ARDA Summer Workshop on Graphical Annotation Toolkit for TimeML

Version: 0.1.0

Release Date: 30-09-03

Authors: James Pustejovsky, Inderjeet Mani, Luc Bélanger, Branimir Boguraev, Bob Knippen, Jessica Littman, Anna Rumshisky, Andrew See, Svetlana Symonenko, James Van Guilder, Linda Van Guilder, Marc Verhagen, Robert Ingria.

Contents

Chapter 1

Executive Summary

1.1 Introduction

From March 1, 2003 through August 13, 2003, 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 some specific annotation and user interface problems that arose in the context of the TERQAS workshop in 2002. In the previous workshop we focused on two efforts, reflecting the major deliverables of that contract:

1. TimeML: Definition and design of 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:

1. Creation of Algorithms for recognizing:
    a. Temporal Expressions,
    b. Event Expressions
    c. Times associated with Events
    d. Ordering of Events and Times
2. Development of a Text Segmented Closure Algorithm
3. Creation of a Semi-graphical Annotation Tool

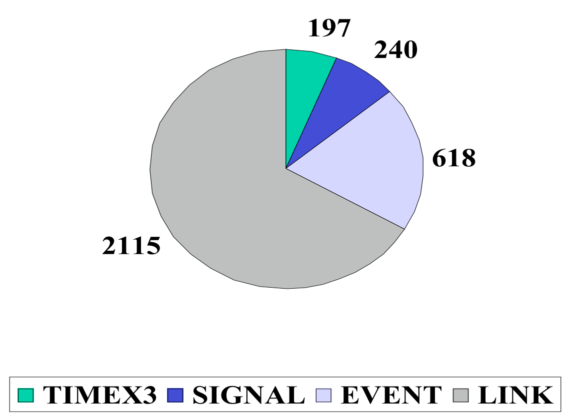
It is the last two of these deliverables above that will form the basis of the present proposal for a graphical annotation toolkit.

The advantages of TimeML have already proven to be quite useful. These include:

1. It provides a robust markup framework for multiple domains and applications;
2. It is compliant and interoperable with emerging Semantic Web standards;
3. Its algorithms can be compared and measured against common TimeML-marked up corpora, starting with TIMEBANK.

However, there are major problems with annotating text to this standard with the currently available annotation tools, such as the Alembic WB.

1. **Inconsistencies**: Annotators frequently input inconsistent information.
2. **Tag Density**: (a) The annotation is very dense, mainly due to link tags, (b) It is hard for annotators to keep track of relationships mentally (see figure below describing the breakdown of elements in a typical document). Notice that this number is only a fraction of the possible temporal links in that document, which is in fact quadratic to the number of events and time expressions.
3. **Speed**: The process is extremely slow, 1K/hour per annotator.
4. **Utility**: (a) Research communities carrying out other tasks need to adopt it, (b) Density and annotation speed is an obstacle.
5. **Invalid Annotation**: Since the current annotation tools were not designed to produce XML, they occasionally produce invalid XML documents.



TimeML tag frequencies in 56.6K bytes (raw) dataset

Figure 1: TimeML Density Information

1.2 Workshop Goals

The goals of the workshop were to:

1. Create a graphical annotation tool for dense annotation tasks (such as TimeML);
2. Embed an interactive closure algorithm into the annotation environment, which helps compute event and temporal relationships automatically.

Given that TimeML-style annotation is a useful (if not necessary) addition to any future efforts in question answering system design, we addressed the above challenges in the following ways:

1. **Inconsistency**: Graphical annotation with closure reveals inconsistent mark up and catches errors.
2. **Density**: The new tool will move away from textual annotation for links towards Graphical Annotation. Visualization is helpful in any link analysis task and reduces the cognitive load of the annotators.
3. **Speed**: We will use radical mixed-initiative architecture, involving massive pre-processing and interactive post-processing (including temporal closure).
4. **Relevance**: This new effort should build links to other communities, by showing value (e.g., Q&A, summarization, MT) with faster and more accurate annotation.
5. **Invalid Annotation**: The new tool will be designed to XML-compliant, so it will only produce well-formed XML.

In this workshop, we designed and built a graphical annotation toolkit, called *TANGO*. The design and functionality of this toolkit took advantage of lessons learned during the TERQAS workshop, for which we developed a semi-graphical annotation tool, built by Marc Verhagen. This tool, called *Event Diagram*, was a first step towards a fully editable timeline annotation tool. It is a semi-graphical annotation tool that displays spatially how a selected event is temporally related to all other events and time expressions in a text. Adding links with this tool proved to be simpler and approximately twice as fast as in Alembic. It was also much easier to change focus and follow a chain of links. In addition, it was possible to zoom in on one segment of a text, ignoring all events and time expressions outside the segment. Event Diagram is an improvement over Alembic-style annotation, but it does not fully solve the problems stated above.

The general features of the proposed toolkit, *TANGO,* are:

1. Graph visualization of temporal and event orderings;

2. Simultaneous Display of Document Text and Graphical Timeline;
3. Standard graph viewing controls;
4. Close coupling with Temporal Closure Algorithm.
5. Native XML output.

The tool has two main widgets representing the text and the timeline. The text widget displays the running text of the document. It highlights temporal entities, events and other relevant tags in the input (signals, states). It should be possible to directly annotate TimeML into the text. It should also be possible to select events or time expressions and drop them on the timeline.

1.3 Workshop Deliverables

The workshop has generated a new toolkit for the annotation of text containing high dependency markup. In particular, TANGO will enable the quick construction of new gold standard texts, overcoming the problems and shortcomings described in section 1 above.

Deliverables:

1. *TANGO* Annotation Toolkit: source code and executables, annotation environment, user instructions;
2. Embedded Temporal Closure Algorithm; as a component of *TANGO*: source code and executables.

Chapter 2

Description of Workshop Activities and Technical Results

2.1 Participants and Conduct of the Workshop

James Pustejovsky	Brandeis University (Team Lead)
Inderjeet Mani	MITRE Virginia, Georgetown U. (Co-Team Lead)
Branimir Boguraev	IBM
Jessica Littman	Brandeis University
Marc Verhagen	Brandeis University
Linda van Guilder	MITRE Virginia
James van Guilder	MITRE Virginia
Robert Ingria	Brandeis University
David Day	MITRE Bedford
Svetlana Symenenko	Syracuse University
Luc Belanger	University of Montreal
Andrew See	Brandeis University
Robert Knippen	Brandeis University

Workshop Schedule

The schedule for the workshop was as follows:

1. April 1-3, 2003 Kick-off and Planning Meeting
2. June 9-26, 2003: Major Workshop Activities
3. June 27, 2003: Midterm Evaluation
4. July 6-August 13, 2003: Code Completion, Evaluation
5. August 15, 2003: Final Evaluation

2.2 Specification of the Tango Tool

2.2.1 Introduction

The Toolkit for graphical annotation of TimeML is used to insert TimeML links in such a way that is intuitive to the annotator. It is used in conjunction with another annotation tool such as Callisto or Alembic. Once a document has been fully tagged with <EVENT>, <TIMEX3>, and <SIGNAL> tags, the Tango tool is used to represent the anchoring and ordering relationships that TLINKs contain, the subordinating relationships that SLINKs signify, and the aspectual information that ALINKs supply. The Tango tool offers several useful features to facilitate the annotation of TimeML links.

2.2.2 Functionality

The Tango user interface (Figures 1 and 2) is split into three parts. The top of the window (the *text window*) contains the text of the document as a reference for the annotator. In the future, this window will most likely be replaced by Callisto, thus making it possible to add non-link tags to the document. Currently, the text window is static, although text that has previously been captured with a TimeML tag is color coded to represent this information. All of these colors can be modified by the user, but, in this document, green text represents events, blue text represents time expressions, and purple text represents signals.

The left side of the Tango window (the *pending window*) contains a list of all of the tags that have not been moved to the graphical annotation palette (the final section of the window). When a document that has no links in it is first opened in Tango, this list will contain all of the events that exist in the document. It may also contain some signal tags. It will not contain any time expressions because they are automatically placed in the palette window.

The final portion of the Tango interface is the graphical annotation palette (the *palette window*). It is here that the annotator draws relationships between time expressions, between event instances, or between a time expression and an event instance. When a document is first opened in Tango, providing it does not already contain links, this window will be empty except for an automatically generated timeline at the top of the window. The top line in this window represents the ordering of those time expressions with comparable ISO values. They are ordered from left to right with the leftmost expression being the earliest ISO value. The document creation time is marked in red with two stars to distinguish it from the rest of the time expressions. Any time expressions that could not fit into this ordering are listed below the initial line. The annotator can discover the ISO value for any time expression in the palette window by scrolling the mouse over the time expression. This results in a tool tip that provides the complete ISO value, if possible, or any other helpful information provided in the specification of the given time expression.

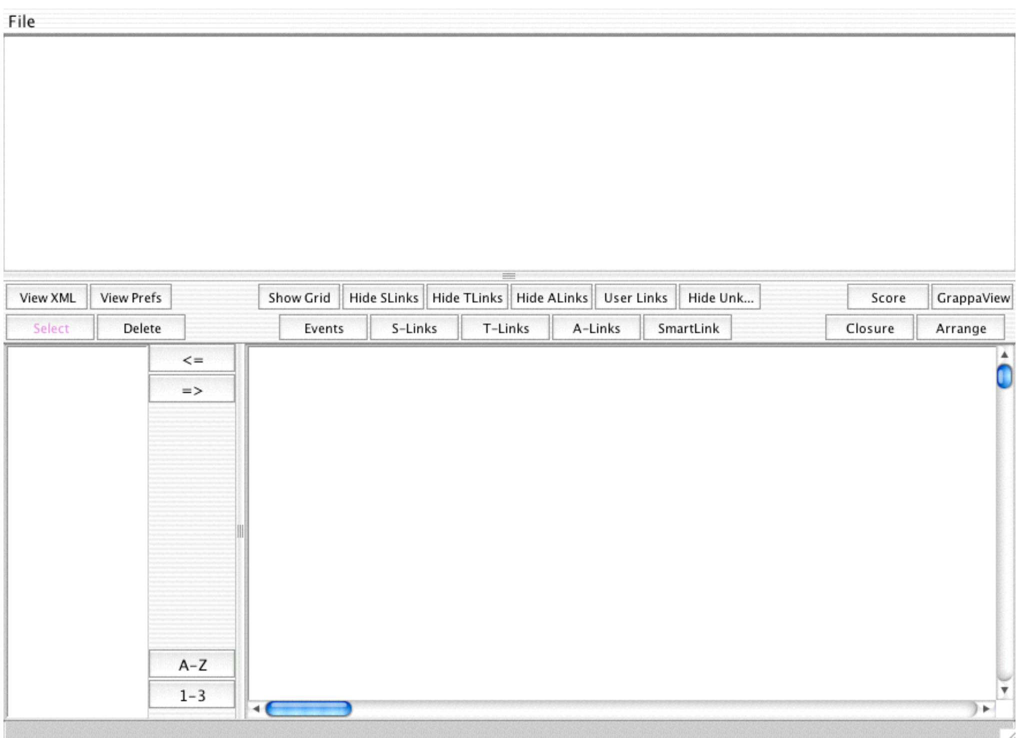


Figure 2: Initial Tango User Interface

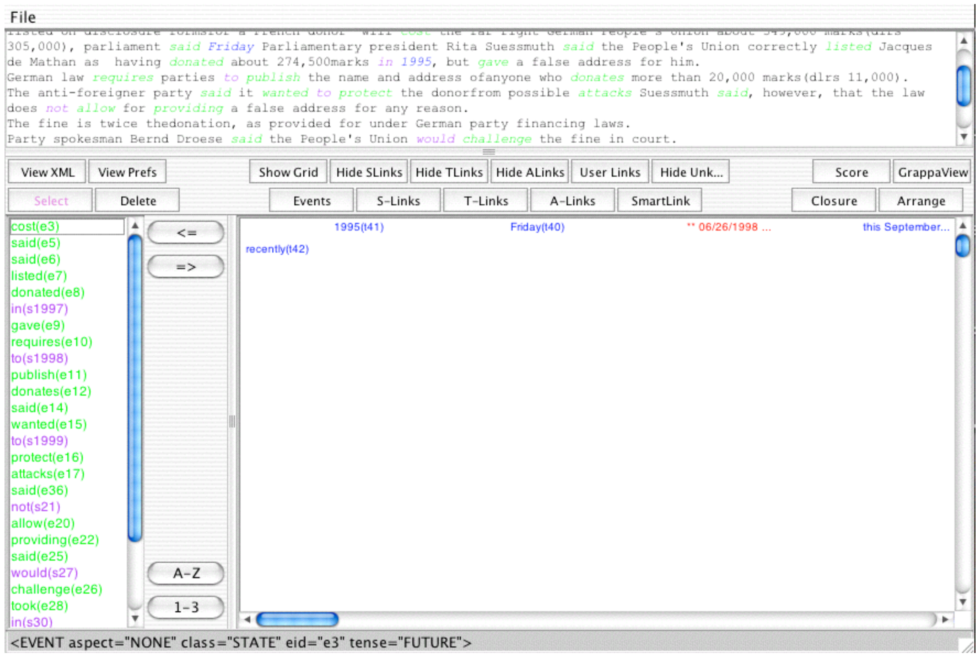


Figure 3: Tango Interface (pre-link insertion)

The primary use of the Tango toolkit is to facilitate the manual insertion of TimeML links. Before a link that involves an event instance can be drawn, the instance must be created by moving the event from the pending list to the palette. This is done by selecting the event from the pending list and clicking the '=>' button (Figure 3). This automatically generates an instance of the event that can now participate in a TimeML link. The user can move the event instance around the palette by selecting the 'Events' button and dragging the instance wherever he or she wants it. The user can also add several event instances at once to the palette by holding the shift or control key while selecting the events from the pending list. Finally, double clicking on an event instance in the palette window has the effect of generating a new instance of that event. However, the multiple instances of that event are not distinguished from each other in the palette. Scrolling over the event instance in the palette window results in a tool tip that provides the class of the event (Figure 4). This information is often useful for the annotator when he or she is manually generating links.

