

1st International Workshop
on Pedagogically-driven
Serious Games



Proceedings of the 1st International Workshop on Pedagogically-driven Serious Games (PDSG 2012)

In conjunction with the
Seventh European Conference on Technology Enhanced Learning
(EC-TEL 2012),
Saarbrücken (Germany), 18th September, 2012

Editors

Stefano Bocconi
(Cyntelix and Vrije University Amsterdam, the Netherlands)

Ralf Klamma
(RWTH Aachen University, Germany)

Yulia Bachvarova
(Cyntelix, the Netherlands)

Introduction

Technology Enhanced Learning is a promising area where Serious Games can have an impact beyond pure entertainment. Despite this potential, two issues stand in the way of achieving learning effects:

- Few pedagogical theories are sufficiently formalized to allow implementation; therefore Serious Games seldom fully utilize the potential of pedagogy.
- Even if present, pedagogical principles are implicitly implemented in the game story, structure, and characters, so that it is very difficult to test different theories on the same game mechanics. This results in low reusability and high costs of pedagogically-driven serious games that hinder their widespread use.

To address these issues, this workshop focuses on:

- Formalizations of pedagogical theories potentially capable to drive the elements in a game's world, such as narrative and characters' behavior.
- Game frameworks based on storytelling, explicit game mechanics and intelligent agents that can provide a programmable environment to implement pedagogical formalizations

Participants to the workshop are researchers from the following areas:

- Serious games and Technology-Enhanced Learning research in general
- Agent technologies (especially in an interactive narrative context). Efforts in the Agent community such as Pogamut that connect agent platforms to games like Unreal Tournament can be the ground where these approaches can be implemented.
- Storytelling and Interactive narrative research; narrative engines such as Brutus, Minstrel and Mexica.
- Pedagogy theories/frameworks especially at a stage close to or already formalized.

Topics of interest include but are not limited to:

- Formalizations of pedagogical theories for the purposes of serious games
- Game frameworks that provide a programmable environment to implement pedagogical formalizations, based on any of the following:
 - Storytelling
 - Explicit game mechanics
 - Intelligent agents
- Approaches to modeling the interdependencies between pedagogy, storytelling and game mechanics in serious games and the translation of these interdependencies into agent behaviour
- Empirical studies addressing the interdependencies between pedagogy, narrative and storytelling and game mechanics
- Architectures for serious games that allow reusability
- Existing programmable platforms, such as Pogamut, that connect agent platforms to games
- Approaches to "programmable" narrative and storytelling in serious games
- Approaches to "programmable" intelligent agents in serious games

International Program Committee

- Stefano Bocconi (Cyntelix and VU Amsterdam, NL)
- Ralf Klamma (RWTH Aachen University, Germany)
- Yulia Bachvarova (Cyntelix, NL)
- Ana Paiva (INESC, PT)
- Frank Dignum (Utrecht University, NL)
- Arnav Jhala (University of California, USA)
- Martin Ruskov (UCL, UK)
- Sandy Louchart (Heriot-Watt University, UK)
- Ido Iurgel (Universidade do Minho, Portugal)
- Milos Kravcik (RWTH Aachen University, Germany)
- Michael D. Kickmeier-Rust (University of Graz, Austria)
- Frank Nack (University of Amsterdam, NL)
- Sara De Freitas (Serious Games Institute, UK)
- Rob Nadolski (Open University, NL)
- Roland Klemke (Open University, NL)

Table of Contents

1. An Ontology for Integrating Didactics into a Serious Training Game Marieke Peeters, Karel Van Den Bosch, John-Jules Meyer and Mark Neerinx	1-10
2. Towards Modeling Educational Objectives in Serious Games Eike M. Hirdes, Niroshan Thillainathan and Jan Marco Leimeister	11-14
3. Open 3D Environments for Competitive and Collaborative Educational Games Roland Klemke and Milos Kravcik	15-18
4. A Competence Performance Analyser Tool for Assessing Players' Activity in Serious Games Elisabetta Parodi and Mauro Vannucci	19-26
5. Using Out-of-Character Reasoning to Combine Storytelling and Education in a Serious Game Jeroen Linssen and Mariët Theune	27-30
6. The TERENCE Smart Games: Automatic Generation and Supporting Architecture Fernando De La Prieta Pintado, Tania Di Mascio, Rosella Gennari, Ivana Marenzi and Pierpaolo Vittorini	31-40
7. Playing for Improving the Reading Comprehension Skills of Primary School Poor Comprehenders Fernando De La Prieta Pintado, Tania Di Mascio, Rosella Gennari, Ivana Marenzi and Pierpaolo Vittorini	41-50
8. Serious Gaming for Complex Decision Making Anja Van Der Hulst and Mark Ruijsendaal	51-60

An Ontology for Integrating Didactics into a Serious Training Game

Marieke Peeters^{1,2,4}, Karel van den Bosch², John-Jules Ch. Meyer^{1,2}, and Mark A. Neerincx^{2,3}

¹ Information and Computing Sciences, Utrecht University

² Training & Performance Innovations, TNO

³ Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

⁴ Corresponding author: m.m.m.peeters@uu.nl

Abstract. Serious games offer high potential for immersive, effective, and autonomous training. However, research has shown that trainees need guidance and structure during training. This could be achieved by means of well-chosen scenarios and targeted adaptations of the storyline based on didactic considerations. This paper discusses some of the challenges posed in adaptive game design. Additionally, the paper outlines the design rationale behind the adaptive game architecture for training (AGAT). An ontology is proposed that serves as a foundation and knowledge base for a system able to orchestrate the game's storyline in a didactically desirable fashion. The ontology's use is versatile: it supports requirements elicitation and refinement, results in traceable underlying assumptions and design choices, and provides the knowledge base used by the system itself. The architecture is illustrated by means of a case study. Future work focuses on the development of a generic set of procedural rules to operate on this ontology and generate user-tailored didactically-driven adaptive scenario content.

Keywords:

ontology, adaptive educational game, game design, instruction, didactics, requirements analysis, situated cognitive engineering, scenario-based training, serious games

1 Introduction

The growing demand for autonomous training has led to an increase in research on intelligent instructional systems, such as serious games [21]. Serious games are designed to offer trainees the opportunity to develop their skills and knowledge in a meaningful and practical, yet virtual, training setting. They carefully balance fun, knowledge transfer, and reality to provide the trainee a meaningful, immersive, and motivating learning experience. A substantial amount of research regarding serious games has involved the use of intelligent agents to control the characters in the storyline, examples of which are the Intelligent Story Architecture for Training (ISAT) [11,10], IN-TALE [22], and Thespian [25].

The use of intelligent agents to control important non-player characters (NPCs) in the scenario allows for training opportunities in the absence of an instructor and without the need for team members being present. For a long time, research has focused on generating believable and adaptive NPC behavior, however, Yannakakis (2012) claims that NPC AI is almost solved [31]. More interesting, novel challenges revolve around the knowledge about users, tailoring the game to those users, and techniques to control automated content generation.

The need for personalization is even more important to the development of *serious* games. In order to warrant the didactical training value, the trainee requires guidance and structured learning content, comparable to following a personalized curriculum guided by a personal coach [8]. Of course, personalization should be grounded in didactical principles, derived from efficacious training forms (e.g., scenario-based training [23]) and instructional design (e.g., 4C/ID [27]) [12,17,18]. Such an approach combines the immersion and appeal of serious games with the structure offered by intelligent tutoring systems [15,28]. Instructional theory promotes a balance of challenge and ability, and the provision of meaningful scenarios to the trainee. But the question remains how these principles should be embedded in AEG design.

Personalization means that the system knows and interprets the trainee's performance and adapts the training exercise to match this performance using didactical strategies [19]. Examples of didactical strategies are the iterative off-line selection of suitable training objectives and topics, the online delivery of variable amounts of support, or adjustments in the pace of training. In addition to the training exercise, the adaptation itself can be altered to fit the trainee's personal preferences or learning style. Note that the trainee is not the only person involved in training with personal preferences or styles; the instructor may also have a preferred didactic strategy. Such requirements for customization and personalization form a serious challenge for AEG design [24]. Preferably, AEG design separates these requirements in an early stage and tackles them in a modular way to promote code reusability over domains, trainees, and instructors.

This paper presents work on an adaptive educational game architecture that fosters reusability of its components and has a strong foundation in didactical principles and instruction theory. Section 2 describes relevant related work in the field of adaptive educational games. Section 3 outlines the reasons for and method of developing an AEG ontology, along with a high/level presentation of the resulting ontology and its intended use. The ideas presented in the paper are discussed in Section 4.

2 Related Work on Adaptive Games

In the past years several researchers started focusing on player-centric adaptive games. As Zook and Riedl (2012) point out, there are two aspects to user-adaptive game design: challenge tailoring (CT) and challenge contextualization (CC) [32]. CT refers to online as well as offline dynamic difficulty adjustment: reasoning about scenario content on a didactic level. This reasoning is based on

didactical principles or strategies leading to decisions about appropriate learning topics and levels of challenge or support. CT requires high resolution player profiles [9], usually specifying the player's skill proficiencies. In contrast, CC refers to the construction of the game world and events that set up the selected learning objective and challenge in the actual game environment. It deals with the reality and believability of the trainee's learning experience.

As mentioned above, challenge tailoring refers to offline as well as online adaptivity. A promising development in offline challenge tailoring is procedural content generation (also mentioned by Yannakakis (2012) [31]) controlled by semantic modeling techniques [5]. By embedding and interpreting higher level semantic annotations in virtual objects and agents, the content generation process can be constrained to create meaningful and realistic content that matches the learner's profile [1,26]. Such higher level constraints can then be fulfilled by equipping objects with the capacity to provide services in the game [6]. For instance, in the CT stage, the scenario is prepared offline by generating a set of constraints that delineate the learning goal (e.g., treat a thermal lesion) and the level of challenge (e.g., beginner). The scenario generator then collects a set of annotated objects that offer the services required to fulfill those constraints (e.g., a hot object, a victim, a water tap, and a first aid kit). A straightforward method to manage online adaptivity is the use of dynamic world elements, such as NPCs and dynamic objects, enhanced with didactically meaningful behavior variations or variable characteristics. In the case of the example, the victim could have two behavior variations, one in which the victim is calm, and one in which the victim is panicking. The ALIGN system [20], for example, uses annotated adaptive elements to enable online adaptivity. It incorporates personalised didactics into a serious game, while separating the pedagogical principles from the game, thereby making it reusable. Peeters et al. (2011) [18] and Westra et al (2010) [30] used scripted NPCs that were able to perform different behavior variations, thereby enabling online scenario adaptation.

2.1 Context of Previous Work by the Authors

In a previous paper, the situated Cognitive Engineering (sCE) method [13] led to the specification of a set of design principles and a high level AEG architecture: the Adaptive Game Architecture for Training (AGAT). The combination of knowledge from different fields (e.g., game research, intelligent tutoring systems, instructional theory, and educational psychology) resulted in an initial requirements baseline ([R1]-[R5]) for an AEG. Each requirement is founded in a set of measurable and testable claims ([C1.1]-[C5.2]), each of which is grounded in literature research and expert knowledge. The interested reader is referred to [19] for the details on the underlying literature review that lead to these claims and requirements.

[R1] Match scenarios to the trainee's skill level (offline).

[C1.1] Presenting scenarios in order of increasing complexity and matching them to the trainee's level of experience prevents cognitive overload.

[R2] Adjust the support level during task performance (online).

- [C2.1] Adjusting the level of challenge to match the trainee's skill level fosters flow and high levels of motivation.
- [R3] Generate authentic scenarios.
 - [C3.1] Authentic training tasks foster transfer.
 - [C3.2] Authentic training tasks foster intrinsic motivation.
 - [C3.3] Engaging in authentic training tasks fosters immersion, and thereby flow and motivation.
- [R4] Generate a wide variety of adaptive scenarios.
 - [C4.1] This will foster transfer and the development of generic solutions.
- [R5] Provide feedback about the task performance during the scenarios.
 - [C5.1] This will foster self-efficacy.
 - [C5.2] This will foster a better understanding of the task domain.

A first experimental evaluation of our prototype was conducted during which domain experts rated video fragments of adaptive and non-adaptive scenarios in terms of learning value. This study revealed that online adjustments ([R2]) of the support level significantly improve the quality of training [18], validating further research on the development of our architecture.

As the research project progressed an additional technical requirement was added to this list:

- [R6] Promote reusability over domains, trainees, and instructors.

3 Adaptive Game Architecture for Training

The requirements mentioned in Subsection 2.1 form the foundation for our AEG architecture. Two important notions led to the design presented below. First of all, as mentioned in the previous section, there are two stages in difficulty adjustment, offline as well as online: 1) challenge tailoring, and 2) challenge contextualization. Second, domains, world content, teaching strategies, and trainees may change over time, and the system should offer ways to handle such changes through reusable components. This requires a clear format for new information and generic procedures able to handle that format. For instance, the system needs to know about the concept of didactical strategies and use this knowledge by employing those strategies using a generic method, instead of employing hard-coded, implicit didactical strategies. We propose the use of an ontology to specify relevant information about the user, the didactical strategies, and the domain. In addition, generic procedural rules are designed to use this information and generate constraints on the procedural content generation process. The training scenarios are first generated on a didactical level (offline) and are then contextualized using semantically annotated objects. In turn, these objects are able to perform several behavior variations to enable online adaptivity.

