Playing for Improving the Reading Comprehension Skills of Primary School Poor Comprehenders

Fernando De la Prieta[1], Tania Di Mascio[2], Rosella Gennari[3], Ivana Marenzi[4], and Pierpaolo Vittorini[5]

[1] U. of Salamanca, Department of Computer Science, Spain, fer@usal.es
[2] U. of L'Aquila, DISIM, Italy, tania.dimascio@univaq.it
[3] Free U. of Bozen-Bolzano, Fac. of CS, Italy, gennari@inf.unibz.it
[4] University of Hanover, L3S, Germany, marenzi@l3s.de
[5] U. of L'Aquila, MeSVA, Italy, pierpaolo.vittorini@cc.univaq.it

Abstract. TERENCE 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, explicitly designed for classes of primary schools poor comprehenders, where classes were created via an extensive analysis of the context of use and user requirements. Its learning tasks are reading stories and playing with games. The games are specialised into smart games, which stimulate inference-making for story comprehension, and relaxing games, which stimulate visual perception and which train the interaction with devices (e.g., PC and tablet PC). The design of the reading and playing tasks is mainly based on the requirements resulting from the study of the context of use, which is made via field studies and expert-based inquiries. In this paper we focus on how we used the pedagogical underpinnings and the acquired requirements to design the games of the system.

Keywords: Formalizations of pedagogical theories, serious games, game frameworks

1 Introduction

More and more young children turn out to be poor (text) comprehenders: they demonstrate text comprehension difficulties related to inference-making skills, despite proficiency in word decoding and other low-level cognitive skills. Deep text comprehension skills develop from the age of 7–8 until the age of 11, when children develop as independent readers. Nowadays, there are several pencil-and-paper reading strategies for improving text reading comprehension, and specifically addressed to poor comprehenders, which could be delivered by an *adaptive learning system* (ALS), that is, a suite of functionalities designed to deliver, track, report on and manage learning content for specific learners [8,9].

TERENCE [19] is a EU project that aims at delivering the first ALS for enhancing the reading comprehension of poor comprehenders, building upon effective pencil-and-paper reading strategies, and framing them into a playful and stimulating environment. Learners are primary school poor comprehenders, hearing and deaf, older than 7.

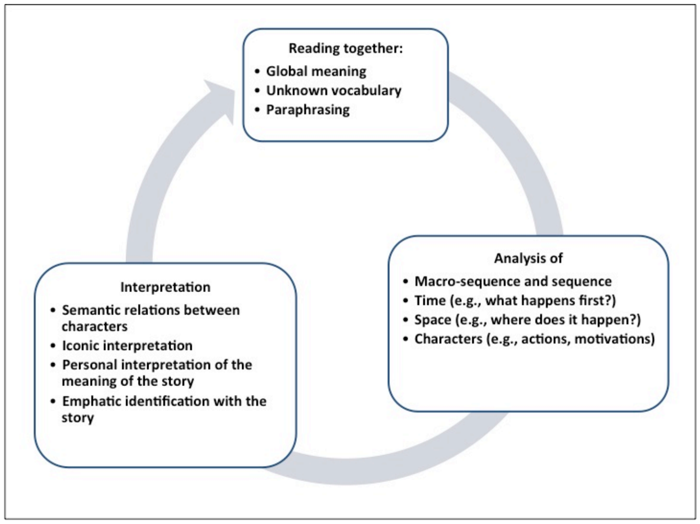

Fig. 1. The pedagogical hermeneutic cycle.

The goal of this paper is to explain how the playing material and tasks of TERENCE are designed and developed on top of an extensive analysis of the requirements of the TERENCE learners.

First, the paper sets the groundwork by presenting the pedagogical theory and approach followed in TERENCE in Sec. 2. Then it outlines the types of data gathered for characterising the TERENCE learners and the analysed effective reading strategies and interventions for the TERENCE learners in Sec. 3. Sec. 4 explains how the design and development of the TERENCE games, in particular, stems from such a knowledge. For space limitations, we focus on the playing material, that is, games and playing tasks.