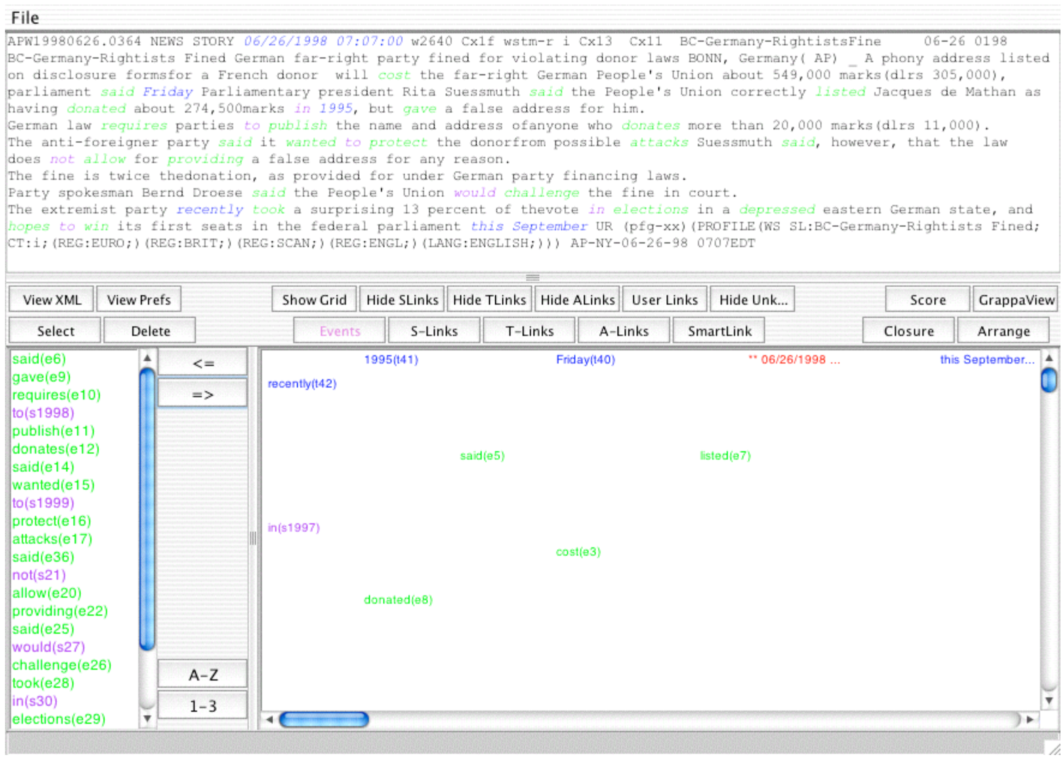


Figure 4: Adding Events and Signals to the Palette

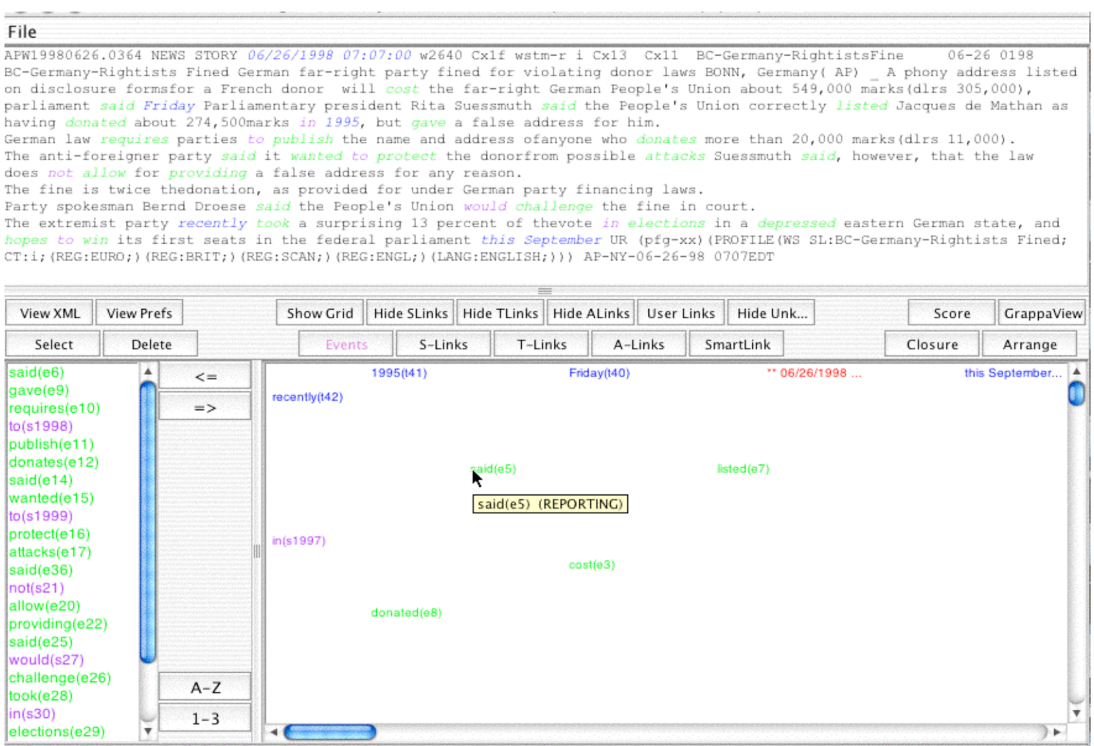


Figure 5: Event Instance Tool Tip

Links can be inserted by selecting the type of link from the tool bar and then simply drawing a line between the two TimeML entities that share the relationship. Once the annotator releases the mouse button, a window will pop up asking for the specification of the link. This gives the annotator the opportunity to specify the relationship type between the two entities (Figure 5). If the link requires a signal ID, the signal must first be moved from the pending list to the palette. Once this is done, the signal will appear in the Signal ID pop-up window. Once a signal has been used in a link, it is automatically removed from the palette (Figure 6).

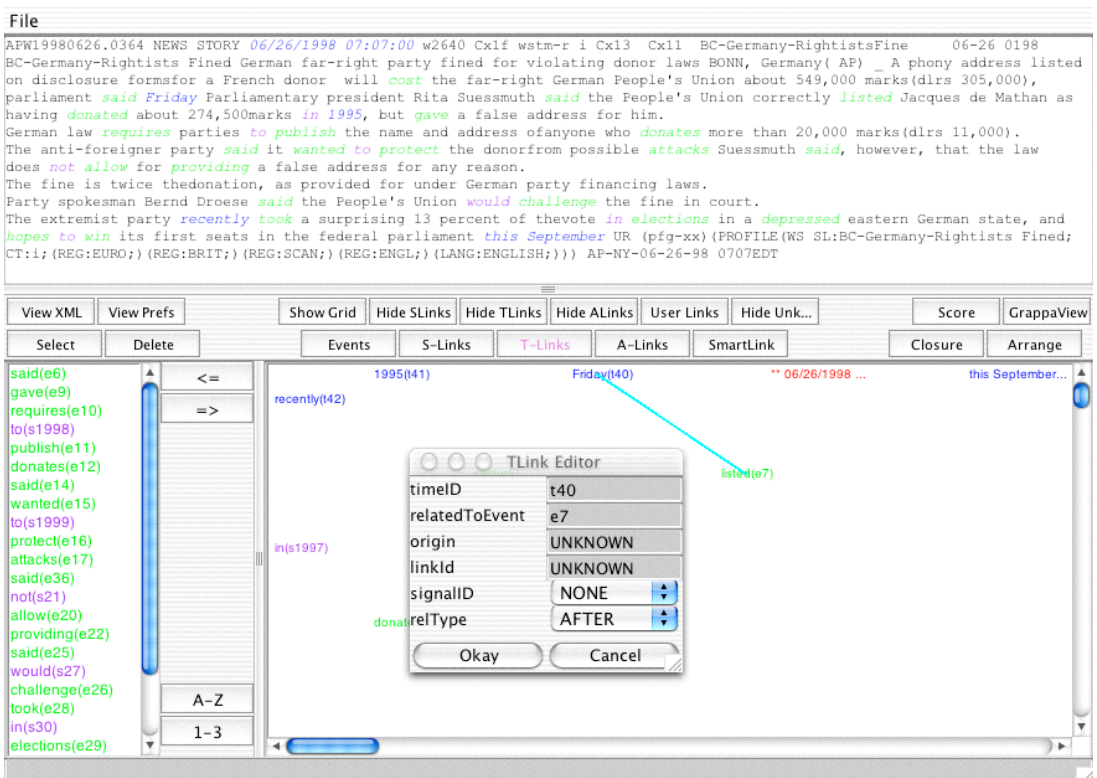


Figure 6: Inserting a TLINK

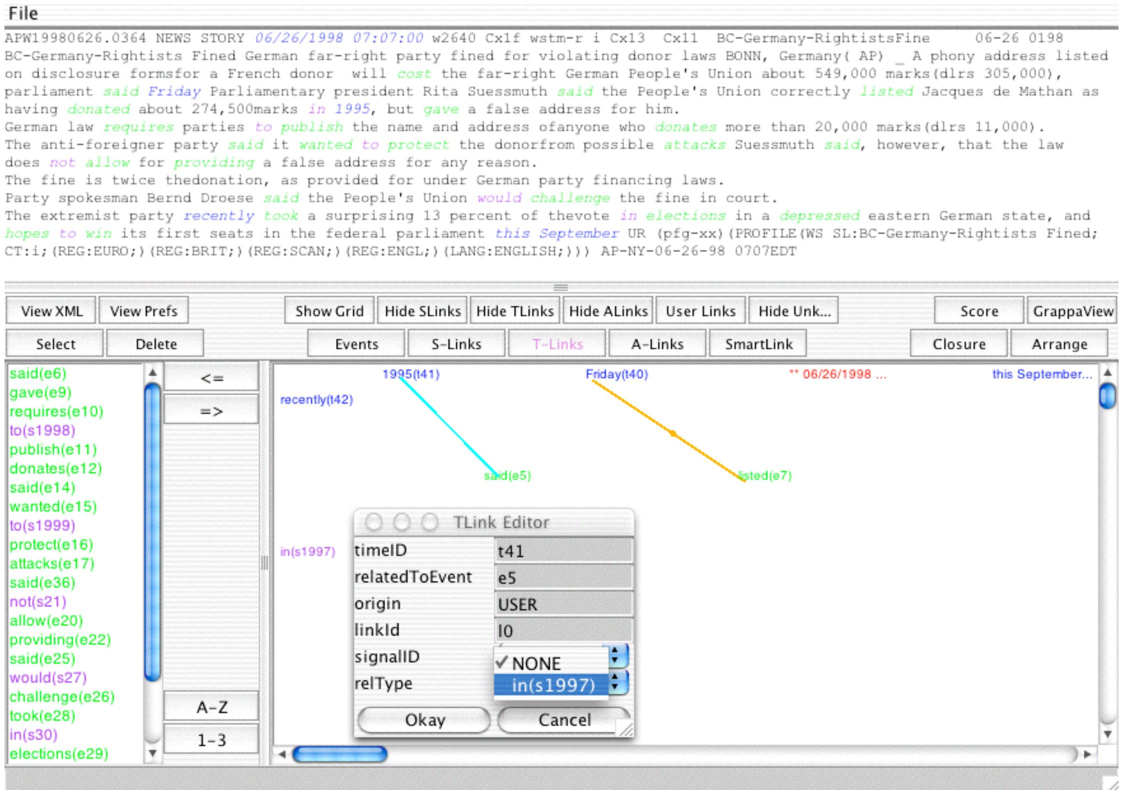


Figure 7: Using a Signal in a Link

Each kind of link in TimeML is color-coded. Orange lines represent TLINKs, red lines represent SLINKs, and aqua lines represent ALINKs (Figure 7). Again, all of these colors can be changed by the user (by selecting View Prefs). Scrolling over a link in the palette window results in a tool tip that explains what the link is (Figure 8). If the annotator wants to change some of the information in a given link, he or she can either delete the faulty link and insert a new one or double click on the current link to bring up the specification window for that particular link.

