State of the art and design of novel annotation languages and technologies
## Document Information

<table>
<thead>
<tr>
<th>ID</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D3.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification</th>
<th>Distribution level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0</td>
<td>Public</td>
</tr>
<tr>
<td>Date of reference M12</td>
<td></td>
</tr>
<tr>
<td>Published date 30 September 2011</td>
<td></td>
</tr>
</tbody>
</table>

Distribution level:
- **Internal**: Circulation is restricted to the partners of the TERENCE Consortium
- **Restricted**: Circulation is restricted to the partners of the TERENCE Consortium and the EU Commission for project evaluation purposes
- **Public**: This document may be distributed to the general public for dissemination and information purposes

<table>
<thead>
<tr>
<th>Editor</th>
<th>Surname and Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Steven Bethard</td>
<td>KUL</td>
</tr>
<tr>
<td></td>
<td>Sara Tonelli, Emanuele Pianta</td>
<td>FBK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reviewers</th>
<th>Surname and Name</th>
<th>Company</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Oleksandr Kolomiyets</td>
<td>KUL</td>
</tr>
<tr>
<td></td>
<td>Rosella Gennari</td>
<td>FUB</td>
</tr>
<tr>
<td></td>
<td>Pierpaolo Vittorini</td>
<td>UnivAQ</td>
</tr>
</tbody>
</table>
Abstract
The TERENCE Adaptive Learning System will offer smart games that help improve reading comprehension for low-literacy and deaf children. Underlying these games will be the rich semantic structure of children’s stories, represented in a machine-accessible format. This format, the TERENCE story annotation scheme, will allow the natural language of children’s stories to be annotated for their semantic content and structure. This document describes the foundations and specifications of the TERENCE story annotation scheme.

Copyright notices
TERENCE Consortium. All rights reserved. This document is a project document of the TERENCE project. All contents are reserved by default and may not be disclosed to third parties without the written consent of the TERENCE partners, except as mandated by the European Commission contract ICT-2010-257410 for reviewing and dissemination purposes.
All trademarks and other rights on third party products mentioned in this document are acknowledged as owned by the respective holders.
Contents

1 Executive summary 9

2 Introduction 10

3 Basic Story Elements 13

3.1 Events 13

3.1.1 State-of-the-Art 13
3.1.2 TERENCE annotation guidelines 14
3.1.3 TERENCE annotation format 15

3.2 Entities (including People and Locations) 16

3.2.1 State-of-the-Art 16
3.2.2 TERENCE annotation guidelines 16
3.2.3 TERENCE annotation format 17

3.3 Times 17

3.3.1 State-of-the-Art 17
3.3.2 TERENCE annotation guidelines 17
3.3.3 TERENCE annotation format 18

4 Coreference between Story Elements 19

4.1 Event coreference 19

4.1.1 State-of-the-Art 19
4.1.2 TERENCE annotation guidelines 19
4.1.3 TERENCE annotation format 20

4.2 Entity coreference 20

4.2.1 State-of-the-Art 20
4.2.2 TERENCE annotation guidelines 21
4.2.3 TERENCE annotation format 21

5 Links between Story Elements 22

5.1 Participant roles 22

5.1.1 State-of-the-Art 22
5.1.2 TERENCE annotation guidelines 24
5.1.3 TERENCE annotation format 24

5.2 Temporal relations 25

5.2.1 State-of-the-Art 25
5.2.2 TERENCE annotation guidelines 26
5.2.3 TERENCE annotation format 27
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Graph representation of the semantics of The bear rose on its hind legs with an angry growl and started after Tuk. The boy turned and ran. He was just a few lengths ahead of the bear when he saw his father coming toward him on the dogsled.</td>
<td>11</td>
</tr>
<tr>
<td>6.1</td>
<td>Screenshot of Callisto interface</td>
<td>32</td>
</tr>
<tr>
<td>6.2</td>
<td>Screenshot of Tango interface</td>
<td>33</td>
</tr>
<tr>
<td>6.3</td>
<td>Screenshot of BAT interface</td>
<td>34</td>
</tr>
<tr>
<td>6.4</td>
<td>Screenshot of CAT interface</td>
<td>35</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Annotator agreement (Krippendorff’s Alpha) for different event annotation schemes. The first three are evaluated on the same set of stories (1-20), the last is on a new set of stories (21-100). ..................................................... 15

5.1 Annotator agreement (Krippendorff’s Alpha) for different temporal link annotation schemes. ................................................................. 27
1 Executive summary

This document is the first deliverable produced by the third Work Package of the TERENCE project. It describes the specifications of the annotation language that will be used to annotate TERENCE stories, and the state of the art research on which this language is based. Stories annotated in this language will provide the semantic content and structure necessary for the smart games in the TERENCE Adaptive Learning System.

More specifically, this deliverable describes the annotation scheme for (1) identifying expressions that refer to basic story elements like people or events, (2) identifying when two or more expressions refer to the same person, event, etc. in the real world, and (3) linking basic story elements together, such as people with the events they participate in, or pairs of events in sequence along the story’s timeline.

This deliverable focuses on the formal description of the annotation language, the format in which the annotation language will be represented, and the tool which will allow manual annotation of stories in the annotation language. The annotation scheme may be further refined in light of the experience gained from the actual manual annotation of texts, from the evaluations of WP7 and the games developed in WP4. Future deliverables will include the software modules for automatically annotating stories with the annotation scheme (D3.2 in month 23) and the repository of annotated stories (D3.3 in month 23).
2 Introduction

Stories for children are often written in simpler language than that of stories directed at an adult audience. For example, they might use more common vocabulary items, shorter sentences, lower syntactic complexity, etc. Nonetheless, even these “simple” stories are still quite complex for software applications that aim to understand them. For example, consider the following snippet from a children’s story:

*The bear rose on its hind legs with an angry growl and started after Tuk. The boy turned and ran. He was just a few lengths ahead of the bear when he saw his father coming toward him on the dogsled.*

Formally representing the semantics of this story involves:

- identifying key characters like *Tuk, Tuk’s father* and *the bear*
- identifying key events like *rose, growl, turned* and *ran*
- understanding the order of the events in the story, like the *growl* being included in when the bear *rose* or that the bear *started after Tuk* before the boy *ran*
- recognizing the role that characters play in events, like *Tuk* being the goal of the *bear’s starting after*
- understanding when different words refer to the same person (a.k.a. coreference), like that *Tuk, The boy* and *He* all mean the same thing

Figure 2.1 shows an example graphical representation of all the story elements that must be identified and the links between them.