The rest of this section describes the first part of the architecture, the ontology, which describes the knowledge areas that are characteristic and relevant to AEG design: the task domain, the trainee, the available didactic strategies, the instructor's personal touch, the game world, and the system's design. First, the

concept of ontologies is explained in Subsection 3.1. Thereafter, it presents and exemplifies the resulting ontology in Subsection 3.2.

3.1 The Need for an Ontology

The motivation to create an ontology that defines all concepts related to AEG design was twofold: 1) by creating an ontology, the system's specification is refined, since it forces the developer to build a solid argument and plan for each functionality, and 2) the ontology contributes to the desire of building a modular system that consists of generic rules imputed by exchangeable (formalized) knowledge bases. An ontology represents the basic concepts relevant to the system's operations, along with their attributes and interrelations, thereby modeling a domain of knowledge. The use of an ontology is beneficial to the design of adaptive systems; it supports a shared understanding of the system's concepts and interrelations [3,4], but also the early refinement and testing of the system's requirements [16]. However, as explained earlier, our main interest in using an ontology is that it can serve as a knowledge base for the system to rely on, that allows for reusability and easy modification [2].

Related Ontologies. Kickmeier-Rust and Albert (2008) agree that serious games should balance challenge and ability to promote flow and motivation [7]. Their ELEKTRA ontology resembles parts of our ontology areas, the most important resemblance being the distinction between task performance and skill proficiency. This distinction is important since it abstracts away from the task, defining the learning content as a higher level ability and understanding. This makes it possible to separate the performance data from the task domain, since the skills to be developed overarch several domains.

The ontology by Van Welie et al. (1998) has served as the starting point for our task domain ontology area [29]. It defines tasks as activities performed to reach a certain goal, and possibly, there are multiple ways to reach it. The goal of a task is a specific state that is reached after successful execution of the task. Since tasks can be performed by a group of people in dynamic environments, Van Welie et al.'s ontology takes roles and events into account.

Ontology Engineering Method. The ontology described below was created using an iterative 4-step process, derived from Noy and McGuiness' Ontology 101 (2001) [14]. It uses First Aid Training as the application domain for clarification purposes. The four steps used during the creation of the ontology were:

- (1) Specify all the terms relevant to the requirements.
- (2) Identify the important properties of the terms specified in Step 1.
- (3) Define the relations between the terms.
- (4) Create domain-specific instances for all of the terms by applying the ontology to the training domain.

The ontology specification process is iterative; each step results in new knowledge about the quality and sufficiency of the terms, attributes, relations and their definitions identified in previous iterations. The main reason for iterative refinement is that the ontology as a whole needs to be cohesive and consistent.

3.2 The AEG Ontology - Description

The ontology serves to answer questions like ‘What will the system teach, and to whom?’, ‘What strategies can the system use to teach?’, ‘What narrative elements can the system use to contextualize the learning content?’, and ‘What higher level design and system constructs does the system use?’. Various sources of information, e.g., observations, interviews, and literature research, were used to answer these questions, meaning that the ontology also serves a purpose of theory development. The analysis resulted in an ontology consisting of 6 main areas: ‘Task domain’, ‘Trainee’, ‘Didactics’, ‘Instructor’, ‘World’, and ‘System’.

- Task Domain - this ontology area refers to concepts involved in the task execution, such as ‘Task’, ‘Role’, ‘Objective’, etc. This ontology area was based on work by van Welie et al. (1998) [29].
- Trainee - these concepts specify all the required knowledge to reason about the trainee and his/her progress during training, e.g., ‘Performance’, ‘Skill’, and ‘Motivation’.
- Didactics - this area includes concepts referring to instructional features of the system, examples of which are ‘Support Level’, ‘Feedback’, and ‘Cognitive Load’.
- Instructor - these concepts deal with the interaction between the system and the *instructor*, and include concepts such as ‘Didactic Strategy’ and ‘Scenario Compilation’.
- World - the concepts in this area refer to all concepts relevant to the game world. It includes concepts such as ‘Object’, ‘Agent’, and ‘Event’.
- System - this area contains concepts that refer to higher level abstractions and to terms relevant to the initial design architecture, specifying, for example, ‘Task Model’, ‘System Component’ and ‘Intelligent Agent’.

The ontology has been implemented in Protégé frames 3.5 alpha ⁵. In the near future it will be reviewed by experts and thoroughly checked for consistency. Due to space limitations the complete ontology is not discussed here. However, the ontology and its use are illustrated by means of a case study in the next subsection.

An Illustrative Case Study This section illustrates the intended architecture and the use for the ontology with a case study.

Jeremy has trouble connecting the topics presented during different training sessions

⁵ <http://protege.stanford.edu>

over time. In a previous session he received instructions on how to deal with non-cooperative patients. Today he will receive instruction on the diagnosis of burns.

To properly teach Jeremy how to perform First Aid, the system will need to meet the specified requirements. For this example we shall discuss just one of them: (1) choose a scenario that is appropriate for the learning goal. To meet this requirement, the system needs to have a proper ‘understanding’ of what it means. To do that it must have knowledge about the meaning of the concepts in the requirement. This knowledge is available in the ontology.

First of all, the system needs to know what a *scenario* is. The ontology defines scenarios on a semantic level: scenarios refer to a set of tasks, and contain an initial world state, and possibly a sequence of necessary events. This initial world state is then defined as a set of specified objects and agents situated in some environmental setting. The ontology also specifies *tasks* and how they should be decomposed and/or performed in the Task ontology area. The storytelling elements, such as settings, objects, and characters along with their actions and the effects thereof, all belong to the World ontology area. Objects are embedded with additional information about their use within the task domain as well as their didactical purposes (e.g., difficulty levels).

Second, the system should recognize the concept *learning goal*, which is defined as an objective that Jeremy should achieve with respect to his *skill* development. To derive Jeremy’s learning goal, the system relies on the knowledge collected in Jeremy’s current *skill graph*. The system can now use this knowledge to derive an appropriate learning goal for Jeremy:

‘Generalize task procedures over contexts.’

To generate a scenario that fits the learning goal, the system must know how to match scenarios to learning goals. However, a learning goal refers to skill development, whereas a scenario refers to a task performance embedded in a storyline. This requires a relation between the concepts ‘task’ and ‘skill’. As mentioned in the discussion of the ELEKTRA ontology, tasks are specified to rely on a (set of) skill(s). This allows for the system to produce a scenario that matches Jeremy’s skill set.

It becomes clear, that there are still a lot of concepts mentioned in this example that need further specification before the system is actually able to reason about them. For now, we will leave this example. The AEG ontology, however, covers a lot more than the content discussed in the example above.

4 Discussion

This paper discusses the design of an adaptive game architecture that promotes reusability and a proper didactical foundation. It proposes the use of an ontology as a knowledge base, combined with a set of generic procedural rules that operate on this ontology. The ontology specifies all the relevant concepts, their attributes, and their interrelations.

The ontology has several purposes: 1) it serves as a vocabulary to use during consultations with stakeholders and domain experts, 2) it explicates choices made

in the design, thereby making them traceable, 3) it can be checked for consistency and coherence, 4) it leads to early refinement and testing of the requirements, and 5) it serves as a declarative knowledge base for the system, promoting reuse of the generic procedural rules on interchangeable knowledge bases regarding for instance different training domains, didactic strategies, and virtual worlds.

The result of our research is a coherent and consistent ontology, forming a solid knowledge base that is useful to the stakeholders, developers, and the system itself. Moreover, the ontology led to a refinement of the requirements, newly discovered requirements, and a way to warrant the system's robustness. The ontology has been implemented and is currently checked for consistency. In short notice, the ontology will be verified by domain experts.

Future work focuses on the further development of a generic procedural rule base that uses the ontology to impute the rules' variables and produce user-tailored, domain specific, and adaptive training scenarios. This process will result in an even more detailed refinement of the system's requirements. Once the ontology and the reasoning rules are finished, an (agent-based) environment will be connected to the system, so the system can be evaluated on its requirements by testing their corresponding claims. The results of that test will lead to a further refinement of the architecture and system design.

Adaptive serious games have mainly focused on the maintenance of the storyline and believability of the characters, lacking didactical principles to adjust the storyline in favor of the learning goals. Alternatively, intelligent tutoring systems rely on didactical principles and result in structured learning content, but often only apply to well-defined training domains, such as computer programming, mathematics, and physics. Bridging the gap between these two research areas would result in highly engaging and effective autonomous training opportunities, however this requires a holistic view on game design: the development of adaptive systems is labour-intensive and reusability of (parts of) the game is important for the future of serious games.

Preferably, an AEG architecture uses several ontologies to draw its knowledge from: one referring to the training domain, one referring to the game world elements, one referring to the trainee, and one referring to the didactic strategies it can use to select and alter training scenarios. By combining semantic modeling, procedural content generation techniques, and adaptive storytelling elements, games may not just become adaptive, but modular and reusable as well.

References

1. Bidarra, R., de Kraker, K., Smelik, R., Tutenel, T.: Integrating semantics and procedural generation: key enabling factors for declarative modeling of virtual worlds. In: Proceedings of the FOCUS K3D Conference on Semantic 3D Media and Content. pp. 51–55 (2010)
2. Bouamrane, M.M., Rector, A., Hurrell, M.: Ontology-driven adaptive medical information collection system. In: An, A., Matwin, S., Ras, Z., Slezak, D. (eds.) Foundations of Intelligent Systems. Lecture Notes in Computer Science, vol. 4994, pp. 574–584. Springer Berlin / Heidelberg (2008)

3. Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What are ontologies, and why do we need them? *IEEE Intelligent Systems* 14(1), 20–26 (1999)
4. Falbo, R.d.A., Guizzardi, G., Duarte, K.C.: An ontological approach to domain engineering. In: *Proceedings of the 14th international conference on Software engineering and knowledge engineering*. pp. 351–358. SEKE '02, ACM (2002)
5. Hullett, K., Mateas, M.: Scenario generation for emergency rescue training games. In: *Proceedings of the 4th International Conference on Foundations of Digital Games*. pp. 99–106. FDG '09, ACM (2009)
6. Kessing, J., Tutenel, T., Bidarra, R.: Services in game worlds: A semantic approach to improve object interaction. In: *Proceedings of the 8th International Conference on Entertainment Computing*. pp. 276–281. ICEC '09, Springer-Verlag (2009)
7. Kickmeier-Rust, M., Albert, D.: The elektra ontology model: A learner-centered approach to resource description. In: Leung, H., Li, F., Lau, R., Li, Q. (eds.) *Advances in Web Based Learning ICWL 2007*. Lecture Notes in Computer Science, vol. 4823, pp. 78–89. Springer Berlin / Heidelberg (2008)
8. Kirschner, P., Sweller, J., Clark, R.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational psychologist* 41(2), 75–86 (2006)
9. Lopes, R., Bidarra, R.: Adaptivity challenges in games and simulations: A survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3(2), 85–99 (2011)
10. Magerko, B., Wray, R., Holt, L., Stensrud, B.: Customizing interactive training through individualized content and increased engagement. In: *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. vol. 2005. NTSA (2005)
11. Magerko, B., Laird, J.E., Assanie, M., Kerfoot, A., Stokes, D.: Ai characters and directors for interactive computer games. In: *Proceedings of the 16th conference on Innovative applications of artificial intelligence. IAAI'04*, vol. 1001, pp. 877–883. AAAI Press (2004)
12. Merrill, M.D.: First principles of instruction. *Educational Technology Research and Development* 50(3), 43–59 (2002)
13. Neerincx, M., Lindenberg, J.: Situated cognitive engineering for complex task environments. In: Schraagen, J., Militello, L., Ormerod, T., Lipshitz, R. (eds.) *Naturalistic Decision Making and Macrocognition*, pp. 373–390. Aldershot, UK: Ashgate Publishing Limited (2008)
14. Noy, N., McGuinness, D.: *Ontology development 101: A guide to creating your first ontology*. Tech. rep., Knowledge Systems Laboratory, Stanford University (2001)
15. Ohlsson, S.: Some principles of intelligent tutoring. *Instructional Science* 14(3), 293–326 (1986)
16. Omoronyia, I., Sindre, G., Stålhane, T., Biffel, S., Moser, T., Sunindyo, W.: A domain ontology building process for guiding requirements elicitation. In: *Requirements Engineering: Foundation for Software Quality*. pp. 188–202. Springer (2010)
17. Oser, R., Cannon-Bowers, J., Salas, E., Dwyer, D.: Enhancing human performance in technology-rich environments: guidelines for scenario-based training. *Human Technology Interaction in Complex Systems* 9, 175–202 (1999)
18. Peeters, M., Bosch, K.v.d., Meyer, J.J., Neerincx, M.: Scenario-based training: Director's cut. In: *Proceedings of the 15th Int. Conf. on AIED*. vol. 15, pp. 264–271 (2011)