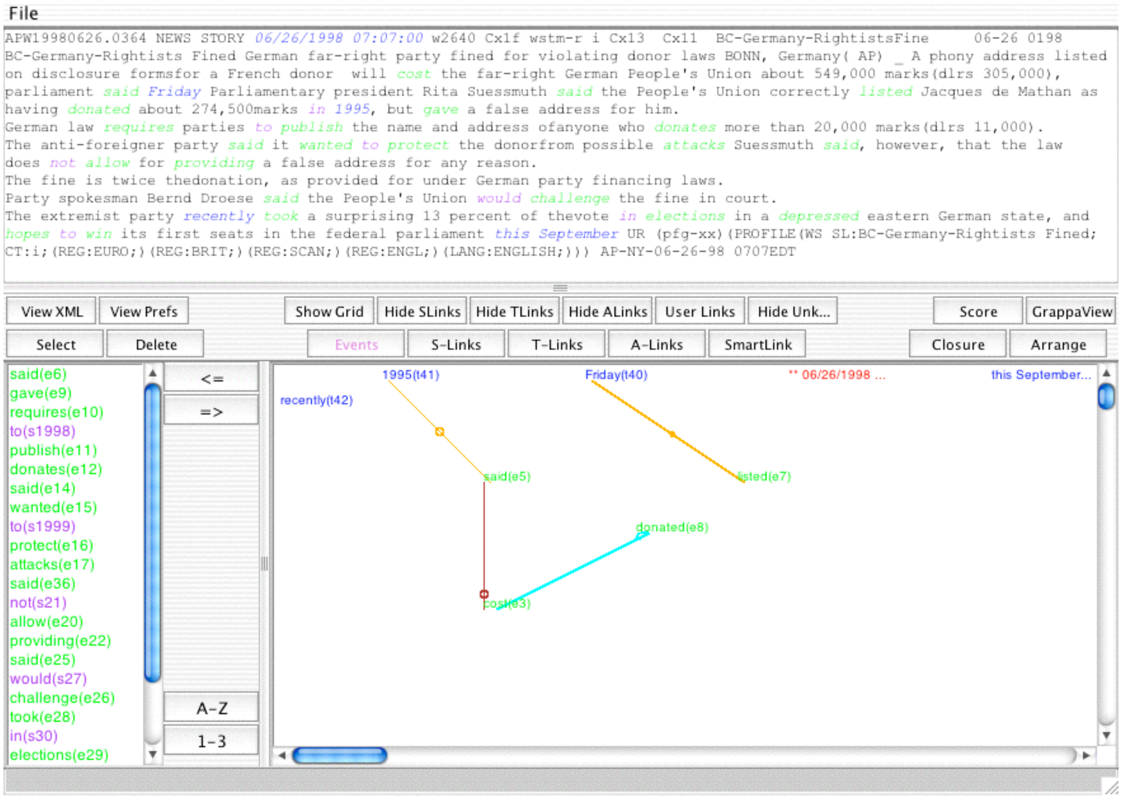


Figure 8: Link Color Coding

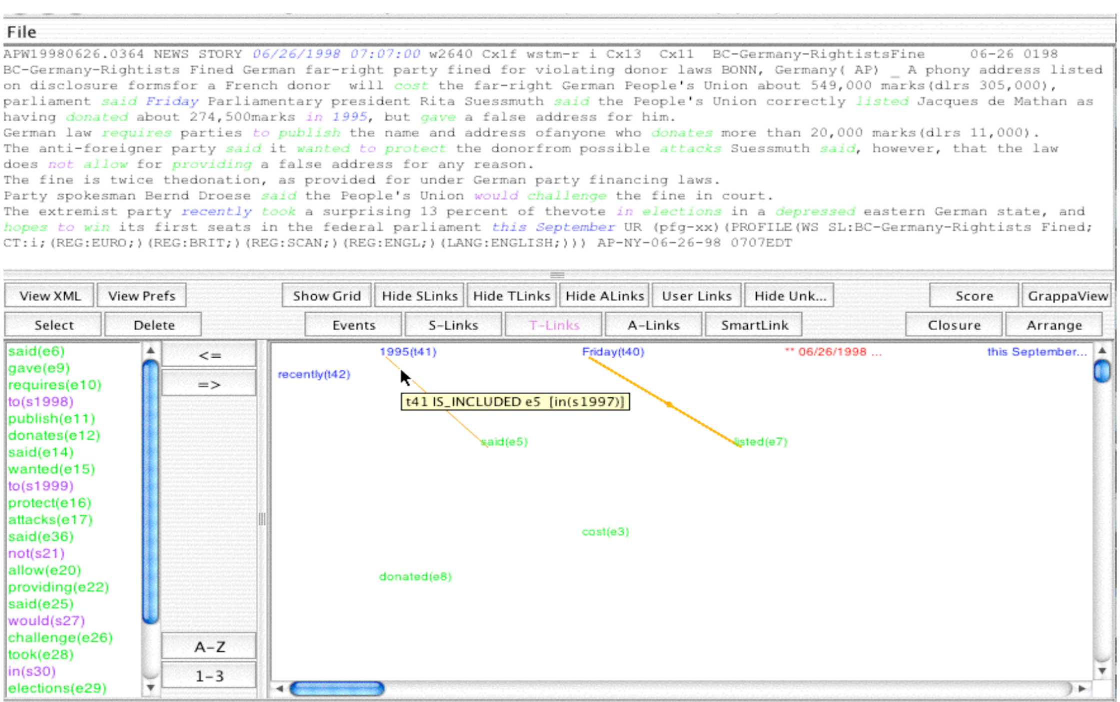


Figure 9: Link Tool Tip

Most of the links in a given document are TLINKs.. Many event instances are anchored to specific times using this tag while temporal orderings between event instances and time expressions are also represented with TLINKs. The SmartLink tool facilitates the creation of these kinds of links. To use this feature, the annotator selects the SmartLink button and then holds the control key while clicking on the item in the palette window that will be included in all of the TLINKs generated by SmartLink. The user can then drag another item in the palette window to the left or right of the selected item. If the second item is dropped to the left, then a BEFORE TLINK is automatically produced. If it is dropped to the right of the selected item, an AFTER TLINK is created. If the second item is dropped below the selected item, SmartLink will pop up the TLINK specification window so that the user can specify exactly what relationship he or she means by this action. Figures 9 through 11 demonstrate the functionality of SmartLink.