The goal of this document is to formally specify how such story graphs will be represented in the TERENCE Adaptive Learning System. As pointed out in D1.1, the general conceptual model proposed in past works (e.g. [8]) for classifying interventions for reading comprehension includes different tasks that a reader should perform to understand a text, i.e. literal comprehension, reorganisation, inference making, prediction, evaluation by giving a global judgement about the text and personal response appealing the learner’s emotional sphere. All these aspects must be taken into account when defining the annotation language and the technologies for the TERENCE system. Questions related to the story timeline, for example, may address both the literal comprehension and the reorganisation task. As for inference making, this can be addressed by questions and activities for children that involve both the temporal and the causal aspect. For this reason, we will explicitly add causal and intentional information to standard TimeML annotation. Besides, text comprehension both at literal and at deeper level requires children to follow and understand what the characters do and how they interact with each other in the stories. Therefore, also the participants in the events will be annotated with specific labels that convey information about their semantic type and coreferential chains.
The bear rose on its hind legs with an angry growl and started after Tuk. The boy turned and ran. He was just a few lengths ahead of the bear when he saw his father coming toward him on the dogsled.

Figure 2.1: Graph representation of the semantics of The bear rose on its hind legs with an angry growl and started after Tuk. The boy turned and ran. He was just a few lengths ahead of the bear when he saw his father coming toward him on the dogsled.
The document is structured as follows: Chapter 3 lays out a formal description of the syntax and semantics of words and phrases that can be tagged as basic story elements (events, characters, etc.). Chapter 4 describes how to annotate coreference between basic story elements, that is, how to indicate when two words or phrases refer to the same thing in the real world. Chapter 5 specifies when and how to annotate links between story elements, e.g., when characters participate in events, or when one event follows the other along the story’s timeline. Finally, Chapter 6 describes the tool that will be used to take stories and annotate them manually with the schemes described in the preceding chapters.
3 Basic Story Elements

The basic elements of a story are the what s, when s, where s and who s that make up the story’s plot. The words that identify what happened are called events, the words that indicate when something happened are called times, the words that indicate where something happened are called location entities, and the words that indicate to whom something happened are called person entities. The following sections describe how each of these categories of story elements will be annotated in the TERENCE project.

3.1 Events

3.1.1 State-of-the-Art

TimeML [21, 23, ?] is an international standard for annotating the event and temporal structure of a text. It provides a standard definition of “event” that has been used in the annotation the TimeBank corpus [22], the Ita-TimeBank corpus [5] and for the annotation of news events in the 2010 TempEval competition [31]. TimeML defines events in the following way:

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).

1. Ferdinand Magellan, a Portuguese explorer, first [event reached] the islands in search of spices.
3. 11,024 people, including local Aeta aborigines, were [event evacuated] to 18 disaster relief centers.
4. “We’re [event expecting] a major eruption,” he said in a telephone interview early today.
5. Israel has been scrambling to buy more masks abroad, after a [event shortage] of several hundred thousand gas masks.

TimeML events may expressed by tensed (erupted) and untensed (expecting) verbs, nominalizations (invasion), predicative clauses (is the president), adjectives (dormant) or prepositional phrases (on board). The English annotation guidelines follow the general principle that only one token should be included in the extent of any event annotation. For instance given an event description such as may not give up, only the head token [event give] will be
annotated, whereas the information contributed by the modal, the polarity item and the particle can be represented as attributes of the event. However when the main event is introduced by a light verb or by and aspectual verb, these are annotated as independent events: e.g. \[\text{[event } \text{demonstrations} \text{]} \text{ have } \text{[event } \text{taken} \text{]} \text{ place, or the private sector } \text{[event } \text{began} \text{]} \text{ [event } \text{establishing} \text{].}

3.1.2 TERENCE annotation guidelines

In TERENCE, we largely follow the TimeML event annotation guidelines. It is well known however that event annotation is a difficult task, and that it can be difficult to reach a high level of inter-coder agreement between annotators. For this reason, we considered some adaptations of the original guidelines in order to make the resulting annotations more reliable in the context of children’s stories.

First, whereas TimeML recommends tagging events within direct speech, we made the hypothesis that in children’s stories such events are less important and are not directly part of the plot.

For example:

- The Fuller \[\text{[event } \text{thanked} \text{]} \text{ him, but } \text{[event } \text{replied} \text{], “I couldn’t think of it, sir: why, everything I take such pains to whiten would be blackened in no time by your charcoal.”} \]
- \[\text{[event } \text{Coming} \text{] and [event } \text{standing} \text{] under the tree he [event } \text{looked} \text{] up and [event } \text{said}, “What a noble bird I see above me!”} \]
- “That’s awkward,” \[\text{[event } \text{said} \text{] the Cat to herself: “the only thing to do is to coax them out by a trick.”} \]

Second, whereas TimeML recommends tagging negated, modal and hypothetical events, we observed that such events can be quite difficult to place along a story timeline. For example:

- Imagining the bird must be made of gold inside, they \[\text{[event } \text{decided} \text{] to kill it in order to secure the whole store of precious metal at once.} \]
- An Old Woman \[\text{[event } \text{made} \text{] an [event } \text{agreement} \text{] with him in the [event } \text{presence} \text{] of witnesses that she should pay him a high fee if he cured her, while if he failed he was to receive nothing.} \]

It would be quite difficult, for example, to describe the temporal order between pay and receive in the second sentence as neither of these events actually occurred.