19. Peeters, M., Bosch, K.v.d., Meyer, J.J., Neerincx, M.: Situated cognitive engineering: the requirements and design of directed scenario-based training. In: Proceedings of the 5th Int. Conf. on ACHI (2012)
20. Peirce, N., Conlan, O., Wade, V.: Adaptive educational games: Providing non-invasive personalised learning experiences. In: Proceedings of the 2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning. pp. 28–35. DIGITEL '08, IEEE Computer Society (2008)
21. Rieber, L.: Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research & Development* 44(2), 43–58 (1996)
22. Riedl, M., Stern, A., Dini, D., Alderman, J.: Dynamic experience management in virtual worlds for entertainment, education, and training. *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning* 4(2), 23–42 (2008)
23. Salas, E., Priest, H.A., Wilson, K.A., Adler, A.B.: Scenario-based training: Improving military mission performance and adaptability. In: Britt, T., Adler, A., Castro, C. (eds.) *Military Life: The Psychology of Serving in Peace and Combat*, pp. 32–53. Praeger Publishers (2006)
24. Salehie, M., Tahvildari, L.: Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 4(2), 14:1–14:42 (2009)
25. Si, M., Marsella, S., Pynadath, D.: Directorial control in a decision-theoretic framework for interactive narrative. In: *Interactive Storytelling*. pp. 221–233. Springer (2009)
26. Smelik, R., Tutenel, T., de Kraker, K., Bidarra, R.: A declarative approach to procedural modeling of virtual worlds. *Computers & Graphics* 35(2), 352–363 (2011)
27. Van Merriënboer, J.: Training complex cognitive skills: A four-component instructional design model for technical training. *Educational Technology Publications* (1997)
28. VanLehn, K.: The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education* 16(3), 227–265 (2006)
29. Welie, M.V., Veer, G.C.V.D., Elins, A.: An ontology for task world models. In: Proceedings of the 5th International Eurographics Workshop on Design Specification and Verification of Interactive Systems. vol. 98, pp. 3–5. Citeseer (1998)
30. Westra, J., Dignum, F., Dignum, V.: Keeping the trainee on track. In: *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. pp. 450–457. IEEE (2010)
31. Yannakakis, G.: Game ai revisited. In: Proceedings of the 9th conference on Computing Frontiers. pp. 285–292. ACM (2012)
32. Zook, A., Riedl, M.: A temporal data-driven player model for dynamic difficulty adjustment. In: Proceedings of the Conference on Artificial Intelligence and Interactive Digital Entertainment. AAAI (2012, to appear)

Towards Modeling Educational Objectives in Serious Games

Eike M. Hirdes, Niroshan Thillainathan, Jan Marco Leimeister

Kassel University, Information Systems, Kassel, Germany

{hirdes, thillainathan, leimeister}@uni-kassel.de

Abstract. When developing serious games the most complex task is the alignment of instructional teaching methods and the game itself. To address this issue, we propose a shared language modeling approach for educational instructors and game developers. The language is based on so called serious game bricks, composites and rules. Combining these pedagogical and story elements allows the domain experts to create serious game patterns. The use of those patterns supports the development of serious games that are both entertaining and present specific educational objectives.

Keywords: serious games, educational objectives, modeling language, pattern, game development

1 Motivation

In many different educational settings, increased usage of digital games to support learning can be observed [1]. The entertaining nature of games incite and motivate users to learn and exercise, and furthermore they can increase the effectiveness of learning processes [2]. Learning objectives are integrated within games, so that users can reach these playfully and learn simultaneously [3]. These games are called serious games. They encompass digital games which entertain and, at the same time, educate or instruct the user [4].

Harteveld et al. state that during the development process of a serious game the toughest challenge is the alignment of learning content and the game itself [5]. Adding to that, game developers need to integrate pedagogy into the digital games' story [6]. Greitzer et al. state that a systematic engineering method is needed to build, understand and analyze serious games, and especially to focus "on pedagogical approaches that provide effective, relevant, and motivating learning experiences" [6]. This position is supported by Zyda, who states a practice to insert learning opportunities into stories needs to be developed and "research must focus on combining instruction with story creation and the game development process" [4].

To facilitate and stimulate the up to now relatively unstructured transition of instructional teaching methods to serious games, we propose a shared language modeling approach between the domain experts, i.e. instructors and game developers, in-

volved. Our aim is to connect both domains to produce reusable patterns for serious games, which enable learners to achieve predictable learning successes in a playful manner. Consequently, the shared language is intended to meet the following goals: (1) Standardize, clarify and simplify the communication between instructors and game developers, (2) create a formal description of patterns to achieve learning objectives within serious games, (3) allow the reusability of the prior mentioned patterns.

The structure of this paper is as follows: Section 2 provides relevant work. Section 3 presents the proposed modeling language. Finally, section 4 presents conclusions, limitations and future work.

2 Related Work

In scientific literature, a plethora of different approaches for developing serious games can be found lacking a standard model. Hartevelde et al. base their game development about levee inspection (Levee Patroller) on underlying design and learning theories [5]. They conclude that three components must be taken into account during serious game development: pedagogy (learning), game (fun) and reality (validity). Their main focus during development lies in matching game contents to pedagogical methods. Kelly et al. developed a serious game (Immune Attack) for teaching immunology [7]. Their approach focuses on three research challenges: game design, integration and multiple scales. Furthermore, learning objectives were used to specify learning outcomes and were connected to gameplay. Muratet et al. have designed and developed a serious game to improve programming skills. In a first step the authors examined what kind of digital game is suitable for the task [8]. In the next step the game was developed based on learning objectives from different points of view to evaluate the learning success. The authors state a learning process will occur, if a serious game is attractive, fun, stimulating, and encourages the player to progress.

While differing in many aspects during development, all three cases highlight the need for a pedagogical approach interwoven within the story of a serious game to achieve and evaluate the learning outcomes. Educational objectives (also known as instructional goals) are outcome statements describing the knowledge, skills and/or attitudes learners have gained upon completion of instructional units. They can be utilized to design instructional units to ensure the focus on learning outcomes. Furthermore, they can be used to communicate instructional aims to learners and serve as a basis for the evaluation of the learning success.

A modeling language is required to develop processes with predictable learning outcomes and to standardize the communication between domain experts. Basically, a modeling language is a Domain Specific Visual Language. Compared with general purpose languages, these languages allow the description of solutions for a problem at the level of abstraction of the domain. A modeling language consists of syntax and semantics. The syntax has elements and rules to construct a correct model. The elements are the building blocks of the modeling language, whereas the rules determine the syntactically correct combination of the elements. The semantics depicts the meaning of the combination of elements or model as a whole.

3 A Modeling Language for Educational Objectives in Serious Games

In this chapter we propose a modeling language that allows the description of pedagogical goals and story aspects for the serious game development process. Our aim is to enable instructors and game developers to combine teaching methods and story elements to obtain reusable serious game patterns for specific educational objectives. As a result, these patterns can be used by game developers as best practices during the development process to ensure predictable learning success in serious games.

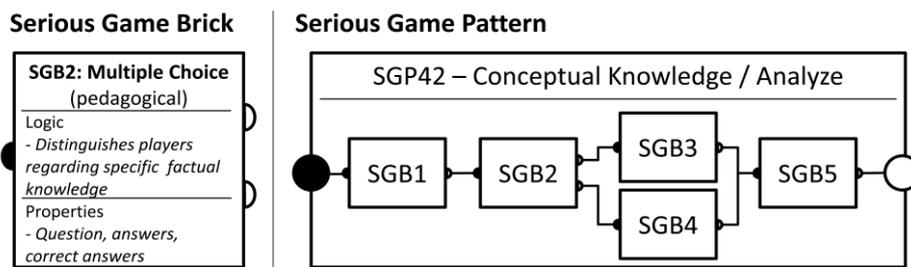


Fig. 1. Serious game brick and pattern (source: own representation)

The modeling language consists of connectors and two kinds of elements: serious game bricks and serious game composites. Connectors describe the control and information flow between elements. Whereas a serious game brick (SGB) represents an indivisible, basic entity of a serious game which fulfills either a pedagogical or a game function. Bricks consist of a name, description, classification, in-/output sockets, logic and properties. The classification specifies whether the SGB fulfills pedagogical or story functions. The in-/output sockets can be used to establish connections using connectors between SGBs. A brick receives data through an input socket, processes the data and sends it updated via the output socket to another element. The logic describes how the data is processed and describes the function of a SGB. Properties are interchangeable parameters to adapt the logic (see Fig. 1). The other elements of our modeling language are called serious game composites (SGC). These elements are representations of a combination of two or more connected bricks to encapsulate several indivisible functions to one reusable complex function. Like a SGB, a composite also consists of a name, description, in-/output sockets and properties. When SGCs achieve educational objectives through integrated instructional teaching methods, then this special kind of composite is referred to as a serious game pattern. By using the revised Bloom's educational objectives taxonomy [9] we apply a knowledge dimension (factual, conceptual and procedural knowledge) and a cognitive process dimension (e.g. apply, create) to each pattern, to define the scope and field of application. To ensure a high degree of reusability, the patterns neither contain learning nor game contents. These contents will be added when applying the patterns.

Basic rules were established, such as an output socket must be connected to an input socket. Each element is connected to at least one input and one output socket. The first and last elements are connected to special start and end elements. Furthermore,

following the aspect of a story driven design for serious games, two as pedagogical classified SGBs should not be placed after each other.

4 Conclusion and Outlook

The objective of this paper is to allow domain experts to create serious game patterns, which combine instructional teaching methods with story elements. We have shown that existing game design approaches highlight the importance of a pedagogical approach in serious games. Therefore, we presented first steps towards a shared modeling language for the domains' pedagogy and game development. The language is based on serious game bricks, composites and rules. We also proposed serious game patterns, a combination of bricks and composites which leads to the achievement of learning objectives. This paper raises several issues for further consideration. First, research is needed to identify the key serious game bricks and modeling language rules to support either pedagogical or story objectives. Second, Bloom's revised taxonomy needs to be assessed for its suitability as a classification. Third, a tool support needs to be implemented and a process developed which enable instructors and game developers to use the modeling language to jointly create and modify serious games patterns. Finally, a tool is needed to develop serious games based on the presented serious game patterns.

The research presented in this article was partially funded by the German Federal Ministry of Education and Research in the project ProduSE (www.projekt-produse.de), FKZ01FL10044.

5 Bibliography

1. Paraskeva, F., Mysirlaki, S., Papagianni, A.: Multiplayer online games as educational tools: Facing new challenges in learning. *Computers & Education*. 54, 498–505 (2010).
2. Prensky, M.: *Digital Game-Based Learning*. McGraw-Hill, New York (2001).
3. Michael, D., Chen, S.: *Serious games: Games that educate, train, and inform*. Thomson Course Technology PTR (2005).
4. Zyda, M.: From visual simulation to virtual reality to games. *Computer*. 25–32 (2005).
5. Harteveld, C., Guimarães, R., Mayer, I., Bidarra, R.: Balancing Pedagogy , Game and Reality Components Within a Unique Serious Game for Training Levee Inspection. 128–139.
6. Greitzer, F.L., Kuchar, O.A., Huston, K.: Cognitive science implications for enhancing training effectiveness in a serious gaming context. *Journal on Educational Resources in Computing*. 7, 10 (2007).
7. Kelly, H., Howell, K., Glinert, E., Holding, L., Swain, C., Burrowbridge, A., Roper, M.: How to build serious games. *Communications of the ACM*. 50, 44–49 (2007).
8. Muratet, M., Torguet, P., Jessel, J.-P., Viallet, F.: Towards a Serious Game to Help Students Learn Computer Programming. *International Journal of Computer Games Technology*. 2009, 1–12 (2009).
9. Krathwohl, D.R.: A revision of Bloom's taxonomy: An overview. *Theory into practice*. 41, 212–218 (2002).

Open 3D Environments for Competitive and Collaborative Educational Games

Roland Klemke¹, Milos Kravcik²

¹ CELSTEC, Valkenburgerweg 177, Heerlen, The Netherlands
roland.klemke@ou.nl

² Advanced Community Information Systems (ACIS), Informatik 5,
RWTH Aachen University, Germany
kravcik@dbis.rwth-aachen.de

Abstract. Educational games have a lot of potential to raise students' motivation and improve the quality of education when applied properly. But finding a suitable game for a particular learning objective is not easy and development of a new one is expensive. In our university course a group of students developed a prototype of a serious gaming tool for architectural design, which is based on the Google Street View environment. With this tool teams can model 3D buildings, place them in real world images, share their results, and rate them. The solution provides a better contextualization of the model and paves a way towards integration with a full 3D environment, which should even more improve the serious gaming experience in the architectural design.

Keywords: game based learning, 3D worlds, open platforms, competitive, collaborative

1. Introduction

Despite the long tradition of games for education and training, their uptake in higher education is very limited - especially, when compared to the boost in the games market [1][2]. The reasons for this are manifold: high technical demands are in conflict with available budgets [3]. Educational games often do not fit in the educational context or they are hard to tailor [4]. It is hard for teachers to support educational games within their educational processes [5]. In a previous paper, we explored and reported on an approach to address the *high technical demands* and the *limited familiarity of teachers* with games based on freely available tools and open platforms [6]. In this paper, we will build on this work and explore an extension of one of the approaches towards 3D-modelling embedded in open environments.

By extending its range of openly accessible productivity tools with open APIs (Application Programmer's Interfaces) that can be used by developers to create services and tools based on Google's suite of technologies, Google simplifies the process of developing specialised applications and services that rely on well tested user interfaces and back-end technologies. Google Street View is one of these tools, offered by Google as an add-on to the popular Google Maps. Street View offers navigable, 3D-like visualisations of the environment, displayed from a user point-of-

view. A user can navigate through Street View as if moving around the actual scenery.