For information concerning the reading material and tasks, see [6]. Moreover, the models for the learning material and learners of the system are described in [4], how the user centred design was used for them is in [2], whereas some of the adaptation rules are outlined in [5]. The game design for all the TERENCE games is in [3] and, finally, the architecture for games and their automatic generation is outlined in [10].

2 The Pedagogical Underpinnings

The theoretical framework underpinning TERENCE is grounded on the constructivistic pedagogical approach [15]. A complementary relationship exists between technology and constructivism, the implementation of each one benefiting the other. Constructivism states that learning takes place in contexts, while technology refers to the designs and environments that engage learners. This approach is committed to the general view that (1) learning is an active process of constructing rather than acquiring knowledge, and (2) instruction is a process of supporting that construction rather than communicating knowledge [11]. Constructivism is a theory of learning that focuses on students being engaged in "doing", rather than passively engaged in "receiving" knowledge. It rests on the idea that learners, armed with suitable strategies, must construct their knowledge through his or her experiences, rather than obtain it passively from the educator. Furthermore, knowledge does not simply arise from experience, but is build through experience over the current knowledge structures.

The educator is required to orchestrate all the resources needed and must guide students in training them how to teach themselves [17]. Scaffolding is offered to the learner as an adequate environment where to find adequate learning material, compelling learning tasks, templates, and guide for the development of cognitive skills [21]. The focus is shifting from the educator directed instruction to a learner centered approach: the learner is at the centre of the learning process.

This yields that the learning material and tasks should be adequate for each learner profile, and that the learner should be guided through the material and tasks so as to achieve the learning goal. The learning goal is to enhance the reading comprehension of poor comprehenders. In order to do so, the TERENCE system has being developed with the *user centred design* (UCD) [16], by involving a relevant number of real learners in the project. The context of use was thoughtfully analysed and specified for characterising the users of TERENCE, and hence stirring the design of the entire system. In this manner, the learning material and tasks get designed so as to be adequate to the real TERENCE learners, and are framed within a pedagogical plan that so serves to guide the TERENCE learning process.

The TERENCE learning material is made of stories and games for primary school children. Both smart and relaxing games are effective to provide a playful environment. When learning takes place in a playful environment, learning involves the learner actively and it increases his or her motivation and engagement. Smart games, in particular, challenge the learner to reason about a story event (or story events), stimulating active knowledge development.

3 Characterisation of Learners for Playing Tasks

By using the UCD, we extensively and deeply analysed the context of use and the learners requirements, thereby specifying classes of learners for the system. The learning material and tasks of the system are designed for those classes of learners. The first part of this section outlines the type of data collected and analysed for specifying the classes of users for the system. The second part outlines the type of data collected and analysed for designing the learning material and the learning tasks.

3.1 Classes of Learners

The types of learners in TERENCE are deaf and hearing learners, distinguished upon their knowledge in relation to the specific learning goal at the start of the project. The TERENCE classes of users refine the types of users on the basis of the results of the analysis of data for the context of use and user requirements. Such data have been gathered via a mix of expert-based method inquiries (e.g., interviews with primary school educators) and user-based method inquiries (e.g., field studies with primary school children by making them play). The learners involved were about 300 in Italy and about 300 in the UK; the educators involved were about 50 in Italy and about 30 in the UK.

Learners are currently represented by five classes in Italy and four classes in UK, see [14] for details. The most significant features related to the characteristics of the users and considered for deriving the TERENCE classes are:

(a) biographical information such as the level of *reading comprehension* (RC), the level of deafness, and the gender;
(b) personality traits such as the management of frustration;
(c) usage of technology, like the preference for certain types of avatars.

All the classes and the features used for deriving the TERENCE classes were then specified using personas, which are explained in details in [14] and outlined in [2].