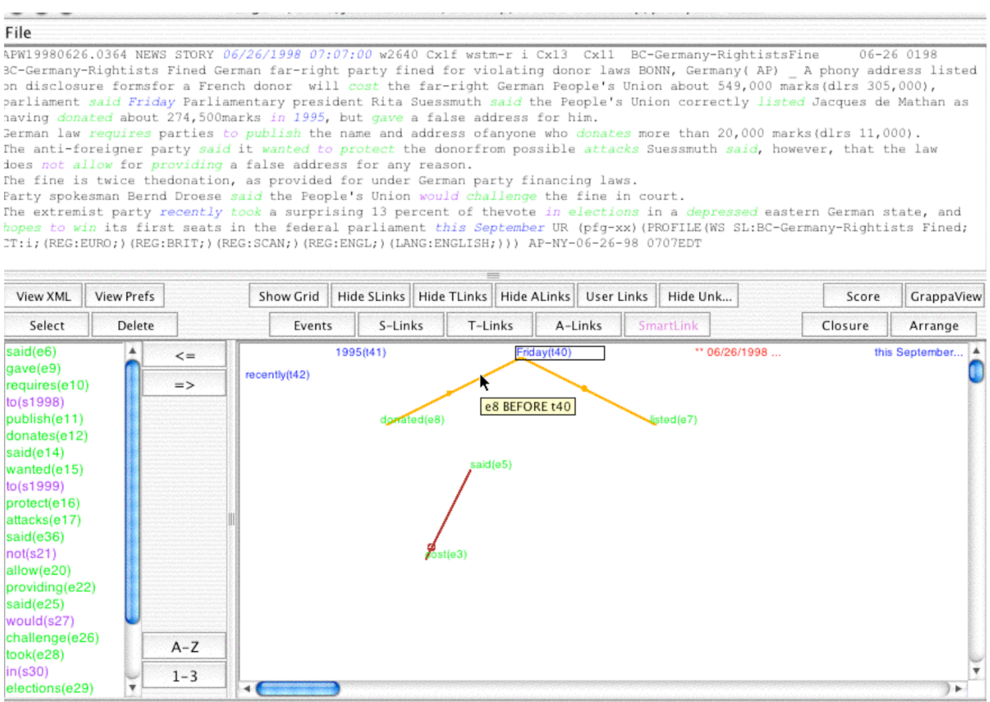


Figure 10: Using SmartLink for BEFORE Links

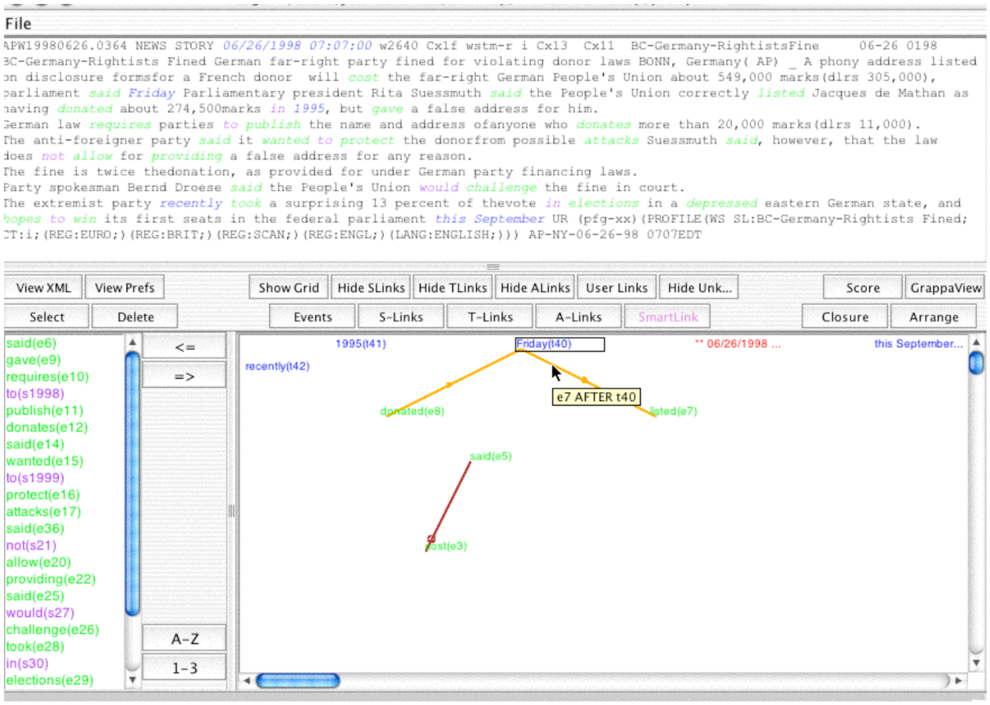


Figure 11: Using SmartLink for AFTER Links

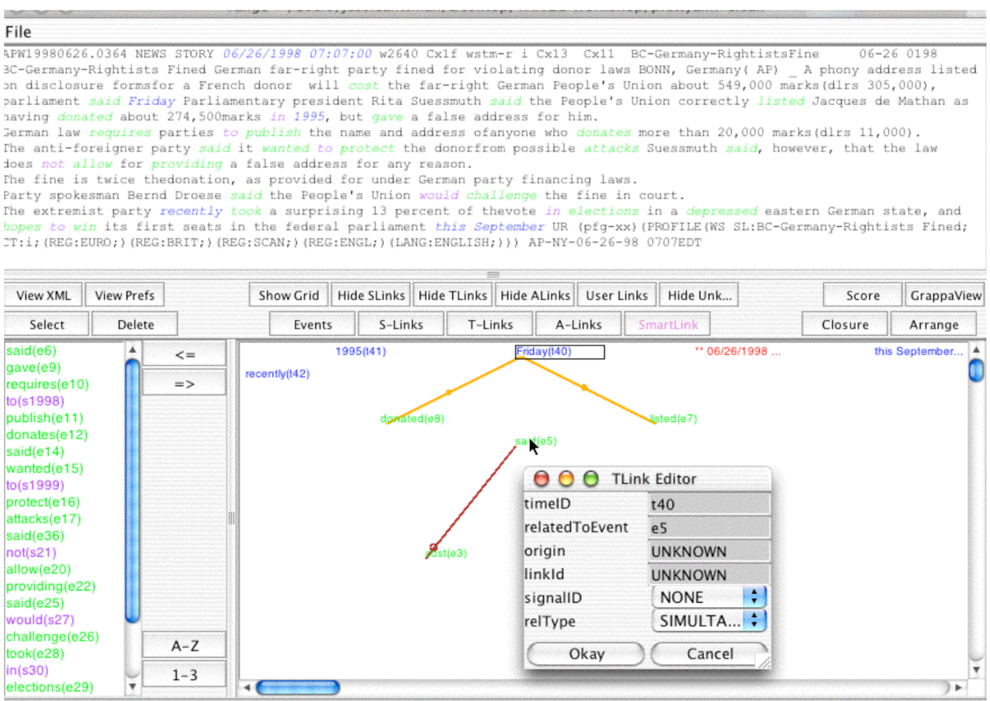


Figure 12: Using SmartLink for Other Link Relations

Once a document has been annotated with links in the Tango tool, the user has several choices for how to view the graphical annotation. Many of these features are also useful throughout the annotation process to remove clutter from the palette while the annotator is working. Buttons for each of these features can be found in the top, center row in the tool bar. The Show/Hide Grid button allows the user to toggle a grid on or off for the palette window (Figure 12). The Hide/Show buttons for each of the link varieties allows the user to clear the screen of a particular kind of link. For instance, a fully annotated document can have so many TLINKs that it is hard to see the SLINKs or ALINKs. Hiding the TLINKs allows the annotator to analyze only these links without disturbing the placement of any TLINKs (Figure 13). These buttons can also be used to analyze only those links that are related to a particular item or set of items in the palette window. To make use of this feature, the user begins by hiding all three kinds of links. Making sure that the 'Select' button in the toolbar is highlighted, the annotator can then select an item in the palette window. Only those links that contain that particular item are shown (Figure 14). To select a set of items when using this feature, the user holds down the control key while making his or her selections (Figure 15).

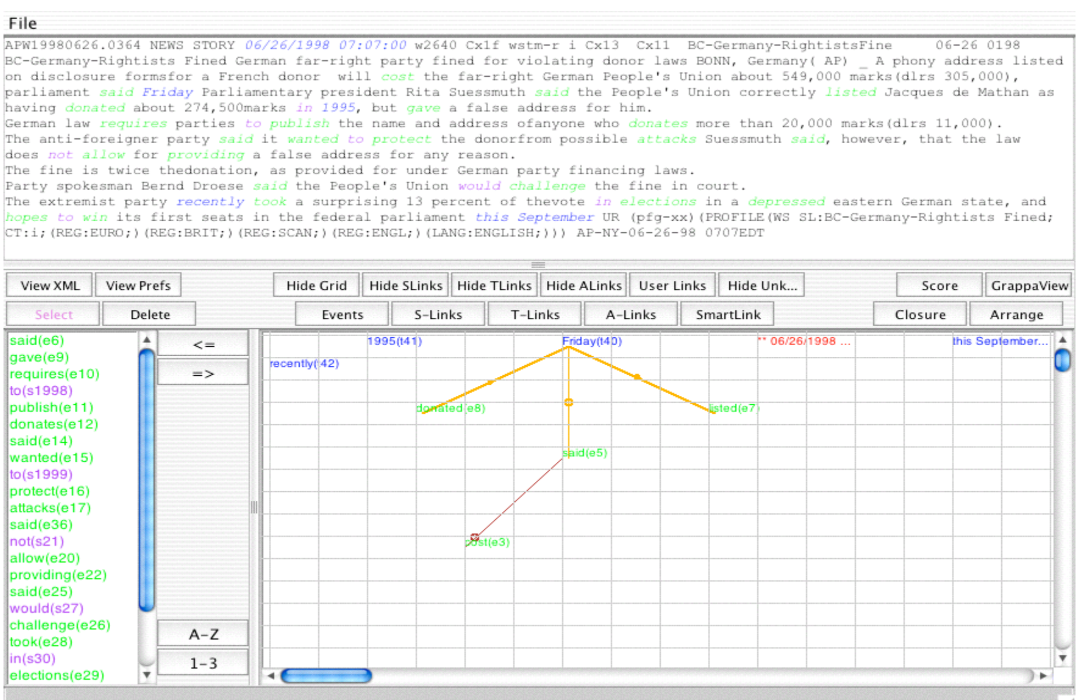


Figure 13: Using Tango with the Grid

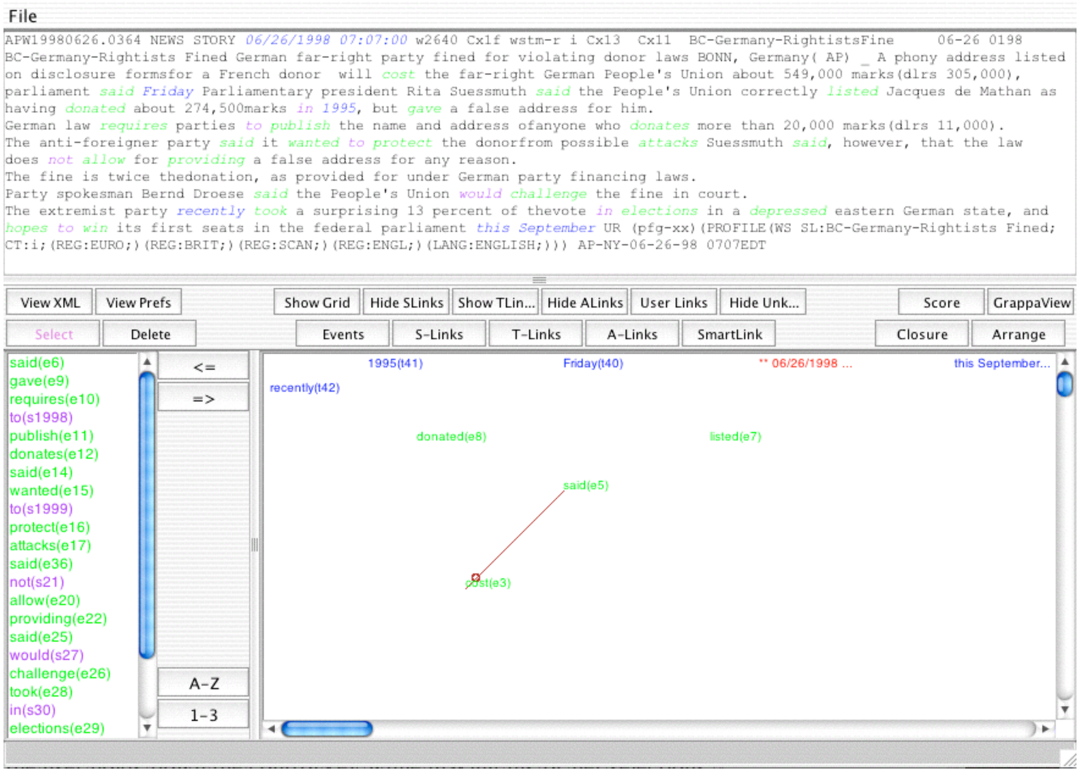


Figure 14: Examining only SLINKs

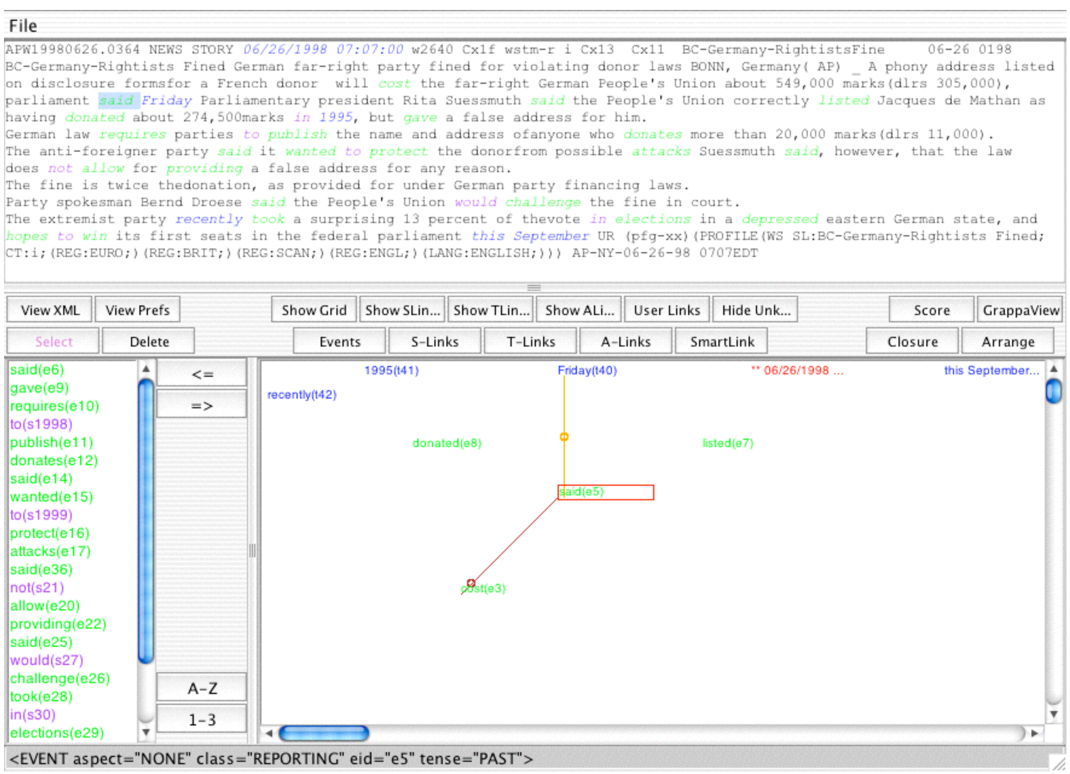


Figure 15: Selecting One Item

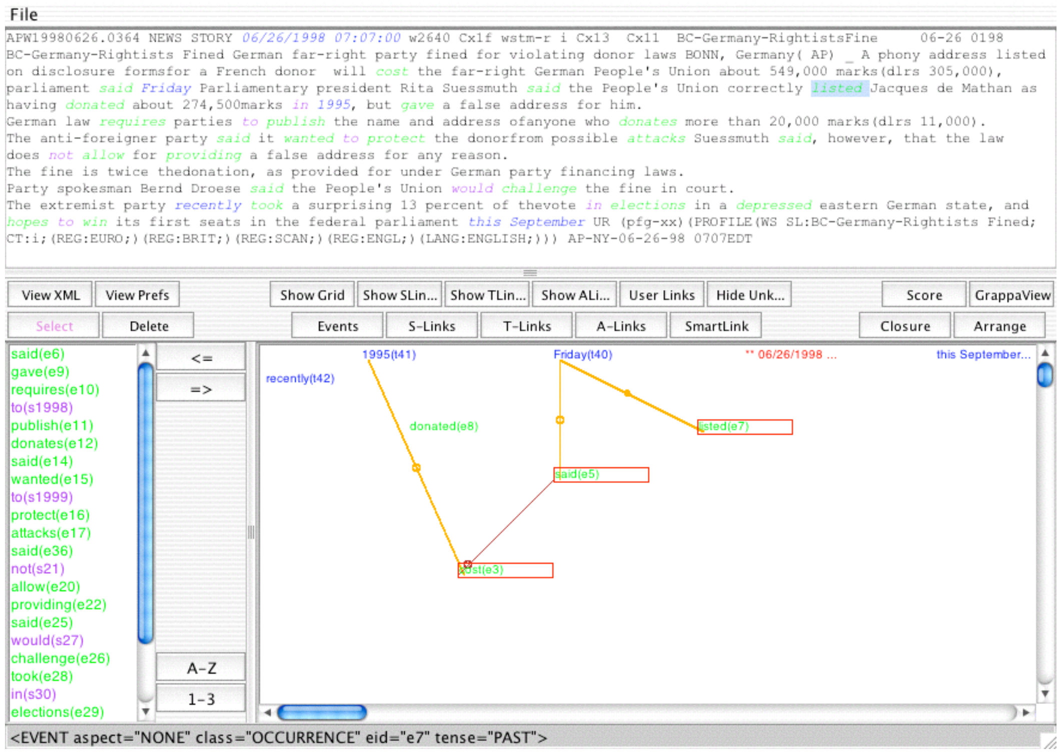


Figure 16: Selecting Multiple Items

2.2.3 Auto-Layout Feature

In addition to the manual link insertion capabilities of the Tango tool, the software is also useful for link analysis of documents that have already been fully annotated with TimeML links using another program or by hand. When such a document is opened in Tango, all of the event instances that participate in links are listed on the left side of the palette window. The timeline at the top of this window is ordered as if the document had been opened in Tango with no links. The annotator has several options for manipulating the layout of the palette window to assist with his or her analysis of the document. By hiding certain link types, the user can more easily see which event instances participate in what links. The user can also manually move each event instance to another part of the palette window to organize the annotation. However, this can be a tedious process, especially with a large, fully annotated document. The auto-layout feature in Tango is a powerful tool for mechanically organizing the palette window for the annotator.

