TERENCE project - ICT FP7 Programme - ICT-2010-257410

State of the art and design of novel intelligent feedback, Deliverable 4.1

Document Version: 11
Created: March 15 2011
Last updated: September 29, 2011
Distribution level: Public

Document Information

Document ID Code	D4.1		
Keywords	Automated reasoning, games, game design, intelligent feedback		
Current classification of this document	*Document Version*: 11	*Official date of reference*: September 30 2011	*Published date*: March 15 2011
Distribution level	Public		

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

	Surname and Name	Company
Editor	Gennari Rosella	LUB
List of contributors	Di Giacomo Dina	UnivAQ
	Di Mascio Tania	UnivAQ
	Gennari Rosella	LUB
	Melonio Alessandra	UnivAQ
	Tifrea Oana	LUB
	Vittorini Pierpaolo	UnivAQ
Reviewers	Bethard Steven	KUL
	de Gasperis Giovanni	UnivAQ
	de la Prieta Pintado Fernando	USAL

Version history		
Date	March 15 2011	Version by Gennari revised with Di Giacomo and Vittorini
	April 4 2011	Version by Gennari
	May 11 2011	Version by Tifrea
	June 1 2011	Version by Gennari
	June 10 2011	Version by Tifrea
	June 11 2011	Version by Gennari revised with Di Giacomo and Vittorini
	June 20 2011	Version by Tifrea
	July 10 2011	Version by Gennari revised after the S&T meeting in Salamanca, July 1 and 2, 2011
	August 5 2011	Version by Gennari, Di Mascio and Melonio revised after the meetings with Saverio (educator and education stakeholder) and Dina Di Giacomo, on August 3rd and 4th respectively
	September 6 2011	Version by Gennari, after the revision by Steve Bethard
	September 20 2011	Version by Gennari, after the revision by Fernando de la Prieta Pintado and Giovanni de Gasperis
	September 29 2011	Version by Gennari, after the S&T meeting in Palermo

Abstract
The automated reasoning module of WP4 is used for resolving and providing feedback for the TERENCE smart games, which are created using the annotations of stories (WP3). The present deliverable is the first of WP4. It reports on: the design of the game prototypes of the first year of the TERENCE project; the state of the art of automated reasoning for the annotations and the games; the intelligent feedback useful for them. It concludes by outlining the impact it can have on correlated working packages.

Copyright notices

Contents

List of Figures

List of Tables

1 Executive Summary

The TERENCE games, introduced in [Alr11], are specialised into smart games, which stimulate inference-making for story comprehension, and relaxing games, which stimualte visual perception and not story comprehension. Figure 1.1 describes the current taxonomy of the TERENCE games based on their stimulation target.

WP4 is concerned with the automated reasoning module of TERENCE. The module is used for resolving and providing feedback for the TERENCE smart games, created using the annotations of stories (WP3). The automated reasoner also complements the annotation process performed in WP3 for some of the TERENCE smart games. This document, the first deliverable of WP4, lays the groundwork for such work that foresees two other two deliverables at month 23, namely, the automated reasoning module (D4.2) and the smart games with intelligent feedback (D4.3).

This deliverable first delves into the current design of the TERENCE smart games, and then tackles the state of the art of temporal reasoning about temporal annotations. The former helps in selecting the relevant material for the latter.

More precisely, Chapter 2 presents the current design of the TERENCE smart games. This work is grounded in interventions by educators or experts, discussed in [SG11], and on-going evaluation work part of WP7. In particular, Section 2.2 analyses and classifies the games according to their specific stimulation target, i.e., whether it stimulates reasoning about a single event or multiple events of the story. Section 2.3 classifies the games according to the formats of the resolutions, e.g., multiple choice. Combined together, the analysis of the stimulation target and of the format allow us to set diverse difficulty indices for the games. Section 2.4 tacks on another perspective over games: their visual format. It firstly, briefly overviews several visual representations found in the literature and amenable for the TERENCE smart games. Secondly, dragging from the two previous sections, it proposes the first visual templates for the TERENCE smart games.

The state of the art of automated reasoning for the annotation process and for the TERENCE smart games is analysed in Chapter 3. The reasoner has thus two main intended types of end users: experts for the annotation process; learners for the smart games. The intelligent feedback is separately analysed for both types in Chapter 4. Both the latter chapters capitalise on the previous exposition of the game design and classification, e.g., the design of certain feedback is made dependent on the type of game a leaner tackles.

Chapter 5 concludes the document with a list of desiderata for the annotation of the TERENCE texts and images alike, rooted in the analysis of the games and of the reasoner's feedback.

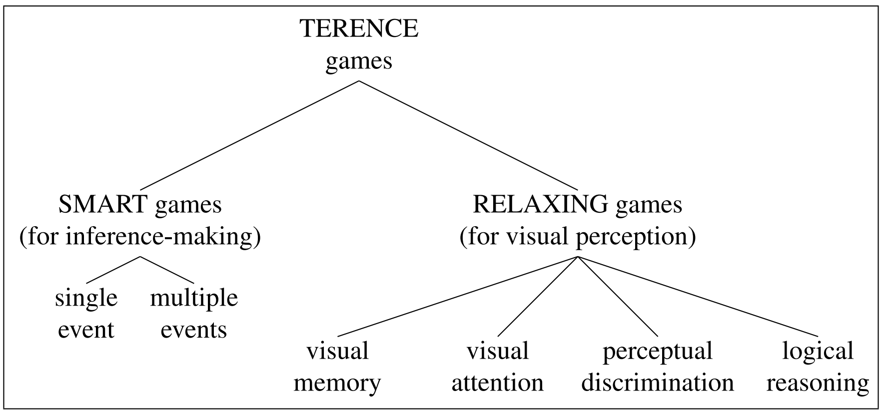


Figure 1.1: Game taxonomy

2 Game Design

2.1 Introduction

This chapter explains the current design of the TERENCE smart games that require the intelligent feedback of the automated reasoning module. Section 2.2 proposes a taxonomy for the TERENCE smart games concerning events of the story, based on the main target of the stimulation. See Figure 2.1. Section 2.3 analysis the formats of the TERENCE answers. Section 2.4 first overviews popular touchscreen games for primary school children, mainly commercial. Based on this, the aforementioned game taxonomy and answers' formats, it then sketches the design of the visual smart games of TERENCE.

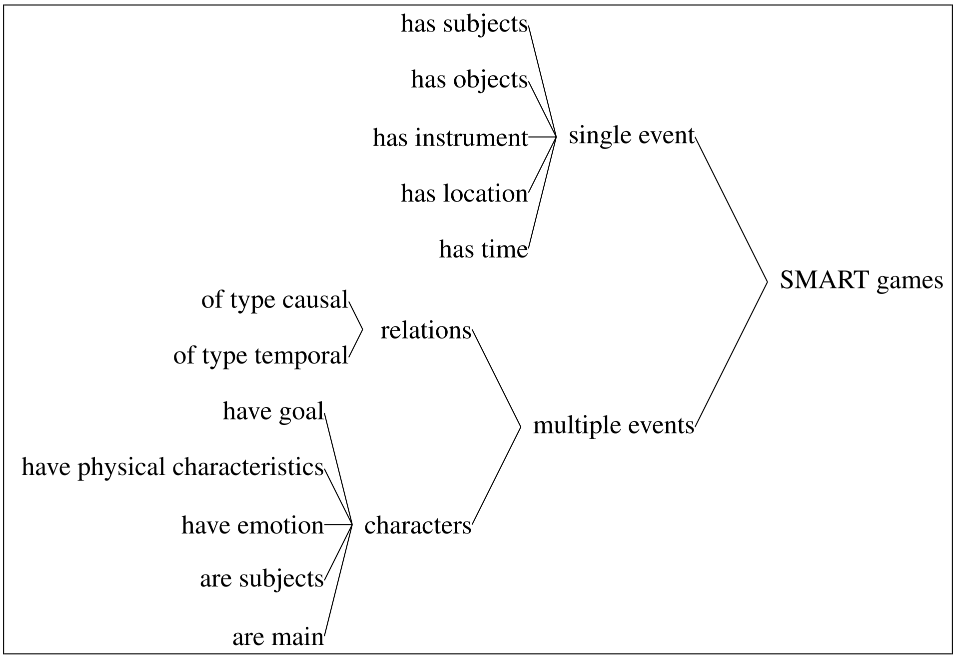


Figure 2.1: Taxonomy of the TERENCE smart games

2.2 Taxonomy of Smart Games

Smart games concerning events of the story are specialised according to their main stimulation targets:

– reasoning about a single event's arguments,

– reasoning about multiple events, and more precisely:

1. their temporal or causal relations;

2. the characters, that is, their relation to story's events or whether they are main characters of the story.

See Figure 2.1. This section presents and delves into them. Subsection 2.2.1 overviews the sources for such games. Subsection 2.2.2 introduces the necessary terminology and game events, thereby setting desiderata for the novel annotation language of [Bet11]. With the preliminary out of the way, Subsection 2.2.3 presents the first type of games, that is, the games for reasoning about the arguments of a game event. Subsection 2.2.4 and 2.2.5 present games for reasoning about multiple events, that is, games for temporal or causal relations, and games concerning the story's characters.

This section also gives a first definition of text difficulty indices, according to the story text and the stimulation target. The indices set some of the desiderata for the annotation language of WP3. Later on, they will be used to create difficulty metrics for smart games. The metrics will allow us to refine the adaptation model of WP2 specified in [Alr11], e.g., for the automatic adaptation of games and stories to the learner.

2.2.1 Sources

The main sources for the classification of questions for the TERENCE smart games were:

- interventions by educators, see Tasks in [SG11],
- inference-making questions from diagnosis tests received from Barbara Arfé and Jane Oakhill, analysed by WP3,
- inference-making interventions from therapy batteries received from Dina Di Giacomo,

all filtered through the user requirements of [SG11] and the on-going evaluation form part of WP7. Usually, questions from diagnosis tests ask about the arguments of a single event. Interventions from therapy batteries, instead, generally target the relations between events in the story.