Third, whereas TimeML recommends tagging light verbs and aspectual verbs as individual events, we considered that these verbs play an ancillary role with regard to the verbs expressing full events. Consider:

- There was once a Dog who used to \[\text{[event } \text{snap} \text{] at people.} \]
- He managed to \[\text{[event } \text{scramble} \text{] on to dry ground.} \]
- He did his best to reach them by \[\text{[event } \text{jumping} \text{].} \]
- So he \[\text{[event } \text{went} \text{] and began to [event } \text{fell} \text{] a tree.} \]
State of the art and design of novel annotation languages and technologies

Table 3.1: Annotator agreement (Krippendorff’s Alpha) for different event annotation schemes. The first three are evaluated on the same set of stories (1-20), the last is on a new set of stories (21-100).

<table>
<thead>
<tr>
<th>Annotation scheme</th>
<th>Stories</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeML guidelines</td>
<td>1-20</td>
<td>0.729</td>
</tr>
<tr>
<td>TimeML, but no speech or modal events</td>
<td>1-20</td>
<td>0.833</td>
</tr>
<tr>
<td>TERENCE guidelines</td>
<td>1-20</td>
<td>0.876</td>
</tr>
<tr>
<td>TERENCE guidelines</td>
<td>21-100</td>
<td>0.847</td>
</tr>
</tbody>
</table>

- She let herself [event hang] down by her hind legs…

In all of these cases the relevant content of the event can be captured by ignoring the ancillary verbs that introduce the main event. So we made the hypothesis that only main events need to be annotated for our purposes.

On the basis of the above observations, we decided to test whether the above mentioned modifications can improve inter-coder agreement on event annotations. The modified guidelines were evaluated based on a pilot annotation study. Two annotators were given 20 stories to annotate, using three different annotation schemes. The first annotation scheme was the original TimeML event annotation guidelines. The second annotation scheme asked annotators to skip events within direct speech and events that were negated, modal or hypothetical, but to otherwise follow the TimeML guidelines. The third annotation scheme (TERENCE guidelines) additionally asked to ignore light and aspectual verbs. After each pass through the stories using one of the annotation schemes, we measured annotator agreement using Krippendorff’s Alpha, a statistical measure of annotator agreement that adjusts for agreement due to chance. The first three rows of Table 3.1 show that the best annotator agreement was achieved using the TERENCE event annotation guidelines.

The last row of Table 3.1 shows the agreement when the annotators were given set of 80 entirely new stories. A high level of agreement was maintained (0.847) using the TERENCE guidelines.

The result of this experiment shows that some modifications to the original TimeML guidelines can improve the reliability of event annotations. Some further investigation needs however to be carried out, in order to verify whether the information that is ignored by adopting the modified TimeML annotation guidelines is indeed irrelevant for the purposes of the Terence games (further input from other WPs is needed on this).

3.1.3 TERENCE annotation format

The Terence annotation format will follow a standoff scheme, based on a number of annotation layers. However, all layers will be included in the same file, so that only intra-file pointers are needed. The most basic annotation layer is the token layer. The event layer is built on top of the event layer. Event annotations are linked by temporal- and causation-relations, belonging to further layers (see below).
3.2 Entities (including People and Locations)

3.2.1 State-of-the-Art

Annotation of people, locations and other named entities for natural language processing is typically traced back to the Message Understanding Conferences (MUC), where the named entity task included identifying proper nouns in a text that referred to organizations, persons and locations [6]. For example:

- [entity Taga Co.]
- the [entity Clinton] government
- [entity Northern California]
- [entity Mips] Vice President [entity John Hime]

The Automatic Content Extraction evaluation (ACE) [7] expanded the MUC set of named entities to be tagged to include facilities (e.g. LaGuardia airport) and geo-political entities (e.g. France), though names of animals and non-human characters (e.g. Snuggle, the fabric softener bear) were explicitly excluded from annotation.

Probably the most recent large scale annotation of entities was performed under the OntoNotes project [82]. OntoNotes expands the ACE set of entities to be tagged to include:

- PERSON: People, including fictional
- NORP: Nationalities or religious or political groups
- FACILITY: Buildings, airports, highways, bridges, etc.
- ORGANIZATION: Companies, agencies, institutions, etc.
- GEO-POLITICAL ENTITY (GPE): Countries, cities, states
- LOCATION: Non-GPE locations, mountain ranges, bodies of water
- PRODUCT: Vehicles, weapons, foods, etc. (Not services)
- EVENT: Named hurricanes, battles, wars, sports events, etc.
- WORK OF ART: Titles of books, songs, etc.
- LAW: Named documents made into laws
- LANGUAGE: Any named language

3.2.2 TERENCE annotation guidelines

Almost all of the OntoNotes entity types can appear in children’s stories in one form or another, though PERSON and LOCATION names will likely be much more common that the other types of names. The one gap in the OntoNotes entity types (and what was an explicit, intentional gap in the ACE entity types) is a type for non-human characters, which are common in children’s stories. For example:

• A dispute arose between the [entity North Wind] and the [entity Sun], each claiming that he was stronger than the other.

To handle these cases we introduce a new entity type called PSEUDO-PERSON, meant to handle all those fictional situations in which a non-human entity behaves like a human. In alternative, we could have extended the semantics of PERSON to include story characters that behave like humans. However note that, although PSEUDO-PERSONs feature human psychological traits, they keep most physical properties of their original type (e.g. a speaking frog will more probably leap than walk, he/it will be green, etc.). So we make the assumption that it is relevant to distinguish between real PERSON and PSEUDO-PERSONs. We also observed that, in children stories, entities such as (non pseudo-) animals, vegetables, and objects may play a crucial role in the development of the story plot. For this reason, we introduce a specific type for each of them: ANIMAL, VEGETABLE, OBJECT. For all other entity types, annotators are instructed to follow the OntoNotes guidelines.

3.2.3 TERENCE annotation format

Entity annotations will be added as a standoff annotation layer on top of the basic token annotation layer. Entity annotations will be linked among them by co-reference relations and will be linked to event annotations by participant relations (see below).

