value="hour" quantity="14" argumentID="tf3"/>
<GetNamedElementOf tfid="tf3" value="day" quantity="every" argumentID="unknown1"/>

<TIMEX3 tid="1" temporalFunction="true" valuefromfunction="tf1" temporalAn-
chorID="unknown1">
To date
</TIMEX3>
<MakeTimePeriod argumentID1="tf1" argumentID2="unknown1"/>
<indefinitefuture tfid="tf1"/>

<TIMEX3 tid="1" temporalFunction="true" valuefromfunction="tf1" temporalAn-
chorID="unknown1">
now
</TIMEX3>
<Identity tfid="tf1" argumentID="unknown1"/>
```

2.8.5 Event Recognition and S-Link builder

Event Recognition uses verbal chunks information to recognize target event heads and temporal/aspectual information.

The following alternatives were also exploited to obtain nominal events:

- Morphological information (POS) ambiguity.
- Contextual clues (presence of signals).
- Semantic Information (Wornet Synset information).

Modal Negation S-Links were also identified using the verbal Chunk information.

2.8.6 Temporal Link Builder

This module modifies XML parse tree built by xmlparse.py using xml.dom and xml.sax packages for Python. It assumes markup of TIMEXes, EVENTs, and EVENT types, as well as shallow parsing of noun phrases. Adds TLinks and SLinks according to TimeML.

Heuristics:

H0:

i. Insert BEFORE TLINK to DCT (document creation time) for all past EVENTS in the document (includes EVENTS marked as present perfect).

ii. Insert AFTER TLINK to DCT for all future EVENTS in the document.

H1: (default heuristic)

i. Insert IS_INCLUDED TLINK from TIMEX to an EVENT immediately to the left of TIMEX

ii. If no EVENTS to the left of TIMEX are found in the same sentence, insert IS_INCLUDED TLINK to the EVENT immediately to the right of TIMEX within the same sentence.

H2:

If a TIMEX is followed by nominal EVENT that belongs to the same noun group as the TIMEX (e.g. The <TIMEX> <nominal event>), insert IS_INCLUDED TLINK between TIMEX and EVENT.

`II. EVENT-to-EVENT SLinks`

Heuristics:

H1: (default EVIDENTIAL)

i. For REPORTING events, insert EVIDENTIAL SLINK to next tensed OCCURRENCE EVENT in the same sentence.

ii. If no OCCURRENCE EVENTS follow the REPORTING EVENT, insert EVIDENTIAL SLINK to the first *preceding* tensed OCCURRENCE EVENT in the same sentence.

H2:

For any infinitival or nominal EVENT, insert MODAL SLINK to the EVENT preceding it in the same sentence, if a "to" SIGNAL is found in between, and no noun group separates the two EVENTS.

2.8.7 Suggested Future Work

- EVENT-to-TIMEX TLINKS

Recognizing types of EVENT-to-TIMEX TLINKS will require a more extensive implementation of SIGNAL recognition as well as the recognition of DURATION TIMEXES.

- Extrasentential Event-Timex Links

The current algorithm will need to be modified to include the mark-up of inter-sentential (sentence-external) EVENT-to-TIMEX TLINKS for TIMEXES other than DCT.

- Event-Event Links

  Intrasentential SLINKS (e.g. Evidential and infinitivals)

  Extrasentential exploiting Discourse information.

The last would require the incorporation of Reference Resolution mechanisms and keep track of Tense sequences.

Completing the previous work would require:

Next Steps

- Complete Timex
- Complete Event Recognition
- Develop Signal Recognition
- Develop Event Class Recognition
- Reference/Anchor Time Recognition
- Evaluation against TimeBank

2.9 Graphical Interface: Display of Results

2.9.1 Introduction

The graphical visualization tool was developed to represent TimeML documents graphically. Other visualization tools are based on the idea that time and event expressions must lie on a timeline, but this is not the case with the tool developed here. The timeline paradigm is a restrictive one and cannot represent all relations annotated with the TimeML annotation language. The graph that is drawn by the visualization tool represents the time in a topological way, not necessarily in a geometrical way as on a timeline.

The main task of the tool is to extract information carried by the annotation of the document. Each tag gives information about the role played by terms and expressions analyzed during the annotation. `EVENT`, `MAKEINSTANCE` and `TIMEX3` tags are used to generate nodes of the graph. Main attributes of each one are inserted in the graphical representation of each node to ease the interpretation of the graph. The other tags, namely `TLINK`, `SLINK` and `ALINK`, are there to link nodes and they can be viewed as relations between temporal elements of the document. When the extraction of elements is done, entities are analyzed to be represented in a internal graph structure from which we will rewrite the graph in the graphviz file format. Once this representation is created, the only process remaining is to send that representation to dot, a tool from the graphviz toolset.

2.9.2 Entities Representation

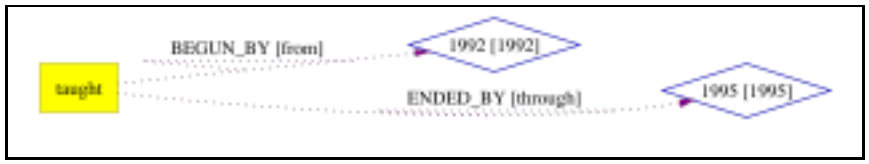
EVENT and MAKEINSTANCE

`EVENT` and `MAKEINSTANCE` tags are drawn as rectangles, with a clear or a solid background according to the relation they have. `EVENT` tags must be treated with caution, since they are closely related to the `MAKEINSTANCE` tags. The `MAKEINSTANCE` tells us that an event happened, it is the realization of an event. Since an `EVENT` can have a single, multiple or an infinity of instances, `MAKEINSTANCE` tags must differ in the way they realize the `EVENT`.

When an `EVENT` have only one `MAKEINSTANCE` related to it, both nodes are merged into one, as in figure 2.3. In this example, there is only one instance of the *taught* `EVENT`.

```
<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="NONE">
taught
```

```
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
```

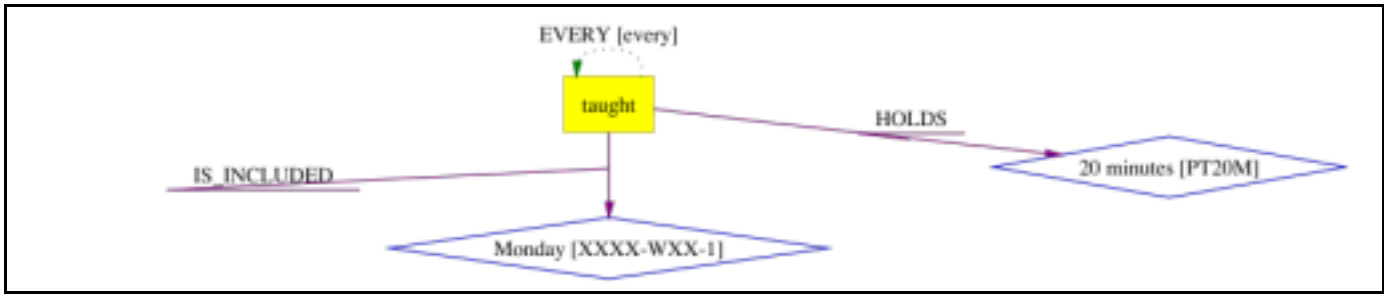


John taught from 1992 through 1995.

Figure 2.3: Example 8

When the EVENT occurs twice or more, one EVENT node is created and the number of instances nodes realizing the EVENT is created. An example of this case can be seen in figure 2.5, where the *taught* EVENT is realized two times on Monday, indicated by the SIGNAL *twice*, and one time on Tuesday. The MAKEINSTANCE link is drawn in green, and dotted when there is a SIGNAL motivating the realization.

Since three instances of the EVENT can have different properties, they must have their own graphical representation. When an EVENT do not have a countable number of instances, the EVENT is drawn as a single node with a solid background and a signal link pointing back. In figure 2.4, *every* is a SIGNAL telling us that the EVENT possesses multiple instances.



John taught 20 minutes every Monday.

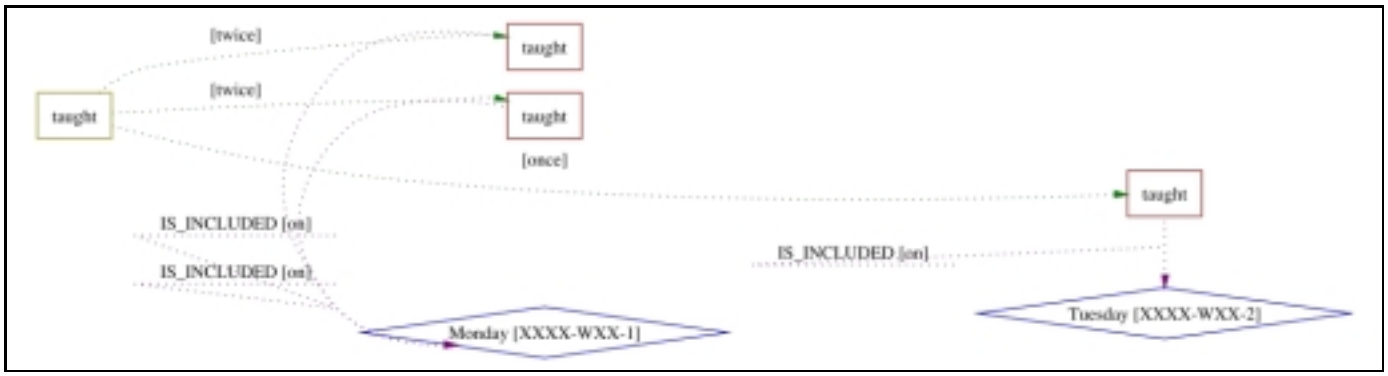
Figure 2.4: Example 4

TIMEX3

A TIMEX3 tag is represented by a diamond. The annotated expression is written inside the diamond and its value is written between square bracket. TIMEX3 tags are all treated equally except for one thing, when the instance of the TIMEX3 node represents the document creation time, the node is drawn as a little house (a flat pentagon), like the one in figure 2.6.

Timex3 tags are the anchor of the graph, their ordering governs the layout of the graph. Their nodes are supposed to be laid out in a chronological way from left to right. For now it is not really the case because the *value* attribute is frequently underspecified. When the value is specified correctly, nodes are drawn in the right order, like the ones in figure 2.5, where the *value* attribute clearly indicates that Monday [XXXX-WXX-1] is before Tuesday [XXXX-WXX-2].