Based on Street View as a front-end component, the *StreetLearn* game engine [6] is designed as a simulated location-based game combining locations, objects, players, and tasks in a 3D-environment representing the real world. Players as well as all objects and tasks are associated with a specific location on the map. The game starts at a specific location, where players are confronted with an initial task description. Typical tasks comprise finding locations, finding/taking objects, retrieving information, and answering questions. Solving a task leads to scores and usually a follow-up task. Players can be organised in competing teams that share tasks. Teams gain a team score, but individual players also score individually. Typical examples for *StreetLearn* games comprise scavenger hunt games, location-based quiz-rallies, or exploration games.



Fig. 1 StreetLearn user interface

Looking at gaming processes and learning processes from a more pedagogical perspective, *StreetLearn* is designed with the learning process being controlled by the gaming process [7].

2. Pedagogical and technological approach

Using Street View as game platform has a specific drawback in educational situations, where real 3D-models offer an additional benefit, such as architectural education: the Street View-based user interface only shows the 2D surface of the environment. Consequently, it is our aim to combine existing 3D models with the *StreetLearn* interface to provide an in depth experience.

In the course of a student development project at RWTH, a group of computer science students participating in the course Hightech Entrepreneurship and New Media (HENM'11) consequently got the task to enhance the *StreetLearn* environment with a 3D visualisation add-on that allows visualising 3D models within the *StreetLearn* environment. This way, the photographic environment of *StreetLearn* can be extended with explorable 3D models of existing or planned buildings. The add-on allows extending the game play of *StreetLearn*: while previously, the existing world serves as a playground, where players could navigate and interact, now it is possible to enhance the environment with virtual entities.

This idea has been taken to a competitive architectural game: several student teams compete in an architectural competition game. Each team represents an architectural firm, which tries to win several projects. Each of these projects is represented by a virtual construction site, represented on the map. The game process introduces the game goal and guides the team around these different sites in order to receive information, retrieve hints, and solve architectural tasks by creating 3D models, which they place in the *StreetLearn* environment. After completion of the tasks, participants

of competing groups can rate the other group's outcomes. This way, the game offers two motivating and pedagogically important principles: collaboration (applied within a group of students) and competition (applied across groups). Collaborative learning [9] fosters engagement of students, who can capitalize on one another's competences. The competitive aspect stimulates the performance of groups. Moreover, the design dimension emphasizes also constructivistic learning principles [10], including active [11], experiential [12], and problem-based learning [13]. This type of learning is an active process of interpreting and constructing individual knowledge representations. It aims at complex problems that do not have a single correct answer and is based on concrete experience. Thus the proposed pedagogical approach cultivates a whole spectrum of cognitive skills from the revised Bloom's taxonomy [10], including the highest ones – analysis, synthesis, and evaluation.

Technically, the students built on the existing StreetLearn object model as described in [6] and extend its MapItem entity, which serves as a general purpose location-based object. As shown in fig. 2, ConstructionItem, representing a construction site on the map, extends MapItem. ConstructionItem can contain a number of BuildingItems, which represent alternative designs created by competing teams. The BuildingStatus allows associating a simple process model to buildings in order to represent several design and construction phases.

This approach allows building on the existing gaming infrastructure of StreetLearn (game process, team play, scoring mechanism) while concentrating on the novel aspects (3D model visualisation, architectural process representation).



Fig. 2 StreetLearn with embedded 3D model (left) and Extension of StreetLearn object model (right)

3. Results and conclusion

Within the student project, we were able to demonstrate the development of a prototypical StreetLearn add-on, which was capable of visualizing uploaded 3D-models on top of the StreetLearn user interface. Also, the collaboration and competition features have been realised within the StreetLearn architecture.

The work presented here is in preliminary status, a full evaluation of the extended StreetLearn module has not yet been performed. First feedback gathered by presenting the prototype to architectural students indicates two main messages: (1) The integration of 3D models in their “natural” environment such as provided by

StreetLearn gives a better contextualisation of the model and the surrounding it may be realised in. (2) The user interface integration of StreetLearn and the 3D-models however feels a bit unnatural, due to the non-3D behaviour of the underlying Street View technology. For a future version of this approach, we consequently think of integration with a full 3D environment such as Google Earth.

Acknowledgments. We want to thank the participating Students of the HENM'11 course for their contributions. Part of the work presented here was co-funded by SURFnet/Kennisnet in their programme 'Innovation of Higher Education 2011'.

References

1. PWC (2010) Global Entertainment and Media Outlook: 2010-2014. <http://www.pwc.com/>
2. National Gaming Survey (2009) Detailed data of gaming in the Netherlands. Newzo, TNS-NIPO. [Online], www.nationaalgamingonderzoek.nl
3. Westera, W., Nadolski, R., Hummel, H. and Wopereis, I. (2008) 'Serious Games for Higher Education: a Framework for Reducing Design Complexity', *Journal of Computer-Assisted Learning*, vol. 24, no. 5, pp. 420-432.
4. Klopfer, E., Osterweil, S. and Salen, K. (2009) *Moving Learning Games Forward, Obstacles Opportunities & Openness*, Cambridge MA: MIT/The Education Arcade, Available: http://education.mit.edu/papers/MovingLearningGamesForward_EdArcade.pdf [2 April 2011]
5. Van Rosmalen, P., Sloep, P., Kester, L., Brouns, F., de Croock, M., and Pannekeet, K. (2008) 'A learner support model based on peer tutor selection', *Journal of Computer Assisted Learning*, vol. 24, pp. 74-86.
6. Van Rosmalen, P., Klemke, R., & Westera, W. (2011) Alleviating the Entrance to Serious Games by Exploring the Use of Commonly Available Tools. In *Proceedings of the 5th European Conference on Games Based Learning*, 20-21 October, Athens, pp. 613-619.
7. Kelle, S., Klemke, R., Gruber, M. and Specht, M. (2011) 'Standardization of Game Based Learning Design', *Proceedings of the International Conference on Computational Science and Applications*. Heidelberg, Berlin, New York: Springer. June 20-23, 2011, Santander, Spain.
8. Dillenbourg, P. (1999). Collaborative Learning: Cognitive and Computational Approaches. *Advances in Learning and Instruction Series*. New York, NY: Elsevier Science, Inc.
9. Anderson, L. & Krathwohl, D. A. (2001) *Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives* New York: Longman.
10. Jonassen, D. (1991). Objectivism vs. constructivism. *Educational Technology. Research and Development*, 39(3), 5-14.
11. Bonwell, C.; Eison, J. (1991). *Active Learning: Creating Excitement in the Classroom. AEHE-ERIC Higher Education Report No. 1*. Washington, D.C.: Jossey-Bass.
12. Kolb, D. A. (1984). Experiential learning: Experience as the source of learning and development. *Journal of Organizational Behavior* (Vol. 8). Prentice-Hall.
13. Hmelo-Silver, C.E. (2004). Problem-based learning: What and How Do Students Learn?. *Educational Psychology Review*, 16(3), 235-266. Springer Netherlands.

A Competence Performance Analyser Tool for Assessing Players' Activity in Serious Games

Elisabetta Parodi¹, Mauro Vannucci¹
¹eXact learning solutions S.p.A., Italy
{e.parodi, m.vannucci}@exactls.com

Abstract. Serious Games are recognized as one of the most promising innovative learning technologies in the short-medium term. Even if it is wide recognized the empowerment of learning they provide, there are few means to trace and measure learners' performances during game sessions. This paper describes a Competence Performance Analyser tool that keep trace of the players' activity in the shape of events in game and basing on these ones assesses the related performances respect to a predefined set of competences.

Keywords: performance assessment, serious games, e-learning, competences, performance indicators

1 Introduction

Game-based learning has grown in recent years as research continues to demonstrate its effectiveness for learning for students of all ages. The greatest potential of games for learning lies in their ability to foster collaboration, problem-solving, and procedural thinking. For a variety of reasons, the realization of this potential is still two to three years away [1].

In the context of the European project TARGET [2], gaming is deemed significant as a conceptual practice with outcomes that enable students to gain skills needed specifically in an information-based culture: a serious game is used to provide work-like learning experiences. The present paper depicts how players' performances are assessed in relation to a set of competences, basing on their observable behaviours in game.

2 The Theory

In the following we outline the background theoretical modelling of game scenarios, competences and performance indicators until the performance assessment model.

2.1 TARGET Scenarios

The project supports three scenarios, all of them dealing with project management. The rationale behind is that after studying on manuals and courses a novel project manager can benefit of experiencing different strategies and behaviours in a safe (consequences-free) environment, such as the serious game one, to approach and face work-life problems, to develop soft skills such as negotiation, trust building, communication. For example, one scenario deals with the need of building a road on a certain land and convincing the owner to sell. Another scenario is about the ability to carry on the products' lifecycle assessment. Last scenario presents the challenges of team recruitment and the player acts as a "Social Architect".

These scenarios are playable stories into a 3D serious game environment, based on Unity 3D¹. Into a realistic context the player can experiment alternative strategies to face every day working problems and challenges, moving across offices and job settings, interacting with colleagues, customers and stakeholders' avatars. The Game platform was extended in such a way to send information (as background, not intrusive events) about specific player's actions and behaviours to the assessment module.

2.2 From Scenarios to Competences and Performance Indicators

A review of the literature, especially about competence modelling for TEL, provided a deeper understanding of the individual competences and the abilities of a person who has the competences; however, this work did not help identify how this ability is affected in different work contexts. This led to the formulation of the OKEI Competence Modelling Framework [3] [4], which identifies different factors of a competence that distinguishes a person's ability to do something, his/her knowledge about something as well as how the ability is exercised by applying the knowledge in a specific context such as within a specific organisation.

The OKEI factors are four:

- **Organisation:** the organizational aspects that influence the work performance and the application of competences, i.e. strategies, values and goals of the organization, work processes, organization structure, roles of people within the organization, the competence profile that one is expected to have is mostly determined by the organization.
- **Knowledge:** the external knowledge resources that could be useful to apply or exercise in the work task at hand, i.e. academic, theoretical or practical knowledge resources.

¹ <http://unity3d.com/>

- Environment: the context outside of the organization, i.e. other companies and industries, networks, public sector and governance, the laws and norms, existing technologies and infrastructure, the market and culture, not to mention the people as consumers, users and citizens.
- Individual: individual and personal factors that may be applied in work situations and that have varying connections to one's performance level, such as knowledge, skills, past experiences, personality traits, mental models, attitudes, motivation, intentions, perceptions and emotions that can either be utilized in work tasks or they influence it in some way.

The OKEI Competence Modelling Framework facilitates the description of competences to the level of detail where elements of the competence can be linked to observable behaviour of people that are able to apply that competence (or reversely, the lack of an ability to apply a competence). Three of the four OKEI Competence Modelling Framework factors, namely the organizational, the knowledge-related and the environmental factors define the "context" in which the competence may be applied. The remaining individual factor describes the competence itself in more detail. Thus, it leads to a specification of the competence and/or to the definition of related sub-competences. Based on specific competences or on more specific sub-competences, it is possible to identify behavioural indicators.

The behavioural indicators, in turn, can be used to derive performance indicators for the learners, which can be used in the formative evaluations of the learners [5]. A performance indicator is a concrete instantiation of a behavioural indicator tailored to the TARGET game to be used to facilitate competence development. The more contextual factors, i.e. mediating variables, are taken into account, and the more possible values for each variable, the more complex the process of operationalization becomes, leading to a formula such as a multiple regression equation.

As an example, the communication competence is calculated basing on trust building, non-verbal and verbal communication sub-competences. Non verbal communication is calculated using the "proxemics" performance indicator. Personal space (or proxemics) [6] can be defined as the area individuals maintain around themselves into which others cannot intrude without arousing discomfort. Which (range of) physical distance between two persons can be seen as appropriate, i.e. which distance doesn't arouse discomfort or stress, is mediated by a great amount of contextual factors, such as cultural background of the other(s), status differences, amount of people, overall available space, etc...

So the appropriate physical distance d_{appr} can be calculated with the following multiple regression formula:

$$d_{appr} = d_{contact} + x_{c_b} (d_{noncontact} - d_{contact}) + x_n * d_{contact} + x_{sd} * d_{std}$$

with the parameters:

$$x_{c_b} = 1 : \text{if the NPC has a noncontact cultural background}$$

$$0 : \text{if the NPC has a contact cultural background}$$

$$x_n = 0 : \text{if } n \leq 4 \quad (= \text{Personal Space})$$

$$5 : \text{if } n > 4 \text{ and } n \leq 8 \quad (= \text{Social Space})$$

$$13 : \text{if } n > 8 \quad (= \text{Public Space})$$

x_{sd} = 1 : if the status of the NPC is higher than the status of the avatar
0 : if the status of the NPC is equal or lower than the status of the avatar

This short introduction to the OKEI model was meant to provide just an overview of the theoretical basis of the CPA module and of course didn't mean and neither could be exhaustive. For further information please refer to the related documentation, as from [3], [4], [5].

3 The Implementation

In this section we describe how the previous concepts have led to the implementation of a Competence Performance Analyser software module.