3.3 Times

3.3.1 State-of-the-Art

The early MUC competitions included recognition of time expressions, limited to dates and times such as January 1990, 5 p.m. EST or last night. The TIDES program [12, 11], and later TimeML [21, 23], expanded the possible expressions to be annotated to include:

- durations, such as 3 hours long
- sets, such as some Thursdays
- approximate expressions, such as more than a decade ago
- fuzzy expressions, such as current

Perhaps more importantly, both TIDES and TimeML include the concept of time normalization, where a normalized form is assigned to each expression based on the ISO 8601 standard for representing dates, times, and durations. So for example, the expression June 7, 2003 would get the normalized form 2003-06-07, and the expression 60 days would get the normalized form P60D.

3.3.2 TERENCE annotation guidelines

While complex time expressions are common in news articles, they rarely appear in children’s stories. Simple expressions like that night or the next morning do appear, but expressions
like *June 7, 2003* are quite rare. And because most children's stories are not fixed to a specific date, time normalization is often impossible – where in a news story *today* could be normalized to a specific date like 2011-07-27, in most children's stories, *today* refers only a date experienced by a character in the story. (In TimeML terms, the *document creation time* does not form a good *anchor time* for most time expressions in children's stories.)

Thus, in TERENCE, annotators are instructed to follow the TimeML guidelines for annotating times, but without attempting to normalize the expressions into the ISO 8601 form.

### 3.3.3 TERENCE annotation format

Time annotations are made on top of the basic token layer. They are linked to event annotations by temporal-relations.
4 Coreference between Story Elements

Two words or phrases in a text are said to co-refer if they are both references the same object of discourse. Discourse objects are mapped by the reader into elements of a mental model, that may or may not correspond to actual entities or events of the real world. For example, a character named Mary may be referred to as her later in the same story, or an event like grew may later be referred to by the phrase the growth. Such coreference should be annotated so that useful questions may be generated around the many events a single entity plays a role in, or around the many entities that participate in a single event.

4.1 Event coreference

4.1.1 State-of-the-Art

TimeML establishes guidelines for annotating coreference between events [26]. TimeML calls this coreference relation IDENTITY and defines it only by giving the following single example:

\textit{John \textcolor{purple}{[event drove]} to Boston. During his \textcolor{purple}{[event drive]} he ate a donut.}

To indicate that drove and drive refer to the same event, TimeML establishes a TLINK between the two events, tagged with the IDENTITY relation. In the ISO-TimeML standoff XML format, this looks like:

```xml
<!-- drove -->
<EVENT xml:id="e1" target="#token2" />
<!-- drive -->
<EVENT xml:id="e2" target="#token8" />
<!-- drove == drive -->
<TLINK target="#range(#e1,#e2)" relType="IDENTITY" />
```

However, in recent years, many variants of TimeML have dropped the IDENTITY link for annotation purposes. For example, both TempEval 2007 [30] and TempEval 2010 [31] limit TLINKs to a smaller set of six relations, excluding the IDENTITY relation.

4.1.2 TERENCE annotation guidelines

In TERENCE, we plan to follow the original TimeML guidelines for annotating event coreference via IDENTITY links. In pilot annotation studies, including this relation did not significantly decrease annotator agreement, and provided additional useful information for reasoning and generating questions. (See Section 5.2 for more details about the TLINK annotation study.)
4.1.3 TERENCE annotation format

See below temporal relation annotations in Sect. 5.2.

4.2 Entity coreference

4.2.1 State-of-the-Art

Annotation of entity coreference can be traced back to the Message Understanding Conferences (MUC) \[15\]. In MUC annotation, all nouns, pronouns and noun phrases were considered as candidates for coreference annotation. For each such markable expression, annotators were instructed to identify its headword (MIN attribute) and link it with its antecedent (REF attribute) if one existed. For example:

```
<COREF ID="100" MIN="Henry Higgins">Henry Higgins</COREF>, who was formerly
sales director for
<COREF ID="101" MIN="director" TYPE="IDENT" REF="100">sales director for</COREF>,
became
<COREF ID="103" MIN="president" TYPE="IDENT" REF="100">president of</COREF>
<COREF ID="104" MIN="Dreamy Detergents">Dreamy Detergents</COREF>
```

The pairwise links between markables would form coreference chains, where all expressions linked in the chain referred to the same real-world entity. In the example above, Henry Higgins, sales director for Sudsy Soaps and president of Dreamy Detergents form a coreference chain. MUC introduced the concept of a grounding instance of a coreference chain, defined as the first proper name element of a coreference chain. In the example, Henry Higgins would be the grounding instance of the chain.

OntoNotes annotation of entity coreference considers as markable all noun phrases, nominal (but not adjectival pre-modifiers) and verbs \[32, 28, 20\]. For example:

- \[entity The Rockford, Ill. maker of fasteners\] also said \[entity it\] expects...
- the \[entity U.S.] economy
- Sales of passenger cars \[entity grew\] 22%. \[entity The strong growth\] followed year-to-year increases.

OntoNotes also differentiates between regular IDENTITY coreference links and APPOSITIVE coreference links:

- She had \[entity a good suggestion\] and \[entity it\] was unanimously accepted by all.
- \[entity [head the PhacoFlex intraocular lens], \[attribute the first foldable silicone lens available for cataract surgery]]
In the first example, OntoNotes considers there to be two separate markables, a good suggestion and it, while in the second example, OntoNotes considers there to be only one markable, the whole phrase, with an additionally marked head and attribute from the appositive relation.

### 4.2.2 TERENCE annotation guidelines

In TERENCE, we plan to follow the OntoNotes guidelines for annotating entity coreference, with the exception that verbs should not be considered markables – event coreference will be handled according to TimeML guidelines, as discussed in Section 4.1.

### 4.2.3 TERENCE annotation format

Entity coreference constitutes a distinct standoff annotation layer linking objects of the entity annotation layer.
5 Links between Story Elements

Though it is important to recognize basic story elements like entities and events, true story understanding can only exist once the relations between these elements are also understood. For example, if a story contains the entities Tuk and the bear, and the event chased, it is crucial to represent in the story structure who was doing the chasing and who was chased, that is, whether The bear chased Tuk or Tuk chased the bear. In TERENCE, four types of relations between basic story elements are recognized: participant relations, linking entities to the events they played a role in; temporal relations, linking pairs of events along the story’s timeline; causal relations, linking events that were causes with the events that were their effects; and intentional relations, linking intended events with their goals and purposes.