3.2 Playing Tasks

The evidence-based practice of the experts responsible for the pedagogical plan requires three main learning tasks in relation to the learning material of the system: (i) reading stories, (ii) playing with smart games for stimulating inference-making about stories, and (iii) playing with relaxing games for relaxing and motivating the learners. The pedagogical goal of relaxing games is to stimulate visuo-perceptual skills that the TERENCE learners are likely to have problem with, according to [14], and to train the TERENCE learners to a specific type of interaction required by a TERENCE smart game. The main pedagogical goal of the smart games is to stimulate the recall and the correlation of the information acquired while reading a story.

All data for the game requirements have been gathered through UCD methods, the results of which are reported in [18] as tasks. In particular, the data for relaxing games are popular causal video games, such as memo, which the TERENCE learners are likely to be familiar with. A casual game is a video-game meant for casual gamers who come across the game and can get into the gameplay almost immediately. This means that the causal game has usually simple rules that are easy to master, and usually it can be played everywhere, anytime and with any device. The data for smart games are mainly diverse reading strategies by pedagogy experts working as therapists with poor comprehenders, cognitive psychologists or educators. More precisely, the main data collected were:

(a) paper-and-pencil inference-making question-answering, with or without picture aids, by cognitive psychologists working on the diagnosis of poor comprehension;
(b) paper-and-pencil puzzle-like games, much relying on visual stimuli, by therapy and pedagogy experts;
(c) diverse strategies of the educators, e.g., analysis of texts in class, drama exercises for stimulating the empathy of the learners with the characters of the story.

The strategies of the educators can be framed in the three stages of the hermeneutic cycle explained in [20] and outlined in Fig. 1. In particular, the explanatory stage can be broken down into the following reading interventions, done in class, mainly using question-answering and drawing:

1. the entire text is discussed with the learners, analysing the vocabulary unknown to the learners and paraphrasing the text;
2. the story is broken down into a sequence of episodes, if possible referring to the story grammar, that is, the story setting, the initiating episode, the culminating episode, the resolving episode, and the final episode;
3. finally, the time, the space and the characters of the story episodes are analysed together.

All the aforementioned interventions were considered for writing the requirements for the TERENCE game design. Constraints of the project triggered a first prioritisation of the requirements. This first sieve left out, for instance, drama exercises or other interventions meant at stimulating the empathy of the learners with the story characters. The remaining interventions refer to the analysis stage of the hermeneutic cycle, with visual aids. They were selected mainly for their expected efficacy for the pedagogy plan, according to the available empirical evidence: they should guide the child to better recall and correlate the information acquired reading the story via adequate visual representations. More precisely, the TERENCE games should propose to reason about the characters and and their participation in the stories; other types of game should propose to reason about temporal relations between events, and more demanding games should propose to reason about causal-temporal relations between events.

The effective interventions relevant for the TERENCE design have thus been hierarchically organised in levels according to their main pedagogical goal:

(1) *time*: interventions for reasoning about temporal relations between events of the story, purely sequential (before-after) or not (all the others);
(2) *causality*: interventions concerning causal-temporal relations between events of the story, namely, the cause of a given event (cause), the effect (effect), or the cause-effect relations between two events (cause-effect);
(3) *characters*: interventions concerning characters, namely, who is the agent of a story event (who), what does a character in the story (what).

The TERENCE smart games were then layered into similar levels, that is, smart games at the entry level for reasoning about characters, games at the intermediate level for time, and games at the top level concerning causality. The following section delves into how the design and development of the smart and relaxing games is carried out via the TERENCE framework.

4 The Design and Development of Games via the TERENCE Framework

According to the game design guidelines presented in [1], the gameplay should detail the following data: the instructions and the overall goal of the game, the initial state of the game, the termination state, the legal actions of the players, and the maximal duration time per action if foreseen. For specifying the gameplay of the TERENCE games we analysed the data for the gameplay of each TERENCE game, then we abstracted the common characteristics in the TERENCE game framework presented in Table 1 and described in Sec. 4.1. The framework serves to specify in a structured manner the above data for the gameplay of the TERENCE smart and relaxing games, essentially, through a timed transition system, with states of the system, and transitions labelled by the player's actions and time constraints. In the following, we first present the framework, and then we sketch how it is used for developing a prototype of a smart game.