2.2.2 Events and Annotations

Events hereby considered are *factual*, some describe varying *physical* characteristics like "the girl became double size" or "the boy wore a black cap", and some are *emotional*, that is, conveying emotions, like "to cry", "to feel", "to laugh", "to tremble". The events of a TERENCE story will be annotated with the TimeML-based language described in [Bet11]. Each event in the story is unique. The annotations of the i-th story event have attributes that specify

- the event's verb in infinite mode, hereby denoted by e_i,
- and, whenever possible, the arguments of e_i, in particular, its subjects, objects, instruments, location (whether an environment, or not), time, all rendered through their unique identifiers in the story. These are denoted by s_i, o_i, m_i, l_i, t_i, respectively.

The *event mention* of e_i is the phrase in the story text of the i-th event, having the aforementioned annotations.

Game events. A *game event* is a phrase with verb, and possibly subjects, objects, instruments, location and time. The verb is usually in present tense, active mode. Game events come in three flavours:

- the *normalised* event of e_i is the (grammatically correct) phrase of the event mention of e_i, with arguments e_i, s_i, o_i, m_i, l_i, t_i;
- the *lexical variant* event of e_i uses lexical variants of any of the arguments or of the verb of both the event mention of e_i and the normalised event of e_i;
- the *pragmatic* event is the phrase that is pragmatically inferred from the annotations of event mentions.

Note that pragmatic events derive from scripturally-implicit or pragmatic questions found in diagnosis tests by psychologists. They are usually who, what, why and often where questions, see [SG11]. For instance, consider the case when the question is "what does John play" and in the story we have the following clue annotations: "Jonh scores the winning goal", "the team", "He plays with his ten mates", so a pragmatic event, consistent with the story, is "John plays football". Such questions are likely to be the most difficult for poor comprehenders according to the literature, e.g., see [YO88], as well as the feedback we received from educators on the evaluation form for stories, part of an on-going work of WP7.

Generation of games. The annotations of event mentions allow us to generate game events, and answer questions about such events, like who questions. Such questions are described in the remainder of this chapter. Hereby, we focus on the generation of the game events. Using the annotations of an event mention, we generate a normalised event sentence, in the form

subject(s)
verb (possibly with modifiers)
object(s)
instrument(s)
location
time.

Lexical variants of e_i are then generated from the normalised event of e_i and its event mention. In TERENCE, the pragmatic events are generated by experts of pragmatic inferences, mainly psychologists. All these events and the related questions described in Subsection 2.2.3 are used by the AR module to generate the textual part of the smart games. See Figure 2.2.

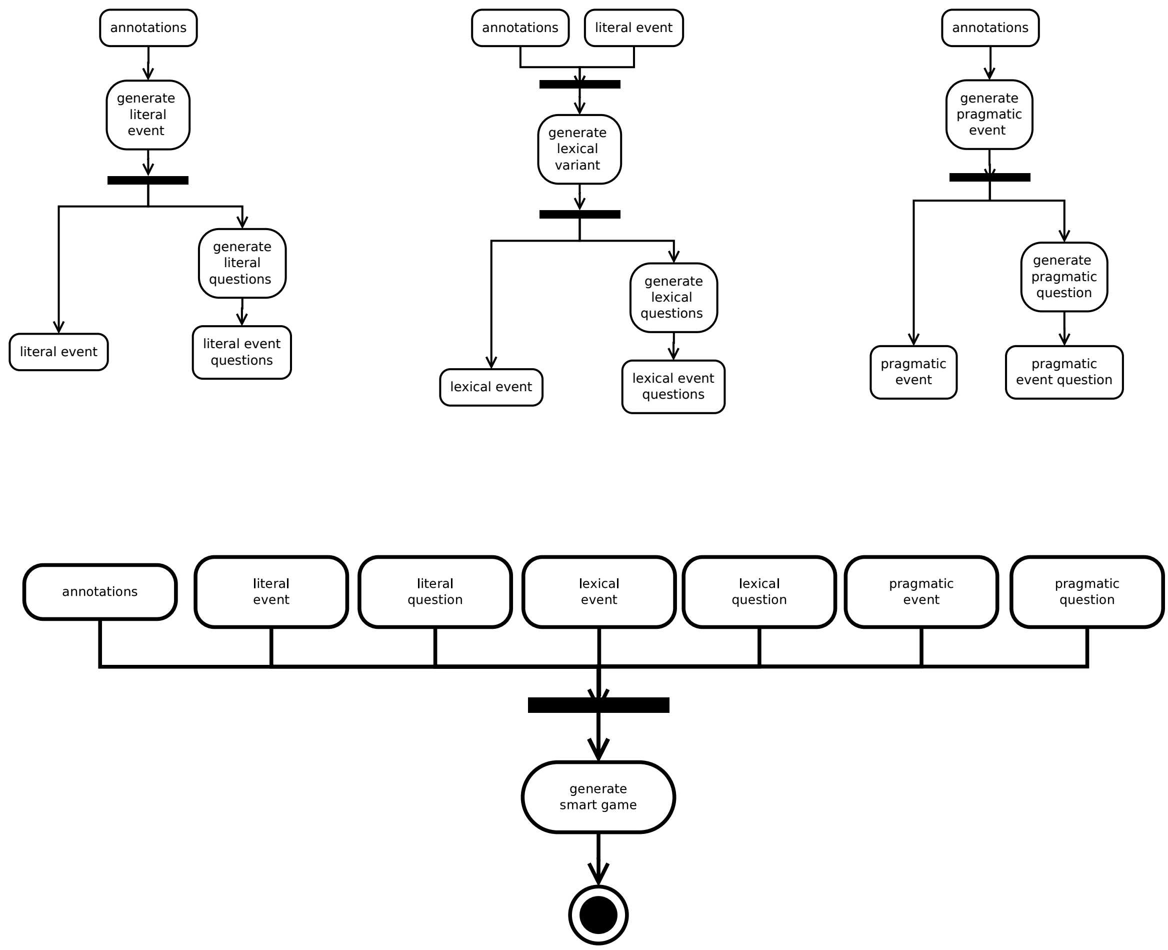


Figure 2.2: Generation of game events for the smart games of TERENCE

Example 1. *In the text in the Appendix we have the story event "One afternoon, she found a new book in the library shelf". The tense of the verb in the syntactic phrase is past remote, and the mode is active. By means of an operation, the system will generate the normalised event "One afternoon, Perla finds 'The Fantastic Circus' in the library shelf". A further operation will generate its lexical variant event "One afternoon, the girl finds 'The Fantastic Circus' in the library shelf". A pragmatic event is "The clowns throw sawdust in book" generated from the annotations of the event mentions "The clowns threw sawdust everywhere" and "a small amount of sawdust falling from the book".*

Annotations for difficulty. The annotation can also be useful for classifying smart games into difficulty levels, according to the user requirements of [SG11], e.g, answering a who question is likely to be more difficult if it requires to resolve an anaphoric expression for the subject and the verb mode is passive in the story text. To this end, the annotation of a event mention should mark, whenever possible:

- the type of event, e.g., factual, physical, emotional,
- only the core factual event of both aspectual and reporting events (e.g., "Perla went" instead of "Perla decided to go"), and that the core factual event stems from an aspectual or reporting event,
- the root form (a.k.a., stem, lemma, infinitive form) of the core event, that is, its verb,
- tense, mode and irregularity of this verb,
- the main arguments of an event, and the role (a.k.a., type) of arguments,
- whether the arguments of an event are characters of the story,
- whether a character is a main character of the story,
- whether the subjects and objects are animated or not,
- whether the location is an environment or not,
- which arguments are implicit, e.g., if the event mention is "She went home", the implicit argument is "she"; whereas the game normalised event is "Johan goes home" using the resolvent "Johan";
- in case any of the argument is implicit, the distance from the closest occurrence of the resolving expression in the story; the distance is equal to the number of intervening words.

2.2.3 Arguments of a Single Event

The games concerning the arguments of a single event ask the learner to reason about the subjects, objects, instruments, locations and temporal occurrence of a game event. They do it by asking who, what, how, where and when questions. They are found in all our sources. However, temporal when-questions are scarcely present in diagnosis tests and therapy batteries of psychologists. This may be due to the length of the test stories, which is similar to that of the TERENCE stories, shorter than 500 words. Moreover, short children stories tend to have no specific time expressions besides day-time or season expressions like "in the morning", "in summer". Therefore the last type of questions are here introduced for they occur in the interventions by teacher but are unlikely to be used in TERENCE.

Textual question format and generation. Questions for the arguments of a game event are in the following forms:

- who/what questions asking for the subjects: a who question in case the subjects are animated, else a what question;
- who/what questions asking for the objects: a who question in case the objects are animated, else a what question;
- how questions asking for the instruments;
- where questions asking for the location (whether environment, or not);
- when questions asking for the time of occurrence.

Such questions are generated from game events: normalised, lexical variant, pragmatic.

Examples. By referring to the story in the Appendix, we have the following normalised questions and answers.

- Who finds "The Fantastic Circus" (in the library's shelf)? Perla.
- What does Perla find in the library's shelf? "The Fantastic Circus".
- How did the clowns enter the circus? By car.
- Where does Perla find "The Fantastic Circus"? In the library shelf.
- When does Perla find "The Fantastic Circus"? In the afternoon.

Then "The little girl discovers 'The Fantastic Circus' in the library shelf" is a lexical variation of "Perla found a new book in the library shelf" (event mention) and "Perla finds 'The Fantastic Circus' in the library shelf". We can have then who, where and what questions like the above concerning such a lexical variation event. A pragmatic question is the "Who throws sawdust in 'The Fantasic Circus'?".

2.2.4 Temporal or Causal Relations

Games for temporal or causal relations between events are often found in the therapy batteries and instructional material alike, and in the form of why or when questions in the diagnosis material of psychologists. By therapists, they are considered in general more difficult than questions concerning a single event because the former questions ask the learner to create semantic relations between events.