```
<TIMEX3 tid="t1" type="DATE" temporalFunction="true" value="XXXX-
WXX-1">
```



John taught twice on Monday but only once on Tuesday

Figure 2.5: Example 6

John taught last week.

Figure 2.6: Example 10

```
Monday
</TIMEX3>
...
<TIMEX3 tid="t2" type="DATE" temporalFunction="true" value="XXXX-
WXX-2">
Tuesday
</TIMEX3>
```

Moreover to represent the temporal relations in a consistent way, the tool need to have a module to calculate the events and temporal marquees order. The problem we have to deal with is the order of magnitude of the annotation. How to compare two values not measured on the same scale, for example *last year* and *at 9 a.m.*. To solve this problem, one approach could be the use of Hobb's temporal math, but this has not been tried yet in our case.

SIGNAL

`SIGNAL`s tags are used to qualify the relation between `EVENT` and `TIMEX3`, they are seen on the graphs as attribute on the relations arcs. When a `SIGNAL` is an attribute of a relation, it is written on the arcs between square brackets. Relations with a signal are drawn as dotted arc, this facilitate the visualization of "signaled" relations.

TLINK

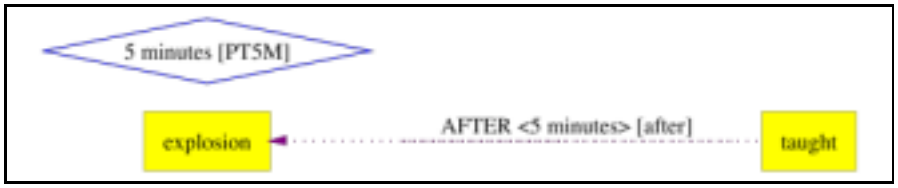
The TLINK is a temporal relationship between EVENTs or between an EVENT and a TIMEX3. TLINKs are part of the ordering procedure since they carry attributes marking relations, such as the inclusion (vertical ordering) and precedence (horizontal ordering). TLINK relations are drawn in purple, and point from the eventInstanceID towards the item that they are related to.

Figure 2.4 shows us two TLINKs, one represents the inclusion of the EVENT node in the TIMEX3 node, and the other represents the fact that the event has a duration. This relation is one of the hardest to represent graphically because we must be able to tell that the period, when the EVENT holds, is included in the other TIMEX3 node.

The other relation we have not seen, is the event ordering one, which is one that is depicted very well in figure 2.7.

```
<TLINK eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="AFTER" magnitude="t1"/>
```

The attribute value ei1 is the *taught* EVENT, s1 is the SIGNAL *after*, which is the type of relation the link carries and the magnitude of the TLINK is *5 minutes*. From the graph, one can easily see that the explosion occurred 5 minutes before the taught event because the purple dotted arc indicates it. The arc also makes explicit the magnitude (5 minutes) of the link, drawn between the angle brackets.



John taught 5 minutes after the explosion.

Figure 2.7: Example 7

SLINK

A SLINK introduces a subordination relation, it can be a positive or a negative relation. SLINKs introducing negative relation are drawn in red. relType's attribute value of NEGATIVE, NEG_EVIDENTIAL or COUNTER_FACTIVE are considered to be of a negative type. Positive SLINK has a relType attribute value of MODAL, EVIDENTIAL, FACTIVE, COUNTERFACTIVE, and is drawn in blue. When the SLINK relation is between an EVENT and a SIGNAL, there will only be an arc pointing back on the EVENT, with the SIGNAL in square brackets. Some examples of SLINK relations can be seen in figure 2.8 and figure 2.9.

ALINK

ALINKs are maybe the easiest to draw. They are established only between two EVENTs and they model relations that can be simplified as an EVENT beginning or ending a second EVENT. They are drawn as orange arc. An INITIATE relation can be seen in figure 2.10 and a TERMINATES relation in figure 2.11.

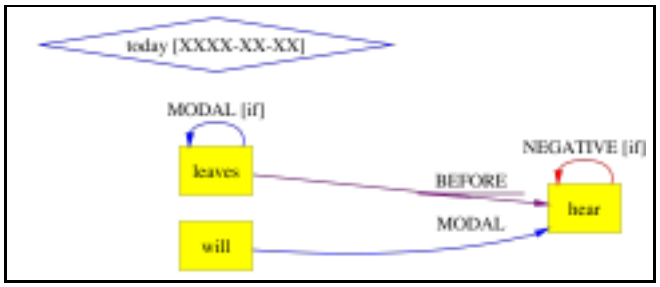


Figure 2.8: Example 13

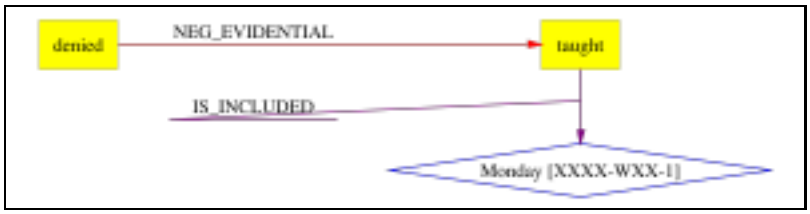


Figure 2.9: Example 14

2.9.3 Conclusion

The graphical tool was not designed to be a ”temporal calculator”, it was first thought and implemented as a XML Stylesheet Transformation (XSLT), but it ended as a Ruby script and its own set of classes. The tool is useful for visualizing small examples. When the document size becomes larger, the number of nodes and relations makes the graph almost unreadable. One of the great utility of the tool is to debug annotation. Since almost all annotation was done by hand, with an annotation tool not supporting directly the TimeML language, many error were found by the analysis stage of the tools. Once the graph is drawn, we can easily find semantic errors made by the annotator.

There is a lot of things missing for this tool to be really useful. First, it must combine the closure tool developed by Marc Verhagen, and second it would be of a great utility to be able to modify the annotation by modifying the graph. Some things that are to be evaluated for a future version of the tool, are a new backend for drawing the graph, and maybe a new way of viewing the graph, such as using a zoom based on the magnitude of the event, a fisheye view or something like that, in order to be able to have a local and a global view at the same time.

2.10 Time Event Closure Algorithm

The temporal annotation of a document should be as informative as possible. Ideally, we want reliable temporal relations between all events and time expressions. That is, the set of temporal relations should be complete and consistent. Unforunately, annotation of temporal relations in a document presents several practical challenges due to the complexity of temporal annotation:

- High Density

  The set of possible temporal relations is quadratic to the number of events and time expressions

The boat began to sink.

Figure 2.10: Example 17

The search party stopped looking for the survivors.

Figure 2.11: Example 18

in a document. A typical TimeBank document contains about 50 temporal objects, which implies 2500 possible temporal relations. Larger documents may allow over 10,000 relations. Achieving completeness with manual annotation alone is clearly impractical.

- Low Markup Speed

  Temporal annotation is slow. It can take several hours to annotate an average TimeBank article.

- Inconsistencies in Annotation

  Even trained annotators introduce inconsistencies regularly, usually as a result of vague or ambiguous temporal relations between events.

In order to tackle these problems we developed two tools: a semi-graphic annotation tool and a closure tool. The annotation tool was described earlier in this report. The closure tool intends to help cope with all three practical challenges above.

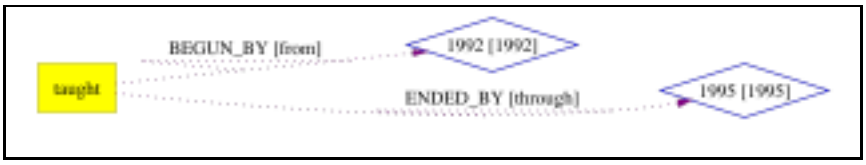
The closure tool helps the annotator achieve partial completeness for the set of temporal relations in a text. The underlying algorithm identifies what the minimal partially complete set of temporal relations is and uses two mechanisms to create that set: (i) computing the transitive closure of the initial set of temporal relations that the annotator added, and (ii) prompting the annotator for new links, followed by another transitive closure computation.

Note that we are only aiming for partial completeness. The complexity of full completeness is quadratic to number of events and time expressions in the document whereas partial completeness is quadratic to number of events and time expressions in the segments, and linear if the segment size is fixed. Full completeness is unpractical even with a closure algorithm that generates up to 90% of all links in a document.

A *locally complete temporal annotation* of a document is defined as an annotation where all events and time expressions are linked to all other events and time expressions within their local context. This relaxed completeness does not require the annotator to fill in all the relations that the closure algorithm cannot derive axiomatically. User prompting is now linear to the number of events instead of quadratic.

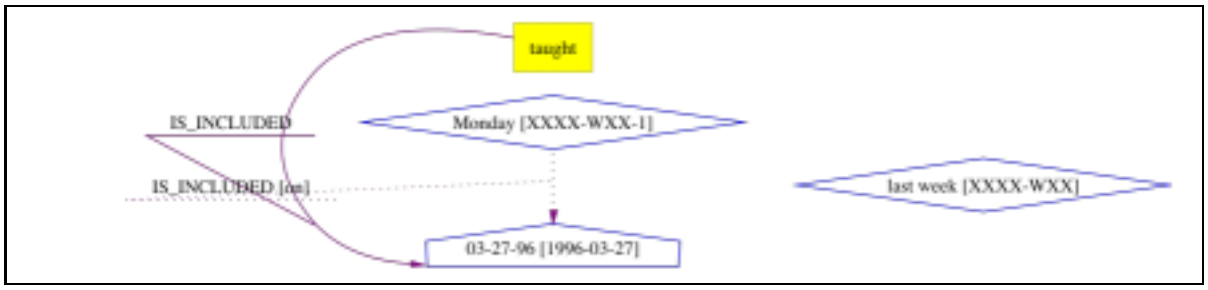
Partial completeness can be achieved using a text segmented closure algorithm. This algorithm is decribed below.

We do not only want our annotation to be partially complete, we also want it to be *consistent*. A consistent annotation does not contain relation pairs that are mutually exclusive. A new relation is consistent with an existing annotation if adding the relation to the annotation generates a consistent annotation. Using



John taught from September to December last year

Figure 2.12: Example 9



John taught last week on Monday.

Figure 2.13: Example 11

temporal closure to generate new links tends to unearth inconsistencies introduced by the annotator and therefore helps increase the quality of the annotation. For example, imagine that an annotator added the following three links:

1. `<x,before,y>`
2. `<y,before,z>`
3. `<x,after,z>`

The closure algorithm will generate the relation `<x,before,z>` which is inconsistent with the third link above.

A final advantage of using closure is that it greatly increases the numbert of links in a document and therefore makes it easier to measure inter-annotator agreement.

2.10.1 Algorithm & Axioms

Recall that TimeML has three kinds of links: TLINKs, ALINKs and SLINKs. The algorithm completely ignores SLINKs. ALINKs and TLINKs are both used to derive new links but new links are always TLINKs.

The input to the closure algorithm is a user-annotated article with TimeML markup. Sentence tags are required since they are used to define local context. All other tags are ignored. The stages of processing are:

1. Perform initial closure on all links added by the annotator.
2. Alert the user to potential identity chains. This is the only occasion where a user may be asked to specify a non-local relation.
3. Create a sliding window of three sentences. Initially, the window will consist of sentences one through three. The sliding window implements the local context. The size of the sliding window can be parameterize.

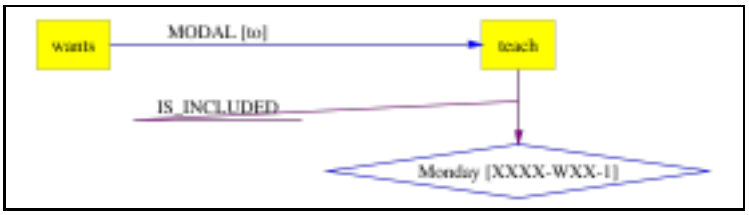


Figure 2.14: Example 15

4. Prompt the user to specify a relation type for two time objects that are not yet linked within the local context. If no temporal relation exists, the annotator may specify ”unknown”.

5. After each added relation, recompute the closure using the new fact. Do this till all time objects within the local context are related.

6. If all objects in the local context are related, move the window up one sentence. For example, if the previous local context was made up of sentences 3-5, then the next local context for the closure algorithm is sentences 4-6. Start prompting the user for the new context.

Closure on a set of links Θ proceeds by parewise comparing all links in Θ. Comparing two links involves checking all axioms and determining whether axiom application generates a new temporal fact. All new derived facts are put on the queue, popped off and then compared to all links in Θ.

Initially, all links provided by the annotator are put on the queue and popped off one by one.

The axioms work with a normalized set of temporal relations. For example, links with the relation types before, after, ibefore, and iafter are normalized as follows:

```
<x,before,y>  => <x,PRE,y>
<x,ibefore,y> => <x,PRE,y>
<x,after,y>   => <y,PRE,x>
<x,iafter,y>  => <y,PRE,x>
```

Other normalized relations are SIM (simultaneity), IDT (identity), and INC (inclusion). Some of the axioms that govern precedence relations (PRE) are listed below. Each axiom, or group of axioms is accompanied by a time line that depicts the events/states in the axiom.

```
PRE1:  [ x PRE y  &  y PRE z  =>;  x before z ]

   ----x----  ----y----  ----z----

PRE2:  [ x PRE y  &  y SIM z  =>  x before z ]
PRE3:  [ x PRE y  &  y IDT z  =>  x before z ]

   ----x----  ----y----
              ----z----

PRE4:  [ x PRE y  &  x SIM z  =>  z before y ]
PRE5:  [ x PRE y  &  x IDT z  =>  z before y ]
```

```
    ----x----  ----y----
    ----z----

PRE6:  [ x PRE y  &  x INC z  =>  z before y ]

    ----x----  ----y----
      --z--
```

Note that the left-hand side of the axioms contain normalized relations, whereas the derived temporal relation is one of the link types defined in TimeML.

Experiments indicated that with closure, an annotator can cover about 65% of all possible links in a document. Of those links, 84% were derived by the algorith:

initial links	36	4%
user prompts	109	12%
derived links	775	84%

Initial links were those put in by an annotator before closure was switched on. The user prompts are the occasions where the algorithm asked the annotator to fill in the holes.

2.11 Annotation Guideline for TimeML

This section describes the annotation guidelines for marking up text according to the TimeML language version 1.0. The section is organized as follows. The first section explains what TimeML tags (XML elements) are and how to annotate them, while also specifying for each tag what its attributes are and provides a BNF definition for the tag and its attributes. 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. A next section provides some fully annotated examples, illustrating all of the interactions between the various entity and relational tags.

2.11.1 Explanation of TimeML Tags and their Attributes

The tag `<EVENT>`

We consider "events" a cover term for situations that *happen* or *occur*. Events can be punctual (1-2) or last for a period of time (3-4). We also consider as events those predicates describing *states* or *circumstances* in which something obtains or holds true (5-6). Not all stative predicates will be marked up, however.

1. *Ferdinand Magellan, a Portuguese explorer, first* **reached** *the islands in search of spices.*
2. *A fresh flow of lava, gas and debris* **erupted** *there Saturday.*
3. *11,024 people, including local Aeta aborigines,* **were evacuated** *to 18 disaster relief centers.*
4. *' 'We're* **expecting** *a major eruption," he said in a telephone interview early today.*
5. *Israel has been scrambling to buy more masks abroad, after a* **shortage** *of several hundred thousand gas masks,*

Events are generally expressed by means of tensed or untensed verbs (1 and 2), nominalizations (3), adjectives (4), predicative clauses (5), or prepositional phrases (6):

1. *A fresh flow of lava, gas and debris* **erupted** *there Saturday.*
2. *Prime Minister Benjamin Netanyahu called the prime minister of the Netherlands* **to thank** *him for thousands of gas masks his country has already contributed.*
3. *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**.
4. *A Philippine volcano,* **dormant** *for six centuries, began exploding with searing gases, thick ash and deadly debris.*
5. *"There is no reason why we would not* **be prepared**," *Mordechai told the Yediot Ahronot daily.*
6. *All 75 people* **on board** *the Aeroflot Airbus died.*

Note that in the previous sentences not all "markables" are tagged. In the first example, for instance, neither *flow* nor *Saturday* are marked.

The annotation of formally simple events (examples 1, 3, 4 and 6 above) is straightforward. Assume for now that events have only an id attribute:

```
A fresh flow of lava, gas and debris
<EVENT eid="e1">
erupted
</EVENT>
there Saturday.
```

However, formally complex events may be sequentally discontinuous in some contexts:

1. *There is no reason why we would not* **be prepared.**
   There is no reason why we would not **be** *fully* **prepared.**
2. *They will definitely* **take into consideration** *our readiness.*
   They will definitely **take** *it* **into consideration**.

In order to avoid problems derived from tagging material of diverse scope, employ the following strategies:

- If the event is expressed by a verbal phrase (*has been scrambling, to buy, were reported*), the EVENT tag will be applied only to its head, which is marked in bold face in the following examples:

  Israel has been **scrambling** *to* **buy** *more masks abroad.*
  No injuries were **reported**.
  The private sector could **establish** *a private agency.*
  If we had been **attacked** *, ...*

  If the main verb falls within the scope of a modal auxiliary, as in the third example above, or a negative particle as in

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

then the EVENT tag is only applied to the main verb, but the modal and/or negation is tagged as a signal indicating a subordinated context – see sections 2.11.4 and 2.11.10.

- If the main predicate is a phrasal verb, only the verbal part (and not its particle) will be marked:

  Additional distribution centers would be **set** *up next week.*

- If the event is expressed by a verbal cluster which consists of an aspectual and a main verb, both verbs will be tagged as independent events. Examples of aspectual verbs are: *begin, stop, end, keep,* etc.

- If the aspectual verb is preceeded by auxiliary forms (as in the last two examples below), the first criterion above will be applied as well. In the following examples, the underlining delimits the verbal cluster, whereas square brackets signal the scope of each EVENT tag:

  The private sector **[began] [establishing]** *a private agency.*
  If US had **[stopped] [interfering]** *in other country policies long ago, ...*
  They probably would have **[began] [responding]** *to President Reagan's 600 ships plan with new construction.*

- If the event is a nominalization which appears as the head of the NP along with other elements (specifiers, complements, modifiers), only the head element will be marked:

  The young industry's rapid **growth** *also is attracting regulators eager to police its many facets.*

- If the event is a nominalization accompanied by some sort of light predicate (as shown by the underlined segments below), we will tag both elements.

  Several pro-Iraq **demonstrations** *have* **taken** *place in the last week.*
  They will definitely **take** *into* **consideration** *our readiness.*

  This proposal takes into consideration the relevance of both verbal and nominal heads with respect to the different kind of eventive information they convey. The two tagged events will be related as `IDENTICAL` by the `relType` attribute in the `TLINK` tag (see section 2.11.10).

- If the event is a predicative clause, only the predicative element (the adjective or the nominal in the following examples) will be tagged. This is done by applying the test of headedness in order to decide what to mark:

  1. If the predicative element is ENDOCENTRIC (it has a head), then we will tag the head only. In the following examples, the predicative element is delimited by square brackets, whereas its head is marked in bold face:

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

  2. If the predicative element is EXOCENTRIC (it has no single head), then we will mark up the entire expression within the TimeML tag.

     All 75 people were **on board** *at 9:00 a.m.*

- If the event is expressed by means of a prepositional phrase (PP), we again use the strategy based on the headedness of the prepositional phrase. Therefore, the following example, where the PP is exocentric, will be tagged as shown:

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

- 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 *rains* event, and should they be tagged as such?

  Three cases can be distinguished:

Case 1 EVENT cause EVENT

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

Case 2 ENTITY cause EVENT

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

We adopt the following solution for causatives predicates in TimeML.

1. If the annotator encounters a predicate denoting a causative relation between arguments in a sentence, then, either:
   (a) If the (logical) subject is an event denoting expression (e.g., *war*, *flood*, *rain*, *meeting*, etc.), then mark it as an event. (CASE 1 ABOVE)
   (b) If the (logical) subject is an individual entity (e.g., *John*, *the woman*, *the company*, etc.), then do nothing with it. (CASE 2 ABOVE)
2. Mark the object event as an event (*flooding*, *fire*).
3. Mark the predicate denoting the causative relation as an event (*caused*).
4. If there is an event introduced by the subject, as in Case 1, then introduce a `TLINK` with a `reltype` attribute of `IDENTITY` identifying the causative relation event with the subject event (see section 2.11.10 for a discussion of the `TLINK` tag and its attributes).
5. Introduce a `TLINK` between the causative predicate event and the event associated with the object position, with `relType` of `BEFORE`.

This solution should be 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 annotator has the option of identifying the discourse marker *and* as a signal (see section 2.11.4) for a `TLINK` introducing `relType BEFORE`, or not. The annotator CAN identify the `TLINK` between the events, but NEED NOT.

Fully annotated versions of the above examples are presented below.

Attributes for EVENT

a. Event ID number (eid) Non-optional attribute. Each event has to be identified by a unique ID number. This will be automatically assigned by the annotation tool every time an EVENT tag is assigned to some string.

b. Class Non-optional attribute. Each event belongs to one of the following classes. Note that while the examples show verbs expressing an event of a given type, it does not follow that every occurrence of the verb in question necessarily expresses an event of the same type. I.e. verbs may be ambiguous with respect to event class.

- REPORTING:

  Reporting events describe the action of a person or an organization declaring something, narrating an event, informing about an event, etc. Some examples: *say, report, tell, explain, state*.

- PERCEPTION:

  This class includes events involving the perception of another event. Such events are typically expressed by verbs like: *see, watch, glimpse, behold, view, hear, listen, overhear.*

- ASPECTUAL:

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

  1. Initiation: *begin, start, commence, set out, set about, lead off, originate, initiate*.
  2. Reinitiation: *restart, reinitiate, reignite* (metaphorically)
  3. Termination: *stop, cancel* (see also I_ACTION class), *end, halt, terminate, cease, discontinue, interrupt, quit, give up, abandon, block, break off, lay off, call off, wind up*.
  4. Culmination: *finish, complete*.
  5. Continuation: *continue, keep, go on, proceed, go along, carry on, uphold, bear on, persist, persevere.*

- I_ACTION:

  An I_ACTION is an Intensional Action. An I_ACTION **introduces an event argument** describing an action or situation **from which we can infer something given its relation with the I_ACTION**. For instance, the events introduced as arguments of the I_ACTIONS in (1) have not necessarily occurred when the I_ACTION takes place. Explicit performative predicates (like those in 5-9, below) are also included here. Note that the I_ACTION class does not cover states (see the description of I_STATES below).

  The following list, where I_ACTIONs are in bold face and the events they introduce are underlined, is meant to be representative (not exhaustive) of the types of events included in this class:

  1. **attempt, try, scramble:**
  2. **investigate, investigation, look at, delve...**
  3. **delay, postpone, defer, hinder, set back:**
  4. **avoid, prevent, cancel**:
  5. **ask, order, persuade, request, beg, command, urge, authorize:**
  6. **promise, offer, assure, propose, agree:**
  7. **swear, vow.**
  8. **name, nominate, appoint, declare, proclaim.**
  9. **claim, allege, suggest.**

- I_STATE:

  I_STATE events are similar to the previous class. This class includes **states that refer to alternative or possible worlds**, which can be introduced by subordinated clauses, nominalizations, or untensed VPs:

  As above, the following list of I_STATEs is just representative:

  1. **believe, think, suspect, imagine, doubt, feel, be conceivable, be sure**:
  2. **want, love, like, desire, crave, lust**:
  3. **hope, expect, aspire, plan**:
  4. **fear, hate, dread, worry, be afraid**:
  5. **need, require, demand**
  6. **be ready, be eager, be prepared**
  7. **be able, be unable**

  All I_STATEs will be annotated, whether they are persistent or not throughout the text being marked-up (see next class).

- STATE:

  States describe *circumstances* in which something obtains or holds true. However, we will only annotate:

  1. States that are identifiably changed over the course of the document being marked up.
  2. States that are introduced by an I_ACTION, an I_STATE, or a REPORTING event.
  3. Predicative states the validity of which is dependent on the document creation time

  Note that the current class, STATE, will not contain states that have been tagged as I_STATEs.

- OCCURRENCE:

  This class includes all other kinds of events describing something that happens or occurs in the world.

c. Additional attributes: EVENT has two additional attributes: **'tense'** and **'aspect'**. They will be introduced during preprocessing, so the human annotator does not have to annotate them. The human annotator need only check for mistakes made by the preprocessor.

2.11.2 Complete annotation of EVENTs

1. *The young industry's rapid growth also is attracting regulators eager to police its many facets.*

```
The young industry's rapid
<EVENT eid="e1" class="OCCURRENCE">
growth
</EVENT>
also is
<EVENT eid="e2" class="OCCURRENCE">
attracting
</EVENT>
```

```
regulators
<EVENT eid="e4" class="I_STATE">
eager
</EVENT>
to
<EVENT eid="e5" class="OCCURRENCE">
police
</EVENT>
its many facets.
```

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

```
A fresh
<EVENT eid="e1" class="OCCURRENCE">
flow
</EVENT>
of lava, gas and debris
<EVENT eid="e2" class="OCCURRENCE">
erupted
</EVENT>
there Saturday.
```

3. *It is conceivable that a larger eruption will take place in few hours.*

```
It is
<EVENT eid="e1" class="I_STATE">
conceivable
</EVENT>
that a larger
<EVENT eid="e2" class="OCCURRENCE">
eruption
</EVENT>
will
<EVENT eid="e3" class="OCCURRENCE">
take
</EVENT>
place in few hours.
```

4. *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.*

```
Israel will
<EVENT eid="e1" class="I_ACTION">
ask
</EVENT>
the United States to
<EVENT eid="e2" class="I_ACTION">
```

```
delay
</EVENT>
a military
<EVENT eid="e3" class="OCCURRENCE">
strike
</EVENT>
against Iraq until the Jewish state is fully
<EVENT eid="e4" class="I_STATE">
prepared
</EVENT>
for a possible Iraqi
<EVENT eid="e5" class="OCCURRENCE">
attack
</EVENT>
.
```

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

```
A Philippine volcano,
<EVENT eid="e1" class="STATE">
dormant
</EVENT>
for six centuries,
<EVENT eid="e2" class="ASPECTUAL">
began
</EVENT>
<EVENT eid="e3" class="OCCURRENCE">
exploding
</EVENT>
with searing gases, thick ash and deadly debris.
```

6. *”There is no reason why we would not be prepared,” Mordechai told the Yediot Ahronot daily.*

```
"There is no reason why we would not be
<EVENT eid="e1" class="STATE">
prepared
</EVENT>
," Mordechai
<EVENT eid="e2" class="REPORT">
told
</EVENT>
the Yediot Ahronot daily.
```

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

```
All 75 people
```

```
<EVENT eid="e1" class="STATE">
on  board
</EVENT>
the Aeroflot Airbus
<EVENT eid="e2" class="OCCURRENCE">
died
</EVENT>
.
```

8. *11,024 people, including local Aeta aborigines, were evacuated to 18 disaster relief centers set up at area schools.*

```
11,024 people, including local Aeta aborigines, were
<EVENT eid="e1" class="OCCURRENCE">
evacuated
</EVENT>
to 18 disaster relief centers
<EVENT eid="e2" class="STATE">
set
</EVENT>
up at area schools.
```

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

```
The agencies
<EVENT eid="e1" class="I_STATE">
fear
</EVENT>
they will be
<EVENT eid="e2" class="I_STATE">
unable
</EVENT>
to
<EVENT eid="e3" class="OCCURRENCE">
crack
</EVENT>
those codes to
<EVENT eid="e4" class="OCCURRENCE">
eavesdrop
</EVENT>
on spies and crooks.
```

2.11.3 The tag <TIMEX3>

Since the exact nature of the tag that TimeML uses to annotate temporal expressions is different in detail both from the TIMEX tag in STAG (Sheffield Temporal Annotation Guidelines – see **?**) and the TIMEX2 tag in TIDES, we here use the tag name TIMEX3 for temporal expressions.

How to annotate TIMEX3s

In order to be as compliant as possible with TIDES TIMEX2 annotation, the `TIMEX3` tag will be applied to TIMEX2 markable expressions. See TIDES(02), section 2, for the particular expressions we intend to cover.

However, TimeML will differ from TIDES in the following ways (the examples given below are adapted from TIDES(02)):

1. **TIMEX2 attributes that will be used:** (related to TIDES(02), sections 3.2 to 3.6)

   TimeML will only take TIDES attributes **VAL** and **MOD** as appropriate for `TIMEX3` elements (TIDES(02), sections 3.2 to 3.4). None of the following features will be considered: SET, PERIODICITY, GRANULARITY, NON_SPECIFIC (explained in TIDES(02), sections 3.5 and 3.6). The information that TIDES conveys by means of the first three attributes will be expressed here by MAKEINSTANCEs.

2. **Extent of the time expression to tag** (issue covered in TIDES(02), section 4).

   The tag extent for `TIMEX3` will be very similar to that proposed in TIMEX2. We follow the schema of TIDES(02), section 4 very closely in order to trace the differences carefully.

   (a) **Appositives:**

   (b) **Range Expressions:**

   (c) **Conjoined Expressions:**

   (d) **Embedded Expressions:**

   (e) **TWO TIMEX3 tags related by a TLINK tag** will be applied to handle cases other than those described above.

   (f) **Temporal expressions containing postmodifiers**

2.11.4 The tag <SIGNAL>

A signal is a textual element that makes explicit either the relation holding between two entities (timex and event, timex and timex, or event and event) or the modality of an event or the fact that one verb refers to two or more separate events. Signals are generally:

- **Temporal prepositions:** *on, in, at, from, to, before, after, during,* etc.
- **Temporal conjunctions:** *before, after, while, when,* etc.
- **Temporal modifiers:** *twice, every, three times,* etc.
- **Negative expressions:** *not, no, none,* etc.
- **Modals:** *might, may, could, should, would.*
- **Special characters:** "-" and "/", in temporal expressions denoting ranges (*September 4-6, Apr. 1999/Jul. 1999,* etc.).

How to annotate SIGNALs

Modals and Negative Expressions In simple cases modal verbs and negative expressions will not need to be tagged by human annotators, nor will the SLINKs they introduce (cf. section 2.11.10), since they will be annotated automatically via preprocessing. Here is such a case and an example of the annotation produced.