3.1 Tracing Player's Performance

While the learner plays, the Game traces his/her behaviour and provides data to a dedicated software module called Competence Performance Analyser (CPA) that elaborates the information and assesses the performance. This means that specific actions of the player are recorded, for example movements into the 3D environment, expression of emotions and text written in chat. These raw data are used by the CPA to calculate performance indicators and in turn, grounding on these ones, to assess performances respect to competences. Which actions have to be monitored and how to combine them to assess the performance was elaborated via the methodology described by the previous section.

3.2 Competence Performance Assessment

The Competence Performance Analyser module implements the assessment of the player's performance as from previously discussed theoretical basis: competences are assessed as a weighted sum of certain performance indicators, that are calculated basing on player's actions in Game.

The CPA module is made up via a number of internal components, as from Fig. 1.

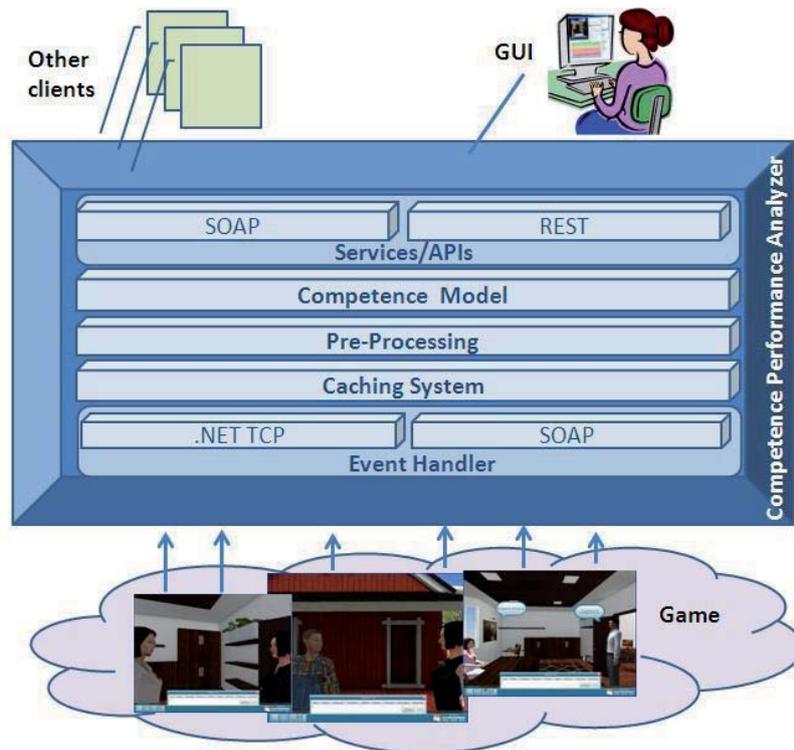


Fig. 1. CPA Architecture

The results of the CPA calculations are presented to the user in a graphical, intuitive manner af from the CPA GUI, that is the higher level in Fig. 1 and is presented in Fig. 2.

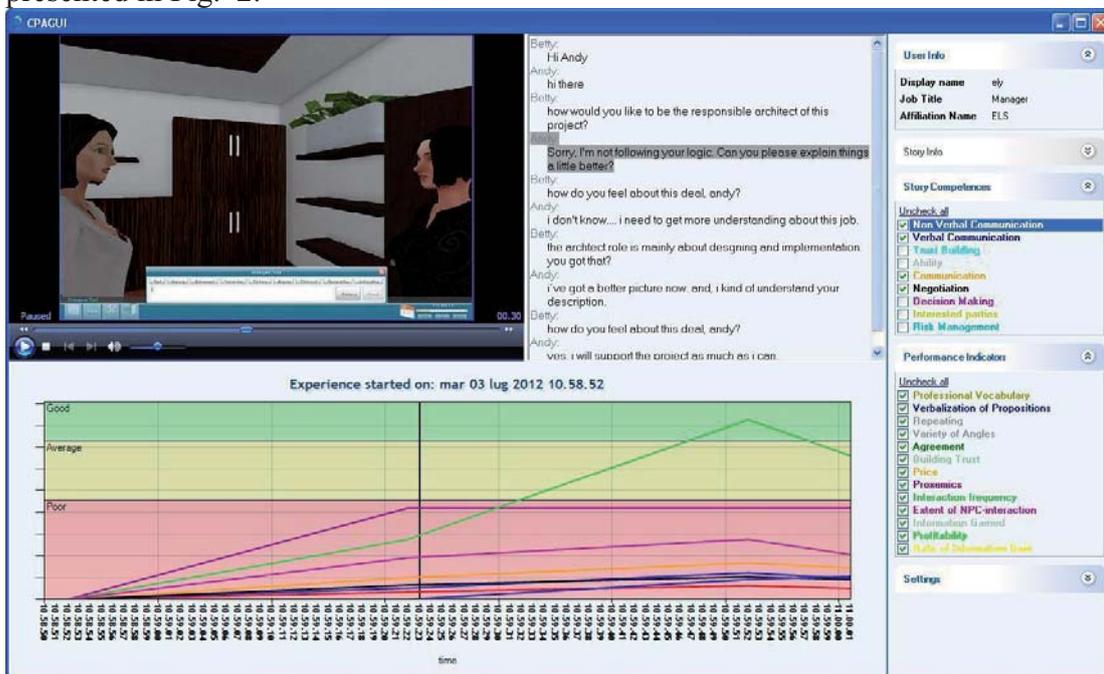


Fig. 2. CPA GUI

The CPA GUI is the interface between the computing back-end software and the human end user. The GUI has to present information in such a way to provide an effective reflection means, where it is straightforward to understand which action/sentence led to a specific assessment and why. So the GUI has three distinct areas:

- Experience Replay (top left), where the user can play-back his game session. The play button functions both as play and pause button and by clicking on a particular point of the progress bar it makes the play-back go to that point. As with this kind of visualization it can be a bit hard to read the chat lines, the text is displayed on the right of the video area and sentences are highlighted synchronously with the replay.
- Performance Graph (bottom), showing how the player's performance evolved along the time for each competence and performance indicator involved in the game scenario. Competences and Performance indicators can be selected and unselected as needed/wished from the lists on the right side.
- User and story information (top right) , with a few data about the "owner" of the experience, such as username and job title. In this way a minimal background about the player is provided (with job title), thus respecting his privacy (with anonymization with username), avoiding a specific identification, but still allowing, for example, browsing his learning path, i.e. by looking for his game experiences, comments and annotations. A synthetic description of the story is provided too.

This kind of visualization was meant to support reflection upon the learning game experience. The replay of the experience presents a video to show what the user was doing at a specific moment into the Game, a highlight of the chat, to show what the user was saying, and a graph showing the assessment of competences and performance indicators, with a bar moving through this graph synchronously with the progress of the other data. The early version of the interface had a quite different layout, the current display mode was re-arranged after an evaluation cycle to grant better usability and effectiveness of the provided information.

Behind the scenes, SOAP and REST APIs grant access to the CPA methods via programming interfaces. Calculations run into the back end thanks to:

- Competence Model (CM) and Pre-Processing modules, responsible for collecting pre-processed data and calculating discrete values for performance indicators and competences. The competence model for a given competence contains all information about the performance indicators used and the formula to assess the performance and trend for the competence and the performance indicator. The Pre-Processing component elaborates raw data and makes them homogeneous. This is needed because raw data arrive at different rates and times: the Pre-Processing component is aimed to interpolate missing data, if possible, and to send back to the competence model comparable data. To complete the process all the data are used to calculate the performance, applying the formula from the competence model Game Status Connector. This process is depicted by Fig. 3.

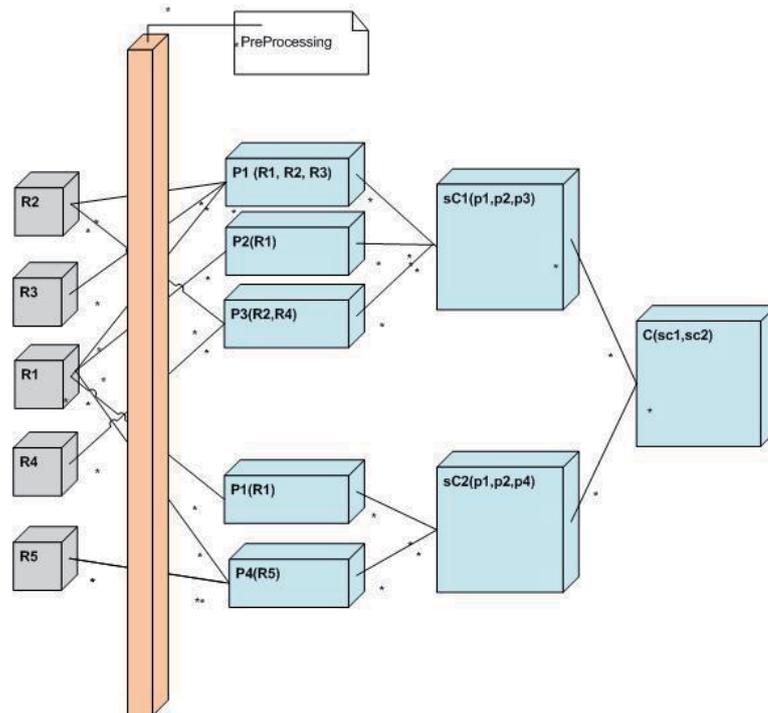


Fig. 3. Data Elaboration inside the CPA.

From Fig. 3 it is possible to understand better how the competence model works. The raw data (R1..Rn) are pre-processed. A function $P_x(R_k, \dots, R_y)$ is applied to these values over the time and the resulting value of the function represents the Performance indicator value at a fixed time. sC_x is a sub competence. A sub competence is itself a competence. The performance related to a competence is calculated as a combination of P_x formula or combination of sub competence formula. When the data are ready an array of pair $\langle \text{time}, \text{value} \rangle$ in the requested period of time is returned to the above component.

- A Caching module, for a faster access to data.
- Event-Handler modules receive and dispatch events from Game. As the Game needs to send a large amount of data (i.e. performance indicator related data, game status, etc..) a .NET WCF RPC based event handler provides a good integration means but it could create integration problems with other components that are not .NET based. For this reason, a second event handler is provided, exposing its interface through a more classical web services based on SOAP protocol.

Data about player's performance can so be accessed in two ways:

- in a graphical manner, via the CPA GUI, showing information and assessment about a specific experience, that is re-proposed in real time
- at service level, via REST and SOAP web services, data can be requested at different levels of granularity. This allows further elaborations, comparisons and evaluations.

4 Conclusion and Further Work

We have presented here the Competence Performance Analyser module developed within the TARGET project. We have presented both the theoretical

basis as well as the implementation criteria and outcomes. The project is now at its final phase. As explained before, early evaluations cycles were already run and gave a good feedback about the CPA module; more evaluations are running at the time of writing, so we can't report more users' feedback.

We think that some ideas at the basics of our work can be further elaborated and/or reused:

- the criteria for usability and effectiveness of the GUI for supporting reflection could be further elaborated and researched
- the basic idea of calculating assessment for competences and performance indicators can be reused and the code can be properly updated to get raw data from different events in different contexts and settings (i.e. data about the user's interaction with online courses instead of serious games, like SCORM tracking ones)
- social aspects could be added, as the possibility for community users of annotating, tagging and commenting specific parts of the experience
- with further extensions, the tool could become configurable enough to allow an average, not developer end user (i.e. a trainer) to provide own formulas for performance indicators and competences – while so far this is coded into the back end
- the assessment data could be compared per single user over time and/or per groups of users with common characteristics, researching i.e. learning evolution, common behaviour patterns, etc... supporting learning analytics
- several users' performance data across the game could be compared to detect frequent problems, errors or even the simplest challenges, to be able to re-design and enhance.

5 Acknowledgement

This research has been co-funded by the European Commission within the IP project TARGET, grant FP7-ICT-231717-TARGET.

References

- [1] L. Johnson, R. Smith, H. Willis, A. Levine, K. Haywood, «The Horizon Report 2011,» Higher Ed., 2011.
- [2] «TARGET - Transformative, Adaptive, Responsive and enGaging Environment,» [Online]. Available: <http://www.reachyourtarget.org>.
- [3] S. A. Petersen, T. Heikura, G. Cerinšek, M. Bedek, «D4.1 Competence Portfolios,» 2011.
- [4] S. H. T. Petersen, «Modelling Project Management and Innovation Competences for Technology-Enhanced Learning,» in *e-challenges*, Warsaw, Poland, 2010.
- [5] V. Lia, E. Parodi, M. Bedek, B. Cowley et al., «D4.4 TARGET Monitoring and Assessment Component,» 2011.
- [6] R. Sommer, *Personal Space. The Behavioral Basis of Design.*, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1969.

Using Out-of-Character Reasoning to Combine Storytelling and Education in a Serious Game

Jeroen Linssen and Mariët Theune

Human Media Interaction, University of Twente, Enschede, the Netherlands
{j.m.linssen,m.theune}@utwente.nl

Abstract. To reconcile storytelling and educational meta-goals in the context of a serious game, we propose to make use of out-of-character reasoning in virtual agents. We will implement these agents in a serious game of our design, which will focus on social interaction in conflict scenarios with the meta-goal of improving social awareness of users. The agents will use out-of-character reasoning to manage conflicts by assuming different in-character personalities or by planning to take specific actions based on interaction with the users. In-character reasoning is responsible for the storytelling concerns of character believability and consistency. These are not endangered by out-of-character reasoning, as it takes in-character information into account when making decisions.