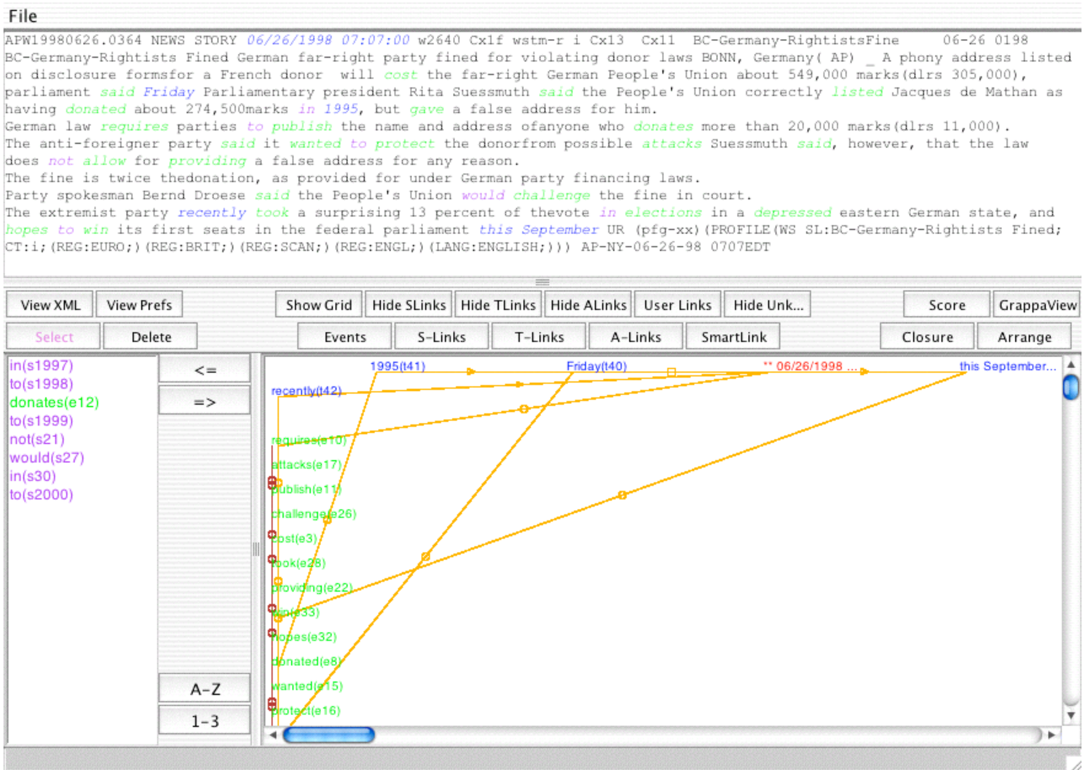


Figure 17: Opening a Fully Annotated Document

The ‘Arrange’ button calls the Auto-Layout routine. It organizes the event instances within the palette window in four ways:

1. The left to right ordering of BEFORE links and their inverses is respected.
2. IDENTITY, SIMULTATENOUS, and INCLUDES links are represented by placing event instances in the same column as the associated time expression.
3. Event instances are placed horizontally with respect to the time expressions with which they are associated.
4. Time expressions that are represented in the same column are sorted in such a way as to reduce link crossing and to make it clear when links are not present.

The resulting palette window after the Auto-layout routine has been run resembles the kind of organization many annotators wish to enforce manually.

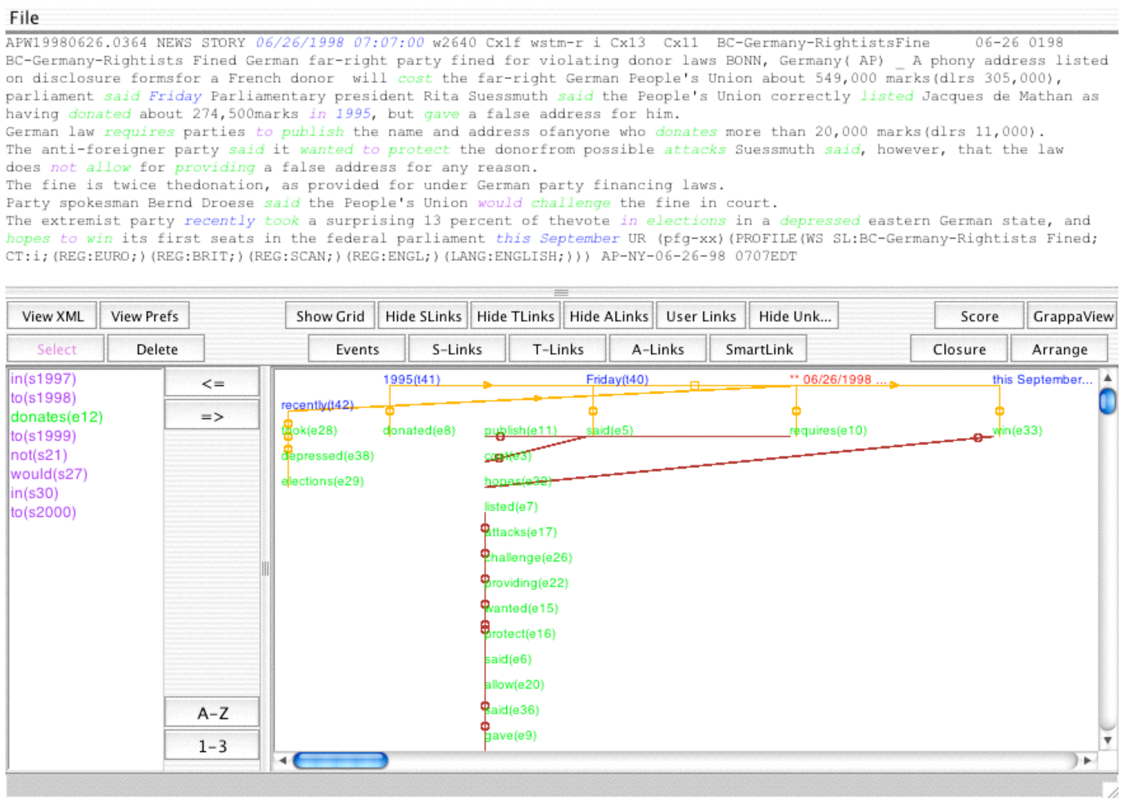


Figure 18: Auto-Layout Organization

The auto-layout algorithm goes through three stages. First, the event tags are assigned to ranks (columns) based on the BEFORE, AFTER, and similar links. This is done by a topological sort. The algorithm is as follows:

```
procedure initRanks:
        For each tag assign tag.degree to be the number of incoming
        AFTER links and outgoing BEFORE links.  Initialize tag.
        rank to 0.

        Insert all tags with degree 0 into queue Q.

        While Q is not empty do:
                dequeue tag t
                foreach link (t BEFORE t2) or (t2 AFTER t) do:
                        t2.rank := t.rank + 1
                        t2.degree := t2.degree – 1
                        if t2.degree = 0 then Q.enqueue(t2)
```

```
        end
    end
```

Next, the event instances are anchored to the time expressions. For all INCLUDES, HOLDS, SIMULTANEOUS and related links to time expressions, event instances are placed directly under the time expression. For BEFORE links from an event instance to a time expression we ensure that the event is to the left of the time expression. Similarly, AFTER links ensure that the event is to the right of the time expression. Event instances linked to other event instances that have moved are then adjusted recursively in the same manner.

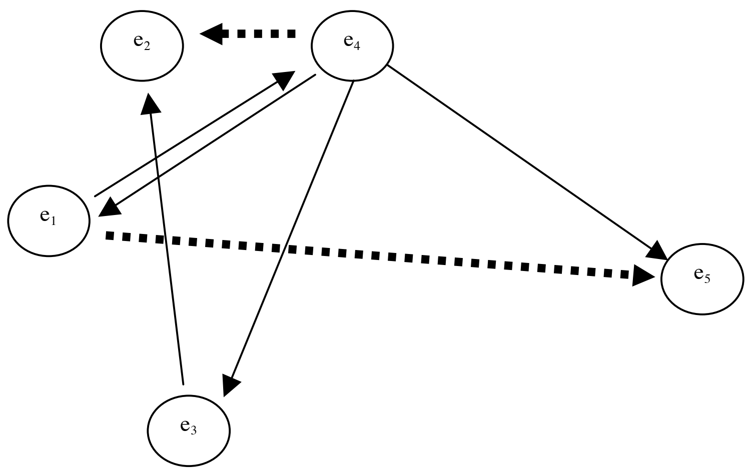
Lastly, within the columns of the graph, tags are sorted to improve the appearance of the graph. Currently, auto-layout attempts to place events that are more directly related to time expressions closer to the top of the panel. Other things that could be done but are not yet implemented include placing identical or simultaneous events closer together in the column or arranging the columns to reduce link crossings between multiple columns.

2.2.4 Closure Feature

The closure algorithm is a complete axiomatic system based on the Warshall algorithm for graph closure. It is used to infer additional links in the event graph, completing the annotation and ensuring that it is consistent.

The input to the closure algorithm is an event graph where each node represents a TimeML event or time expression and each arc represents a TimeML temporal link, for example, the following subgraph:

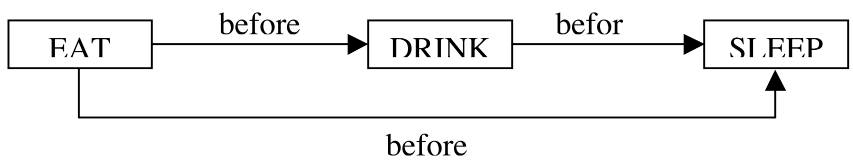
The Warshall algorithm closes graphs by processing each node one-by-one. When some node *i* is processed, it checks for paths from *x* to *i* to *z* (that is, it checks to see if *i* has both incoming and outgoing links). If so, it generates an arc from *x* to *z*. For example, in the following graph, nodes e3 and e4 each have both incoming and outgoing arcs, so the Warshall algorithm would close the graph by processing e4 and adding and arc from e1 to e5, and then processing e3 and adding an arc from e4 to e2.



While the Warshall algorithm can add an arc between any two nodes using this procedure, the event graph closure algorithm has to be able to determine whether the relation types involved license such an addition. Thus, the complete axiom set contains axioms expressed as properties of nodes. For example, it contains the following axiom, which allows a BEFORE link to be added any time an event is found which has both an incoming BEFORE link and an outgoing before link:

IN: BEFORE
OUT: BEFORE
NEW: BEFORE

So when the event graph closure algorithm processes the subgraph above, it finds that the DRINK node contains both incoming and outgoing arcs, so it has to check whether there is an axiom which allows a link between EAT and SLEEP. Since they are both BEFORE links, the axiom above says it is possible to generate a BEFORE link from EAT to SLEEP:



Generating the axiom set

Because of the number of different TimeML relations, and the nature of those relations, he complete set of axioms need to perform temporal closure would be very difficult to construct by hand. Thus, an algorithm was designed which uses the properties of partial orderings (transitivity of equality and precedence) to automatically generate all the closure axioms needed. To generate the complete set of axioms, each TimeML relation must first be converted into a relation between two intervals, allowing us to derive precedence and equality statements between the begin and end points of the two intervals:

That is, the x BEFORE y relation is reduced to the following sorts of statements about points in time:

> $x1 < x2$
> $y1 < y2$
> $x2 < y1$

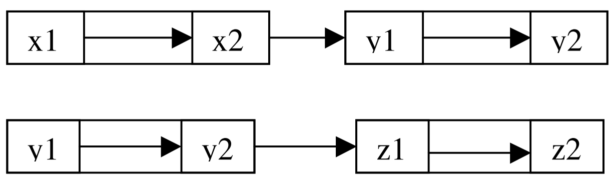
Thus, each relation is, in effect, reduced to a partial ordering. This allows one to use the transitivity of equality and precedence to infer more relations between the points involved:

x1 < y2

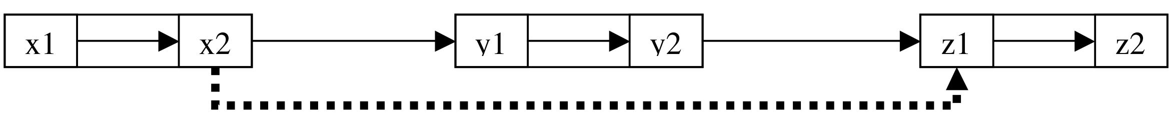
x2 < y2

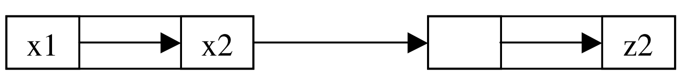
…

All possible pairings of TimeML relations are generated and reduced to relations between points. Then a graph closure algorithm is used to compute all possible point-to-point relations for each pair. The new point-to-point relations are checked to see if any are equivalent to a TimeML relation. If so, a new TimeML relation is inferred. Since a new TimeML relation is generated based on assuming two other relations, an axiom can be generated. For example, from x BEFORE and y BEFORE z, the following set of point relations are generated:



From these relations, the following inference can be made via transitivity of equality and precedence:





Since the second set of statements translates straightforwardly into the TimeML relation x BEFORE z, the axiom

IN: BEFORE

OUT: BEFORE

NEW: BEFORE

can be automatically derived. By using the algorithm described above, the complete set of all possible axioms over TimeML relations (638 axioms) was generated.