4.1 The Framework

Given its aim, the TERENCE framework is less general than other frameworks like [12] and, clearly, less general purpose than game patterns like [13]. Therefore it better lends itself to implementation of single-player casual and puzzle games, where time sets constraints on the players' actions.

Name	name of the game	
Instructions	instructions concerning the game, for the learner: specific to the game instance; motivational; concerning the rules	
Choices	the choices available to the learner; their availability is state dependent	
Solutions	**correct**	**wrong**
	which choices are correct solutions	which choices are wrong solutions
Consistency f.	**correct**	**wrong**
	a yes-message for correct solutions	a no-message for wrong solutions
Explanatory f.	**for correct**	**for wrong**
	explanatory message for correct solutions	explanatory message for wrong solutions
Solution f.	a message consisting in the correct solution	
Smart points (e.g., coins)	$K.P(\theta)$, where θ is the underlying ability of the learner for the game, and K is a constant ranging over natural numbers	
Relaxing points (e.g., stars)	M, that is, a natural number from 1 up to N	
Avatar	the states of the avatar	
Time	resolution time t_r	
Rules	the rules for the game mechanics, specifying the states of the system, the learners' actions and the transitions from state to state through the learner's actions	

Table 1. The TERENCE game framework

The *instructions* for the game are: questions specific to the game instance; of motivational type and usually related to the learner avatar; concerning the rules.

The available *choices* may change from state to state of the game: at the beginning all the choices are available; when the play starts, some choices may become unavailable. The *solutions* for the game list the choices or their combination that form a correct solution to the game (correct), and those that do not (wrong).

The feedback for the game is specialised into a *consistency* feedback (yes, no), an *explanatory* feedback for finding a correct solution (for correct) or for spotting what is wrong in the current solution (for wrong), and a *solution* feedback (the correct solution).

Smart points are the points a learner with a specific reading comprehension level can gain in a smart game. These points can be calculated using the IRT [7], so that the more difficult a game is (assessed to be) for a learner, the more points the learner can gain in resolving correctly that game. Relaxing games have *relaxing points* instead of smart points. Relaxing points should be easy to cumulate, so as to motivate the learner to keep on playing and, in so doing, earning attributes for the avatar.

The states of the *avatar* in the gameplay are of two kinds: happiness for the correct solution, disappointment for the wrong solution. The resolution *time* is a constant.

Now, like points, rules are different for smart games and relaxing games.

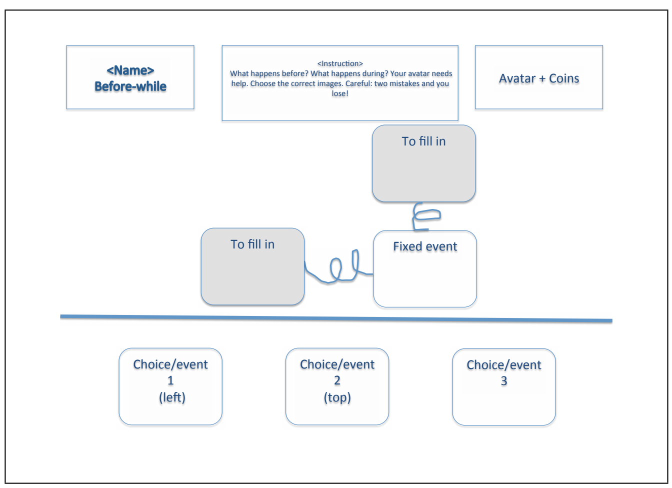

Fig. 2. The visual template of the before-while smart game.