5.1 Participant roles

5.1.1 State-of-the-Art

Links from entities to the events they participate in have been annotated in a variety of different forms. The Penn TreeBank [17] annotated some basic predicate argument structure, including the subjects, times and locations of verbal events by adding “functional tags” to syntactic constituents (e.g. -SBJ, -TMP, -LOC). For example:

(S (NP-SBJ-1 The ball)  
 (VP was  
  (VP thrown  
   (NP *-1)  
    (PP by  
     (NP-LGS Chris)))))

Predicate argument structure:
throw(Chris, ball)

When a syntactic dependency formalism is applied, rather than a syntactic constituency formalism, such simple predicate argument structure is often visible directly in the dependencies. For example, the Stanford typed dependencies [10] would produce as a syntactic analysis structures such as:
Bills on ports and immigration were submitted by Senator Brownback, Republican of Kansas

The FrameNet project [2] aimed for a more direct representation of the predicate argument semantics (a.k.a. frame semantics), without the use of a syntactic tree. In FrameNet, groups of predicates are associated with a frame, which has several frame elements, or arguments that the predicates are typically associated with. For example, the driving frame has the frame elements DRIVER, VEHICLE, RIDER and CARGO, inherits the frame element PATH from the TRANSPORTATION frame, and would be used to annotate a sentence like:

Now [mover Van Cheele] was [target driving] [ride his guest] [path back to the station]

PropBank [19] picks a middle path between the syntactic treebank formalisms and the FrameNet semantic formalisms. PropBank defines a variety of roles shared across many predicates, ARG0 (roughly equivalent to AGENT) and ARG1 (roughly equivalent to PATIENT), ARGM-LOC (for locations), ARGM-TMP (for times), ARGM-MNR (for manners), etc. PropBank also defines a frame file for each verb that assigns any additional roles for that predicate to the labels ARG2 through ARG5. This is similar in nature to the FrameNet approach, though there is no effort in PropBank to cluster together semantically similar verbs. However, unlike FrameNet, PropBank annotates semantic roles on top of an existing syntactic tree annotation, so that PropBank roles can only be annotated matching constituents in the syntactic tree.

Treebank annotation:
(S (NP-TPC John)
, (NP-SBJ I)
(VP like
 (NP him)
 (NP-ADV a lot)))

Propbank annotation:
Rel: like
Arg0: I
Arg1: him
ArgM-MNR: a lot
ArgM-DIS: John
5.1.2 TERENCE annotation guidelines

For asking questions about children’s stories, the most crucial relations are the **subject** and **object** relations. Thus, as a minimum requirement for TERENCE stories, syntactic dependency relations between story events and story entities will be provided. This is acceptable as automatic dependency parsers exist for both English and Italian. However, FrameNet or PropBank-style semantic role parsers (and the extensive data required for training them) are only currently available in English. Thus in TERENCE, the annotation scheme will require dependency relations, but semantic role relations will be optional.

When annotating stories manually, TERENCE annotators are instructed to follow the Stanford typed dependencies manual [9] guidelines for annotating subjects and objects:

- **NSUBJ**: A nominal subject is a noun phrase which is the syntactic subject of a clause.
  
  *Clinton defeated Dole.*
  
  **NSUBJ**(defeated, Clinton)

- **AGENT**: An agent is the complement of a passive verb which is introduced by the preposition *by* and does the action.
  
  *The man has been killed by the police.*
  
  **AGENT**(killed, police)

- **CSUBJ**: A clausal subject is a clausal syntactic subject of a clause, i.e., the subject is itself a clause.
  
  *What she said makes sense.*
  
  **CSUBJ**(makes, said)

- **CSUBJPASS**: A clausal passive subject is a clausal syntactic subject of a passive clause.
  
  *That she lied was suspected by everyone.*
  
  **CSUBJPASS**(suspected, lied)

- **XSUBJ**: A controlling subject is the relation between the head of an open clausal complement (xcomp) and the external subject of that clause.
  
  *Tom likes to eat.*
  
  **XSUBJ**(eat, Tom)

- **DOBJ**: The direct object of a VP is the noun phrase which is the (accusative) object of the verb.
  
  *She gave me a raise.*
  
  **DOBJ**(gave, raise)

- **IOBJ**: The indirect object of a VP is the noun phrase which is the (dative) object of the verb.
  
  *She gave me a raise.*
  
  **IOBJ**(gave, me)

5.1.3 TERENCE annotation format

Participant annotations constitute a specific standoff annotation layer linking entity to event annotations.
5.2 Temporal relations

5.2.1 State-of-the-Art

Temporal relations between events have primarily been annotated in the form of TimeML [21, 26, 23]. In TimeML, temporal relations are annotated via TLINKs. Each event (or time) is assigned a unique identifier, and these identifiers are used by TLINK annotations to assign one of the temporal relations BEFORE, AFTER, INCLUDES, IS_INCLUDED, DURING, DURING_INV, SIMULTANEOUS, IAFTER, IBefore, BEGINS, BEGUN_BY, ENDS or ENDED_BY. For example:

A major earthquake struck Los Angeles.

<!-- earthquake -->
<EVENT xml:id="e1" target="#token3" />
<!-- struck -->
<EVENT xml:id="e2" target="#token4" />
<!-- earthquake immediately before struck -->
<TLINK target="#range(#e1,#e2)" relType="IBEFORE" />

Mia visited Seoul to look me up yesterday.