The games concerning relations between game events ask the learner to reason about:

- the causal relation between the given events,
- the temporal relations between the given events.

Examples. By referring to the story in the Appendix, we have the following example questions.

- Perla laughs because the clowns throw water and sawdust everywhere. True/false?
- Two acrobats entered pirouetting on the circus floor and then there was a big loud BANG. True/false.

Annotation for difficulty. Besides those listed in Subsection 2.2.2, other annotations useful for classifying such games into difficulty levels, according to the user requirements of [SG11], are the following. In case of normalised or lexical variant events of e_i and e_j, the annotation for the temporal or causal relation between the event mentions of e_i and e_j must have attributes for specifying:

- whether the relation is rendered in the story via an explicit causal signal or is not. Examples follow:
  1. "She went home because she felt bad"; "She felt bad. She went home";
  2. "She felt bad after she went to school";"She went to school. She felt bad";
  - in case it is an explicit temporal relation, the order (reverse, non reverse) is also annotated (see [Pas11]);
- if the event mentions of e_i and e_j occur in different non-adjacent sentences.

2.2.5 Characters

Games concerning a story's characters are often found in the therapy batteries and instructional material alike. By therapists, they are considered in general more difficult than questions concerning a single event because the former questions ask the learner to create semantic relations between events and their characters.

The TERENCE games concerning given characters are specialised as follows:

C-1 given the game event of e_i of which the characters are subjects and the causally related game event e_j, ask for the goal, physical attributes or emotions of the characters in relation to the game event of e_j,

C-2 ask for the main events of which the characters are subjects,

C-3 ask which of the characters are main characters of the story.

Such games are typical of interventions by teachers and books for teachers, see Tasks in [SG11].

The games in item C-1 ask about the relations of a character's goals, physical characteristics or emotions in relation to events of the story. As for goals, the situation model of [Zwa99] also measures whether two actions are related to a protagonist's *global* goal. However, this is common of fables and not of the TERENCE stories. In these, the story's main characters may have different goals according to the events they are involved in. Consequently, we need C-1 games that ask the learner to decide which event is a goal of an event with characters as subjects. Physical characteristics or emotions of characters in relation to the story's events are typically questioned by teachers at school, and sometimes they are the focus of questions by psychologists. Hereby we consider those that explicitly ask about the physical characteristics or emotions of story's characters. The emotions we consider in the TERENCE games are the seven of [Ekm73] [1]. In order to generate such questions from the game events of e_i and e_j, the annotation for the event mention of e_i must mark that this is a physical or an emotional event. The question's difficulty depends on whether or not the event mention or game events of e_i explicitly state anyone of those emotions.

[1] They are *a)* neutral, *b)* happy, *c)* sad, *d)* fearful, *e)* angry, *f)* disgusted, *g)* or surprised.

Examples. By referring to the story in the Appendix, we have the following example questions.

- Perla ducks so that she avoids the cake. True/false?
- Perla feels happy (generated from the game event "Perla laughs") when/after the clowns throw water and sawdust everywhere. True/false?
- What does Perla do in the story? Perla finds "The Fantastic Circus" in the library shelf...
- Which of the following is a main character of the story? Perla, Gianna, the acrobats, the clowns.

2.3 Item-based Answers

The Computer Assisted Assessment (CAA) literature, e.g., see [MB99], is the main source for the analysis of the formats of the games' resolutions, a.k.a. answers, that are amenable for the automated feedback.

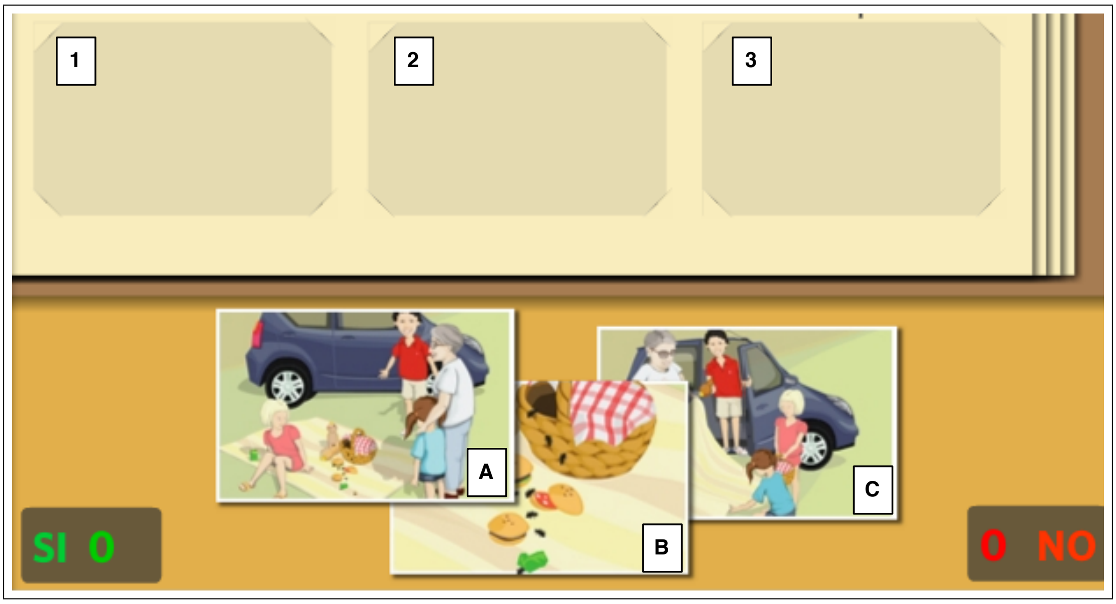


Figure 2.3: A sequencing game

TERENCE prefers item-based answers to free-text answers. Firstly, the correctness of free-text answers may be biased by the ability of the TERENCE learners to write their answers in a grammatically correct format. Given their age and grammar problems (see [SG11]), this is an unrealistic assumption. Secondly, item-based answers are more amenable to the automatic assessment and hence to the automatic adaptation of TERENCE, see also [McA02]. Thirdly, current educational material for primary school children heavily uses item-based answers, in UK and Italy alike, and hence the TERENCE learners are used to them.

Given this, the TERENCE answers to the above questions are usually items to choose among, or to be matched with other items or buckets. Formally, given a question/task, we have

- a set $I \times B$ of answers, where I is a set of $|I| > 0$ items to choose among and B is a set of $|B| > 0$ buckets/items for the items from I,
- a nonempty subset CIB of $I \times B$ of correct answers.

In the remainder, we refer to $I \times B$ as the set of *classify* answers, consistently with [Alr11]. For instance, consider the sequencing game in Figure 2.3. Its answers are classify answers. The $|I| = 3$ items (labelled with 1, 2, 3) are illustrations of events of a story; the $|B| = |I| = 3$ buckets (labelled with A, B, C) are the place-holders for such illustrations. The subset CIB of correct (classify) answers is $(1, C), (2, B), (3, A)$. Other well-known formats for answers are given by text-based multiple-choice, graphical hotspot, and matching games. See also [MB99].

The set $I \times B$ of classify answers can be specialised according to the number $|B|$, as explained below.

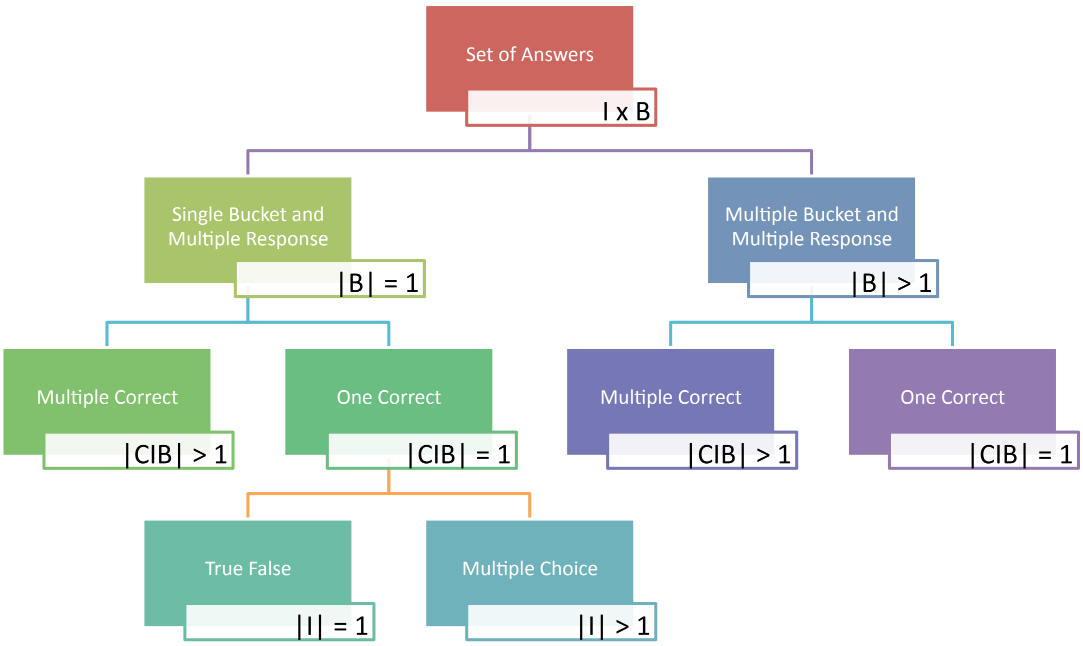


Figure 2.4: Formats of answers

1. If $|B| = 1$ (single bucket), then we have classical (single bucket) multiple response answers. In turn, multiple response answers can be of the following types:
   - if $|CIB| = 1$ (more than one correct answers) they are named multiple correct,
   - else (exactly one correct answer) they are named one correct, and
     - if $|I| = 1$ (only one item to choose) they are true false,
     - else (more than one item to choose) they are multiple choice.
2. If $|B| > 1$ (many buckets) then we have multiple-bucket multiple-response answers. These are specialised as follows: if CIB is a bijective function (that is, there is precisely one bucket for each item), then they are named one correct; else they are named multiple correct.