By making temporal information explicit, closure will simplify and optimize event graph querying, event graph comparison, and timelining, all of which are basic ways of answering temporal questions.

2.2.5 Scoring Feature

The goal of the scorer is to compare the annotation of links from the current document against the ones from a reference document. The document opened in the Tango GUI is the current document, the reference document is chosen by the user with a file chooser dialog box when he or she presses the score button. One limitation of the scorer is that the two documents must have the same annotation for the EVENT and TIMEX3 tags. This is because the link tags are the only ones added by Tango and it does not make sense to score links when the underlying EVENT and TIMEX3 tags are not the same.

When the reference document is chosen, the scorer uses it as a baseline to compute precision, recall, and the F-measure of each of the link tags. For each link of the reference document, the algorithm takes the two endpoints of the link, which are TIMEX3 IDs or IDs of an instance of an event, and tries to find a link in the current document with the same endpoints. When this is the case, the link is counted as correct and the algorithm proceeds to check if the relation type is the same for the two links. After all the links (TLINK, SLINK and ALINK) are processed, a new window is created showing the results as follows.

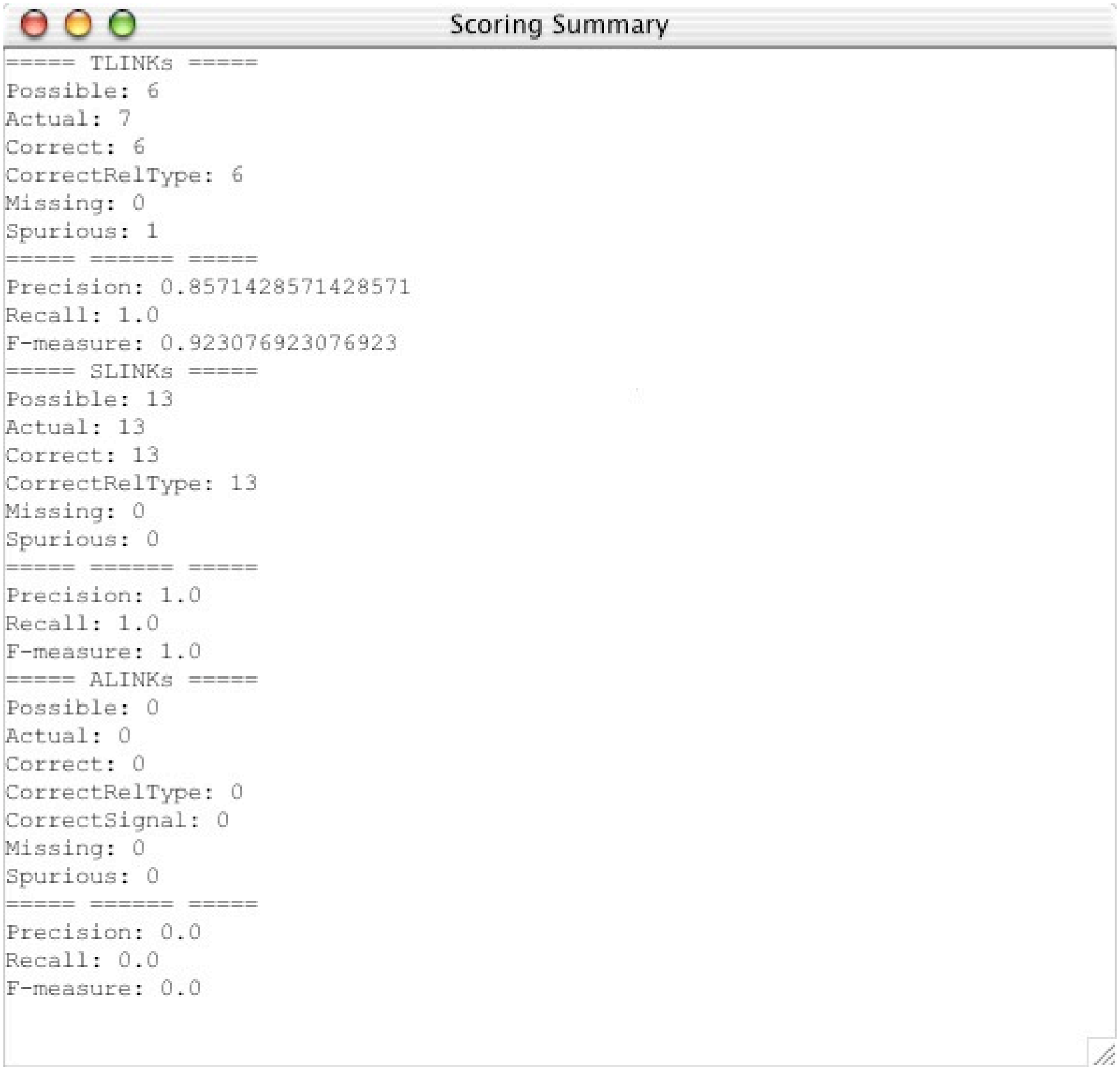


Figure 19: Resulting Window from Score Feature

2.3 Exploratory Evaluation of the Tango Tool

An exploratory valuation of the Tango toolkit took place before the mid-term of the workshop while the software was still being developed. This evaluation was exploratory, as a sanity check and a guide for further development. It should not be viewed as a substitute for a more full-fledged evaluation. The evaluation proceeded as follows:

1) Seven documents from TimeBank were checked for accuracy. This included examination of the events and time expressions, as well as the quality of the

current links. These clean documents are used in the evaluation as references. Versions of the documents were created without links for the evaluation task.

2) One document was chosen as a training article. Each annotator was given half an hour to familiarize themselves with the task in both Alembic Workbench and Tango.
3) Each annotator was given one hour to annotate their documents in a predetermined tool. The annotators were instructed to insert as many links as possible within that hour. Each annotator alternated which tool and document he or she began with. After the first hour, the annotators began with a new document on the next annotation tool for one additional hour.
4) After the task was complete, the annotators were offered the opportunity to make comments comparing Alembic and Tango.

Three annotators participated in the evaluation. All had roughly the same amount of training and experience using Tango. In addition, all three had previous training using Alembic with varying levels of experience. An accuracy metric was defined as the percentage of TLINKs in common with the reference annotation (shared TLINKs) that had the same relType. A TLINK was considered shared if it was found directly in the reference annotation, or if it could be derived from the reference document by closure. In short, the accuracy metric measured the correctness of the type and direction of TLINKS.

The results of this informal evaluation (Figure 19) showed that annotators were more accurate using TANGO (naïve: 36% more; expert: 19% more). Naïve annotators (those participants who were not trained as linguists, shown with 'N' in the legend in Figure 19) were just as fast using TANGO, but the expert was slower with TANGO, perhaps due to the increased focus on ordering encouraged by TANGO.

At the time of this mini-evaluation, the participants offered some helpful commentary on the tool. They reported that the automatic placement of time expressions and the graphical nature of the annotation made the task more intuitive and allowed for a more "holistic picture of the narrative". They acknowledged that these aspects of Tango facilitated connections between different parts of the document and allowed for higher accuracy. The participants suggested the use of multiple annotation modalities, citing their belief that a tabular interface is useful when dealing with dense texts and nonnarrative discourse. Such an interface would also be helpful for sorting links and might allow for quicker creation of local links. The integration of the new Callisto software offers a solution to this problem. Lastly, the annotators suggested that it would be helpful if a section of the graph could be selected from text view and if events could be laid out automatically. The former has not yet been implemented because the text view in Tango should eventually be replaced by Callisto. The latter suggestion, auto-layout of events, has been implemented fully into the Tango toolkit.

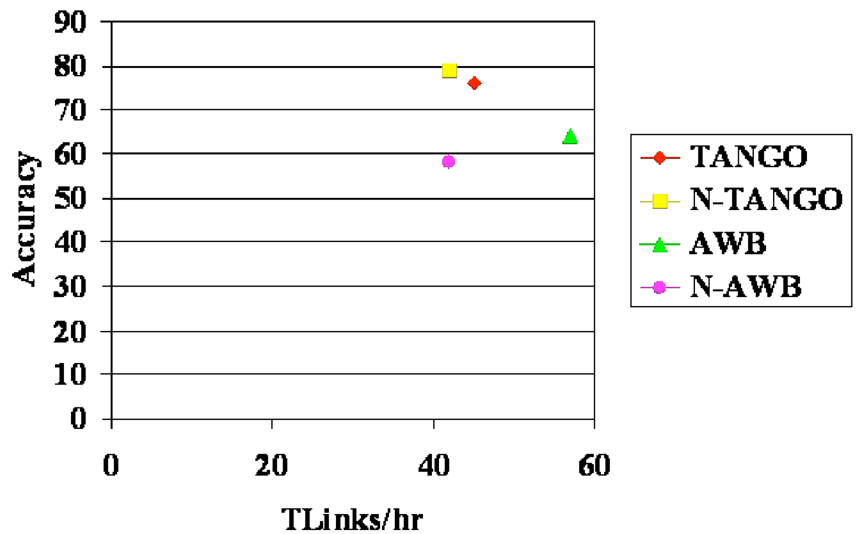


Figure 19: Accuracy and Speed using TANGO and Alembic

2.4 Update to TimeML

2.4.1 Introduction

While the Tango tool was being created, it became apparent that the specification of TimeML could be updated to provide a stronger language for the annotation of documents with regards to temporal entities. Some of the changes in the new specification, called TimeML 1.1, simply involve changing the name of some attributes. These modified names better reflect the intended use of the attributes. In addition to these minor adjustments, the following modifications were made:

1. The EVENT tag has fewer attributes, while further importance is given to the MAKEINSTANCE tag.
2. Text that was tagged as a signal in TimeML1.1 because it generated a modal or negative SLINK is no longer captured with a TimeML 1.2 tag.
3. Such information is now reflected in the MAKEINSTANCE tag rather than an SLINK.
4. Anchored durations can now be represented in a TIMEX3 tag with non-text consuming TIMEX3s that represent the implicit begin point or end point of the duration.
5. The TIMEX3 tag can now capture information about a set of time such as "every Monday".

2.4.2 TimeML 1.1 Specification

2.4.2.1 Explanation of Naming Conventions and Examples

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.