Smart rules. At a high level, smart games all have the same *rules* imposed by the pedagogical plan. In other words, the pedagogical plan establishes requirements for the actions that the learner can take, the states the system can be in, and constraints on them. In the following, we sketch the actions, the states and the constraints for smart games.

- *Actions.* The pedagogical plan sets that the learner should be allowed to choose no solution, choose a correct solution or choose wrong solutions. This means that the main actions the learner can take are as follows:
  - `no_solution`, that is, the learner chooses no solutions or no exit options;
  - `wrong`, the learner chooses the wrong solution;
  - `correct`, the learner chooses the correct solution;
  - `skip`, the learner chooses an exit option.

  The allowed exit options depend on the pedagogical plan, e.g., the learner can choose another story.
- *Constraints.* The pedagogical plan sets constraints. The pedagogical plan sets that the learner should be allowed to choose a wrong solution until the correct solution "becomes obvious". This means that the probability of guessing a correct solution for the game sets the maximum number of attempts that learners have at their disposal for choosing wrong solutions in the game. We refer to this as the *wrong attempts' limit*. The pedagogical plan also sets temporal constraints on playing with smart games, and hence the following time constant: the game resolution time, that is, how long the learner can spend on the smart game instance.
- *States*. The plan also recommends diverse types of feedback if the learner makes a wrong choice and still the learner can play the game: first, a no-consistency feedback for signaling that the solution is wrong, and then an explanatory feedback are given. Finally, the plan suggests a solution feedback, that is, it displays the solution in case the learner chooses no solution within the resolution time or the number of wrong solutions overcomes the wrong attempts limit. Given all this, the main states the system can be in are as follows:
  - the initial state, in which the learner score s and resolution time t are set to 0, the smart points for the learner are computed as a function of the learner ability in the game, all the choices are set as available, and the number of wrong answers is set to 0;
  - a terminal state reachable via a `correct` action, in which a yes-consistency feedback is given, the score is displayed and the avatar is in the happy status;
  - a terminal state reachable via a `skip` action, in which the solution feedback is given, the null score is displayed and the avatar is in the displeased status;
  - a state, reachable via a `wrong` action, in which a no-consistency feedback is given, an explanatory feedback is given, the set of available choices is updated, and the number of wrong answers is updated;
  - a terminal state reachable via a `wrong` action, in which the no-consistency feedback is given, the solution feedback is given, the null score is displayed and the avatar is in the displeased status.

Relaxing rules. At a high level, relaxing games all have the same rules as well, based on common rules for casual games found in the literature. In the initial state, the score and resolution time are set to 0. From any non-terminal state, we can have the following: let N be the number of relaxing points that can be cumulated in a relaxing game:

1. if the score is less than N and, within the game's resolution time,
   - the learner chooses a correct solution, then the system shows the yes-consistency feedback, and the score gets increased by 1,

– but, if the learner chooses a wrong solution, then the system shows the no-consistency feedback, the game terminates and the system shows the disappointed avatar;

2. otherwise, the system terminates the game, shows the score and the happy avatar.

4.2 A Prototype

The development of the prototypes of smart games, like the one in Fig. 3, relied on the TERENCE framework as follows. Firstly, the TERENCE game framework is instantiated for a specific level of games, like before-while games. Then a visual template is realised, e.g., see Fig. 2, and built on top of the resulting framework. Finally, the prototype is developed by means of the visual template. The development procedure, from the framework via the visual template to the prototypes, is reported in [3].

Fig. 3. An instance of a prototype of a before-while smart game.

5 Conclusions

In this paper we explained how the playing material and tasks of TERENCE are designed and developed on top of an extensive analysis of the requirements of the TERENCE learners.

Acknowledgments. The authors' work was supported by TERENCE project, funded by the EC through the FP7 for RTD, Strategic Objective ICT-2009.4.2, ICT, TEL. The contents of the paper reflects only the authors' view and the EC is not liable for it. Gennari work was also funded through the CRESCO and DARE projects, financed by LUB and the Province of Bozen-Bolzano.