```
John <SIGNAL sid="s1"> might </SIGNAL> teach Monday
```

In cases where the surface expression is too complex for the preprocessor to identify the modal or negative and the associated subordinated verb, the annotator will have to tag the signal explicitly. For example, in sentences such as

John might, other things being equal, teach Monday.

the signal and associated SLINK may need to be manually added. In general material of any arbitrary nature and extent may appear between the modal or negative particle and the subordinated verb, and no preprocessing short of perfect parsing could correctly process all cases.

Temporal Prepositions, Conjunctions and Modifiers As for the other expressions, they will be tagged as illustrated in the following simple examples:

```
John taught <SIGNAL sid="s1"> on </SIGNAL> Monday

All passengers died <SIGNAL sid="s1"> when </SIGNAL>
the plane crashed into the mountain.
```

When two distinct signals appear side by side, they must be annotated separately, if they belong to different signal classes as listed above. Otherwise they must be annotated as a single SIGNAL. Contrast the two following examples:

```
John taught <SIGNAL sid="s1"> three times </SIGNAL>
<SIGNAL sid="s2"> on </SIGNAL> Monday

They will investigate the role of the US
<SIGNAL sid="s1">before, during and after</SIGNAL> the genocide
```

In this second case, 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 section 2.11.10).

2.11.5 Attributes for SIGNAL

SIGNAL has only one, non-optional, attribute: `sid`, the signal's unique id. It is automatically assigned by the annotation tool each time a SIGNAL is marked up.

2.11.6 The tag <MAKEINSTANCE>

We distinguish between event **tokens** and event **instances** or realisations – MAKEINSTANCE creates the actual realisation of an event. The motivation is examples like *John taught on Monday and Tuesday*, where one verb represents two events. In order to be able to annotate such cases, it is necessary to create two **instances** of *taught*, representing the two different events. MAKEINSTANCEs are created **in addition** to the event annotation (which marks up the event token).

In most cases, only one MAKEINSTANCE is needed, and will be automatically created by the annotation tool, so the annotator only has to insert **additional** MAKEINSTANCE for the second (and further, if necessary) instance or realisation of an event. There are also cases, where the annotator can either create as many MAKEINSTANCEs as motivated by the text or one MAKEINSTANCE which carries a cardinality value. Examples are the following:

1. *John taught twice on Monday.*
2. *John taught 150 times last year.*

In the first example, the annotator can either create two MAKEINSTANCEs or one MAKEINSTANCE with cardinality two. In the second example, the only practical solution is to create one MAKEINSTANCE with cardinality 150.

2.11.7 How to annotate MAKEINSTANCEs

We will use examples to demonstrate how to annotate MAKEINSTANCEs. In these examples we will not give a detailed annotation of events, times and signals, so please refer to the appropriate sections for instructions on how to annotate them. We also only show the mark-up for those entities which are relevant to the examples.

A MAKEINSTANCE has to be created for each instance or realisation of an event – as many as are motivated by the text. Alternatively, the cardinality attribute may be used, e.g. for *every* and for large numbers. When the cardinality is small enough (e.g. *twice*) , the annotator has a choice of creating one MAKEINSTANCE with the appropriate cardinality or creating as many MAKEINSTANCES as appropriate. We recommend using only one MAKEINSTANCE, unless individual events are referred to later in the text. Note that one MAKEINSTANCE will be automatically created for each event by the annotation tool, so the annotator only has to create additional MAKEINSTANCEs when more than one event instance is referred to in the text. This is done by annotating or 'swiping' the event as many times as needed. Each 'swipe' will automatically create a new instance ID.

In each of the following examples the event *taught* is marked up as follows:

```
<EVENT eid="e1"> taught </EVENT>
```

1. *John taught on Monday*

   This leads to one automatically created MAKEINSTANCE:

   ```
   <MAKEINSTANCE eiid="ei1" eventID="e1"/>
   ```

2. *John taught on Monday and Tuesday*

   Since the teaching event has two different time values in this example, two MAKEINSTANCEs are necessary. The first one will be automatically created, but the second one has to be created by the annotator:

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<MAKEINSTANCE eiid="ei2" eventID="e1"/>
```

3. *John taught twice on Monday*

   Since there is only one time expression in the text, the annotator has the choice of either creating a second MAKEINSTANCE or using the cardinality attribute. If two MAKEINSTANCE are created then they would look as follows (where *twice* is the signal with signalID="s1"):

```
<MAKEINSTANCE eiid="ei1" eventID="e1" signalID="s1"/>
<MAKEINSTANCE eiid="ei2" eventID="e1" signalID="s1"/>
```

   If one MAKEINSTANCE were created instead then it would look as follows:

```
<MAKEINSTANCE eiid="ei1" eventID="e1" signalID="s1" cardinality="2"/>
```

4. *John taught every Monday*

   The MAKEINSTANCE that is automatically introduced has to be changed to the following, where the SIGNAL *every* is referred to by signalID="s1"):

```
<MAKEINSTANCE eiid="ei1" eventID="e1" signalID="s1" cardinality="EVERY">
```

   Currently, the annotation tool allows the annotator to create different MAKEINSTANCEs by introducing an identification number (followed by a comma) in the last column of the links table, whenever a T-, S- or ALINK involving that particular instance is introduced.

2.11.8 Attributes for MAKEINSTANCE

```
attributes ::= eiid eventID [signalID] [cardinality]
```

2.11.9 BNF for the MAKEINSTANCE tag

```
eiid ::= ei<integer>
eventID ::= e<integer>
signalID ::= s<integer>
cardinality ::= <integer> | 'EVERY' | ...
```

2.11.10 The link tags: <TLINK>, <SLINK>, and <ALINK>

There are three types of link tags. The function of each will be introduced here, before we move on to explaining in detail how links are annotated.

- **TLINK:**

  A TLINK or Temporal Link 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:

1. **Simultaneous**
   Two events are judged simultaneous if they happen at the same time, or are temporally indistinguishable in context, i.e. occur close enough to the same time that further distinguishing their times makes no difference to the temporal interpretation of the text.
2. One **before** the other:
   As in the following example between the events *slayings* and *arrested*:
   > *The police looked into the* **slayings** *of 14 women. In six of the cases suspects have already been* **arrested**.
3. One **after** the other:
   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.
4. One **immediately before** the other:
   As in the following sentence between *crash* and *died*.
   > *All passengers* **died** *when the plane* **crashed** *into the mountain*
5. One **immediately after** than the other:
   This is the inverse of the preceding relation.
6. One **including** the other:
   As is the case between the temporal expression and the event in the following example:
   > *John* **arrived** *in Boston* **last Thursday**.
7. One **being included** in the other:
   The inverse relation to the preceding one.
8. One **holds** during the other:
   Specifically applicable to states or events that persist throughout a duration, for example:
   > *James was* **CTO** *for* **two years**.
   > *John* **taught** *for* **20 minutes** *on Monday.*
9. One being the **beginning** of the other:
   As holds between the first of the temporal expressions and the event in the following example:
   > *John was* **in the gym** *between* **6:00 p.m.** *and 7:00 p.m.*
10. One being **begun by** the other:
    The inverse relation to the one just introduced.
11. One being the **ending** of the other:
    > *John was* **in the gym** *between 6:00 p.m. and* **7:00 p.m.**.
12. One being **ended by** the other:
    The inverse relation to the one just introduced.

In addition, `TLINK`s are also used in the three following situations:

1. **Event identity**
   Event identity is also annotated via the `TLINK`. **Note that event identity is a very important relationship, which will not be picked up during the closure part of the annotation. So please make sure that all identity links are annotated.** E.g.:

John **drove** *to Boston. During his* **drive** *he ate a donut.*

2. **When a 'set/subset' relationship occurs in the text.**
   An example is:

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

   With the current version of TimeML we cannot capture this 'set/subset' relationship. What will be done for now is to create a `MAKEINSTANCE` for each of the 'sets', i.e. one `MAKEINSTANCE` for *slayings* with cardinality plural (or 14?) and one `MAKEINSTANCE` for *cases* with cardinality plural (or 6?) and then link these two instances via a `TLINK` with the temporal relationship IS_INCLUDED or INCLUDES.
3. **When two events that are temporally ordered are separated by an explicit duration of time.**
   In the following example, for instance, the sending of the report occurred two months before the attack:

   Two months before the attack, a report was sent.

   A `TLINK` will be used in order to relate *sent* and *attack*. The interval of time will be conveyed by means of the attribute `magnitude` (see section 2.11.12), which will refer to the time ID of the temporal expression *two months*.
   The fully annotated version of examples like this above are in Chapter 2.11.14, numbers: 5-9. Compare them with examples 1-4, 10 and 11.