<!-- document creation time -->
<TIMEX3 xml:id="t0" type="DATE" value="2009-10-20"
    functionInDocument="CREATION_TIME"/>
<!-- visited -->
<EVENT xml:id="e1" target="#token2" class="OCCURRENCE" tense="PAST"/>
<!-- look up -->
<EVENT xml:id="e2" target="#range(#token5,#token7)"
    class="OCCURRENCE" tense="NONE" vForm="INFINITIVE"/>
<!-- yesterday -->
<TIMEX3 xml:id="t1" target="#token8" type="DATE" value="2009-10-19"/>
<!-- visited before document creation time -->
<TLINK target="#range(#e1,#t0)" relType="BEFORE"/>
<!-- visited on or before yesterday -->
<TLINK target="#range(#e1,#t1)" relType="ON_OR_BEFORE"/>
<!-- look up is included in yesterday -->
<TLINK target="#range(#e2,#t1)" relType="IS_INCLUDED"/>

TimeML annotations were applied to the news stories in the TimeBank [22], but agreement was low with annotators agreeing on which pairs of events and times to link only 55%, and agreeing on the relation type only 77% of the time. The TempEval competitions [30, 31] therefore tried to simplify the annotation scheme, annotating only temporal relations in certain syntactic constructions (e.g. the main events in adjacent sentences) and adopting a simpler relation set: BEFORE, AFTER, OVERLAP, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE. While requiring an explicit set of syntactic constructions resulted in 100% agreement on which pairs of events and times to tag, agreement on which temporal relation to assign was still low, between 65% and 72%.
5.2.2 TERENCE annotation guidelines

In children’s stories, it is important to annotate the full temporal structure of the story. In particular, all events in the plot should be linked and ordered. Thus in TERENCE, we cannot take the TimeBank approach, where annotators were allowed to annotate whatever temporal relations they liked, and typically left most stories with an incomplete and disconnected temporal relation graph. We also cannot take the TempEval approach, because a restricted set of syntactic constructions also results in gaps in the temporal graph, and because it resulted in lower agreement, probably because “it did not give the annotator the option to ignore certain pairs of events and made it therefore impossible to skip hard-to-classify temporal relations” [30].

Instead, in TERENCE annotators are instructed to annotate a “temporal dependency structure”, that is, to link each event with (at least) one other event in the story, aiming to create a fully connected temporal graph. For example, given the story:

So she [event considered] a while, and then [event climbed] up the wall and let herself [event hang] down by her hind legs from a peg, and [event pretended] to be dead. By and by a Mouse [event peeped] out and [event saw] the Cat hanging there.

An annotator should construct a timeline like:

![Timeline Diagram]

When deciding which other event in the story to link an event to, annotators are instructed to look for the most informative relation that is supported by the language in the story. So, for example, while it would also be possible to identify the relation climbed BEFORE saw, it is better to identify the relation peeped BEFORE saw. There is not much linguistic support for the climbed relation (the events are quite far apart in the story), and the climbed relation would be less informative than the peeped relation because it could be inferred automatically from the peeped relation via the transitivity of the BEFORE and includes relations (but not vice versa).

In TERENCE, annotators are also given a set of relations to choose from that is a compromise between the 13 relations in full TimeML and the 6 relations in TempEval: BEFORE, AFTER, INCLUDES, IS INCLUDED and OVERLAP. This is essentially the TempEval relations minus the disjunctive relations (e.g. BEFORE-OR-OVERLAP) and plus the inclusion relations (INCLUDES and IS INCLUDED). The disjunctive relations are not necessary because in the dependency approach to temporal link annotation, annotators are not forced to annotate temporal relations where they are not every informative. We choose to add the inclusion relations because they substantially improve the ability to reason over the resulting annotations. For example, if the cat and mouse example above had been annotated with OVERLAP relations instead of INCLUDES relations, we could no longer infer that climbed is BEFORE peeped.

These adapted TimeML TLINK annotation guidelines were evaluated based on a pilot annotation study. Two annotators were given 20 stories to annotate using two versions of these
Annotation scheme | Stories | Event pairs | Relation labels
--- | --- | --- | ---
Temporal dependency tree (BEFORE AFTER OVERLAP) | 1-20 | 0.856 | 0.653
Temporal dependency tree (all TERENCE types) | 1-20 | 0.854 | 0.629
Temporal dependency tree (all TERENCE types) | 21-100 | 0.812 | 0.687

Table 5.1: Annotator agreement (Krippendorff’s Alpha) for different temporal link annotation schemes.

guidelines. In the first version, only the relations BEFORE, AFTER and OVERLAP were allowed. In the second version, the relations BEFORE, AFTER, INCLUDES, ISINCLUDED, OVERLAP, and IDENTITY (see Section 4.1) were allowed. We measured annotator agreement using Krippendorff’s Alpha, first over all possible pairs of events which could have been linked, and then over the temporal relation labels assigned to the selected event-event links. The first two rows of Table 5.1 show annotator agreement was high (0.854) for deciding which pairs of events to link, and moderately high (0.629) for deciding which temporal relation to assign. The table also shows that agreement on temporal relation labels (though not on which events to link) is slightly lower when the full set of relations is allowed than when only BEFORE, AFTER and OVERLAP are allowed. However, the difference is small, and the added benefit of better reasoning support (and event coreference support) was deemed worth it.

The last row of Table 5.1 shows the agreement when the annotators were given set of 80 entirely new stories. A high level of agreement was maintained (0.812 for which event pairs to link and 0.687 for which relation type to assign to a pair of events) using the TERENCE guidelines.

5.2.3 TERENCE annotation format

Temporal relation annotations link couples from the event annotation layer (event-to-event relations), and elements of the event annotation layer to elements of the time annotation layer.

5.3 Event Causation

Causal relations between causes and effects are a complex phenomenon with roots in philosophy, psychology and linguistics. Understanding causal relations is a vital part of cognition. In the scope of TERENCE, besides temporal relations, automated detection of causal relations in text is one of the research issues.

5.3.1 State-of-the-Art

Several attempts have been made to annotate causal relations in text. A common approach is to look for specific cue phrases like because or resulted or to look for verbs that contain a cause
as part of their meaning, such as break (cause to be broken) or kill (cause to die) \[16, 25, 13\]. Causal relations have also been annotated in the form of predicate-argument relations, for example in PropBank [3], where ARGM-CAU is used to annotate “the reason for an action”, for example:

\[
\ldots \text{will [PREDICATE remain] on the list [ARGM-CAU because of an interim review]}
\]

Finally, causal relations have been annotated as relations between events in a restricted set of linguistic constructions [27] and between clauses in text from novels [14].

