

# The TERENCE smart games revision guidelines and software tool

Cofini V<sup>1</sup>, Di Mascio T<sup>2</sup>, Gennari R<sup>3</sup>, and Vittorini P<sup>1</sup>

**Abstract** TERENCE is an FP7 ICT European project, highly multidisciplinary, that is developing an adaptive learning system for supporting poor comprehenders and their educators. The paper introduces the automatic smart games generation process in TERENCE and presents the guidelines for the manual revision as well as the software system supporting it.

**Key words:** Smart games, manual revision, guidelines

## 1 Introduction

TERENCE [10] is an FP7 ICT European project, highly multidisciplinary, that is developing an adaptive learning system for supporting poor comprehenders and their educators. Its learning material are stories and games. The games are specialised into relaxing games, which stimulate visual perception and not story comprehension, and smart games [7], which in TERENCE stimulate inference-making for story comprehension [2].

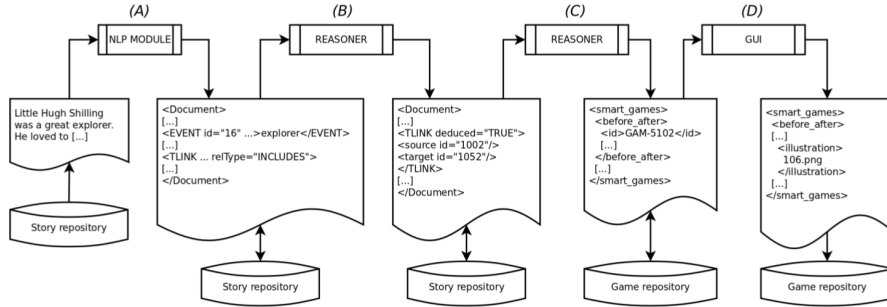
---

<sup>1</sup> Cofini V, Vittorini P  
Dep. of Life, Health and Environmental Sciences, University of L'Aquila, 67100, L'Aquila, Italy  
e-mail: vincenza.cofini@cc.univaq.it  
e-mail: pierpaolo.vittorini@univaq.it

<sup>2</sup> Di Mascio T  
Dep. of Engineering, Information Science and Mathematics, University of L'Aquila, 67100, L'Aquila, Italy  
e-mail: tania.dimascio@univaq.it

<sup>3</sup> Gennari R  
Faculty of Computer Science, Free University of Bozen-Bolzano (FUB), 39100 Bolzano, Italy  
e-mail: gennari@inf.unibz.it

The smart games stimulates the comprehension in terms of: the story characters (who games), the events that take place in the story (what games), the temporal ordering of such events (before/after, while/after, before/while, before/while/after games), and the causal relations (cause, effect, cause/effect games) [9, 3]. The adopted game framework is a generalisation of the EMAPPS one [5].



**Fig. 1** The smart games generation workflow

Fig. 1 summarises the smart games generation process. The process is made up of the four phases described below:

**Phase A.** Stories are stored as flat text in a story repository. From a story text, an NLP module generates an annotated story. The annotations follow a variant of the TimeML language [11], that was extended in [8] so to include new tags useful for the TERENCE smart games. For instance, the ENTITY and CLINK tags aim, respectively, at representing the entity related to an event (the EVENT tag is already present in the TimeML standard), and the causal relation between two events. After the annotation process, the annotated story is stored in the same repository;

**Phase B.** The output of the annotation module is initially checked by a reasoner in terms of consistency of the annotations, i.e., by detecting any eventual temporal inconsistency (e.g., if a relation between two events  $E_1$  and  $E_2$  is that  $E_1$  is before  $E_2$ , there cannot exist other relations stating that  $E_1$  is after  $E_2$ ). If inconsistencies are found, the annotations cannot be stored<sup>1</sup>. Otherwise, the annotations are enriched by adding deduced temporal relations as further TLINK tags [6]. This new consistent and enriched story is also stored in the story repository;

**Phase C.** Starting from the consistent and enriched story, the reasoner module generates automatically instances of smart games. For instance, for a WHO-game (i.e., who is doing something in the story) related to a certain event, hereafter called *fixed event*:

<sup>1</sup> Techniques of relaxation are under development, that will lead to a “self-reparation” process concerning the inconsistent annotations.