Keywords: Interactive Storytelling, Drama Management, Autonomous Agents, Emergent Narrative, Serious Games, Conflict, Social Behaviour.

1 Introduction

A well-known issue in the field of interactive storytelling is the trade-off between users' freedom of action and a pre-authored plot: the narrative paradox. We wish to extend previous work on this paradox by addressing it in the context of serious games. Such games are defined by having educational meta-goals that need to be attained. In our opinion, the necessity of this attainment is similar to having a pre-authored plot as both require the story to evolve in a particular way. Thus, we focus on finding a balance between freedom of action and achieving a story that supports attainment of educational meta-goals.

We will develop a game dealing with social interaction in the domain of law enforcement by police officers. The meta-goal of our game is improving users' social awareness and it focuses on social behaviour in conflict situations. Scenarios include settings in which police officers are required to interact with civilians, e.g., loitering juveniles.

We base our research on work done on the Virtual Storyteller (VST) [6], a story generation system that uses an *emergent narrative* approach. This enables characters in a storyworld to carry out actions autonomously so that a story emerges from their joint behaviour. Such an agent-based approach offers much more freedom of action than simple branching narratives. For instance, in a scenario in our serious game that involves loitering juveniles, a police officer has

a variety of options to confront them, e.g., by rapidly approaching them, demanding respect with a loud voice or by taking a more calm, submissive stance towards the youngsters. These approaches will have different effects on the juveniles' reactions and, accordingly, on the emergent story. For a first prototype of our serious game, we will build on the recently developed interactive version of the VST architecture [1]. This allows users to control one or more characters in the story, while other characters are controlled by autonomous agents (as in previous versions). The first prototype will have menu-based interaction and a 2D graphical interface. Ultimately, components of our research will be used in a multi-modal 3D training environment.¹

Currently, the VST uses techniques from improvisational theatre, in which the agents controlling the non-player characters can reason *out-of-character* to determine which actions to take *in-character*, i.e., as a character in the story-world. We aim to extend this mechanism to equip virtual agents with the means to reason about meta-goals and make decisions in such a manner that educational prerequisites are satisfied by the emergent narrative, while maintaining character believability.

Generally, narratives revolve around a central conflict between a protagonist and an antagonist that emerges from their disagreement on some topic. This idea of conflict coincides with the domain of our serious game; therefore, we assume that a conceptual understanding and formalisation of conflict can assist out-of-character reasoning in agents.

2 Related Work

The concept of out-of-character (OOC) reasoning was first explored and implemented in the context of FATiMA, the architecture underlying the serious game FearNot! The FATiMA architecture allows agents to choose to perform actions that have the highest emotional impact [5]. Thus the emergent narrative can be guided in a distributed manner.

OOC reasoning in the VST continues this line of work by using several improv techniques, in particular that of *late commitment* [6]. This lets the system's agents assert facts of any kind that were until then undecided upon, if they are of use for story progression. For instance, an agent can assert OOC that its character has a key in his possession to open a locked door so that the narrative does not become dull or end at this point.

Because of the importance of conflict for narratives, recent work [2, 7] has attempted to formalise the concept of conflict in order to implement it in serious games and story generation systems, respectively. However, the former only deals with resource management for different characters and does not tackle user-character conflicts, while the latter only considers planning as a cause for conflict.

¹ See <http://www.commit-nl.nl/projects/interaction-for-universal-access/> for a description of the Dutch national project that is the context of our research.

3 Using OOC Reasoning to Combine Diegesis and Pedagogy

In this paper, we argue for an emergent narrative approach that splits agents' minds into in-character (IC) and OOC parts. By doing so, we can separate diegetic, i.e., storytelling, and pedagogic concerns between these parts—the IC part being responsible for the former and the OOC part for the latter. Nonetheless, the two are linked as an agent's OOC part can access IC information and take it into account when making decisions such as taking a particular action, to ensure that character believability and consistency are maintained. For example, an agent can use late commitment to explain possibly inconsistent character actions.

The distributed nature of the emergent narrative approach has several advantages over a centralised approach. Because each of the agents has its own beliefs about the world, it can behave in a believable manner that is unique to itself. Its behaviour is motivated by its own experiences and is thus consistent. This would not necessarily be the case if it was directly controlled by a central entity. Thus, emergent narrative lets the agents' IC reasoning account for diegetic concerns of consistency and believability.

Conversely, the pedagogical concerns can be satisfied by letting the agents reason about the educational meta-goals OOC. To improve users' social awareness, they require feedback on the possible effects of their social behaviour. We postulate that this can be done by letting agents reason OOC to let the conflict escalate. Dutch law enforcers are currently training conflict scenarios with the help of a theory about social interaction called the interpersonal circumplex [3]. Central to this theory is the notion that certain personalities oppose each other. We wish to incorporate this idea in our system by letting agents reason OOC, e.g., by letting them use late commitment, to adopt IC personalities that oppose that of the user so that the conflict intensifies.

After this OOC intervention, agents' IC behaviour should be able to let the conflict develop in a natural way, i.e., reaching a climax and having positive or negative resolution, which should reflect the effectiveness of the user's (change in) approach. As conflicts may not always emerge spontaneously following this structure, agents monitor the goings-on OOC to guide the story where necessary, e.g., by exaggerating their IC personality to stir up the conflict. We believe that our OOC reasoning mechanism should draw from the fields of narrative and sociological research by combining their definitions and theories of conflicts, cf. dramatic *story arcs* such as Freytag's Pyramid [4, pp.99–101] and the concept of a *conflict cycle* [2]. This will enable agents to monitor the conflict so that it develops in a natural way. For example, an agent may incite a conflict by letting its character take offence at a certain utterance of the user. This can be explained by letting it use late commitment OOC and adopt a personality that is consistent with this behaviour. If the user does not adapt his behaviour hereafter, he seemingly did not learn from the juveniles' reactions as his approach triggered the conflict. Then, the agents could reason OOC to introduce a 'bad' ending, e.g., the juveniles running away, continuing to loiter in another place. If

the user had adapted his approach to the juveniles, a better resolution would have ensued.

4 Conclusions

The issue of conjoining diegesis and pedagogy seems similar to overcoming the narrative paradox. Therefore, we aim to expand previous forays into OOC reasoning—proposed as a way to overcome the narrative paradox—by equipping virtual agents with the means to reason OOC about conflicts so that the meta-goal of our serious game—improving social awareness—can be attained by users. Our idea is that agents can use OOC reasoning to influence their IC personalities through late commitment and monitor the conflict as it develops to take additional actions if necessary to provide a learning example. By taking IC information into account during OOC reasoning, agents can choose IC actions that do not lead to a decrease in character believability and consistency. As the story evolves in reaction to players' actions, they become aware of their own social behaviour and learn that different individuals require different approaches.

We believe that, by implementing this distinction between IC and OOC reasoning in agents, the game will offer a high degree of interactivity and freedom with consistent and believable characters, while at the same time ensuring that the educational meta-goals are reached. We have recently started development of the game prototype in which we intend to implement our ideas. Future work will focus on a more developed theory of conflict based on narrative and sociological research, combining this with the interpersonal circumplex theory.

Acknowledgements. This publication was supported by the Dutch national program COMMIT.

References

1. Alofs, T., Theune, M., Swartjes, I.: A tabletop board game interface for multi-user interaction with a storytelling system. In: Proc. of INTETAIN '11. pp. 123–128 (2011)
2. Cheong, Y.-G., Khaled, R., Grappiolo, C., Campos, J., Martinho, C., Ingram, G., Paiva, A., Yannakakis, G.N.: A computational approach towards conflict resolution for serious games. In: Proc. of FDG '11. pp. 15–22 (2011)
3. Gurtman, M.B.: Exploring personality with the interpersonal circumplex. *Soc. and Personal. Psychol. Compass* 3(4), 601–619 (2009)
4. Lethbridge, S., Mildorf, J.: *Basics of English studies: An introductory course for students of literary studies in English*. Freiburg University, Freiburg im Breisgau (2003)
5. Louchart, S., Aylett, R.: Building synthetic actors for interactive dramas. In: Proc. of the AAAI Fall Symposium on Intelligent Narrative Technologies. pp. 63–71 (2007)
6. Swartjes, I.: *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*. Ph.D. thesis, University of Twente, Enschede (2010)
7. Ware, S.G., Young, R.M.: Modeling narrative conflict to generate interesting stories. In: Proc. of AIIDE '10 (2010)

The TERENCE Smart Games: Automatic Generation and Supporting Architecture

Fernando De la Prieta¹, Tania Di Mascio², Rosella Gennari³, Ivana Marenzi⁴, and Pierpaolo Vittorini⁵

¹ U. of Salamanca, Department of Computer Science, Plaza de la Merced s/n, 37008, Salamanca, Spain

² U. of L'Aquila, DIEI, Via G. Gronchi 18, 67100, L'Aquila, Italy

³ Free U. of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy

⁴ L3S, University of Hanover, Germany

⁵ U. of L'Aquila, Dep. of MeSVA, V.le S. Salvatore, Edificio Delta 6, 67100, L'Aquila, Italy

Abstract. TERENCE is an FP7 ICT European project 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 smart games, which stimulate inference-making for story comprehension, and relaxing games, which stimulate visual perception and not story comprehension. The paper focuses on smart games. It first shows the current prototypes, then it describes the TERENCE system architecture, thus it delves into the generation of smart games instances, by highlighting the role of the constraint-based module therein. Finally, it ends with short conclusions about the planned improvements.

Keywords: Game frameworks, serious games architectures, reusability, user centred design

1 Introduction

Nowadays, circa 10% of young children are estimated to be poor (text) comprehenders [14]: they are proficient in word decoding and other low-level cognitive skills, but they show problems in deep text comprehension [8, 15]. TERENCE [18] is a European ICT multidisciplinary project, for the technology enhanced learning (TEL) area, that is developing the first *adaptive learning systems* (ALS, [6, 7]) for improving the reading comprehension of 8–10 year old poor comprehenders.

The learning material of TERENCE is made of stories and games, written in the two languages of the project, Italian and English. The learning material was designed following the user centred and evidence-based design, see [9]. The models for the learning material and learners of the system are in [2], and the first adaptation rules are in [3], whereas [4] explains how the models and learner model, in particular, stem from an extensive context of use and requirement analysis [17, 11]. In particular, the TERENCE games are distinguished by the evidence-based pedagogical plan into smart and relaxing. Smart games in TERENCE consist of reasoning tasks about the characters and events of a story, as well as on their relations, and are designed on effective pedagogical interventions by educators as well as pedagogy and therapy experts for assessing

and stimulating story comprehension, see [10]. As such, smart games are cognitively demanding for the learner, whereas relaxing games allow the learner to have breaks.

This paper, which is of a technical nature, focuses on the automatic generation of smart games using the TERENCE framework, as well as on the underlying architecture. Sec. 2 introduces background concepts, so as to make the reader grasp what the smart games are like. With such background and intuitions, the paper shows prototypes of smart games (Sec. 3), then delves into the TERENCE architecture and the methodology for the automatic generation of smart games from stories (Sec. 4).

2 Background: Pedagogically-driven Reading Interventions for Smart Games

The TERENCE smart games mainly serve for stimulating the learner in enhancing reading comprehension. How the design of the games is based on effective reading interventions, selected by the TERENCE pedagogical plan, is described in [12, 10]. In brief, the plan guides the process of deep text comprehension by proposing increasingly demanding tasks: firstly, it makes the learner reason about the characters that are in the story, then about the events that take place in the story, hence about temporal relations among the events, and finally about causal-temporal relations among events. For instance, let us consider the following story excerpt.

Perla took a book from the library shelf [...] She watched the clowns in the circus. They were throwing cakes here and there [...] The little girl ducked just in time to avoid the flying cake.

Typical reading interventions, selected by the plan, ask the reader to infer that:

- (a) “Perla watches the clowns in the circus” (the anaphoric resolution “She” → “Perla” is needed),
- (b) “Perla takes a book” and “Perla watches the clowns in the circus” (though, for instance, Perla does not throw cakes),
- (c) “Perla takes a book from the library shelf” before “Perla watches the clowns in the circus”, and
- (d) “Perla ducks” because “The clowns throw cakes”.

The TERENCE smart games embed such interventions in game-like environments. They are organised as in the taxonomy in Fig. 1. The children of the root are so defined:

- (1) *characters*: smart games concerning characters, namely, who an agent of a story’s event is (who-game, e.g. the example in point (a) above), what a character does in the story (what-game, e.g. the example in point (b) above);
- (2) *time*: smart games for reasoning about temporal relations between events of the story, purely sequential (before/after-game, e.g. the example in point (c) above) or not (all the others);
- (3) *causality*: smart games 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-game, e.g. the example in point (d) above).