2.4.2.2 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

eid ::= ID
{eid ::= EventID
EventID ::= e<integer>}
class ::= 'OCCURRENCE' | 'PERCEPTION' | 'REPORTING' | 'ASPECTUAL' |
'STATE' | 'I_STATE' | 'I_ACTION'
```

<MAKEINSTANCE>

MAKEINSTANCE is a realization link; it indicates different instances of a given event. Since different instances can have different attribute values, the tense and aspect of the event are represented within this tag. In addition, if the instance is modified by a negation or modal operator, this is represented in the appropriate attributes within this tag. 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 tense aspect negation [modality] [signalID]
[cardinality]

eiid ::= ID
{eiid ::= EventInstanceID
EventInstanceID ::= ei<integer>}
eventID ::= IDREF
{eventID ::= EventID}
tense ::= 'PAST' | 'PRESENT' | 'FUTURE' | 'NONE'
aspect ::= 'PROGRESSIVE' | 'PERFECTIVE' | 'PERFECTIVE_PROGRESSIVE' |
'NONE'
negation ::= 'true' | 'false'
{negation ::= boolean}
modality ::= CDATA
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.

We expect that the `tense` and `aspect` attributes will have their values filled in by a pre-processing program, according to the following paradigm:

Active Voice
`tense="PRESENT"`

Verb group	**`aspect=`**
teaches	`"NONE"`
is teaching	`"PROGRESSIVE"`
has taught	`"PERFECTIVE"`
has been teaching	`"PERFECTIVE_PROGRESSIVE"`

tense="PAST"

Verb group	**aspect=**
taught	"NONE"
was teaching	"PROGRESSIVE"
had taught	"PERFECTIVE"
had been teaching	"PERFECTIVE_PROGRESSIVE"

tense="FUTURE"

Verb group	**aspect=**
will teach	"NONE"
will be teaching	"PROGRESSIVE"
will have taught	"PERFECTIVE"
will have been teaching	"PERFECTIVE_PROGRESSIVE"

Passive Voice

Note: Forms marked with (?) do not seem fully acceptable. They are included to show the full logical paradigm.

tense="PRESENT"

Verb group	**aspect=**
is taught	"NONE"
is being taught	"PROGRESSIVE"
has been taught	"PERFECTIVE"
has been being taught (?)	"PERFECTIVE_PROGRESSIVE"

tense="PAST"

Verb group	**aspect=**
was taught	"NONE"
was being taught	"PROGRESSIVE"
had been taught	"PERFECTIVE"
had been being taught (?)	"PERFECTIVE_PROGRESSIVE"

tense="FUTURE"

Verb group	**aspect=**
will be taught	"NONE"
will be being taught (?)	"PROGRESSIVE"
will have been taught	"PERFECTIVE"
will have been being taught (?)	"PERFECTIVE_PROGRESSIVE"

`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 twice on Monday but only once on Tuesday” 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.

The values of `negation` and `modality` are determined by modifiers found near the event in the text. Formally, this information was annotated using a SIGNAL and a SLINK. Some examples:

```
(1) should have bought

should have
<EVENT eid=”e1” class=”OCCURRENCE”>
bought
</EVENT>
<MAKEINSTANCE eiid=”ei1” eventID=”e1” tense=”PAST” aspect=”PERFECTIVE”
negation=”false” modality=”SHOULD”/>

(2) did not teach

did not
<EVENT eid=”e1” class=”OCCURRENCE”>
teach
</EVENT>
<MAKEINSTANCE eiid=”ei1” eventID=”e1” tense=”PRESENT” aspect=”NONE”
negation=”true”/>

(3) must not teach twice

must not
<EVENT eid=”e1” class=”OCCURRENCE”>
teach
</EVENT>
<SIGNAL sid=”s1”>
twice
</SIGNAL>
<MAKEINSTANCE eiid=”ei1” eventID=”e1” tense=”PRESENT” aspect=”NONE”
negation=”true” modality=”MUST” signalID=”s1” cardinality=”2”/>
```

<TIMEX3>

The TIMEX3 tag is primarily used to mark up explicit temporal expressions, such as times, dates, durations, etc. It is modeled 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] [beginPoint] [endPoint]
[quant] [freq] [temporalFunction] (value | valueFromFunction) [mod]
[anchorTimeID]

tid ::= ID
{tid ::= TimeID
TimeID ::= t<integer>}
type ::= 'DATE' | 'TIME' | 'DURATION' | 'SET'
beginPoint ::= IDREF
{beginPoint ::= TimeID}
endPoint ::= IDREF
{endPoint ::= TimeID}
quant ::= CDATA
freq ::= CDATA
{value ::= 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}
```

`functionInDocument`, an optional attribute, indicates the function of the TIMEX3 in providing a temporal anchor for other temporal expressions in the document. If this attribute is not explicitly supplied, the default value is `"NONE"`. The non-empty values take their names from the temporal metadata tags in the Prism draft standard (available at http://www.prismstandard.org/techdev/prismspec1.asp), and are intended to have the same interpretations:

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

Table 4: Elements for Time and Date Information

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

Note that there can be as many instances of TIMEX3s containing a `functionInDocument` attribute with a non-empty value as there are TIMEX3s that express different functions. In practice, there will probably be no more than two, one with `CREATION_TIME` and another with `PUBLICATION_TIME`, since these are likely to be the only attributes that will appear in the text of documents to be annotated. Note that `RELEASE_TIME` does not indicate when the document was actually released. It is a specification of when the document is allowed to be released. This comes up in documents that are syndicated and where the issuing organization wants to delay publication by syndicators, so as not to be scooped.

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

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

The values specified for the `value` attribute---`duration`, `dateTime`, `time`, `date`, `gYearMonth`, `gYear`, `gMonthDay`, `gDay`, and `gMonth`---are the XML time datatypes based on the ISO 8601 standard. See http://www.w3.org/TR/xmlschema-2/ for the definitions of these and the other built-in XML schema datatypes. Since the TIDES guidelines, which we follow in this area, extend the ISO 8601 values, we will need to extend these data types to include these additional values.

`mod` is an optional attribute adopted from TIDES. It is used for temporal modifiers that cannot be expressed either within `value` proper, or via links or temporal functions. Some examples:

```
(4) no more than 60 days

<TIMEX3 tid="t1" type="DURATION" value="P60D" mod="EQUAL_OR_LESS">
no more than 60 days
</TIMEX3>

(5) the dawn of 2000

<TIMEX3 tid="t2" type="DATE" value="2000" mod="START">
the dawn of 2000
</TIMEX3>
```

`anchorTimeID` is used to point to another TIMEX3 in the case of expressions such as “last week”, which have a functional interpretation. The value of `anchorTimeID` provides the reference point to which the functional interpretation applies.

`quant` and `freq` are used to specify sets that that denote quantified times in a TIMEX3. `quant` is generally a literal from the text that quantifies over the expression. `freq` contains an integer value and a time granularity to represent any frequency contained in the set, just as a period of time is represented in a duration (without the leading letter ‘P’). Some examples:

```
(6) twice a month

<TIMEX3 tid="t3" type="SET" value="P1M" freq="2X">
twice a month
</TIMEX3>

(7) three days every month

<TIMEX3 tid="t4" type="SET" value="P1M" quant="EVERY" freq="3D">
three days every month
</TIMEX3>

(8) daily

<TIMEX3 tid="t5" type="SET" value="P1D quant="EVERY">
daily
</TIMEX3>
```

`beginPoint` and `endPoint` are used to anchor durations to other time expressions in the document. If there is no explicit `tid` to assign to one of these values, then a non-consuming TIMEX3 is created to represent the unspecified point. Conversely, if both the beginning and end points of a duration are explicitly stated in the document, a non-consuming TIMEX3 is created to represent the unspecified duration. Some examples:

```
(9) two weeks from June 7, 2003

<TIMEX3 tid="t6" type="DURATION" value="P2W" beginPoint="t61"
endPoint="t62">
two weeks
</TIMEX3>
<SIGNAL sid="s1">
from
</SIGNAL>
<TIMEX3 tid="t61" type="DATE" value="2003-06-07">
June 7, 2003
</TIMEX3>
<TIMEX3 tid="t62" type="DATE" value="2003-06-21"
temporalFunction="true" anchorTimeID="t6"/>

(10) 1992 through 1995

<TIMEX3 tid="t71" type="DATE" value="1992">
1992
</TIMEX3
<SIGNAL sid="s1">
through
</SIGNAL>
<TIMEX3 tid="t72" type="DATE" value="1995">
1995
</TIMEX3>
```

```
<TIMEX3 tid="t7" type="DURATION" value="P4Y" beginPoint="t71"
endPoint="t72" temporalFunction="true"/>
```

2.4.2.3 <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:

1) 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).
2) indicators of temporal quantification such as "twice", "three times", etc.

2.4.2.4 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.

<TLINK>

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

```
attributes ::= [lid] [origin] (eventInstanceID | timeID) [signalID]
(relatedToEventInstance | relatedToTime) relType

lid ::= ID
```

```
{lid ::= LinkID
LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
timeID ::= IDREF
{timeID ::= TimeID}
signalID ::= IDREF
{signalID ::= SignalID}
relatedToEventInstance ::= IDREF
{relatedToEventInstance ::= EventInstanceID}
relatedToTime ::= IDREF
{relatedToTime ::= TimeID}
relType ::= 'BEFORE' | 'AFTER' | 'INCLUDES' | 'IS_INCLUDED' | DURING
                'SIMULTANEOUS' | 'IAFTER' | 'IBEFORE' | 'IDENTITY' |
                'BEGINS' | 'ENDS' | 'BEGUN_BY' | 'ENDED_BY'
```

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

Examples:

```
(11) John taught 20 minutes every Monday.

John
<EVENT eid="e1" class="OCCURRENCE">
taught
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
negation="false"/>
<TIMEX3 tid="t1" type="DURATION" value="P20TM">
20 minutes
</TIMEX3>
<TIMEX3 tid="t2" type="SET" value="xxxx-wxx-1" quant="EVERY">
every Monday
</TIMEX3>
<TLINK timeID="t1" relatedToTime="t2" relType="IS_INCLUDED"/>
<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="DURING"/>

(12) John taught twice on Monday but only once on Tuesday.

John
<EVENT eid="e1" class="OCCURRENCE">
taught
</EVENT>
<SIGNAL sid="s1">
twice
</SIGNAL>
<SIGNAL sid="s2".
on
</SIGNAL>
<TIMEX3 tid="t1" type="DATE" value="xxxx-wxx-1">
```

```
Monday
</TIMEX3>
but only
<SIGNAL sid=”s3”>
once
</SIGNAL>
<SIGNAL sid=”s4”>
on
</SIGNAL>
<TIMEX3 tid=”t2” type=”DATE” value=”xxxx-wxx-2”>
Tuesday
</TIMEX3>
<MAKEINSTANCE eiid=”ei1” eventID=”e1” tense=”PAST” aspect=”NONE”
negation=”false” signalID=”s1” cardinality=”2”/>
<MAKEINSTANCE eiid=”ei2” eventID=”e1” tense=”PAST” aspect=”NONE”
negation=”false” signalID=”s3” cardinality=”1”/>
<TLINK eventInstanceID=”ei1” signalID=”s2” relatedToTime=”t1”
relType=”IS_INCLUDED”/>
<TLINK eventInstanceID=”ei2” signalID=”s4” relatedToTime=”t2”
relType=”IS_INCLUDED”/>
```

(13) John taught 5 minutes after the explosion.

```
John
<EVENT eid=”e1” class=”OCCURRENCE”>
taught
</EVENT>
<MAKEINSTANCE eiid=”ei1” eventID=”e1” tense=”PAST” aspect=”NONE”
negation=”false”/>
<TIMEX3 tid=”t1” type=”DURATION” value=”PT5M” beginPoint=”t2”
endPoint=”t3”>
5 minutes
</TIMEX3>
<SIGNAL sid=”s1”>
after
</SIGNAL>
the
<EVENT eid=”e2” class=”OCCURRENCE”>
explosion
</EVENT>
<MAKEINSTANCE eiid=”ei2” eventID=”e2” tense=”NONE” aspect=”NONE”
negation=”false”/>
<TIMEX3 tid=”t2” type=”TIME” value=”xxxx-xx-xx” temporalFunction=”true”
anchorTimeID=”t1”/>
<TIMEX3 tid=”t3” type=”TIME” value=”xxxx-xx-xx” temporalFunction=”true”
anchorTimeID=”t1”/>
<TLINK eventInstanceID=”ei2” signalID=”s1” relatedToTime=”t1”
relType=”BEGINS”/>
<TLINK eventInstanceID=”ei2” relatedToTime=”t2” relType=”IS_INCLUDED”/>
<TLINK eventInstanceID=”ei1” relatedToTime=”t3” relType=”IS_INCLUDED”/>
```

Treatment of Temporal Functions:

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

```
John
<EVENT eid=”e1” class=”OCCURRENCE”>
taught
</EVENT>
<MAKEINSTANCE eiid=”ei1” eventID=”e1” tense=”PAST” aspect=”NONE”
negation=”false”/>
<SIGNAL sid=”s1”>
from
</SIGNAL>
<TIMEX3 tid=”t1” type=”DATE” value=”xxxx-09”>
 September
</TIMEX3>
<SIGNAL sid=”s2”>
to
</SIGNAL>
<TIMEX3 tid=”t2” type=”DATE” value=”xxxx-12”>
December
</TIMEX3>
<TIMEX3 tid=”t5” type=”DURATION” value=”P4M” beginPoint=”t1”
endPoint=”t2” temporalFunction=”true”/>
<TIMEX3 tid=”t3” type=DATE” value=”1995” temporalFunction=”true”
anchorTimeID=”t4”>
last year
</TIMEX3>
<TIMEX3 tid="t4" type="DATE" value="1996-03-27"
functionInDocument="CREATION_TIME">
03-27-96
</TIMEX3>
<TLINK timeID=”t1” signalID=”s1” relatedToTime=”t5” relType=”BEGINS”/>
<TLINK timeID=”t2” signalID=”s2” relatedToTime=”t5” relType=”ENDS”/>
<TLINK eventInstanceID=”ei1” relatedToTime=”t5” relType=”HOLDS”/>
```

```
(15) John taught last week.
```

```
John
<EVENT eid="e1" class="OCCURRENCE">
taught
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense=”PAST” aspect=”NONE”
negation=”false”/>
<TIMEX3 tid="t1" type="DATE" value="XXXX-WXX" temporalFunction="true"
anchorTimeID="t2">
last week
</TIMEX3>
<TIMEX3 tid="t2" type="DATE" value="1996-03-27"
functionInDocument="CREATION_TIME">
03-27-96
</TIMEX3>
<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
```

Note: The TLINK relates Timex3 expressions. This is the only representation that will adequately express the temporal anchoring of this event.

```
(16) John taught last week on Monday.
```

```
John
```

```
<EVENT eid="e1" class="OCCURRENCE">
taught
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
negation="fasle"/>
<TIMEX3 tid="t1" type="DATE" value="XXXX-WXX" temporalFunction="true"
anchorTimeID="t2">
last week
</TIMEX3>
<SIGNAL sid="s1">
on
</SIGNAL>
<TIMEX3 tid="t3" type="DATE" value="XXXX-WXX-1" temporalFunction="true"
>
Monday
</TIMEX3>
<TIMEX3 tid="t2" type="DATE" value="1996-03-27"
functionInDocument="CREATION_TIME">
03-27-96
</TIMEX3>
<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"/>
<TLINK timeID="t3" signalID="s1" relatedToTime="t2"
relType="IS_INCLUDED"/>
```

<SLINK>

This is a subordination link that is used for contexts involving negation, modality, evidentials, and factives. An SLINK is used in cases where an event instance subordinates an event type. These are cases where a verb takes a complement and subordinates the event instance referred to in this complement.