- the ENTITY that participates in the fixed event with a role of protagonist is selected as the correct answer;
- other two entities that are not related to the fixed event are added as wrong answers;
- the question that must be asked to the learner is generated through a text-generation module (e.g. if the fixed event is that “Ernesta is riding<sup>2</sup> a bike”, the question will be “Who is riding a bike?”).

The resulting games are then stored in the game repository;

**Phase D.** Finally, a manual revision of the generated smart game instances takes place through a dedicated GUI (Sec. 2), by following an ad-hoc written set of guidelines (Sec. 3).

## 2 GUI for manual revision

The GUI is designed with the twofold aim of

1. providing the context (i.e. the story from which the games were generated) to the experts during the revision;
2. keeping a link from the game under revision and the location (in the story text) of both the fixed event and choices.

Accordingly, the GUI is designed as follows: the story text is shown on leftmost area of the screen, and the interface where the revisions actually take place is placed on the right (as of point 1 above). Moreover, when either the fixed event or the choices are focused, the position of the related event/entity is highlighted in the story text (as of point 2 above).

Fig. 2 depicts the current interface structure. Few notes about the part of the interface concerning the actual smart games revision follow. The interface shows the game type on top and enables to change the game question (only for who-games). Right below (in a greyed box), it is possible to change the fixed event (on the right the corresponding word is highlighting, see dotted arrow) as well as its text. Furthermore, on the bottom, the choices are listed. For each choice, the right/wrong events/entities, the related text, and the correctness of the choice can be modified.

To enable such a kind of flexibility, the GUI is developed as a web application written by using the jQuery library<sup>3</sup>. Furthermore, the smart games are taken from a repository organised in terms of books/stories/versions, placed on a server under revision control, and – when finalised – stored into the story repository.

---

<sup>2</sup> The verb “to ride” will be annotated as event.

<sup>3</sup> jQuery is JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

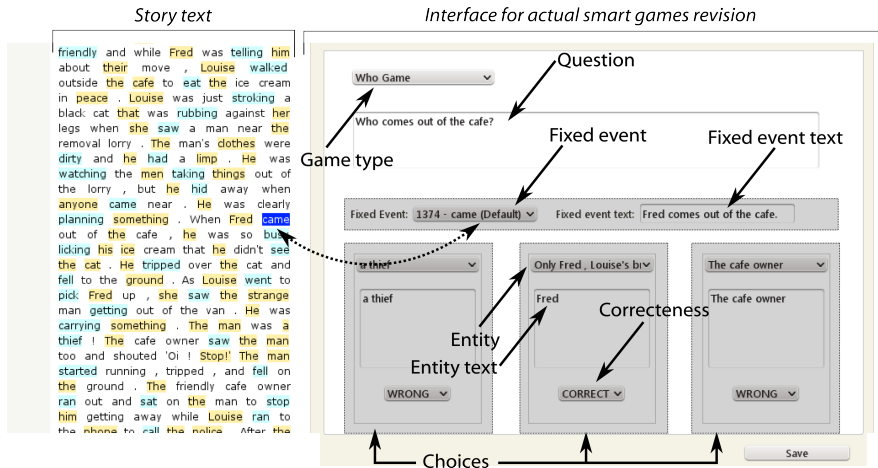


Fig. 2 GUI structure

### 3 Guidelines for manual revision

TERENCE provides stories rewritten into four different levels of increasing difficulty. Level 1 is written for deaf learners, level 2 for skilled deaf learners, level 3 for poor comprehenders, and level 4 for skilled learners [1]. However, the smart games are all generated from story level 1, so to be comprehensible to all our types of learners. Accordingly, also the guidelines detailed below are written so that the resulting smart games shall at least suit the deaf audience (e.g. by preferring short sentences, possibly without subordinates and coordinates, possibly minimising the referential expressions especially if complex).