- **SLINK:**

An SLINK or Subordination Link is used for contexts introducing relations between two events, or an event and a signal. SLINKs are of one of the following sorts:

1. **Modal:**
   This relation is introduced mostly by modal verbs (**should, could, would,** etc.), which will be marked as SIGNALS (2.11.4), but also by events that introduce a reference to a possible world – mainly I_STATEs:

   John **should** *have bought some wine.*
   Mary **wanted** *John to buy some wine.*
2. **Factive:**
   Certain verbs introduce an entailment (or presupposition) of their argument's veracity. They include **forget** in the tensed complement, **regret**, **manage**:

   John **forgot** *that he was in Boston last year.*
   Mary **regrets** *that she didn't marry John. John* **managed** *to leave the party*
3. **Counter-factive:**
   Contrary to the previous relation, in this case the event introduces a presupposition about the non-veracity of its argument: **forget (to), unable to** (in past tense), **prevent, cancel, avoid, decline**, etc.

   John **forgot** *to buy some wine.*
   Mary was **unable** *to marry John. John* **prevented** *the divorce*.
4. **Evidential:**
   Evidential relations are typically introduced by REPORTING or PERCEPTION events:

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

5. **Negative evidential:**
   Introduced by REPORTING (and PERCEPTION?) events conveying negative polarity:

   John **denied** *he bought only beer.*

6. **Negative:**
   Introduced only by negative particles (*not, nor, neither*, etc.), which should be marked as SIGNALs, with respect to the events they are modifying:

   *John did***n't** *forget to buy some wine.*
   John did **not** *want to marry Mary.*

   For each REPORTING or PERCEPTION event, an SLINK has to be introduced. The SLINK expresses the relation between the REPORTING or PERCEPTION event and the main event in its subordinate clause. In the following example, the REPORTING and PERCEPTION events are in bold face, whereas the subordinate events are underlined:

   John **said** *that he taught on Monday*
   John **saw** *the plane crash into the building*

   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):

   Israel has been **scrambling** *to buy more masks abroad.*
   The local government **hopes** *that residents will soon return to their homes*

- **ALINK:**

  An `ALINK` or Aspectual Link represents the relationship between an aspectual event and its argument event. Examples of the aspectual relations to be encoded are:

  1. **Initiation**:

     John **started** *to read*

  2. **Culmination**:

     John **finished** *assembling the table.*

  3. **Termination**:

     John **stopped** *talking*.

  4. **Continuation**:

     John **kept** *talking*.

2.11.11 How to annotate LINKs

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 section for instructions on annotating these. Also, we only show the mark-up for those entities which are relevant to the examples.

NOTE that a link that originates in an event always links via an **event instance ID** (i.e. the ID used in the `MAKEINSTANCE`) rather than via the ID of the event token itself.

- **TLINK:**

  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:

  1. *John taught on Monday*

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

     ```
     <TLINK eventInstanceID="ei1" relatedToTime="t1" signalID="s1"
            relType="IS_INCLUDED"/>
     ```

  2. *John taught on Monday and Tuesday*

     As explained in section 2.11.6 above, two MAKEINSTANCEs represent the two instances of *taught*:

     ```
     <MAKEINSTANCE eiid="ei1" eventID="e5"/>
     <MAKEINSTANCE eiid="ei2" eventID="e5"/>
     ```

     The temporal relationship holding between those two events and the two time expressions are annotated by introducing the following two TLINKs:

     ```
     <TLINK eventInstanceID="ei1" relatedToTime="t1" signalID="s1"
            relType="IS_INCLUDED"/>

     <TLINK eventInstanceID="ei1" relatedToTime="t2" signalID="s1"
            relType="IS_INCLUDED"/>
     ```

  3. *John taught every Monday*

     The MAKEINSTANCE representing the multiple instances of *taught* looks as follows:

     ```
     <MAKEINSTANCE eiid="ei9" eventID="e4" signalID="s5"
         cardinality="EVERY"/>
     ```

     The TLINK representing the temporal relation holding between the event(s) and the temporal expression looks like this:

     ```
     <TLINK eventInstanceID="ei9" relatedToTime="t7"
       relType="IS_INCLUDED"/>
     ```

  4. *John taught for 20 minutes on Monday.*

     The MAKEINSTANCE representing the event *taught* looks as follows:

     ```
     <MAKEINSTANCE eiid="ei4" eventID="e4"/>
     ```

     Two TLINKs have to be introduced. One TLINK captures the fact that the *taught* event holds throughout the *20 minutes*, and one TLINK captures the fact that the *taught* event is included in *Monday*.

     ```
     <TLINK eventInstanceID="ei4" relatedToTime="t2" signalID="s5"
            relType="HOLDS"/>

     <TLINK tid="t2" relatedToTime="t3" signalID="s6"
         relType="IS_INCLUDED"/>
     ```

5. *John drove to Boston. During his drive he ate a donut.*
   The MAKEINSTANCEs presenting the events *drove* and *drive* look as follows:

   ```
   <MAKEINSTANCE eiid="ei1" eventID="e1"/>

   <MAKEINSTANCE eiid="ei2" eventID="e2"/>
   ```

   The TLINK that has to be introduced to represent the identity of these two events looks as follows:

   ```
   <TLINK eventInstanceID="ei1" relatedToEvent="ei2" relType="IDENTITY"/>
   ```

   Please note that we did not include the TLINK that would represent the temporal relationship between *ate* and *drive*.

6. *The police looked into the slayings of 14 women. In six of the cases suspects have already been arrested.*
   The MAKEINSTANCEs representing the events *slayings* and *cases* look as follows:

   ```
   <MAKEINSTANCE eiid="ei1" eventID="e1" cardinality="14"/>

   <MAKEINSTANCE eiid="ei2" eventID="e2" cardinality="6">
   ```

   The TLINK that has to be introduced to represent the relationship between these two events looks as follows:

   ```
   <TLINK eventInstanceID="ei1" relatedToEvent="ei2" relType="INCLUDES"/>
   ```

- **SLINK:**

  SLINKs that relate two events link an event instance and an event type. On the other hand, SLINKs can relate SIGNALs (modals or negation particles) and event types. There will NOT be SLINKs linking two events and a SIGNAL.

1. *John said that he taught on Monday*
   The two MAKEINSTANCEs for the two events are the following:

   ```
   <MAKEINSTANCE eiid="ei2" eventID="e2"/>

   <MAKEINSTANCE eiid="ei3" eventID="e3"/>
   ```

   To express the fact that the *taught* event is reported by the *said* event, the following SLINK is created:

   ```
   <SLINK eventInstanceID="ei2" subordinatedEvent="e3"
     relType="EVIDENTIAL"/>
   ```

2. *John might teach on Monday*
   The MAKEINSTANCEs for the events are the following:

   ```
   <MAKEINSTANCE eiid="ei4" eventID="e3"/>
   <MAKEINSTANCE eiid="ei5" eventID="e4"/>
   ```

The SLINK that has to be created to represent the modality of *teach* looks as follows:

```
<SLINK eventInstanceID="ei4"
    subordinatedEventInstance="e5" relType="MODAL"/>
```

3. *John did not teach on Monday*
   The MAKEINSTANCE for the event is the following:

   ```
   <MAKEINSTANCE eiid="ei4" eventID="e3"/>
   ```

   The SLINK that has to be created to represent the fact that the polarity of the *teach* event is negative looks as follows:

   ```
   <SLINK eventInstanceID="ei4" signalID="s2" relType="NEGATIVE"/>
   ```

- **ALINK:**

  ALINKS represent the relationship between aspectual verbs (and other expressions) and the event they refer to. Examples are:

1. *John started to read*
   The two MAKEINSTANCEs for the two events are the following:

   ```
   <MAKEINSTANCE eiid="ei5" eventID="e2"/>
   <MAKEINSTANCE eiid="ei6" eventID="e3"/>
   ```

   The ALINK that has to be created between the aspectual verb *started* and the event *teach* is the following:

   ```
   <ALINK eventInstanceID="ei5" relatedToEvent="ei6"
     relType="INITIATES"/>
   ```

2. *John finished reading*
   The two MAKEINSTANCEs for the two events are the following:

   ```
   <MAKEINSTANCE eiid="ei5" eventID="e2"/>

   <MAKEINSTANCE eiid="ei6" eventID="e3"/>
   ```

   The ALINK that has to be created between the aspectual verb started and the event teach is the following:

   ```
   <ALINK eventInstanceID="ei5" relatedToEvent="ei6"
     relType="TERMINATES"/>
   ```

2.11.12 Attributes for LINKs

- **TLINK:**

1. **eventInstanceID** or **timeID**
   Obligatory attribute (one or the other of these needs to be present). This is the ID of the eventInstance or the timeID involved in the temporal link.

2. **signalID**

   Optional attribute. If the temporal relation holding between the entities is explicitly signalled in the text, then the ID of that signal needs to be supplied here.

3. **relatedToEvent** or **relatedToTime**

   Obligatory attribute (one or the other of these needs to be present). This is the ID of the entity that is being related to the event instance with `ID=eventInstanceID` or time expression with `ID=timeID`.

4. **relType**

   Obligatory attribute. This is the temporal relation holding between the entities. Possible values are: `BEFORE, AFTER, INCLUDES, IS_INCLUDED, HOLDS, SIMULTANEOUS, IAFTER, IBEFORE, IDENTITY, BEGINS, ENDS, BEGUN_BY, ENDED_BY` . These are assigned according to the instructions given at the beginning of section 2.11.10, on `TLINK`s. There will be only one relation assigned per `TLINK`.

5. **magnitude**

   Optional attribute.

   Attribute used when relating two events separated by a duration of time that is made explicit in the text. E.g.,

   One month before the attack a report was sent.

   `magnitude` ranges over the ID of `TIMEX3` expression that represent the magnitude of the relation between the attack and the sending of the report (*one month*, etc.).

- **SLINK:**

1. **eventInstanceID**

   Optional attribute. This is the ID of the `eventInstance` involved in the subordination link. **Note:** `eventInstanceID` is optional because an event can be subordinated (e.g. in a conditional or when an event is negated) without being subordinated to a particular event.

2. **subordinatedEvent** or **subordinatedEventInstance**

   Obligatory attribute (one of them needs to be present). This is the ID of the subordinated event or event instance that the event instance with `ID=eventInstanceID` is related to.