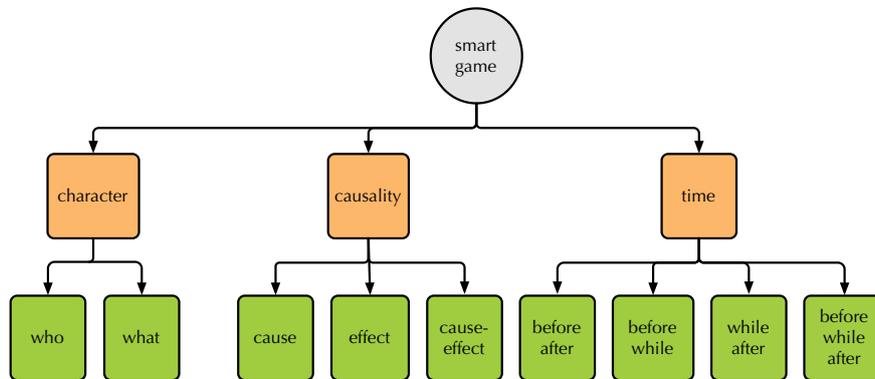


Fig. 1. Hierarchy of the TERENCE smart games

3 Prototypes of Smart Games

For giving the reader a concrete feeling of how reading interventions were turned into smart games in TERENCE, this brief section shows prototypes of smart games. However, we only pause on the features of the prototypes that are relevant for explaining the generation of smart games, which is the focus of the remainder of this paper.

Fig. 2 and 3 show two prototypes, at different states. In all prototypes, the interface is split in two areas. The top area contains the points for the game, and the avatar of the learner. The bottom area is divided into two parts. One part contains the main question. The other part shows the choices available to the learner in the current state. The feedback can be different, see [12]. For instance, in the shown prototypes, the reader can see a consistency feedback, in the form of a yes/no visual message, displayed on top of the available choices; an explanatory feedback for wrong solutions is shown in the form of a red light bulb between two events in case the learner places them in a temporal relation that is inconsistent with the story.



Fig. 2. Screenshots of a what game prototype.

Now, more in details, the left screenshot in Fig. 2 shows an intermediate state of a multiple-choice *what* game: the question posed to the learner is “What happens in the story?”. The learner has to choose and drag one of the choices as a key, and see if it opens (or not) the cupboard. The right screenshot in Fig. 2 shows the feedback concerning the (in)consistency of the choice with the story in the form of a *no* visual message. If the resolution is correct, the locker opens and a reward drops. Notice that the visual metaphor are adapted to the age of the learner, e.g., a locker is used for 9-11 learners whereas a cupboard is used for 7–9 learners.

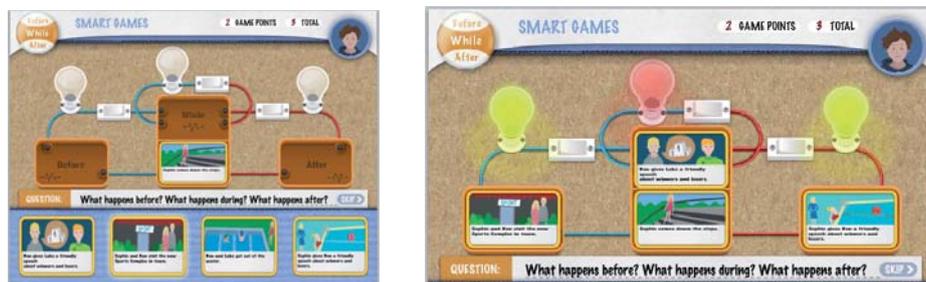


Fig. 3. Screenshots of a time game prototype.

The left screenshot in Fig. 3 shows the initial state of an ordering time game: the learner has to establish before, while and after relations with the event displayed in the centre of the top area. To do so, the learner has to choose and drag events from the bottom area, and drop them into the appropriate empty container in the top area. The right screenshot in Fig. 3 shows an explanatory feedback in the top area: correctly placed relations are shown with yellow greens; the wrongly placed relation is signalled with a red bulb. Here as well, a different metaphor is used with younger learners, with water and mill-wheels in place of electricity and light bulbs.



Fig. 4. The smart games activity: instructions and avatar feedback.

Finally, the left screenshot in Fig. 4 shows the instructions presented to a learner before a smart game starts. In order to maximise the chances that the learner reads the instructions, five seconds of a fake loading activity are added. Afterwards, a “Play game” button is shown. The right screenshot in Fig. 4 shows that, after each game, the avatar gives a further feedback to the learner: in case the learner solves the game correctly, an happy avatar is shown; otherwise a sad avatar is displayed.

4 System Architecture

Fig. 5 shows the main components of the TERENCE system. Although the figures display them in layers, all work independently and the information is exchanged through RESTful web services. In this manner, high cohesion and low linkage is ensured, as well as the re-usability of each component.

An overview of the whole system, organized in layers, is shown below.

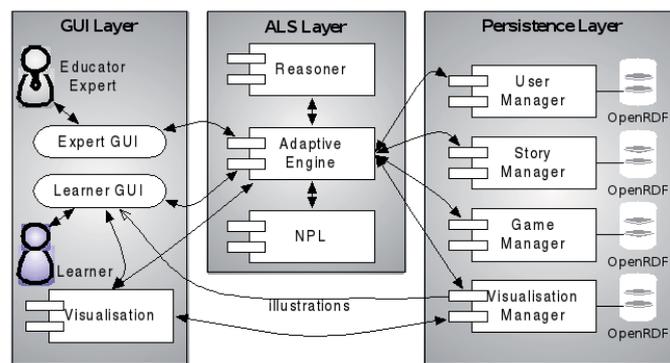


Fig. 5. The architecture.

GUI Layer. This functional layer is divided into three main components:

- the Educator GUI, that is a Rich Internet Application (RIA) built on top of the Vaadin framework [19] that provides the necessarily facilities to manage the TERENCE learning material;
- the Learner GUI, that is designed to be used in normal PCs as well as tablets, and provides a Flash environment where learners read stories and play with the TERENCE smart games;
- the Visualisation module that is in charge of preparing the visual information (e.g., illustrations, templates) that are used in the Learner GUI. For instance, concerning the smart game in Fig. 2, the visualisation module provides the background image, the boxes, the question and the avatar.

ALS Layer. This layer contains the core system components:

- the NPL module, which is in charge of annotating the stories with tags useful for generating the TERENCE smart games;
- the Reasoner module that (i) checks the consistency of the annotations of each story, (ii) enriches the story with further temporal annotations missed during the annotation process, and (iii) automatically generates smart game instances from the enriched annotations;
- the Adaptive Engine, which orchestrates the other components and establishes adaptive sessions for each learner, by taking into account their profiles.

Persistence Layer. This layer is divided into four components, that is, the User, Story, Game and Visualisation Manager, each with its own repository. The repositories are based on RDF ontologies. This schemaless model has two main advantages: it enables the possibility of evolving the system (adding more functionalities or refining existing ones) without large efforts, and allows for further reasoning services about the stored data.

Among the various functionalities of the system, we focus on the management of the smart games, and specifically on how the smart games are generated. Fig. 6 shows an overview of the entire process.

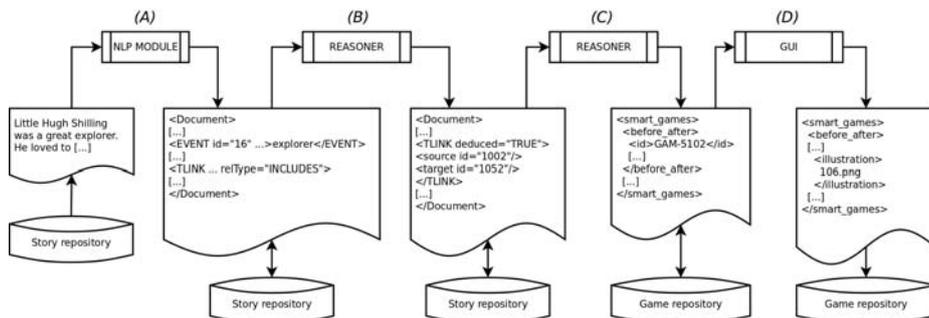


Fig. 6. Smart games generation process

Phase A. Firstly, from a story text contained in the story repository, the NPL module generates a story annotated with a variant of the TimeML language that was extended in [16] with tags for information that is relevant for the TERENCE smart games, e.g., the ENTITY and CLINK tags, that aim, respectively, to represent the entity related to an event, and the causal-temporal relations between two events. The annotated story is then stored in the same repository.

Phase B. Then, the reasoner checks the consistency of the annotations, detects the eventual temporal inconsistencies, and enriches the annotations by adding deduced temporal relations as further TLINK tags. This new consistent and enriched story is also stored in the story repository.

Phase C. Afterwards, starting from the enriched story, the reasoner module generates automatically instances of smart games that are stored in the game repository.

Phase D. Finally, a manual revision of the generated smart game instances takes place, where the related visuals (e.g. background illustrations, buttons) are also specified.

In the following, we delve into phase C of the process. Starting from a story s , annotated and then enriched as explained above, the smart games generation goes as follows:

Algorithm 1 The algorithm for generating smart game instances.

Require: story s , number of events n , number of games k

```

1: foreach event  $e$  in  $s$  do
2:   generate all types of games for event  $e$ 
3: end for
4: sort events
5: keep the games for the first  $n$  events
6: reduce the total number of games to fixed number  $k$ 

```

Algorithm 1 initially iterates among all events, tagged with the EVENT tag in story s . Iteratively, an event e is selected as the *fixed event* for the generation process. Then, the algorithm generates instances of smart games with e and other events (lines 1–3).

For example, let us consider a time before-after games, shown in Fig. 3. The fixed event is displayed as the central even in the figure. Then the algorithm, using specific heuristics, finds an event that happens before the fixed event, and one that happens after the fixed event, and a further event that does not happen before or after the fixed event in the story. Accordingly, all possible before-after games for the fixed event e are generated as follows:

Algorithm 2 The before-after games generation algorithm.

Require: event e , story s , story $s_o \neq s$

```

1: foreach tlink  $t_1$  in  $s$ , that has  $e$  as target do
2:   foreach tlink  $t_2$  in  $s$ , that has  $e$  as source do
3:     select a random event  $w$  from story  $s_o$ 
4:     create a before/after game that has:
5:        $e$  as fixed event
6:       the source event of  $t_1$  as before
7:       the target event of  $t_2$  as after
8:        $w$  as wrong event
9:   end for
10: end for

```

Algorithm 2 initially selects all pairs of temporal relations, that is, TLINKS that have e as target/source in the enriched annotated story. For each pair $[t_1, t_2]$, it first tries

to select an event that is related with a while relation with e . In case of failure, it then selects a fake event w that does not happen in the story. For further details on how each type of game is generated, please refer to [1].

After all possible games are generated, Algorithm 1 produces an ordered list of fixed events (line 4) according to the following heuristics. Given two fixed events e_1 and e_2 , in order to decide if $e_1 > e_2$, we compare the related number of generated games, weighting these according to their difficulty level, established by the stimulation plan. In other words, $e_1 > e_2$ if the number of cause-effect games for e_1 is higher than for e_2 . If equal, $e_1 > e_2$ if the number of effect games for e_1 is higher than for e_2 . If still equal, we compare the number of cause games, etc.

After the ordering, two pruning phases take place. The first keeps only the games for the first n fixed events (line 5) in the ordered list. The second pruning is concerned with the total number of smart games, reduced to a fixed number k . For each game of a given level, the algorithm keeps with different reasoning complexity, e.g., before-after games with both “deduced” events, implicit in the text, as well as who-games and what-games with both “protagonists” and “secondary” characters as agents (line 6). For further details on how games are removed from the list, please refer to [13].

Fig. 7 shows an excerpt of an XML document that contains the textual instance of a before-after game (lines 4–24), conforming to the conceptual models defined in [5], generated as described above. The game contains the fixed event (line 22), and three choices representing the event that happens before the fixed event (lines 17–21), the event that happens after the fixed event (lines 12–16), and the wrong choice (lines 7–11). Note that each choice contains:

- its correctness, i.e. if the choice is (i) always correct, (ii) correct if selected as before, after, or while, (iii) correct if selected as cause or effect; for instance, line 13 states that the second choice is correct if selected as the event that happened after the fixed event;
- the event it represents (e.g. line 14 for the second choice), and
- the text that will be shown to the learner (e.g. line 15 for the second choice).

Note that:

- the textual sentences/questions contained in the example (i.e., instructions, textual labels of the illustrated events) require manual rewritings so to become comprehensible to learners. For instance, the textual label for the game fixed event (line 23) is simply “asked”¹. Since not enough explicative, it must be manually rewritten, e.g., into “Little Hugh asked the little boy why the streets were empty”. However, it is planned that the next release of the NLP module will implement an automated functionality that generates such textual sentences;
- the heuristic for selecting the relevant events has still to be assessed and improved, e.g., by adjusting the heuristic through a comparison of the results with manually selected events.