The resulting answer formats are in Figure 2.4. The formats give us further difficulty indices, that is, the difficulty of a game also depends on $|B|$ and $|I|$.

2.4 Visual Games

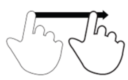
Subsection 2.4.1 sketches popular visual games for touchscreen tablets. Starting from this and the taxonomy for answers in Section 2.3 and smart games in Section 2.2, we conclude sketching visual templates for the TERENCE smart games.

2.4.1 Popular Games

Popular games for touchscreen tablets are outlined in [DMM11]. Hereby, we consider those games that, currently, seem amenable for the visualisation of the smart games of TERENCE introduced in Chapter 2.

A *tiling or puzzle* game asks the user to drag the shapes in the right place in order to complete the tiling or puzzle. A *shade* game asks the user to drag the right figure in the shadow. A *correlate* game asks to correlate items to buckets. Examples of such games taken from [DMM11] are in Figure 2.5. A *sequencing game* ask to order illustrations in a given order. An example is in Figure 2.4.

The gestures used in such games are recapped in Figure 2.1.

Gesture name	Textual description	Iconic description
Tap	briefly touch surface with fingertip	
Double tap	rapidly touch surface twice with fingertip	
Press	touch surface for extended period of time	
Drag	move fingertip over surface without losing contact	
Flick	quickly brush surface with fingertip	

Rotate	touch surface with two fingers and move them in a clockwise or counterclockwise direction	
Pinch/spread	touch surface with two fingers and move them close/apart	
Combined gestures	combine some of the above gestures	
Tilt	move the tabletop holding it with two hands	[image]

Table 2.1: Gesture for touchscreen tablets and its iconic representation

2.4.2 Visual Templates

Games About the Arguments of a Single Event. Games concerning the arguments of an event have answers in single-bucket multiple-response format ($|B| = 1$), where the bucket is the given event or a given feature (e.g., subject) of the event. Remember that e_i is in general the main verb of the event i; s_i are its subjects, if any; o_i are its objects, if any; l_i gives its location, if any; m_i gives its instruments, if any; t_i gives its temporal occurrence, if any. See Figure 2.6.

Games About Relations. Games concerning causal relations, intentions or temporal relations can be in single-bucket multiple-response format ($|B| = 1$) as well as multiple-bucket multiple response format ($|B| = 1$). See Figure 2.7.

Games About Characters. Games concerning attributes of characters are envisioned in single-bucket multiple-response format ($|B| = 1$), where the singleton B only contains a character. See Figure 2.8.

Figure 2.5: Examples of visual games

Single Event

	Who/what	Whom/what	Where	How
Task	Find the correct subjects	Find the correct objects	Find the correct location	Find the correct means
Description	A shade-like game (e.g., as in Geronimo Stilton): the event illustrated without the subjects that are replaced by a gray box	A shade-like game like for objects: now, the objects are replaced by a gray box	The subjects of the event are illustrated; the possible locations are illustrated	A shade-like game: now, the possible means are replaced by a gray box
Answer	The child needs to place the correct subjects (items) in the gray box (bucket)	The child needs to place the correct objects (items) in the gray box (bucket)	The child need to correlate the subjects of the given event with the correct location	The child needs to place the correct means (items) in the box (bucket)

Figure 2.6: Visual templates for games about a single event

Multiple Events - Relations

Question	Causal	Temporal
Task	Identify the causal relations	Identify the temporal relations
Description	Single bucket. A correlate game with one event and the possible events which that event causes Multiple buckets. A sequence-like game with events to be ordered: the event on the left causes the event on the right	Single bucket A. A correlate game with one event and the possible events which that event are temporally correlated to. Single bucket B. Tiling game with 4 empty boxes; four illustrated events, of which two are overlapping , another one is before and the remaining one is after the overlapping events. Multiple-buckets. A sequence-like game with events to be ordered: the event on the left is before the event on the right
Story Example	Perla laughs ***because*** the clowns threw water and sawdust everywhere	Two acrobats entered pirouetting on the circus floor ***and then*** there was a big loud BANG
Answer	Single bucket. The child needs to correlate the the given event (bucket) with the correct cause event correct (item) Multiple-buckets. The child order the events in the correct sequence	Single bucket A. The child needs to correlate the given event (bucket) with the correct sequential/overlapping event (item) Single bucket B. The child needs to correctly place the events (items) in the boxes (buckets). Multiple-buckets. The child order the sequential events in the correct sequence

Figure 2.7: Visual templates for games about causal or temporal relations

Multiple Events - Characters

Question	Have goal	Have emotion	Are subjects	Are main
Task	Identify the goal	Identify the emotion	Identify the events of which the characters are subjects	Find the main characters
Description	A causal-like 1single-bucket game	A shade-like game: the event (bucket) triggering the emotional event is illustrated without the face of the characters, subjects of the emotional event; the faces with possible facial expressions (items) of these chararcters are illustrated	An events of which the characters are subjects is illustrated; other possible events, from the same book, of which the characters are not argument are illustrated, and their subjects are replaced by the given characters	A game like Bugs Rush Lite: characters from the same book are floating, among them are the main characters of the story
Story Example	Perla bends ***so that*** she avoids the cake	The object of "Perla finds "The Fantastic Circus in the library shelf""	The correct event is "Perla finds the "The Fantastic Circus" in the library shelf", a wrong event is "Perla made a double somersault"	The means of "The clowns enter the circus tent with their battered car"
Answer	The child needs to correlate the given event (single bucket) with the correct cause event (item).	The child needs to flick the faces with the correct facial expression (items) into the empty faces of the event (bucket)	The child needs to double tip the correct event	The child needs to press the correct characters (items)

Figure 2.8: Visual templates for games about characters

3 State of the Art of Automated Reasoning About Temporal Annotations

3.1 Introduction

Traditionally, in *artificial intelligence* (AI), temporal reasoning consists of "formalising the notion of time and providing means to represent and reason about the temporal aspects of knowledge" [Sch98]. In other words, it means choosing:

- *representation:* a time granularity and structure, and a formal language for them;
- *reasoning:* a reasoning system, amenable to automation, with specific reasoning tasks that, ideally, are computationally tractable.

A third facet of time is often neglected in AI, and confined to *human computer interaction* (HCI):

- *visualisation:* the visualisation of temporal information.

However, this is also a crucial facet in the development of a usable tool for representing and reasoning about temporal relations, given that graphics seem to be a relevant support for humans in semantics oriented tasks [Pai91].

In this document, we consider the three aforementioned aspects of automated temporal reasoning all together in relation to the TimeML annotation language, which is a temporal markup language that aims at capturing the richness of time information in written documents, and is considered as the starting point for the annotation language described in [Bet11].

3.2 Representation of Events and Qualitative Temporal Relations

There are different temporal structures, for instance, linear, cyclic, or branching [Sch98]. Linear time corresponds to our natural perception of time (in Western culture) as being ordered collections of temporal primitives, e.g., time has a direction, and proceeds from the past to the future [Haj96]. Temporal events can thus be assimilated to either *time points* or *time intervals*. A time point can be considered as an instantaneous event. A time interval is a continuous event with a start and a different end. The chosen granularity depends on the application domain.

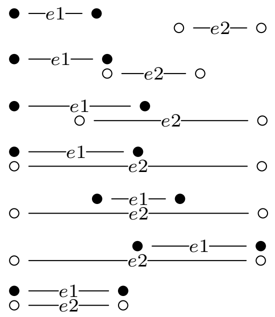
Exhaustive, mutually exclusive qualitative relations are possible among time points and among time intervals of a linear structure, see Figure 3.1 for the latter case—note that such relations can be extended to non-linear structures, see [Haj96]. Other relations are possible between a time point and a time interval [Mei95]: before and its inverse, starts and its inverse, during.

A qualitative approach to time is embedded in TimeML [Tim11]. As for temporal relations, TimeML defines a TLINK tag that links tagged events to other events. Their BNF representation is as follows:

relType ::= BEFORE | AFTER | INCLUDES | IS_INCLUDED |
DURING | DURING_INV | SIMULTANEOUS | IAFTER | IBEFORE |
IDENTITY | BEGINS | ENDS | BEGUN_BY | ENDED_BY

The TLINK relations are based on the atomic Allen relations, according to [MWVP07]. We introduce the Allen relations in Figure 3.1, and then compare their expressive power with that of TLINKs.

In his seminal paper [All83], Allen motivated his time representation as follows: "This representation is designed explicitly to deal with the problem that much of our temporal knowledge is relative, and hence cannot be described by a date (or even a fuzzy date)". In the Allen representation, the only events are time intervals. Between any two pairs of events, there is precisely one *atomic Allen relation*, namely, a relation at of the form before, meets, during, overlaps, starts, during, finishes, equals, or its inverse at^{-1}. See Figure 3.1 for their (standard) interval interpretation, not naming the starting and ending points of intervals.



$e1$ before $e2$, $e2$ before$^{-1}e1$
$e1$ meets $e2$, $e2$ meets$^{-1}e1$
$e1$ overlaps $e2$, $e2$ overlaps$^{-1}e1$
$e1$ starts $e2$, $e2$ starts$^{-1}e1$
$e1$ during $e2$, $e2$ during$^{-1}e1$
$e1$ finishes $e2$, $e2$ finishes$^{-1}e1$
$e1$ equals $e2$, $e2$ equals $e1$
(note that equals^{-1} = equals)

Figure 3.1: The atomic Allen relations.

As Allen arguments, his representation of time allows for "significant imprecision" whenever, as in the TERENCE stories, it is often the case that temporal knowledge is relative without relations to absolute dates. Indefinite information can be represented by means of disjunctions (unions) of the Allen atomic relations through $\vee$. Then an *Allen relation* rel is a disjunction of atomic relations. An example is before $\vee$ meets.

The set of Allen relations forms the *Allen Interval Algebra* (IA) with conjunction (intersection), inverse $^{-1}$ and composition $\bowtie$, e.g., see [LM94].