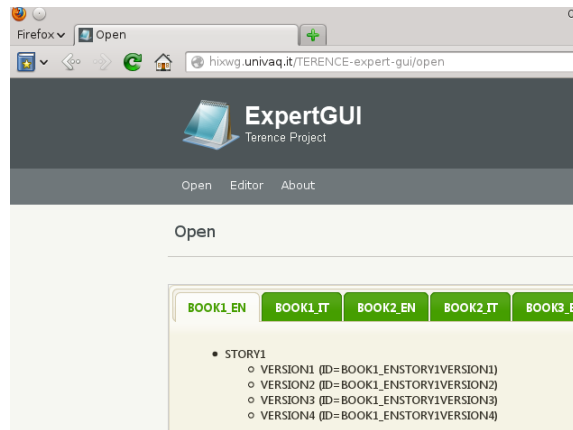


Fig. 3 Smart games selection

The last updated GUI version is available at <http://hixwg.univaq.it/TERENCE-expert-gui/>. In order to load the games for a story, the reviewer must act as follows:

1. first, click on “Open”, choose a book from the tab menu in your language (EN, IT), then click on VERSION 1 (only) of a story <sup>4</sup> (Fig. 3);
2. then, the “Editor” link actually opens the editor (Fig. 2), that enables the following three main functionalities:
  - a. revise existing games;
  - b. create new smart games;
  - c. delete a game.

Each is explained in details below.

### 3.1 Revision

*Task 1.* Revise the sentences according to the following general guidelines, whenever possible:

1. the sentence must be related to the selected event;
2. prefer short sentences, possibly without subordinates and coordinates, possibly minimising the referential expressions especially if complex (example: “drag the two cards below those over them”);
3. add
  - a. location, or
  - b. a temporal/causal/final subordinate.

though only when (a) or (b) above are mandatory for identifying unequivocally the event described in the sentence. For example: by given the sentence “Victor approaches Ugolino and Pietro”, it should be revised as “Victor approaches Little Hugh and Peter before bringing them the cake” in order to distinguish the event “approaching before bringing the cake” from the event “approaching for bringing the cake” (the two “approaches” take place in different moments of the story);

4. prefer present tense, active form;
5. substitute direct speech with indirect speech. Example: “Giovanni says: “drink the milk!”” should be rewritten as “Giovanni says to drink the milk”;
6. report the revision in a spreadsheet file.

*Task 2.* If the game is wrong (e.g., the solution chosen as BEFORE is wrong), proceed as follows, whenever possible:

1. select through the drop down list, for revision

<sup>4</sup> In TERENCE, the stories are available in different levels of difficulties. Version 1 is the easiest, while Version 4 is the hardest to comprehend. The smart games in TERENCE are always generated from the simplest story version.

- a. the entity, e.g., characters, that is right or wrong, in case of who games;
- b. the event, that is right or wrong, in case of the other games.

by always preferring

- a. characters to other entities, in case of who games,
  - b. action verbs than description verbs (better “do, eat, knock, bring” than “say that, tell that, seems that”...) in case of other games;
  - c. events that have a duration span in the story that is rather clear to the child.
2. revise the sentence possibly in the same way (same meaning) in the different games of the story;
  3. report the new choice in a spreadsheet file.

*Subtask 2.1.* In case of WHO games,

1. revise the CORRECT solution by taking it from the drop-down menu that comes before “Other”; if the correct entity (e.g., Ben) is missing in the menu, use on its behalf another entity (e.g., “child”) that does not appear in another WHO game;
2. revise the WRONG solution by taking it from the part that starts with “Other” in the drop-down menu.

*Subtask 2.2.* In case of all the other smart games, revise a WRONG solution as follows:

1. if possible, keep the same event, otherwise select a WRONG event from the same story that must be different than
  - a. the correct solution and the other wrong solution in case of WHAT games,
  - b. the BEFORE, AFTER, WHILE solutions in the time games,
  - c. the CAUSE, EFFECT solutions in the causality games.
2. rewrite the related sentence by changing the subject/object. For instance, let us suppose that in a WHAT game the expert GUI displays
  - the event “bring” with sentence “Victor brings the cake” as WRONG (however, the event actually takes place in the story, therefore it should not be marked as WRONG),
  - the event “put” with sentence “Little Hugh put the fork into his mouth” as CORRECT,
  - the event “wear” with sentence “Little Hugh wears spectacles” as the other WRONG solution (and this is indeed wrong, as it happens in other story).