Several types of annotation guidelines for causal relations have been presented, with varying degrees of reliability. One of the simpler approaches asks the annotator to ask whether the sentence they’re reading can be paraphrased using a connective phrase like and as a result or and as a consequence [27]. For example:

\[
\text{Fuel tanks had [event leaked] and [event contaminated] the soil}
\]

This approach is relatively simple for annotators, but agreement is only moderate (kappa of 0.556), in part because there are both causal and non-causal readings of such connective phrases.

Another approach to annotating causal relations tries to combine linguistic tests with semantic reasoning tests. In [14], the linguistic paraphrasing suggested by [27] is augmented with rules like:

- If the potential cause occurs after the potential effect, the example is not causal.
- If the potential effect would probably have happened in the absence of the potential cause, the example is not causal.
- If it is difficult to choose which event is the cause and which is the effect, then the example is not causal.

Despite these clarifications, causal relations were still difficult to annotate, and high agreement (kappa of 0.84) was only achieved by the combined voting of four annotators.

5.3.2 TERENCE annotation guidelines

The annotation of causal relations can be a very complex task, because the notions of cause and result are highly complex and often controversial, as is demonstrated by the fact that they have deserved centuries of philosophical inquires. For instance, one could try to distinguish between causal and enabling relations, or analyze those cases where the expectation for a certain cause-effect relation is violated (despite relations). We could analyze situations in which a certain event causes the non-occurrence of another event (preventive relations), or we could consider that causal relations may obtain not only between events, but also between events and non-eventive entities, as in the sentence [cause A big asteroid] [effect killed] the Dinosaurs, where the non-eventive entity asteroid can be analyzed as the cause of the killing of the Dinosaurs. Also, it should be noted that often causal relations are not overtly expressed
in text, although they can be inferred on the basis of detailed world knowledge. For instance, given the sentence *He tried to [cause open] the door and [effect struck] John in the back*..., one can infer a causal relation between the *opening* and the *struck* events, even if this is not explicitly expressed.

Addressing all these issues, even if only at the level of manual annotation, not to speak about automatic annotation, is clearly outside the scope and the resources of the TERENCE project. Thus, we will follow here a clearly limited and pragmatic approach. First of all we will take into consideration as candidate causes and effects, only those events that have already been identified for the purposes of the previous annotation layers. As a consequence, given the above example, we will not be able to annotate *asteroid* as the cause of *killing*. Second, we will only consider causal relations that are explicitly expressed in text by conjunctions and adverbials such as “because”, “due to”, “as a consequence”, “because of that”, etc. In this way we can also hope that the manual annotations that we will perform will reach a satisfactory level of inter-coder agreement.

Given this approach we can still aim at distinguishing causal, despite, and prevent relations in those cases when the semantic of the relation is explicitly expressed in the text.

### 5.3.3 TERENCE annotation format

Causal relations are introduced as a standoff annotation layer linking event annotations.

### 5.4 Intentional relations

#### 5.4.1 State-of-the-Art

Only a small amount of work has been done on the annotation of intentional relations between events in text. “Purposes” have been annotated in PropBank using the ARG_M-PRP link, which identifies when a constituent is “the motivation for some action”, for example:

*Edison could [PREDICATE raise] its rates [ARG_M-PRP to pay for the plant]*.

Events that indicate some intention are also recognized by the TimeML annotation scheme, which gives events like *attempt, avoid and expect* semantic classes like *INTENTIONAL_ACTION* and *INTENTIONAL_STATE*. TimeML does not include a intentional or purpose-like link between events, but SLINKs can at least indicate when an *INTENTIONAL* event introduces a new possible world or indicates that a subordinate event must be true or false.

#### 5.4.2 TERENCE annotation guidelines

TERENCE must go beyond the annotation schemes of PropBank’s ARG_M-PRP and TimeML’s SLINKS, which only annotate certain clause structures, to capture all intended events regardless of the syntactic structure in which they appear.
In TERENCE, an intentional relation is a relation between an intentional event (or an intentional state) and another subordinated event, called the intent, which denotes the purpose of the intentional event. The intent is commonly expressed by a subordinated clause, a nominalization, or an untensed verbal phrase. A intentional relation can be seen as the purpose of the (intentional) event. For example:

\[
\text{He [event \textit{jumped}] to [intent \textit{reach}] ...}
\]

In the example above the intent of \textit{reaching} is the reason for \textit{jumping}.

As a counter-example of intentional relations, consider the following example:

\[
\text{And they [event \textit{decided}] to [intent \textit{kill}] it}.
\]

In this example the intent of \textit{killing} is not the reason for \textit{deciding}, therefore no intentional relation will be annotated between the two events.

### 5.4.3 TERENCE annotation format

Similar to causal relations, intentional relations are introduced as a standoff annotation layer linking event annotations.

### 5.5 Link signals

#### 5.5.1 State-of-the-Art

Sometimes links such as temporal relations are signaled explicitly in the language, e.g. by the use of a temporal preposition like \textit{before}. TimeML defines a \texttt{SIGNAL} annotation for such words:

\texttt{SIGNAL} is used to annotate sections of text, typically function words, that indicate how temporal expressions or eventualities are to be related to each other. The material marked by \texttt{SIGNAL} constitutes several types of linguistic elements: indicators of temporal relations such as temporal prepositions (e.g. \textit{on}, \textit{during}) and other temporal connectives (e.g. \textit{when}) and subordinators (e.g. \textit{if}).

Once identified, a \texttt{SIGNAL} can then be linked to the relation that it evokes:

\[
\text{John left 2 days before the attack.}
\]

\[
<!-- left -->
<\text{EVENT} \texttt{eid="e1" target="#token2"} />
</!-- 2 days -->
<\text{TIMEX3} \texttt{tid="t1" target="#range(#token3,#token4)"} />
</!-- before -->
<\text{SIGNAL} \texttt{sid="s1" target="#token5"} />
</!-- attack -->
\]
5.5.2 TERENCE annotation guidelines

For generating questions that test whether a reader has mastered both explicit and implicit cues about story structure, it is important to identify in the annotations when there was or was not an explicit signal of a relation. Thus in TERENCE, annotators are instructed to tag signals as per the TimeML guidelines. In particular, annotators should identify all temporal uses of the signals before, after, when, while, as and then (in Italian: prima, dopo, quando, mentre e poi). In addition to signals for temporal relations, as in TimeML, signals will be marked also for causal relations. More specifically, annotators should identify all causal uses of because, thus and so (in Italian: perché, quindi and cosi).