3. **signalID**

   Optional attribute. If the subordination relation holding between the events is explicitly signalled in the text, then the ID of that signal needs to be filled in here.

4. **relType**

   Obligatory attribute. This is the temporal relation holding between the entities. Possible values are: `MODAL, NEGATIVE, EVIDENTIAL, NEG_EVIDENTIAL, FACTIVE, COUNTER_FACTIVE`. These are assigned according to the instructions given at the beginning of section 2.11.10, on `SLINK`s.

- **ALINK:**

1. **eventInstanceID**

   Obligatory attribute. This is the ID of the (aspectual) eventInstance involved in the aspectual link.

2. **signalID**
   Optional attribute. If the aspectual relation holding between the events is explicitly signalled in the text, then the ID of that signal needs to be filled in here. An example for a signal is *to* in *The ship began to sink*.
3. **relatedToEvent**
   Obligatory attribute. This is the ID of the event related to the aspectual event.
4. **relType**
   Obligatory attribute. This is the temporal relation holding between the entities. Possible values are: `INITIATES, CULMINATES, TERMINATES, CONTINUES`

2.11.13 BNF for the link tags

- **TLINK:**

```
attributes ::= (eventInstanceID | timeID) [signalID]
               (relatedtoEvent | relatedtoTime) relType [magnitude]
eventInstanceID ::= ei<integer>
timeID ::= t<integer>
signalID ::= s<integer>
relatedToEvent ::= ei<integer>
relatedToTime ::= t<integer>
relType ::= 'BEFORE' | 'AFTER' | 'INCLUDES' | 'IS_INCLUDED' | 'HOLDS'
             'SIMULTANEOUS' | 'IAFTER' | 'IBEFORE' | 'IDENTITY' |
             'BEGINS' | 'ENDS' | 'BEGUN_BY' | 'ENDED_BY'
magnitude ::= t<integer>
```

- **SLINK:**

```
attributes ::= [eventInstanceID]
               (subordinatedEvent | subordinatedEventInstance)
               [signalID] relType [polarity]
eventInstanceID ::= ei<integer>
subordinatedEvent ::= e<integer>
subordinatedEventInstance ::= ei<integer>
signalID ::= s<integer>
relType ::= 'MODAL' | 'NEGATIVE' | 'EVIDENTIAL' | 'NEG_EVIDENTIAL' |
            'FACTIVE' | 'COUNTER_FACTIVE'
```

- **ALINK:**

```
attributes ::= eventInstanceID [signalID] relatedToEvent relType
eventInstanceID ::= ei<integer>
signalID ::= s<integer>
eventID ::= e<integer>
relType ::= 'INITIATES' | 'CULMINATES' | 'TERMINATES' | 'CONTINUES'
```

2.11.14 Completely annotated examples

Assume for all the examples that the document creation time (DCT) is marked up as a TIMEX3 expression with tid="t0".

Complex TIMEX Examples

1. *John left 2 days before yesterday.*

```
John
<EVENT  eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
left
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<TIMEX3 tid="t1" type="DATE" value="2002-07-08"
    anchorTimeID="t0" temporalFunction="true"
    valueFromFunction="tf1">
2 days before yesterday
</TIMEX3>

<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
```

   We don't split the temporal expression into smaller components (i.e., "two days" and "yesterday"). Similarly, we do not tag "before" as a SIGNAL, but as part of the TIMEX3 expression instead.

   The type attribute of the TIMEX3 denotes a DATE (the date in which John left), which can be computed by a temporal function relative to the temporal anchor "t0" (the DCT).

2. *I'm leaving on vacation two weeks from next Tuesday*.

```
I'm
<EVENT  eid="e1" class="OCCURRENCE" tense="FUTURE" aspect="PROGRESSIVE">
leaving
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
on vacation
<TIMEX3 tid="t1" type="DATE" value="2002-07-23"
    anchorTimeID="t0" temporalFunction="true"
    valueFromFunction="tf1">
two weeks from next Tuesday
</TIMEX3>

<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
```

3. *A major earthquake struck Los Angeles three years ago today.*

```
A major
<EVENT  eid="e1" class="OCCURRENCE" tense="NONE" aspect="NONE">
earthquake
```

```
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<EVENT  eid="e2" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
struck
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
Los Angeles
<TIMEX3 tid="t1" type="DATE" value="1999-07-12"
    anchorTimeID="t0" temporalFunction="true"
    valueFromFunction="tf1">
three years ago today
</TIMEX3>

<TLINK eventInstanceID="ei1" relatedToEvent="ei2" relType="IBEFORE"/>
<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
```

4. *John left 2 days ago.*

```
John
<EVENT  eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
left
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<TIMEX3 tid="t1" type="DATE" value="2002-07-08"
    temporalFunction="true" anchorTimeID="t0"
    valueFromFunction="tf1">
2 days ago.
</TIMEX3>

<TLINK  eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
```

Note that *ago* 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”, which necessarily relate two events and thus introduce a TLINK with the magnitude attribute. This can be observed in the 3 following examples.

5. *John left 2 days before the attack.*

```
John
<EVENT  eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
left
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<TIMEX3 tid="t1" type="DURATION" value="P2D" temporalFunction="false">
2 days
```

```
</TIMEX3>
<SIGNAL sid="s1">
before
</SIGNAL>
the
<EVENT  eid="e2" class="OCCURRENCE" tense="NONE" aspect="NONE">
attack
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>

<TLINK  eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="BEFORE" magnitude="t1"/>
```

6. *5 days after he came back Mary got sick.*

```
<TIMEX3 tid="t1" type="DURATION" value="P5D" temporalFunction="false">
5 days
</TIMEX3>
<SIGNAL sid="s1">
after
</SIGNAL>
he
<EVENT  eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
came
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
back Mary
<EVENT  eid="e2" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
got
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
sick.

<TLINK  eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="BEFORE" magnitude="t1"/>
```

7. *Two months before the attack, a report was sent.*

```
<TIMEX3 tid="t1" type="DURATION" value="P2M" temporalFunction="false">
Two months
</TIMEX3>
<SIGNAL sid="s1">
before
</SIGNAL>
the
<EVENT  eid="e1" class="OCCURRENCE"  tense="NONE" aspect="NONE">
```

```
attack
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
a report was
<EVENT  eid="e2" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
sent
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>

<TLINK eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="AFTER" magnitude="t1"/>
```

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.

2.11.15 Complex TLINK and SLINK Examples

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

```
The
<EVENT eid="e1" class="OCCURRENCE" tense="NONE" aspect="NONE">
attack
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
was
<SIGNAL sid="s1">
not
</SIGNAL>
<EVENT eid="e2" class="I_STATE" tense="PAST" aspect="PERFECTIVE">
expected
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
at all, although a report had been
<EVENT eid=e3>
sent
</EVENT>
<MAKEINSTANCE eiid="ei3" eventID="e3"/>
<TIMEX3 tid="t1" type="DURATION" val="P2M" temporalFunction="false">
2 months
</TIMEX3>
<SIGNAL sid=s2>
before
</SIGNAL>
```

```
<SLINK eventInstanceID="ei2" signalID="s1" relType="NEGATIVE"/>
<SLINK eventInstanceID="ei2" subordinatedEvent="e1" relType="MODAL"/>
<TLINK eventInstanceID="ei1" relatedToEvent="ei3" relType="AFTER"
    magnitude="t1" signalID="s2"/>
```

2. *Mary arrived yesterday but John left 2 days before.*

```
Mary
<EVENT  eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
arrived
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<TIMEX3  tid="t1" type="DATE" value="2002-07-09"
    temporalFunction="true" anchorTimeID="t0"
    valueFromFunction="tf1">
yesterday
</TIMEX3>
but John
<EVENT  eid="e2" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
left
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
<TIMEX3 tid="t2" type="DURATION" value="P2D" temporalFunction="false">
2 days
</TIMEX3>
<SIGNAL sid="s1">
before.
</SIGNAL>

<TLINK  eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="AFTER" magnitude="t2"/>
<TLINK  eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
```

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

3. *She was sick after the play.*

```
She was
<EVENT eid="e1" class="STATE" tense="NONE" aspect="NONE">
sick
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<SIGNAL sid="s1">
after
</SIGNAL>
the
```

```
<EVENT  eid="e2" class="OCCURRENCE"  tense="NONE" aspect="NONE">
play
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>

<TLINK  eventInstanceID="ei1" signalID="s1" relatedToEvent="ei2"
relType="AFTER"/>
```

The TLINK does not introduce magnitude.

4. *She was sick for 2 hours after the play.*

```
She was
<EVENT eid="e1" class="STATE" tense="NONE" aspect="NONE">
sick
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<SIGNAL sid="s1">
for
</SIGNAL>
<TIMEX3 tid="t1" type="DURATION" value="P2H" temporalFunction="false">
2 hours
</TIMEX3>
<SIGNAL sid="s2">
after
</SIGNAL>
the
<EVENT  eid="e2" class="OCCURRENCE"  tense="NONE" aspect="NONE">
play
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>

<TLINK  eventInstanceID="ei1" signalID="s1" relatedToTime="t1"
relType="HOLDS"/>
<TLINK  eventInstanceID="ei1" signalID="s2" relatedToEvent="ei2"
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. Since there is no explicit reference to the duration of the interval between the two events, the second TLINK does not introduce the magnitude attribute.

5. *John taught for 20 minutes every Monday.*

One EVENT, two TIMEX3s and two SIGNALs need to be created. In addition, the following tags are needed:

(a) One MAKEINSTANCE with cardinality *every*, as signaled by the expression *every*.

(b) One TLINK linking the duration *20 minutes* to the event.

(c) One TLINK linking the TIMEX3 *Monday* to the TIMEX3 *20 minutes*.