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).

```
attributes ::= [lid] [origin] [eventInstanceID] [signalID]
subordinatedEventInstance relType

lid ::= ID
{lid ::= LinkID
LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= IDREF
{eventInstanceID ::= EventInstanceID}
subordinatedEventInstance ::= IDREF
{subordinatedEventInstance ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relType ::= 'MODAL' | '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.

The following EVENT classes interact with SLINK:

```
REPORTING
I_STATE
I_ACTION
```

Some lexical notes:

Verbs that introduce `I_STATE` EVENTs that induce SLINK:

want, desire, crave, lust
believe, doubt, suspect
hope, aspire
intend
fear, hate
love
enjoy
like
know

Verbs that introduce `I_ACTION` EVENTs that induce SLINK:

attempt, try
persuade
promise
name
swear, vow

Examples:

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

<SIGNAL sid="s1">
if
</SIGNAL>
Graham
<EVENT eid="e1" class="OCCURRENCE">
leaves
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PRESENT" aspect="NONE"
negation=”false”/>
<TIMEX3 tid="t1" type="DATE" value="XXXX-XX-XX" temporalFunction="true"
>
today
</TIMEX3>
he will not
<EVENT eid="e2" class="OCCURRENCE">
hear
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense="FUTURE" aspect="NONE"
negation=”true” modality=”WILL”/>
Sabine.
<SLINK subordinatedEventInstance="ei1" signalID="s1" relType="MODAL"/>
```

```
<TLINK eventInstanceID="ei1" relatedToEventInstance="ei2"
relType="BEFORE"/>
<SLINK eventInstanceID=”ei1” subordinatedEventInstance=”ei2”
reltype=”MODAL”/>
```

(18) Bill denied that John taught on Monday.

```
Bill
<EVENT eid="e1" class="I_ACTION">
denied
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense=”PAST” aspect=”NONE”
negation=”false”/>
that John
<EVENT eid="e2" class="OCCURRENCE">
taught
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense=”PAST” aspect=”NONE”
negation=”false”/>
<SIGNAL sid="s1">
on
</SIGNAL>
<TIMEX3 tid="t1" type="DATE" value="XXXX-WXX-1">
Monday
</TIMEX3>
<TLINK eventInstanceID="ei2" signalID=”s1” relatedToTime="t1"
relType="IS_INCLUDED"/>
<SLINK eventInstanceID="ei1" subordinatedEventInstance="ei2"
relType="NEG_EVIDENTIAL"/>
```

(19) Bill wants to teach on Monday.

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

```
(20) Bill attempted to save her.

Bill
<EVENT eid="e1" class="I_ACTION">
attempted
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
negation="false"/>
<SIGNAL sid="s1">
to
</SIGNAL>
<EVENT eid="e2" class="OCCURRENCE">
save
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense="NONE" aspect="NONE"
negation="false"/>
her
<SLINK eventInstanceID="ei1" signalID="s1"
subordinatedEventInstance="ei2" relType="MODAL"/>
```

<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 ::= [lid] [origin] eventInstanceID [signalID]
relatedToEventInstance relType

lid ::= ID
{lid ::= LinkID
LinkID ::= l<integer>}
origin ::= CDATA
eventInstanceID ::= ID
{eventInstanceID ::= EventInstanceID}
signalID ::= IDREF
{signalID ::= SignalID}
relatedToEventInstance ::= IDREF
{relatedToEventInstance ::= EventInstanceID}
relType ::= 'INITIATES' | 'CULMINATES' | 'TERMINATES' | 'CONTINUES' |
'REINITIATES'
```

Some examples:

```
(21) The boat began to sink.

The boat
<EVENT eid="e1" class="ASPECTUAL">
began
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
negation="false"/ >
<SIGNAL sid="s1">
```

```
to
</SIGNAL>
<EVENT eid="e2" class="OCCURRENCE" >
sink
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense="NONE" aspect= "NONE"
negation="false"/>
<ALINK eventInstanceID="ei1" signalID="s1" relatedToEventInstance="ei2"
relType="INITIATES"/>

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

The search party
<EVENT eid="e1" class="ASPECTUAL">
stopped
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
negation="false"/>
<EVENT eid="e2" class="OCCURRENCE">
looking
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense="NONE" aspect="PROGRESSIVE"
negation="false"/>
<ALINK eventInstanceID="ei1" relatedToEventInstance="ei2"
relType="TERMINATES"/>
for the survivors
```

2.4.2.5 Other Tags

<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:

```
(23) The TWA flight
<EVENT eid="e1" class="OCCURRENCE">
crashlanded
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE"
negation=”false”/>
on Easter Island
<TIMEX3 tid="t1" type="DURATION" value="P2W" beginPoint=”t2”
endPoint=”t3”>
two weeks ago
</TIMEX3>
<TIMEX3 tid=”t3” type=”DATE” value=”1999-12-06” temporalFunction=”true”
anchorTimeID=”t1”/>


...

<TIMEX3 tid="t2" type="DATE" functionInDocument="CREATION_TIME"
value="1999-12-20">
12-20-1999
</TIMEX3>
<TLINK eventInstanceID=”ei1” relatedToTime=”t3” relType=”IS_INCLUDED”/>
```

If we wanted to indicate that we were unsure that we had annotated “two weeks ago” correctly, we could add this annotation:

```
(23') <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:

```
(23'') <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.

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.

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

1) The choice of confidence values should be as clearly defined as possible to cover the options; this relates to the granularity and orders of magnitude as presented by Jerry Hobbs as well. This would suggest a selection from a small set (e.g. low, mid, high; not_sure, sure, abolutely_sure). These could be interpreted or rescaled to a (0,1] range, if need be, for subsequent inference.
2) There should be a default value specified (at high (=1)) so that it is not necessary to annotate all links and attributes for them with a confidence.

The constraint on human annotators to a subset of the possible values should be documented in the annotation guidelines and implemented in the annotation tool. And it would probably be best if the annotation tool did not present numbers but rather natural language descriptions such as those suggested above, which would be represented in the underlying annotation numerically. For example, the annotator might pick “moderately certain”, which would enter the annotation as .5.

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

<TimeML>

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

```
<?xml version="1.0"?>
<!DOCTYPE TimeML SYSTEM "TimeML.dtd">
<TimeML>
FAMILIES SUE OVER AREOFLOT CRASH DEATHS

   The Russian airline Aeroflot has been
<EVENT eid="e1" class="OCCURRENCE">
hit
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PRESENT"
aspect="PERFECTIVE" negation="false"/>
with a writ for loss and damages,
<EVENT eid="e2" class="OCCURRENCE">
filed
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense="PAST" aspect="NONE"
negation="false"/>
in Hong Kong by the families of seven passengers
<EVENT eid="e3" class="OCCURRENCE">
killed
</EVENT>
<MAKEINSTANCE eiid="ei3" eventID="e3" tense="PAST" aspect="NONE"
negation="false"/>
<SIGNAL sid="s1">
in
</SIGNAL>
an air
<EVENT eid="e4" class="OCCURRENCE">
crash
</EVENT>
<MAKEINSTANCE eiid="ei4" eventID="e4" tense="NONE" aspect="NONE"
negation="false"/>.


   All 75 people
<EVENT eid="e7" class="STATE">
on board
</EVENT>
<MAKEINSTANCE eiid="ei7" eventID="e7" tense="NONE" aspect="NONE"
negation="false"/>
<TLINK eventInstanceID="ei7" relatedToEvent="ei5" relType="INCLUDES"/>
the Aeroflot Airbus
<EVENT eid="e5" class="OCCURRENCE" >
died
</EVENT>
<MAKEINSTANCE eiid="ei5" eventID="e5" tense="PAST" aspect="NONE"
negation="false"/>
<TLINK eventInstanceID="ei5" signalID="s2" relatedToEvent="ei6"
relType="IAFTER"/>
<SIGNAL sid="s2">
when
```

```
</SIGNAL>
it
<EVENT eid="e6" class="OCCURRENCE">
ploughed
</EVENT>
<MAKEINSTANCE eiid="ei6" eventID="e6" tense="PAST" aspect="NONE"
negation=”false”/>
<TLINK eventInstanceID="ei6" signalID="s3" relatedToTime="t2"
relType="IS_INCLUDED"/>
<TLINK eventInstanceID="ei6" relatedToEvent="ei4" relType="IDENTITY"/>
into a Siberian mountain
<SIGNAL sid="s3">
in
</SIGNAL>
<TIMEX3 tid="t2" type="DATE" value="1994-04">
March 1994
</TIMEX3>.

...

<TIMEX3 tid="t1" type="DATE" value="1996-03-27">
03-27-96
</TIMEX3>

<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="BEFORE"/>
<TLINK eventInstanceID="ei2" relatedToEvent="ei1" relType="BEFORE"/>
<TLINK eventInstanceID="ei3" relatedToEvent="ei2" relType="BEFORE"/>
<TLINK eventInstanceID="ei3" signalID="s1" relatedToEvent="ei4"
relType="IS_INCLUDED"/>

</TimeML>
```

2.5 Update to TimeBank Corpus

The update of the TimeBank corpus consisted of normalizing the existing corpus as well as making the appropriate changes to reflect the new specification of TimeML. The normalization process was done both manually and by scripts. Once TimeBank was effectively “clean”, its contents were modified according to the TimeML 1.1 schema.

The update to TimeBank is an ongoing process. During this workshop, the following changes were made:

1) Attribute names were changed where appropriate.
2) All modal and negative SLINKs that consisted only of a subordinatedEventInstanceID and a signalID were converted to attributes of the MAKEINSTANCE tag.

3) TLINKs containing the 'magnitude' attribute were converted into new TLINKs with relType 'BEGINS' and 'ENDS', respectively. Where possible, beginPoint and endPoint times for the associated duration TIMEX3 were inferred.

At this point, TimeBank is not completely converted into TimeML 1.1. The following changes are yet to be made:

1) Missing beginPoint or endPoint times in a DURATION have not been calculated. Such values can be inferred based on the ISO value of the duration and the supplied begin or end point.
2) Implied durations such as "from 1992 to 1995" have not been extracted.
3) The value attribute in the TIMEX3 tag is not well regulated. Currently, the type this attribute expects is a string, but this may allow faulty values to be assigned to a TIMEX3.

2.6 Conclusion

This document summarizes the research and findings of the TANGO workshop, held from March through August, 2003. The most significant milestones and achievements from the workshop are listed as follows:

1. **Tango Toolkit**: Development of software for graphical annotation using TimeML, including a built-in closure feature
2. **TimeML 1.1**: Enhancement of TimeML 1.0, a markup language for time and events
3. **TimeBank 1.1**: Normalized TimeBank corpus with some TimeML 1.1 modifications

Chapter 3

Catalog of Software, Data, Reports, and Presentations

The following items can be found at www.timeml.org/tango/deliverables.

3.1 Software and Data

1. Tango Toolkit
2. TimeBank 1.1 Corpus

3.2 Documentation and Presentations

1. TimeML 1.1 Specification
2. Workshop Documentation
3. Workshop Presentations