References

1. Adams, E.: Fundamentals of Game Design. New Riders (2010)
2. Alrifai, M., de la Prieta, F., Di Mascio, T., Gennari, R., Melonio, A., Vittorini, P.: The Learners' User Classes in the TERENCE Adaptive Learning System. In: Proc. of the ICALT2012 (2012)
3. Alrifai, M., Gennari, R.: Deliverable 2.3: Game Design. Tech. Rep. D2.3, TERENCE project (2012)
4. Alrifai, M., Gennari, R., Tifrea, O., Vittorini, P.: The User and Domain Models of the TERENCE Adaptive Learning System. In: Vittorini, P., Gennari, R., Marenzi, I., De la Prieta, F., Corchado, J.M. (eds.) International Workshop on evidenced-based Technology Enhanced Learning. Advances in Intelligent and Soft Computing series, Vol. 151, pp. 83-90, (2012)
5. Alrifai, M., Gennari, R., Vittorini, P.: Adapting with Evidence: the Adaptive Model and the Stimulation Plan of TERENCE. In: Vittorini, P., Gennari, R., Marenzi, I., De la Prieta, F., Corchado, J.M. (eds.) International Workshop on evidenced-based Technology Enhanced Learning. Advances in Intelligent and Soft Computing series, Vol. 151, pp. 75-82, Springer (2012)
6. Arfé, B.: Deliverable 2.2: Repository of Stories. Tech. Rep. D2.2, TERENCE project (2012)
7. Baker, F., Seock-Ho, K.: Item Response Theory. Dekker Media (2012)
8. Brusilovsky, P.: Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education. In: Lecture Notes In Computer Science,. 5th international Conference on intelligent Tutoring Systems (2000)
9. Brusilovsky, P., Karagiannidis, C., Sampson, D.: Layered evaluation of adaptive learning systems. International Journal of Continuing Engineering Education and Lifelong Learning 1(14), 402–421 (2004)
10. De la Prieta, F., Di Mascio, T., Gennari, R., Marenzi, I., Vittorini, P.: The TERENCE Smart Games: Automatic Generation and Supporting Architecture. In: Proc. of the PDSG 2012 workshop co-located with EC-TEL 2012. CEUR-WS (2012)
11. Duffy, T.M., Cunningham, D.J.: Constructivism: Implications for the design and delivery of instruction. Jonassen, D. H. (1996), New York: Simon & Schuster Macmillan
12. EMAPPS consortium: EMAPPS Game Framework. Retrieved January 2012 from http://emapps.info/eng/Games-Toolkit/Teachers-Toolkit/Games-Creation/Framework-for-Game-Design
13. Kelle, S., Klemke, R., Specht, M.: Design Patterns for Learning Games. Journal of Technology Enhanced Learning 3(6), 555–569 (2011)
14. Mascio, T.D.: First User Classification, User Identification, User Needs, and Usability goals, Deliverable D1.2. Tech. rep., TERENCE project (2012)
15. Nanjappa, A., Grant, M.M.: Constructing on constructivism: The role of technology. Electronic Journal for the Integration of Technology in Education 2 (2003)
16. Norman, D.: The design of everyday things. Doubleday, New York (1998)
17. Rizzo, R.: Multimodal and Multimedia aspects of English Language Teaching and Studies in Italian Universities. Ibis (2009), Como
18. Slegers, K., Gennari, R.: Deliverable 1.1: State of the Art of Methods for the User Analysis and Description of Context of Use. Tech. Rep. D1.1, TERENCE project (2011)
19. TERENCE Project Web Site: `http://www.terenceproject.eu/`
20. Valeriani, A.: Ermeneutica retorica ed estetica nell'insegnamento verso l'oriente del testo. Andromeda (1986)
21. Yelland, N., Masters, J.: Rethinking scaffolding in the information age. Computers and Education 48, 362–82 (2007)