Note that disjunctions of TLINKS relations are not foreseen in TimeML. This can be rather restrictive when annotating stories for children, due to inherent imprecision of data (e.g., "Early in the afternoon, Perla went to the library") or different text interpretations by the annotators (e.g., knowledge dependent information). Therefore, in this setting, one may need a more expressive language than TLINKS. One could use relations of a subalgebra of the Allen one, say, the *continuous-endpoint subalgebra* (CA), that is computationally tractable—we will specify what we mean by a tractable subalgebra below, after introducing the necessary details. This is a widely investigated subalgebra, already used in several applications of NLP to narratives [van92], that allows for expressing vague information such as before $\vee$ meets and before $\vee$ meets $\vee$ overlaps. The annotation language described in [Bet11] goes into this direction.

Another historically relevant qualitative representation is the one by Freska, see [Fre91]. This is based on the notion of conceptual neighbourhood: relations between semi-intervals are used instead of relations between intervals. However, relations such as before $\vee$ meets seem to be missing in such calculi, and we found such relations in stories for children for expressing information such as "at some point before" no further specified in the text.

In case time points instead of intervals were the primitive entities, then one could resort to the *point algebra* (PA). PA relations are conjunctions of relations between end-points of intervals of the form: (1) $x = y$, (2) $x \leq y$, and (3) $x \neq y$.

This said, the expressive power of the fragment of the annotation language for events and temporal relations described in [Bet11] should be balanced by tractable reasoning tasks that, possibly, can exploit existing automated reasoning tools. We review some of them in the next section.

3.3 Reasoning about Qualitative Temporal Relations via Constraints

The constraint literature has a number of studies on subalgebras of IA, and algorithms for different reasoning tasks. In the remainder of this chapter, we introduce some of such subalgebras, which are relevant for TERENCE, and the related reasoning tasks with their computational complexity, primarily, the so-called consistency checking and deduction tasks [Gen98]. For the entire list of all the maximal tractable subalgebras of IA, we refer the reader [KJJ05]. In the end of this section, we also outline some of the current reasoning tools relevant for TERENCE.

3.3.1 Reasoning Tasks

What we mean by reasoning with qualitative relations is best explained by introducing (binary) constraint problems. In the following, we restrict our presentation to the case of interest, namely, to a subset A of Allen relations. In essence, an *(Allen binary) constraint (satisfaction) problem* P over A is given by

– a finite sequence of variables, $e_1, e_2, \ldots, e_n$, where each e_i represents an event and ranges over a finite collection D_i of intervals of reals,
– and a (binary) constraint $C(i, j) \in A$ for each pair of variables (e_i, e_j) with $0 \leq i < j \leq n$.

This is the classical encoding of a qualitative temporal (satisfiability) problem with Allen relations into a constraint problem, albeit others are possible, e.g., see [Apt03].

In the following, without sacrificing generality, we will assume that there is precisely one constraint for each pair of events e_i and e_j, with $i < j$. In the literature, this means saying that our constraints problems are normalised, see [Apt03].

With a slight abuse of notation, we will write $I_i\, C(i, j) I_j$ if (I_i, I_j) is an interval interpretation (as in Figure 3.1) of one of the atomic disjuncts of $C(i, j)$. A tuple of intervals $(I_1, \ldots, I_n)$ of $D_1 \times \cdots \times D_n$ is a *solution* to the constraint problem P if $I_i\, C(i, j) I_j$ for each $C(i, j)$ of P. More generally, P is *satisfiable* or *consistent* if it has a solution, *unsatisfiable* or *inconsistent* otherwise.

Example 2. *Consider the constraint problem with two events, namely, e_1 and e_2, and constraint $C(1, 2)$ equal to* before $\vee$ meets*. A solution to this constraint problem is the pair $([0, 1], [2, 3])$. Another solution is $([0, 1], [1, 2])$. This problem is thus consistent. The problem with events e_1, e_2 and e_3, constraints $C(1, 2)$ equal to* before $\vee$ meets*, $C(1, 3)$ equal to* after*, and $C(2, 3)$ equal to* before *is inconsistent.*

We will say that $rel \in A$ for (e_i, e_j) is *deduced* if $I_i\, rel\, I_j$ holds for all solutions $(I_1, \ldots, I_n)$ to P. Let DC_{ij} be the set of all the deduced relations for (e_i, e_j). The *deductive closure* of P is the set of all such DC_{ij}, for $0 \leq i < j \leq n$. Clearly, a problem is consistent if and only if its deductive closure is different than the empty set.

Example 3. *Consider the constraint problem with three events, namely, e_1, e_2 and e_3, constraints $C(1, 2)$ and $C(1, 3)$ equal to* before $\vee$ meets*, whereas $C(2, 3)$ is equal to* before*. The deductive closure of this problem has* before $\vee$ meets *for e_1 and e_2, and* before *for e_1 and e_3 as well as for e_2 and e_3.*

If there is a PTIME algorithm that can decide about the satisfiability of any problem over A, then we say that A is a *tractable subalgebra*. In case the tractable subalgebra A contains all the atomic relations, the deductive closure of any problem over A can be computed in PTIME by resorting to the algorithm for A satisfiability [NB95].

For instance, let us consider the CA subalgebra of IA. This is tractable. CA relations can be represented as PA relations of the form (1) and (2). Checking the consistency of a constraint problem over CA can be done in quadratic time in the number of events by means of the algorithm for PA developed in [van92]. Computing the deductive closure of a problem over CA can be done in cubic time in the number of events with the path consistency algorithm, which is a constraint propagation algorithm. Constraint propagation algorithms monotonically search the input problem for the minimum problem that satisfies a so-called local consistency property and has the same solutions as the input problem. In case of the path consistency algorithm, the local consistency property is that, for each triple of distinct events e_i, e_j and e_k, we have

$$C(i, j) = C(i, k) \bowtie C(k, j)$$

where $\bowtie$ is the composition operation over IA. The algorithm then works by enforcing such a property by using the operations of intersection, inverse and composition as shown in Table 3.1, which is the PC-2 algorithm of [Mac77] and that we took from [Apt03].

In turn, this algorithm can be used to decide about the consistency of the maximal tractable subalgebra that contains CA, namely, the ORD-Horn subalgebra [NB95]. Computing the deductive closure of the ORD-Horn subalgebra can be done in time $O(n^5)$ by resorting to the path consistency algorithm, with n equal to the number of events.

However, neither the ORD-Horn subalgebra and, hence, nor CA allow for expressing disjointness, as in "before or after". Notice that Sp and Ep are the only maximal tractable subalgebras that allow for it [KJJ05]: Sp can viewed as the set of relations obtained by replacing each of the basic relations meets, overlaps, during, finishes and their inverses with their disjunction with before; Ep can viewed as the set of relations obtained by replacing each of the basic relations meets, overlaps, during, starts and their inverses with their disjunction with before.

Constraint problems of the aforementioned type are also referred to as *hard* or *classical* constraint problems: a constraint between two events either holds (true) or it does not hold (false). In case one needs to have other truth values, then *soft* or

$V_0 := \{(e_i, e_j, e_k) | i < j\}$;
$V := V_0$;
while $V \neq \emptyset$ **do**
  choose $(e_i, e_j, e_k) \in V$;
  $C(i,j) = C(i,j) \cap \big(C(k,i)^{-1} \bowtie C(k,j)\big)$;
  if $C(i,j)$ changed **then**
    add to V the subset of triples of V_0 that contain e_i and e_j;
  else
    remove (e_i, e_j, e_k) from V;
  fi
od

Table 3.1: A path consistency algorithm

non classical constraint problems come to hand, e.g., see the c-semiring framework of [BMR97]. In brief, each constraint can now be associated to a value (taken from a c-semiring) that is different than 0 (false) and 1 (true). Such a value can represent vagueness, likelihood as well as desirability. In other words, fuzzy constraint problems, probabilistic constraint problems, weighted constraint and constraint problems over conditional preference (CP) nets are all soft constraint problems.

Of particular interest for this deliverable are the probabilistic temporal frameworks of [Taw02] and [RT04], as well as the fuzzy framework of [BG00], which extend the Allen algebra with probabilities and fuzzy values, respectively. In case of c-semiring based constraints, consistency checking and deduction can be performed by combining the soft values associated to the constraints. A general framework for the related constraint propagation algorithms is presented in [BGR03].

Another interesting work for TERENCE is [RVW04] in case we need to represent and reason about the simultaneous preferences of several agents, say the annotators, over the same relations. To aggregate the agents' preferences, one can query each agent in turn and collect together the results. In the aforementioned work, this is formalised as a voting problem and encoded as a constraint problem over CP nets. Consistency checking means now aggregating the votes of all the agents and finding an optimal result. Deducing means finding all the optimal results. Besides returning that the relations are inconsistent, the reasoner can try backtracking to the 'preferred' explanation of the inconsistencies and interact with humans for relaxing inconsistent relations, see [Jun04].

3.3.2 Reasoning Tools

For (hard) temporal constraint problems, nowadays, there are efficient automated reasoning tools. Two well known such tools are: TimeGraph for temporal relations among points; the more general Generic Qualitative Reasoner (GQR) for relations among points or intervals. They are outlined as follows.

Temporal Automated Reasoners for Qualitative Relations. TimeGraph I (TG-I), TimeGraph II (TG-II) and TimeGraph III (TG-III) are three temporal reasoning systems accepting as input PA relations, in which the primitive entities are time points, see [GS95]. In all the three versions of TimeGraph, temporal relations are represented through graphs whose vertices represent points and whose edges represent temporal relations. Given a chain, which is a set of linearly ordered points, a graph-based structure is used to guide the search of a solution across the chain. Given a set of temporal relations S, the main temporal reasoning tasks are determining the consistency of a problem over S and providing the strongest relation entailed by S between (any) two time points of a problem.