In such a case, the reviewer can:

- a. select the event “sit” with sentence “Little Hugh and Peter sit at the table” (that does not occur in another WHAT game for the chosen story and is different than the correct solution and the other wrong solution for this game),
- b. rewrite the sentence as “Victor sits at the table” (this is an event that does NOT happen in the story).

Similarly we proceed in temporal/causal games for WRONG events so as to make them plausible wrong events and minimise the work of illustrators.

*Task 3.* Save using the SAVE button.

### 3.2 Creation

*Task 1.* In order to create a new game, click on the ADD NEW button.

*Task 2.* Select the game type.

*Task 3.* Insert a new game id as follows: *bookX-storyY-versionI-new-N*, where *X* is the book number, *Y* is the story number and *N* is the number for the new game:  $N=1$  if it is the first new game,  $N=2$  if it is the second new game, etc.

*Task 4.* Proceed as of “Revision” (Subsec. 3.1).

### 3.3 Delete

Avoid to delete a game. Try to revise it, if necessary, explain the reasons for deletion in the note field.

*Task 1.* Use the DELETE button to delete a game.

*Task 2.* Create a new game of the same type (e.g., WHAT) as in “Creation” (Subsec. 3.2).

## 4 Conclusion

The paper summarised the software and methodological basis for the manual revision process that takes place after the automatic generation process. The paper introduces the rationale behind the guidelines that were used by the staff that took care of actually revising the smart games, whose results are reported in [4].

**Acknowledgements** This work was supported by the TERENCE project, funded by the EC through FP7 for RTD, ICT-2009.4.2.

**Special thanks to the developer:** A. Paolucci.

## References

1. Arfé, B.: Deliverable 2.2: Repository of Stories. Tech. Rep. D2.2, TERENCE project (2012)
2. Cofini, V., Di Giacomo, D., Di Mascio, T., Vittorini, P.: Evaluation Plan of TERENCE: when the User-centred Design Meets the Evidence-based Approach. In: Proc. of the ebTEL 2012 workshop co-located with PAAMS 2012. Springer (2012)

3. Cofini, V., de la Prieta, F., Di Mascio, T., Gennari, R., Vittorini, P.: Design Smart Games with requirements, generate them with a Click, and revise them with a GUIs. *Advances in Distributed Computing and Artificial Intelligence Journal* 1(3), 59–68 (2012)
4. Cofini, V., Gennari, R., Vittorini, P.: The manual revision of the TERENCE Italian smart games. In: *Proc. of the ebTEL 2013 workshop co-located with PAAMS 2013*. Springer (2013), submitted
5. EMAPPS consortium: EMAPPS Game Framework. Retrieved January 2012 from <http://emapps.info/eng/Games-Toolkit/Teachers-Toolkit/Games-Creation/Framework-for-Game-Design>
6. Gennari, R.: Deliverable 4.1: M.State of the Art and Design of Novel Intelligent Feedback Deliverable 4.1. Tech. rep., TERENCE project (2011)
7. Hedberg, S.R.: Executive Insight: Smart Games: Beyond the Deep Blue Horizon. *IEEE Expert* 12(4), 15–18 (1997)
8. Moens, S.: Deliverable 3.1: State of the Art and Design of Novel Annotation Languages and Technologies. Tech. Rep. D3.1, TERENCE project (2012)
9. Pintado, F.D.L.P., Mascio, T.D., Gennari, R., Marenzi, I., Vittorini, P.: The TERENCE Smart Games: Automatic Generation and Supporting Architecture. In: *1st International Workshop on Pedagogically-driven Serious Games* (2012-09-01)
10. TERENCE Consortium: TERENCE web site, <http://www.terenceproject.eu>
11. TimeML working group: Retrieved August 2011 from <http://www.timeml.org/> (2011)