5.5.3 TERENCE annotation format

Signal annotations are carried out directly on top of the basic token annotation layer. They are linked to relations between events and times and to relations between causes and results.
6 Annotation Tools

6.1 State-of-the-Art

Three tools are currently available in order to perform TimeML-compliant annotation. One is actually the combination of two systems, namely Callisto (http://callisto.mitre.org) and Tango (http://www.timeml.org/site/tango/tool.html). The first is a general-purpose tool for performing linguistic annotation of documents in any Unicode-supported language, and it is used to annotate events, time expressions and signals (see Fig. 6.1). The second was specifically developed for adding TLinks, ALinks and SLinks. As shown in Fig. 6.2 the visualization of relations with Tango may become unclear if many relations have been annotated on the same text.

![Figure 6.1: Screenshot of Callisto interface](image)

Another tool, which overcomes the problem of performing annotation using two different tools, is BAT (Brandeis Annotation Tool, http://timeml.org/site/bat/).

The tool does not need installation because it can be accessed through a web browser after obtaining an administrator account from the developers. The TimeML annotation task is divided into a set of subtasks, which however do not have to be necessarily performed in sequence, given that all dependencies are observed:

- Annotation of Event extents
Each document is assigned to two or more annotators, and after the completion of the task a final adjudication phase is performed. This process is particularly useful when several annotators are supposed to work at the same document. However, the use of this tool is affected by some shortcomings due to the annotation levels: if a level has not been completed, it is not possible to start annotating the next one, and if some modifications have to be carried out on a level that has already been completed, it is not straightforward to go back to the previous levels without deleting some annotated information. Furthermore, BAT was developed for TempEval-2, which implies that TLink annotation is divided into four different levels and SLink/ALink annotation is not supported.

A third tool is the CELCT Annotation Tool (CAT), which has been developed in the framework of the Italian TimeBank project [5]. This standalone tool supports ISO-TimeML format and can be easily extended in order to handle additional annotated elements and relations. Given that in TERENCE stories information about events’ participants and coreference will be needed as well, CAT is the tool that best meets the project requirements and that will be used to perform annotation (for further details, see Section 6.2).

The advantage of CAT is that it was primarily implemented for performing TimeML annotation, which means that common TimeML standards as well as file formats are supported. This
TERENCE project – ICT FP7 Programme – ICT-2010-257410
State of the art and design of novel annotation languages and technologies

Figure 6.3: Screenshot of BAT interface

Figure 6.3: Screenshot of BAT interface
does not happen with other general-purpose annotation tools like MMAX2 [18], which are intended to support almost any kind of linguistic annotation and therefore must be customized in order to be employed for a specific annotation task. Using CAT, instead, we avoid making a time-consuming adaptation effort. Other tools for semantic annotation, for example SALTO [4], have been designed to annotate an additional information layer on top of parse trees for tasks such as role and discourse relation annotation, but again they would need additional adaptation work to support all information encoded in TimeML as well as the ISO-standard format.

6.2 TERENCE Design

In order to annotate documents in TERENCE, we employ the CELCT Annotation Tool (hereafter CAT), a standalone system designed with the aim of optimizing task decomposition and flexibility. This tool can potentially support many types of annotation due to the number of configurable features available, although it has been primarily employed for TimeML-like annotation [5]. The basic idea behind the tool is that different information layers should be annotated in a text without particular interdependencies, i.e. it should be possible to add an annotation layer without waiting for the accomplishment of the others. Besides, a user is free to easily add new elements, attributes and relations to those already available through the interface.

The tool interface (see Fig. 6.4) is based on four main components, namely the Corpus Panel (top left), where all files are listed, the Layers Panel (bottom left), where the annotation layers, the empty tags and the lexical categories are listed, if present, the Text Panel (top right), with plain text source data, and the Relations Panel (bottom right), where the available relations are displayed and can be deleted, if necessary.

The objects to be annotated are called Markables, and correspond to a text string, either a single token or a sequence of words. In TERENCE, events, temporal expressions and entities involved in events are associated with a markable. For each of them, several attributes can be defined using a radio button label, a text box or a dropdown menu. A screenshot of the graphical interface for event and attribute annotation is displayed below. The attribute
window contains all options encoded in TimeML, such as pos, tense and aspect. Further options can be easily added by the user.

Markables can be connected to each other by means of Relations, which in standard TimeML annotation correspond to TLinks. In TERENCE, other relations have been added, for example the participants’ relation between an entity and an event, and the anaphoric one between co-referring entities. A screenshot of the graphical interface for relation annotation is displayed below. In this case, a temporal link is set between the event guardava (looked at) and the
temporal expression *notti* (*nights*), after that *nelle* (*in the*) has been annotated as signal. This directional relation is marked as an ‘IS_INCLUDED’ relation.
7 Conclusions

In this document we present the annotation language adopted for TERENCE stories, which is an extension of TimeML standard with anaphoric and participants relations. We distinguish three basic story elements to annotate, namely events, entities and time expressions. For each of them, we briefly describe how they are represented in TimeML standard, and how we handle them in the framework of this project. Besides, we present CAT, a tool developed for TimeML annotation which supports all TERENCE extensions and will be employed in the annotation of the project stories.

We plan to start the manual annotation of the stories in English and Italian as soon as the story repository is complete. During this, we will possibly refine the the annotation scheme in light of the evaluations of WP7 and the games developed in WP4. The annotation is needed on the one hand to provide ‘gold’ information used for a first development of story games, and then to provide training material for the implementation of natural language processing modules that will annotate new stories automatically.
Bibliography