TG-I was originally developed in the context of natural language comprehension of narratives and was integrated into a series of knowledge representation and reasoning systems, e.g., EPILOG, a computational system for Episodic Logic, a very expressive NL-like logic. TG-II is an extension of TG-I that generalised the timegraph approach to a wider class of applications including planning and scheduling. Both TG-I and TG-II are written in Common Lisp. TG-III is a C reimplementation of TimeGraph-II with some extensions. The current TG-II's interface allows the assertion of all the relations in PA and of all the 188 interval relations of the related Allen subsets, extended with relations expressing point-interval exclusion and disjointness relations such as "before or after", which are representable through binary disjunctions of inequalities. Only TG-II is currently publicly available.

GQR [GWW08] is a solver for binary qualitative temporal problems. GQR takes a calculus description and one or more qualitative problem as input, and processes the problems using an efficient implementation of the PC-3 path consistency algorithm combined with backtracking. In contrast to specialised reasoners like TimeGraph, it offers reasoning services, like consistency checking and deduction, for arbitrary binary qualitative calculi. New calculi can be added to the system by specifications in a simple text format or in an XML file format. GQR is developed in C++. It is freely usable and distributable under the terms of the GNU General Public License. Its flexibility and open source choice makes it a suitable choice for TERENCE.

Relations in GQR are stored as bit vectors based on templates. The extensive use of templates allows the reasoner to be recompiled from the same source. GQR allows for the precomputations of the composition and inverse operations. It supports both full and partial precomputations. The latter comes in two version and the reasoner automatically chooses which to use, according to given memory limits for composition and converses. Adding new precomputation methods for specific calculi is also possible.

Annotation Tools with Reasoning Capabilities. Annotation tools with some automated reasoning capabilities were also developed in the TimeML community. See also [Bet11]. The most noticeable is TANGO, an annotation tool for TLINKS of TimeML, now no more in use. Since TimeML forbids disjunctions in TLINKS, the deductive-closure algorithm of TANGO is not complete for the composition operation as specified in IA. For instance, the algorithm cannot compute the Allen composition of before and its inverse, since the result is the disjunction of all the Allen atomic relations.

3.4 Visualisation of Events and Qualitative Temporal Relations

The visualisation of time involves representational and perceptual issues as well, and it must be intuitive for the intended end users. In the following, we describe different visualisations of qualitative relations between intervals, which are mainly studies in the HCI literature. We divide tools and visualisation techniques in two classes, according to their possible employment:

1. for the visualisation of a single pair of temporal events and their relations;
2. for the visualisation of temporal events and their relations, with more than two events.

3.4.1 Visualisation of a Single Pair of Events

The classical visualisation between a pair of events is given by their interval interpretation displayed in Figure 3.1. However, this visualisation is at loss when disjunctions of relations must be considered for expressing indefinite information, e.g., “before or immediately before”.

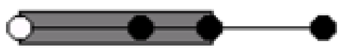


Figure 3.2: A visualisation of a disjunctive relation

The authors of [HR97] propose an interesting solution: the first interval, say A, is represented by a gray bar; the second interval, say B, is represented by a segment bounded by two circles, a white circle for the left end and a dark circle for the right hand. Then interval B can terminate in different positions with respect to the termination of interval A, and this allows for the representation of CA relations or, more generally, PA relations. Figure 3.2 is an example taken from [CC01].

The authors of [CC01] surveys other approaches by considering different aspects in visualising temporal information, such as, the time granularity (points versus intervals), and the capability of expressing disjunctive relations between a pair of intervals or points. According to their survey, time points are usually associated to some graphical objects, such as circles, boxes, or ad-hoc icons; objects are located with reference to a time axis, which is usually represented as a horizontal line. The usual graphical elements for intervals are boxes or lines; temporal location and extent of intervals are displayed with reference to a (possibly implicit) time axis, as for time points.

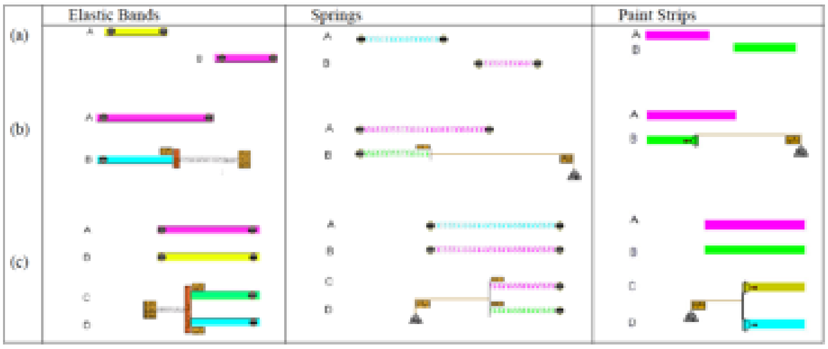


Figure 3.3: Examples of the three concrete visualisations of Allen relations

Then [CC01] proposes three alternative visual metaphors for the representation of intervals and their relations, which can scale up to the visualisation of relations in a network with more than two intervals. Their metaphors are based on concrete objects and phenomena from the physical word: elastic bands, springs and paint strips, see Figure 3.3. In the (a) row, the atomic Allen relation before is represented in those three different manners. In the (b) row, the disjunction of the atomic relations starts^{-1}, equals, starts is represented in the three different manners.

3.4.2 Visualisation of More than One Pair of Events

The three visual metaphors proposed in [CC01] can also render constraint problems with more than two events, and their relations, see the (c) cases of Figure 3.3 with four events, namely, A, B, C, and D. However, such a visualisation does not seem to scale up well to a problem with a number of events, as it is the case for the events and relations found in stories.

The most famous tool for temporally annotating texts with TLINKs, namely TANGO (see Subsection 3.3.2), uses Graphviz [wp] in order to visualise temporal relations among events.

In [Ver05], for improving over the TANGO visualisation, the author introduced T-BOXes: they allow annotators to select a whole group of events, and use only one link to state that every event in this group stands in a specific temporal relation with another even.

See Figure 3.4 for a comparison of the two visualisations.

Other examples are found in the medical fields, where, however, events come with a starting and an ending point. For instance, in the PatternFinder tool [FS06], temporal relations are visualised using rows. Figure 3.5 represents an example of the Pattern visualisation, where each row is a single pattern match for a patient.

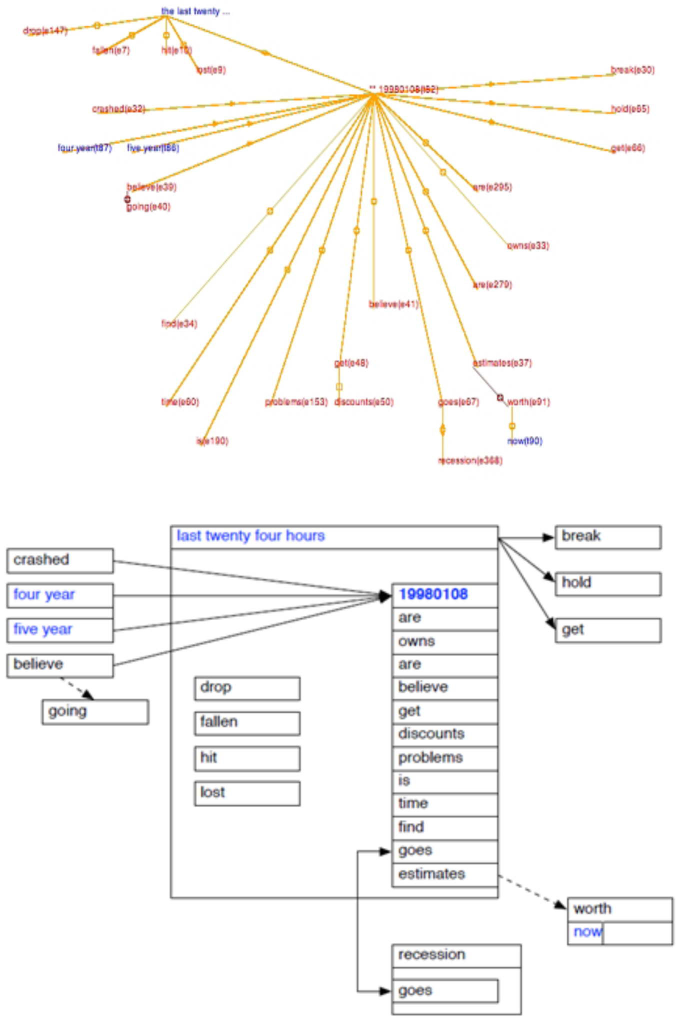


Figure 3.4: The TANGO (left more) and the T-BOX (right more) visualisations.

Figure 3.5: The PatternFinder visualisation

4 Intelligent Feedback Design

The TERENCE system interacts with three main users: educators; experts; leaners. For all the users, the automated reasoning module states whether a game resolution is correct or not. For experts and leaners, the reasoner can provide other types of feedback. Therefore this chapter focuses on the design of the feedback first for experts and then for learners.

The design of the feedback for experts and learners uses the analysis of the state of the art of temporal automated reasoning in Chapter 3 and the design of the TERENCE games in Chapter 2. It is rooted in the field studies of WP1, reported in [SG11], and in on-going work concerning the stimulation plan of WP7. Therefore, following the UCD, the design and development of the feedback may be updated due to the results of these WPs.

4.1 Feedback for Experts

The annotations of a story are useful for generating the TERENCE games as outlined in Subsection 2.2.2. The automated reasoning module of TERENCE will interact with the experts annotating the TERENCE stories. In this process, the annotations of a story, returned by the NLP module, are checked and completed with information that the automated reasoning module can add. See also Figure 4.1.

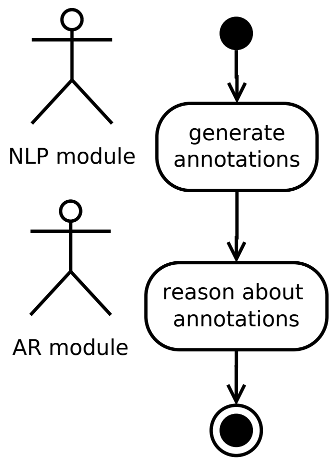


Figure 4.1: The annotation process

The feedback that the reasoner can give to the experts during this process can be the result of classical reasoning tasks like consistency checking and deduction, or non classical reasoning tasks. By selecting the relevant work presented in Chapter 3, we briefly outline them both in the following.