```
John
<EVENT eid="e4" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
taught </EVENT>
<SIGNAL sid="s5">for</SIGNAL>
<TIMEX3 tid="t2" type="DURATION" value="PT20M" anchorTimeID="3">
        20 minutes</TIMEX3>
<SIGNAL sid="s6">every</SIGNAL>
<TIMEX3 tid="t3" type="DATE" value="XXXX-WXX-1">Monday</TIMEX3>
<MAKEINSTANCE eiid="ei4" eventID="e4" signalID="s6" cardinality="EVERY"/>
<TLINK eventInstanceID="ei4" relatedToTime="t2" signalID="s5"
       relType="HOLDS"/>
<TLINK timeID="2" relatedToTime="t3" relType="IS_INCLUDED"/>
```

6. *John left between Monday and Wednesday*

```
John
<EVENT eid="e1" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
left
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<SIGNAL sid="s1"/>
between
</SIGNAL>
<TIMEX3 tid="t1" type="DATE" value="2002-07-15"
        temporalFunction="true" anchorTimeID="t0"
        valueFromFunction="tf3"/>
Monday
</TIMEX3>
and
<TIMEX3 tid="t2" type="DATE" value="2002-07-17"
        temporalFunction="true" anchorTimeID="t0"
        valueFromFunction="tf3"/>
Wednesday
</TIMEX3>

<TLINK eventInstanceID="ei1" relatedToTime="t1" signalID="s1" relType="IAFTER"/
<TLINK eventInstanceID="ei1" relatedToTime="t2" signalID="s1" relType="IBEFORE
```

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

7. *John taught from 1994 through 1999.*

In this case, one EVENT and the two TIMEX3s need to be created. In addition, the following tags are needed:

(a) One automatically created MAKEINSTANCE for the event.

(b) One TLINK to capture the fact that the event started in 1994.

(c) 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.

```
John
<EVENT eid="e4" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
taught</EVENT>
<SIGNAL sid="s5">from</SIGNAL>
<TIMEX3 tid="t2" type="DATE" value="1994">1994</TIMEX3>
<SIGNAL sid="s6">through</SIGNAL>
<TIMEX3 tid="t3" type="DATE" value="1999">1999</TIMEX3>

<MAKEINSTANCE eiid="ei4" eventID="e4"/>
<TLINK eventInstanceID="ei4" relatedToTime="t2" signalID="s5"
       relType="BEGUN_BY"/>
<TLINK eventInstanceID="ei4" relatedToTime="t3" signalID="s6"
       relType="ENDED_BY"/>
```

8. *John did not leave on Monday but on Tuesday.*

   One EVENT, **two event instances**, and three SIGNALs need to be created:

```
John did <SIGNAL sid="s5">not</SIGNAL>
<EVENT eid="e4" class="OCCURRENCE" tense="PAST" aspect="PERFECTIVE">
leave</EVENT>
<SIGNAL sid="s6">on</SIGNAL>
<TIMEX3 tid="t3" type="DATE" value="XXXX-WXX-1">Monday</TIMEX3>
but
<SIGNAL sid="s7">on</SIGNAL>
<TIMEX3 tid="t4" type="DATE" value="XXXX-WXX-2">Tuesday</TIMEX3n>
<MAKEINSTANCE eiid="ei1" eventID="e4"/>
<MAKEINSTANCE eiid="ei2" eventID="e4"/>
<SLINK subordinatedEventInstance="ei1" signalID="s5"
       relType="NEGATIVE" polarity="false"/>
<TLINK eventInstanceID="ei1" relatedToTime="t3" signalID="s6"
       relType="IS_INCLUDED"/>
<TLINK eventInstanceID="ei2" relatedToTime="t4" signalID="s7"
       relType="IS_INCLUDED"/>
```

Causative Examples

1. *The rains caused the flooding.*

```
The
<EVENT eid="e1" class="OCCURRENCE">
 rains
</EVENT>
```

```
<EVENT eid="e2" class="OCCURRENCE">
caused
</EVENT>
the
<EVENT eid="e3" class="OCCURRENCE">
Flooding
</EVENT>

<TLINK eventInstanceID=e1 relatedtoEvent=e3  relType="BEFORE" />
<TLINK eventInstanceID=e1 relatedtoEvent=e2  relType="IDENTITY" />
```

2. *John caused the fire.*

```
John
<EVENT eid="e1" class="OCCURRENCE">
caused
</EVENT>
the
<EVENT eid="e2" class="OCCURRENCE">
fire
</EVENT>
<TLINK eventInstanceID=e1 relatedtoEvent=e2  relType="BEFORE" />
```

3. *Kissinger secured the peace at great cost.*

```
Kissinger
<EVENT eid="e1" class="OCCURRENCE">
secured
</EVENT>
the
<EVENT eid="e2" class="OCCURRENCE">
peace
</EVENT>
<TLINK eventInstanceID=e1 relatedtoEvent=e2  relType="BEFORE" />
At great cost.
```

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

   Discourse relations acting as a causative, we are currently unable to handle as causatives, but we can certainly annotate the temporal relation that exists in the text.

```
He
<EVENT eid="e1" class="OCCURRENCE">
Kicked
</EVENT>
The ball
And
```

```
it
<EVENT eid="e2" class="OCCURRENCE">
rose
</EVENT>
Into the air
<TLINK eventInstanceID=e1 relatedtoEvent=e2  relType="BEFORE" />
```

2.12 Conclusion

This document summarizes the research and findings of the TERQAS workshop, held from January through September, 2002, and funded by ARDA through the NRRC at MITRE. The most significant milestones and achievements from the workshop are listed as follows:

1. TIMEML: Markup language for time and events
2. TIMEBANK: Gold standard annotated against this language
3. TIME/EVENT ALGORITHMS: For recognizing events and times in text
4. CLOSURE ALGORITHM: For inferring hidden temporal relations between events
5. DISSEMINATION OF KNOWLEDGE: U. of Chicago Book, journal articles, symposium, and courses on TimeML.

It is hoped that the research and findings of this workshop will generate interest in the specification language developed for time and event markup. Further, it is hoped that the goldstandard will continue to grow in order to account for multiple domains and languages.

Chapter 3

Catalog of Software, Data, Reports, and Presentations

3.1 TimeBank

1. raw data:
   location: /workshops/terqas/data/doc-corpora/
2. alembic preprocessed data
   location: d̃oc-corpora/doc-corpora-training/al-preprocessed
3. pre-annotation preprocessed data
   location: d̃oc-corpora/doc-corpora-training/Event-preprocessed
4. TimeBank location: d̃oc-corpora/doc-corpora-training/TimeBank
5. Timex3, Event and Link Recognizer Output:
   location: d̃oc-corpora/doc-corpora-training/T3PO

3.2 Query

1. Query Collection:
   /workshops/terqas/data/question-queries/

3.3 Software

1. closure tool:
   location: /workshops/terqas/software/closure
2. alembic to-TimeML script:
   location: /workshops/terqas/software/alembic-to-TimeML
3. Event preprocessing software for alembic:
   location: /workshops/terqas/software/TimeMLRecognizers
4. Timex3 processor (MetaCarta)
   location: /workshops/terqas/software/Timex3

5. Event and S-Link Recognizers
   location: /workshops/terqas/software/TimeMLRecognizers

6. visualization:
   location: /workshops/terqas/software/visualization

7. Link Builder:
   location: /workshops/terqas/software/TimeMLRecognizers

3.4 Documentation and Presentations

1. documentation:
   location: /workshops/terqas/documentation

2. presentations:
   location: /workshops/terqas/documentation

Bibliography

Valerio Allegranza, Paul Bennett, Jacques Durand, Frank van Eynde, Lee Humphreys, Paul Schmidt, and Erich Steiner. Section 3.2.5, 'the semantics of tense and aspect'. In C. Copeland, J. Durland, S. Krauwer, and B. Maegaard, editors, *Linguistics for Machine Translation: The Eurotra Linguistic Specifications*. Office for Official Publications of the Commission of the European Community.

C. Aone and M. Ramos-Santacruz. Rees: A large-scale relation and event extraction system. In *Proceedings of ANLP 2000*, 2000.

L. Chittaro and C. Combi. Extending the event calculus with temporal granularity and indeterminacy. In *AAAI'2000 Workshop on Spatial and Temporal Granularity*, pages 53–59, 2000. AAAI Tech Report WS-00-08.

David Day. Alembic workbench user's guide. Technical report, The MITRE Corporation.

Elena Filatova and Eduard Hovy. Assigning time-stamps to event-clauses. In *Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing, Toulose, France, 6-11 July*, pages 88–95, 2001.

Beth Sundheim Lisa Ferro, Inderjeet Mani and George Wilson. Tides temporal annotation guidelines. version 1.0.2. Technical report MTR 01W0000041, MITRE.

A. Montanari, E. Maim, E. Ciapessoni, and E. Ratto. Dealing with time granularity in the event calculus. In *Proceedings of the 1992 International Conference of Fifth Generation Computer systems (FGCS-92)*, pages 702–712, 1992.

James Pustejovsky. Events and the semantics of opposition. In C. Tenny and J. Pustejovsky, editors, *Events as Grammatical Objects*. CSLI Publications, 2000.

Frank Schilder and Christopher Habel. From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing, Toulose, France, 6-11 July*, pages 65–72, 2001.

Andrea Setzer. Guidelines for annotating events and temporal information in newswire texts. Technical report, University of Sheffield.

Andrea Setzer. *Temporal Information in Newswire Articles: an Annotation Scheme and Corpus Study*. PhD thesis, 2001.

Andrea Setzer and Robert Gaizauskas. Annotating events and temporal information in newswire texts. In *Workshop: Information Extraction Meets Corpus Linguistics, LREC 2000*, 2000a.

Andrea Setzer and Robert Gaizauskas. Building a temporally annotated corpus for information extraction. In *Workshop: Information Extraction Meets Corpus Linguistics, LREC 2000*, 2000b.

Andrea Setzer and Robert Gaizauskas. A pilot study on annotating temporal relations. In *ACL 2001, Workshop on Temporal and Spatial Information Processing*, 2001.

Andrea Setzer and Robert Gaizauskas. On the importance of annotating temporal event-event relations in text. In *Workshop on Annotation Standards for Temporal Information in Natural Language, LREC 2002*, 2002.

Beth Sundheim. Preliminary rdc guidelines for time attributes. version 1.2. Technical report, SPAWAR Systems Center, 2001.

Bibliography