Classical feedback. Checking the consistency of a the TLINKs of a story can be performed by a tool like GQR. This can help in avoiding annotation inconsistencies, likely to arise when comparing the annotators' work and the system. The deduction of new temporal relations from the existing annotations is another type of feedback and reasoning task that can be automated, and thus release the annotations of deducible relations from the annotators. GQR can be used for that as well. Note that the deduced annotations can be disjunctions of relations. Whether disjunction and which disjunctions are allowed depends on the specifications of the TLINKs in [Bet11]. If these form an Allen subalgebra, then they are closed for deduction, and hence the deduction will return TLINKs. Otherwise approximations like in [RL05] or human intervention are required.

Non classical feedback. Another interesting way of supporting the annotation process consists in considering the weights on the annotated temporal relations and reasoning about them. Useful in this respect is the work in the area of soft temporal reasoning like [Taw02] and [RT04]. Moreover, reasoning about inconsistencies can be a useful feature for annotators as well. Besides returning that the annotations are inconsistent, the reasoner can try backtracking to the preferred explanation for the inconsistency and ask the annotator which inconsistent annotations can be relaxed. Interesting work in this respect, following an interactive approach, is in [Jun04].

In order to assess which of the aforementioned reasoning tasks, besides consistency checking and deduction, can give other useful feedback for the annotation process, it is necessary to analyse and specify the requirements of the NLP experts working on the annotations, in particular, whether the reasoning tool should be used as an interactive tool interleaving with the annotation work or not. What visual metaphor to adopt for the annotation process will clearly depend on the annotation language and the performed reasoning tasks.

4.2 Feedback for Learners

In order to understand the feedback for the TERENCE learners, we need to first outline the stimulation plan of TERENCE. This section first presents the stimulation plan and then, within it, the intelligent feedback for the TERENCE learners.

4.2.1 Stimulation Plan

Consistently with what sketched in [Alr11], the learner's activities are: reading stories; playing games. Hereby, we sketch the parts of the stimulation plan, part of the on-going design of the WP7 evaluations, and that are relevant for the intelligent feedback of WP4.

Cycles and sessions of activities. A stimulation is divided into *cycles*. A cycle should last from 2 to 3 months, and should include also a brief suspension; the longer the cycles, the shorter the suspension, the stronger the intervention. Clinical practice suggests a stimulation organised in terms of cycles lasting 10 weeks, with 2 weeks of suspension.

Each week should include two or three *sessions*: the higher the number of sessions in a week, the stronger the intervention. Each session is divided into circa five *sets of reading and playing activities*. A set consists in reading a story of a book in its entirety, and then resolving the correlated games. As for these, first a subset of smart games is proposed. Then the learner plays with relaxing games, which are unrelated to the story and have a relaxing and distracting effect. Then, another subset of smart games is proposed. Alternating games and episodes of the story is likely to badly affect the story's comprehension of the poor comprehenders, who will tend to read the episodes as different unrelated stories.

Measures of activities and performances. Each activity has different types of measures, described as follows.

The reading activity has diverse logged times. The *reading time* is the time spent by a learner in reading a story of a set, whereas the *maximum reading time* of a story is the maximum time allowed for reading the story, independently of the learner. The *average reading time* for a learner is the average of the reading times the learners has spent in reading stories.

With specific regard to playing games, we should at least take care of three different types of measures:

- the *accuracy* ratio for a learner is measured as the number of games correctly solved, divided by the total number of games;
- the *omission* ratio for a learner is measured as the number of unresolved games by the *maximum resolution time*, divided by the total number of games;
- the *resolution time* of a game of a set is the time spent by the learner for resolving the game, whereas the *maximum resolution time* of a game is the maximum time allowed for resolving the game, independently of the learner. The *average resolution time* for a learner is the average of the resolution times the learner takes for resolving games.

Given the aforementioned measures, clinical practice suggests that the *performance* of a learner on games of a set increases if, in order, firstly the omission ratio decreases, secondly the accuracy ratio increases, thirdly the average resolution time decreases, and finally the average reading time decreases. The average reading time can also be considered for measuring the overall performance of a leaner in a set.

Performances and order of smart games. Smart games should address the story's events in the same order in which they are presented in the story. The difficulty level of a story and a type of game in the first set should be the same as the learner's reading comprehension level, or even slightly inferior than this, so as not to frustrate the learner. For the same reason, the system can propose the type of games in which the learner is likely to succeed, and the more demanding ones later on. According to the learner's performance in a set, the system can then increase the difficulty level of the story and games in the subsequent sets.

The first session is likely to be slightly different than the subsequent sessions, because in that session the learners need to acquaint with the system. Accordingly, the first session will mainly consist of the training to the system. For instance, the child will get familiar with the system's interface, and the type of interactions required by the TERENCE games. In particular, the resolution times during the first session and the correctness of the resolutions are likely to depend on the concurrent training with the system. During the other sessions, the resolution times of the session's sets should become independent from this aspect.

If, during a session, the learner makes a significant number of mistakes in resolving the smart games of the story (that is, the learner's accuracy is low or the learner's omission is high), the system:

– shall propose the same story in a simplified version, possibly after further sessions with the system;
– shall propose easier games, or games with other feedback than the correctness of the resolution.

Another important part of the stimulation consists indeed in the feedback, which is the focus of the forthcoming subsection.

4.2.2 Intelligent Feedback

From the GUI point of view, for all the learners and sessions, the intelligent feedback should be rendered through the avatar and the increase in points, see [Alr11]. In the following, we analyse the type of feedback the reasoner can give according to the learner's performance.

Whether the game is resolved correctly or not is a recommended feedback that the reasoner can automatically give to all its learners, independently of their performance. For all the games, this amounts to a consistency checking task, that is, whether the resolution to a game is consistent with the annotations of the story. As outlined in Section 4.1, in case of temporal relations, this can require deducing further relations than those annotated by the NLP module, and then the encoding all of them as possible answers to the game.

Clinical practice suggests to also use other types of intelligent feedback only after the first cycle, in case the learner does not show progresses in his/her performances.

In particular, in case the omission rate of a learner is high during a stimulation cycle, during the subsequent cycle another basic feedback could consist in also giving the right answer after any omission.

In the following, we sketch other types of textual feedback that the avatar may give to leaners with a non increasing performance according to the type of smart game that the leaners play. This is rooted in the work of WP1 and WP7. The textual feedback of the automated reasoning module may later be integrated by a visual feedback, depending on the visual template sketched in Section 2.4. At the time of writing and to the best of our knowledge, there are no studies supporting what visual feedback may be of aid for poor comprehenders on the types of games envisioned in TERENCE. This may become part of an evaluation of WP7.

Games about a Single Event. A textual feedback for games about a single event of the story consists in pointing out the sentence in the story in which the related event mention is narrated, and then highlighting the clue annotations for resolving the game. For instance, in the game asking about the subjects of the game event "One afternoon, Perla finds 'The Fantastic Circus' in the library shelf", a textual feedback consists in pointing out the story's semantic event "One afternoon, she found a new book in the library shelf". A further clue consists in highlighting the clue annotation "she". The subsequent feedback consists in highlighting the nearest resolving expression, "Perla".

Games about Multiple Events. A textual feedback for games about causal or temporal relations between events of the story consists in pointing out the sentences in the story in which the related event mentions are narrated. A textual feedback for games about characters' relations in events is similar to that for the games about causal or temporal relations.

5 Conclusions

The previous chapters analyse the current smart games, some of their difficulty indices, and the tasks of the reasoners in relation to the smart games of TERENCE. This chapter outlines their possible impact on other WPs, mainly, WP3 (annotation of stories) and WP5 (illustrations for stories and games).

5.1 Impact on WP3

Questions can be created and graded in difficulty according to the annotations. To this end, Section 2.2 lists the main desiderata for the annotation language of [Bet11], resulting from the current game design. The main desiderata for the annotation language are recapped in Table 5.1.

Besides such desideta, there are other requirements that depend on the reasoning module. As put forward in Chapter 3, classical reasoners for qualitative temporal relations also deduce non-atomic relations. For this and expressivity reasons, the annotation language of [Bet11] should allow for disjunctions. Another useful feature would be to have weights over disjuncts so as to allow the reasoner to reason about such weights as well.

5.2 Impact on WP5

Hereby, we sum up the desiderata for the illustrations of WP5, resulting from the current game design.

Illustrations. For each story, we need illustrations for

1. the main characters of the story; in relation to the "The Fantastic Circus" story, they are precisely Perla (the girl), Gianna (the librarian), the two acrobats (precisely two as in the story), the two clowns (precisely two as in the story), the public of the circus; the other characters or elements can be there for describing the environment but should be in the background (e.g., the elephant should stay in the background as the public in order to describe the circus environment);
2. the objects of the story (e.g., trapeze, car, working table);
3. the environments; in relation to the "The Fantastic Circus" story, they are precisely the library and the circus.

Such illustrations will be realised in layers so as to be reused for the illustration of the events that become part of the TERENCE games. The layers could be then annotated with the appropriate tags of the annotation language, e.g., the illustration of the subjects of an event can be annotated with the related tag.

Illustrations of characters. Having questions concerning characters of the story, namely, their emotions and their role, WP5 should provide illustrations of the characters of the story along with

1. illustration of their facial expressions that the young and old learners of TERENCE are used to, e.g., see [Ekm73], *a*) neutral *b*) happy *c*) sad *d*) fear *e*) anger *f*) disgust *g*) surprise,
2. variations on their physical characteristics, object of games.

Reasoning about	Main requirements for the annotation language	Reference
event mention in the story	1. the type of event, e.g., factual, physical, emotional 2. only the core factual event of both aspectual and reporting events (e.g., "Perla went" instead of "Perla decided to go"), and that the core factual event stems from an aspectual or reporting event 3. the root form (a.k.a., stem, lemma, infinitive form) of the core event, that is, its verb 4. tense, mode and irregularity of this verb 5. the main arguments of an event, and the role (a.k.a., type) of arguments 6. whether the arguments of an event are characters of the story 7. whether a character is a main character of the story 8. whether the subjects and objects are animated or not 9. whether the location is an environment or not 10. whether an argument is explicit (that is, it is the id of the argument) or implicit (e.g., it is another referential expression) 11. if implicit, the distance from the resolvent	Subs. 2.2.2
relations between events	1. the type of relation, that is, temporal or causal 2. implicit/explicit signal 3. if an explicit temporal signal, the order 4. if the annotated events occur in the same sentence/adjacent sentences of the story, or not	Subs. 2.2.4

Table 5.1: Desiderata for the annotation language, resulting from the current game design.

Envisioned risks. The smart games of TERENCE serve to stimulate and also measure the text comprehension of the TERENCE leaners. According to several experts interviewed for [SG11], poor comprehenders and in particular deaf children tend to rely on images rather than on the text when reading a story. Therefore the illustrations of stories should not become a shortcut for resolving the TERENCE smart games.

In particular, the reconstruction of the temporal flow through the viewing of the story's images should be avoided as much as possible so that the user is compelled to read the text. The same problem was encountered in [CDMG11]. Therein, the authors decided to use illustrations that do not present any visual clue concerning the temporal flow of the stories on purpose. However, since the visual component of the application must be appealing and comply with the standard of printed books for children, where illustrations function as memory-reinforcement and attention-catalysts, the illustration of an episode characterised the actors and the spatial locations of the episode's main events.

Having games concerning emotions also implies that, in the story, the facial expression of the subjects of emotional game events should not be illustrated. E.g., Perla is represented backwards when she laughs because the clowns throw water and sawdust everywhere.

A note of warning is also due for pragmatic events, which are inferred from annotations in the story. As such they can be problematic for TERENCE, e.g.,

- in case their visualisation requires a visual representation different than those of the events of the story,
- in case they question an argument of an event that would be illustrated by the story's illustrations.

Let us given an example for the latter point. Think of a story concerning a guy playing with a ball and scoring the winning goal in the final match[1]. A pragmatic inference-making question is about the game played by the guy, which is football. Now, it is difficult to illustrate the story without conveying the fact that the game played is football.

The envisioned risks suggest that the game events for a story be known before the final versions of the story's illustrations are released.

[1]Personal communication with Jane Oakhill, University of Sussex, 2011.

Appendices

The Fantastic Circus

Perla loved books. One afternoon she found a new book in her library's shelf: it was entitled 'The Fantastic Circus'. Gianna, the librarian, looked at it with sparkling eyes, and recommend it enthusiastically to Perla.

The girl sat at the work table and opened it. She turned the first page of the book and started reading the first lines... Immediately, she found herself in an enormous circus tent. It was all so real!

Two acrobats entered pirouetting on the circus floor. Quickly, they climbed along the ropes until they reached the trapezes, going up, and up, and up... Perla watched them holding her breath. One of the two was dangling from the trapeze with his knees. The other made a double somersault, and the first caught him. Then everybody clapped enthusiastically.

After a short while, there was a big loud BANG! The clowns entered the circus floor on their battered car; the car's doors were falling apart and the engine was chugging. The clowns threw water and sawdust everywhere, going round and round along the circus floor. Perla laughed until she cried. The girl was still laughing when she saw a cake flying over the audience. She ducked just in time to avoid it!

The kid was smiling when she closed the book. She thanked Gianna for the beautiful adventure, and the librarian mysteriously winked at her. When Perla placed the book on the shelf, she saw a small amount of sawdust falling from the book...

Weird, isn't it? Where do you think the sawdust came from?

Bibliography

[All83] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *ACM Comm.*, 26:832–843, 1983.

[Alr11] Mohammad Alrifai. Conceptual Model's Specification. Technical Report D2.1, TERENCE project, 2011.

[Apt03] Krzysztof R. Apt. *Principles of Constraint Programming*. Cambirdge University Press, 2003.

[Bet11] Steven Bethard. State of the Art and Design of Novel Annotation Languages and Technologies. Technical Report D3.1, TERENCE project, 2011.

[BG00] S. Badaloni and M. Giacomin. Flexible Temporal Constraints. In *Proc. of IPMU 2000*, pages 1262—1269, 2000.

[BGR03] S. Bistarelli, R. Gennari, and F. Rossi. General Properties and Termination Conditions for Soft Constraint Propagation. *CONSTRAINTS*, 8(1), 2003.

[BMR97] S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based Constraint Solving and Optimization. *Journal of ACM*, 44(2):201–236, 1997.

[CC01] L. Chittaro and C. Combi. Representation of Temporal Intervals and Relations: Information Visualization Aspects and their Evaluation. In *Proc. of TIME*, pages 13–20. IEEE Press, 2001.

[CDMG11] M. Carlini, T. Di Mascio, and R. Gennari. Reading as Playing: a new Tutoring Multimedia Tool for Children with Text Comprehension Problems. In *Proc. of HCI 2011*. IADIS, 2011.

[DMM11] T. Di Mascio and A. Melonio. Popular games for touchscreen tabletops. 2011.

[Ekm73] Paul Ekman. *Cross-cultural Studies of Facial Expression*. New York: Academic Press, 1973.

[Fre91] C. Freska. Conceptual Neighbourhood and its Role in Spatial and Temporal Reasoning. In *Proc. of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, pages 181–187, 1991.

[FS06] Karlson A. Shahamat L Fails, J.A. and B. Shneiderman. A Visual Interface for Multivariate Temporal Data: Finding Patterns of Events across Multiple Histories. In *Proc. of Visual Analytics Science And Technology, IEEE Symposium*, 2006.

[Gen98] Rosella Gennari. Temporal Constraint Programming: a Survey. *CWI Quarterly Report*, 1998.

[GS95] A. Gerevini and L. Schubert. Efficient Algorithms for Qualitative Reasoning about Time. *Artificial Intelligence*, 74(2):207–248, 1995.

[GWW08] Z. Gantner, M. Westphal, and S. Wölfl. GQR - A Fast Reasoner for Binary Qualitative Constraint Calculi. In *AAAI'08 Workshop on Spatial and Temporal Reasoning*, 2008.

[Haj96] E. Hajnicz. *Time Structures: Formal Description and Algorithmic Representation*. LNCS. Springer, 1996.

[HR97] S. Hibino and E.A. Rundensteiner. User Interface Evaluation of a Direct Manipulation Temporal Visual Query Language. In *Proc. of the ACM Conference on Multimedia*, pages 99–107, 1997.

[Jun04] U. Junker. QUICKXPLAIN: Preferred Explanations and Relaxations for Over-Constrained Problems. In *Proc. of AAAI04*, 2004.

[KJJ05] A. Krokhin, P. Jeavons, and P. Jonsson. Reasoning about Temporal Relations: The Tractable Subalgebras of Allen's Interval Algebra. *Journal of ACM*, 50(5):591–640, 2005.

[LM94] P. Ladkin and R. Maddux. On Binary Constraint Problems. *Journal of ACM*, 41(2):435–469, 1994.

[Mac77] A.K. Mackworth. Consistency in Network of Relations. *Artificial Intelligence*, 8, 1977.

[MB99] C. McKenna and J. Bull. Designing Effective Objective Test Questions: an Introductory Workshop. Technical report, CAA Centre, 1999.

[McA02] Mhairi McAlpine. A Summary of Methods of Item Analysis. Technical report, CAA Centre, 2002.

[Mei95] I. Meiri. Combining Qualitative and Quantitative Constraints in Temporal Reasoning. *Artificial Intelligence*, 1995.

[MWVP07] I. Mani, B. Wellner, M. Verhagen, and J. Pustejovsky. Three Approaches to Learning TLINKS in TimeML. Technical Report CS-07-268, Brandeis University, 2007.

[NB95] N. Nebel and H.-J. Bürckert. Reasoning About Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *Journal of ACM*, 42(1):43–66, 1995.

[Pai91] A. Paivio. Dual-coding Theory: Retrospect and Current Status. *Canadian Journal of Psychology*, 45:255–287, 1991.

[Pas11] M. Pasini. Working document 1.1. Technical report, TERENCE project, 2011.

[RL05] J. Renz and G. Ligozat. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *Proc. of CP 2005*, LNCS, pages 534–548. Springer, 2005.

[RT04] A. Ryabov and A. Trudel. Probabilistic Temporal Interval Networks. In *Proc. of TIME 2004*, 2004.

[RVW04] F. Rossi, K. B. Venable, and T. Walsh. mCP Nets: Representing and Reasoning with Preferences of Multiple Agents. 2004.

[Sch98] Schwalb, E., and Vila, L. Temporal Constraints: a Survey. *CONSTRAINTS*, 1998.

[SG11] Karin Slegers and Rosella Gennari. State of the Art of Methods for the User Analysis and Description of Context of Use. Technical Report D1.1, TERENCE project, 2011.

[Taw02] A.Y. Tawfik. Towards Temporal Reasoning Using Qualitative Probabilities. In *Proc. of FLAIRS-02*, 2002.

[Tim11] TimeML, working group. Retrieved August 2011, from http://www.timeml.org/, 2011.

[van92] van Beek, P. Reasoning about Qualitative Temporal Information. *Artificial Intelligence*, 58(1-3):297—326, 1992.

[Ver05] M. Verhagen. Drawing TimeML relations with TBox. In *Proc. of Annotating, Extracting and Reasoning about Time and Events*, pages 7–28, 2005.

[wp] Graphviz web page. Retrieved August 2011 from `www.graphviz.org/`.

[YO88] N. Yuill and J. Oakhill. Effects of inference awareness training on poor reading comprehension. *Applied Cognitive Psychology*, 2(33), 1988.

[Zwa99] R.A. Zwaan. *Narrative Comprehension, Causality, and Coherence: Essays in Honor of Tom Trabasso*, chapter Five Dimensions of Narrative Comprehension: the Event-indexing Model. Erlbaum, 1999.