¹ So far, the field is always initialised with the event text.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <smart_games>
3   ...
4   <before_after>
5     <id>GAM-5102</id>
6     <type>TERENCE_SG_BEFORE_AFTER</type>
7     <choices>
8       <correct>WRONG</correct>
9       <event id="550" modality="NONE" morphology="NONE"> stand </event>
10      <text> stand </text>
11    </choices>
12    <choices>
13      <correct>AFTER</correct>
14      <event id="644" modality="NONE" morphology="NONE"> blackened </event>
15      <text> blackened </text>
16    </choices>
17    <choices>
18      <correct>BEFORE</correct>
19      <event id="166" modality="NONE" morphology="NONE"> looked </event>
20      <text> looked </text>
21    </choices>
22    <event id="558" modality="NONE" morphology="NONE"> asked </event>
23    <text> asked </text>
24  </before_after>
25  ...
26 </smart_games>

```

Fig. 7. An example of a textual instance of a smart game

5 Conclusions

The paper described the architecture supporting the management of the TERENCE smart games, starting from their generation until their usage in the prototypes currently developed. The design of the current prototypes is being revised in light of the results of small-scale usability evaluations, expert based evaluations by psychologists and interaction designers. On-going work is considering the adaptation of the feedback to the learner, the automatic generation of the explanatory feedback by the reasoning module, improvements in the automatic generation of the textual instances of the smart games, and improvements in the performances of the overall architecture.

Acknowledgments. This work was supported by the TERENCE project, funded by the EC through FP7 for RTD, ICT-2009.4.2. The contents of the paper reflects only the authors' view and the EC is not liable for it. The authors thank MOME for producing the images of events in snapshots of games. Gennari work was also funded through the CRESCO and DARE projects, financed by LUB and Province of Bozen-Bolzano.

References

1. Alrifai, M., Gennari, R.: Deliverable 2.3: Game Design. Tech. Rep. D2.3, TERENCE project (2012)

2. Alrifai, M., Gennari, R., Tifrea, O., Vittorini, P.: The Domain and User Models of the TERENCE Adaptive Learning System. In: Vittorini, P., Gennari, R., Marenzi, I., de la Prieta, F., Corchado, J.M. (eds.) Proc. of the ebTEL 2012 workshop co-located with PAAMS 2012. Soft Computing, Springer (2012)
3. 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.) Proc. of the ebTEL 2012 workshop co-located with PAAMS 2012. Soft Computing, Springer (2012)
4. 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 ICALT 2012. IEEE Press (2012)
5. Alrifai, P.: Deliverable 2.1: Conceptual Model's Specification. Tech. Rep. D2.1, TERENCE project (2011)
6. 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)
7. Brusilovsky, P., Karagiannidis, C., Sampson, D.: Layered evaluation of adaptive learning systems. International Journal of Continuing Engineering Education and Lifelong Learning (14), 402–421 (2004)
8. Cain, K., Oakhill, J.V., Barnes, M.A., Bryant, P.E.: Comprehension Skill Inference Making Ability and their Relation to Knowledge. Memory and Cognition 29, 850–859 (2001)
9. Cofini, V., Di Giacomo, D., Di Mascio, T., Necozone, S., Vittorini, P.: Evaluation plan of terence: when the user-centred design meets the evidence-based approach. In: Vittorini, P., Gennari, R., Marenzi, I., de la Prieta, F., Corchado, J.M. (eds.) Proc. of the ebTEL 2012 workshop co-located with PAAMS 2012. Soft Computing, Springer (2012)
10. De la Prieta, F., Di Mascio, T., Gennari, R., Marenzi, I., Vittorini, P.: Playing for improving the reading comprehension skills of primary school poor comprehenders. In: Proc. of the PDSG 2012 workshop. CEUR-WS (2012)
11. Di Mascio, T., Gennari, R., Melonio, A., Vittorini, P.: The user classes building process in a tel project. In: Vittorini, P., Gennari, R., Marenzi, I., de la Prieta, F., Corchado, J.M. (eds.) Proc. of the ebTEL 2012 workshop co-located with PAAMS 2012. Soft Computing, Springer (2012)
12. Gennari, R.: Deliverable 4.1: State of the Art and Design of Novel Intelligent Feedback. Tech. rep., TERENCE project (2011)
13. Gennari, R.: Deliverable 4.2: Automated Reasoning Module. Tech. Rep. D4.2, TERENCE project (2012)
14. Lyon, G., Fletcher, J., Barnes, M.: Child Psychopathology, chap. Learning Disabilities. NY: The Guilford Press (2003)
15. Marschark, M., Sapere, P., Convertino, C., Mayer, C., W.L., Sarchet, T.: Are Deaf Students' Reading Challenges Really About Reading? (In Press)
16. Moens, S.: Deliverable 3.1: State of the Art and Design of Novel Annotation Languages and Technologies. Tech. Rep. D3.1, TERENCE project (2012)
17. 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)
18. TERENCE Consortium: TERENCE web site, <http://www.terenceproject.eu>
19. Vaadin Ltd: Vaadin framework, <https://vaadin.com/home>. Last accessed: July 30, 2012.

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

Fernando De la Prieta¹, Tania Di Mascio², Rosella Gennari³, Ivana Marenzi⁴, and Pierpaolo Vittorini⁵

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.

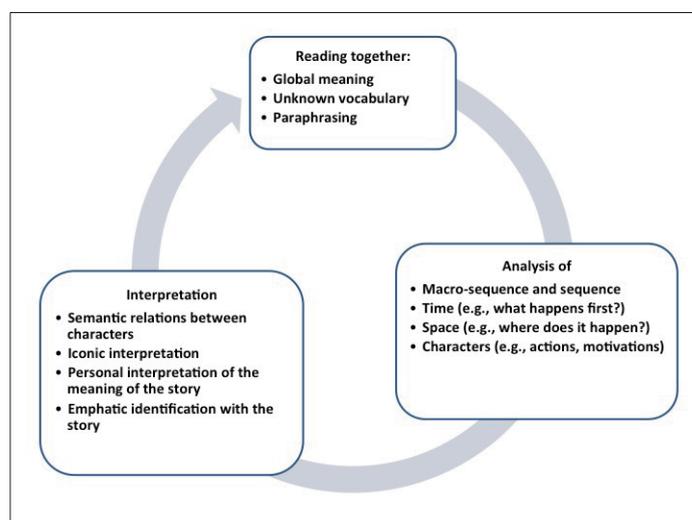


Fig. 1. The pedagogical hermeneutic cycle.

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 casual 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 casual 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 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.

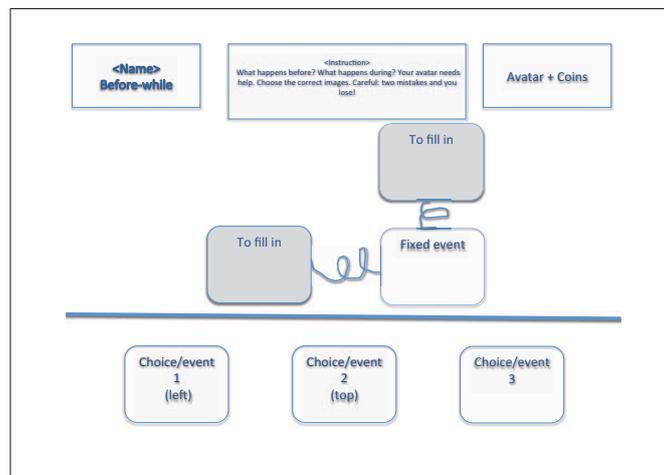


Fig. 2. The visual template of the before-while smart game.

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].

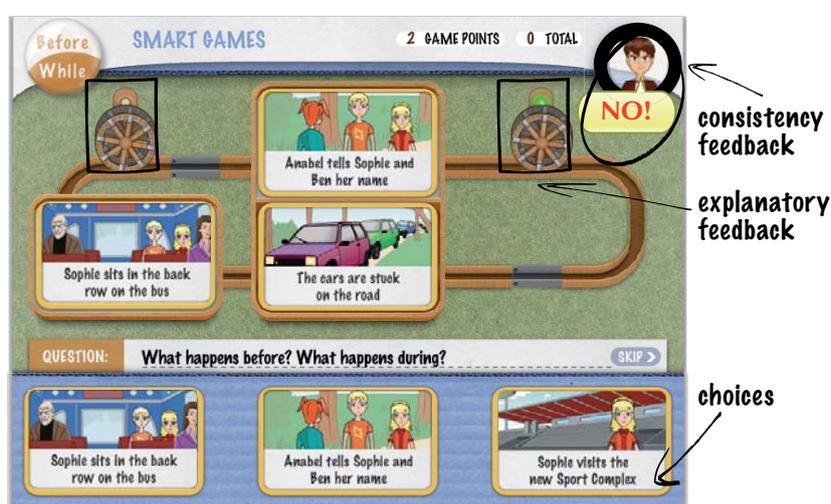


Fig. 3. An instance of a prototype of a before-while smart game.

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)

Serious Gaming for Complex Decision Making

Anja v.d. Hulst & Mark Ruijsendaal

TNO, P.O.Box 23, 3769 ZG Soesterberg, The Netherlands

`anja.vanderhulst@tno.nl mark.ruijsendaal@tno.nl`

Abstract.

Tactical- and strategic decision making in the safety domain is a form of ‘complex decision making’ with Naturalistic Decision Making as the predominant line of research. At the heart of the Decision Making expertise are ‘situation assessment’ capabilities, the most ‘intuitive’ aspect of complex decision making. In training it is also the most neglected. Particularly for developing the highly intuitive assessment skills, substantial task experience is indispensable. This makes serious gaming an attractive alternative to live training sites for tasks that are dangerous, hard or just too expensive. However, gaming requires a dedicated training approach like Job Oriented Training to be effective. We learned that implementing JOT in realistic settings, many lower level design issues emerge. Design choices are found to have substantial impact on the effects of training. Unsolved design issues are: level of fidelity, scenario progression design and designing for flow.

Keywords: Serious Gaming, Complex Decision Making, Situation Assessment, Job Oriented Training, Design Issues

1 Complex decision making

Tactical- and strategic decision making in the safety domain is highly situational, cognitively complex and is performed under demanding circumstances. As such, it is typically a form of ‘complex decision making’ (CDM) (Orasanu & Connolly, 1993). That is, decision making under circumstances that can be characterized by dynamic and continually changing conditions, uncertainty and ambiguity, ill-defined tasks, time constraints and most important, high stakes, multiple actors and significant personal consequences of mistakes (Klein, 2003).

From the 1960s, it became clear that a substantial part of failure in military operations resulted from inadequate situation awareness and decision making by commanders. From then on, the US Army began funding decision making research during the mid-1980s. The U.S. Navy became involved following the 1988 USS Vincennes shoot-down incident, in which a U.S. Navy Aegis cruiser destroyed an Iranian

commercial airliner, mistaking it for a hostile attacker (Klein, 1998). TNO participated extensively almost right from the beginning in this research, e.g. by the work of Schraagen (1997, 2008). At present, in the field of military- and safety research, a vast body of research is emerging that aims at understanding complex decision making to better prepare commanders for the demanding circumstances of the safety domain.

The predominant line of research in complex decision making is that of Naturalistic Decision Making (NDM) (Klein, 1998, 2003, Zsombok et al., 1997). Research in NDM aims at developing models that describe how experienced decision makers actually function under demanding circumstances. The analyses done in the field of NDM reveal, for instance, that at the tactical and strategic level, very few general procedures exist. For example, Kahneman et al.(1982) observed that expert decision makers did not adhere to the principles of optimal performance; they relied on heuristics as opposed to algorithmic strategies. Also, it was observed that decision making is largely situated. That is, proper decision making is highly dependent on awareness of characteristics of the local terrain, infrastructure, population, the local risks as well as mandates and responsibilities of the actors involved.

2 Situation Assessment

Even more important, one of the essential findings in NDM is that ‘situation assessment’ (SA) capabilities are at the heart of the expertise. SA is the most ‘intuitive’ aspect of complex decision making and it takes most time and experience to develop to an expert level (Stehouwer et al., 2005). What distinguishes an expert from a novice in situation assessment is that the expert has acquired a comprehensive repertoire of patterns (Klein, Calderwood, & Clinton-Cirocco, 1986). ‘These patterns describe the primary causal factors operating in the situation. The patterns highlight the most relevant cues, provide expectancies, identify plausible goals, and suggest typical types of reactions in that type of situation’ (Klein, 2008, p. 457). Decision makers recognize and categorize situations on the basis of the patterns they have acquired, hence the label ‘Recognition Primed Decision Making’ (RPDM).

Situation assessment may be essential to expertise in CDM, sadly, in training it is also the most neglected (Stehouwer, 2005). Particularly for developing the highly intuitive assessment skills, substantial task experience is indispensable (Klein, 1998). That is, essential to acquiring a sufficient repertoire of situated patterns, a commander has to experience a vast amount of relevant situations. Generally, the common field exercises (FX) fail to provide such practice. FX are generally limited by the features of the dedicated exercise terrains, infrastructure, maneuvering- and weapon-platforms available. In addition, FX are costly, logistically demanding and generally not very efficient as there is always a great deal of waiting.... Worse, crisis situations are generally large scale incidents which are dangerous by nature and are hard if not impossible to mimic properly during a FX.

3 Serious Games for Situation Assessment

Simulation and Serious Games (SG) broaden the range of tasks as well as circumstances under which can be trained and have great potential for the training of situation assessment. Simulations have been here for more than 40 years, but it is the gaming industry that presently defines the progress. Civilian commercial entertainment technology development coincides with the emergence of an impressive military gaming community. The costs of production of content (e.g. terrains, platforms and weapon models) are being shared among the various international military users. As a result, military serious gaming technology undergoes an extremely rapid evolution, more and more commercial off-the-shelf SGs are used in military training courses.



Fig. 1. Tarin Kowt, Afghanistan Port of Rotterdam

Even more important, the recent development of techniques for the (semi) automated generation of terrain databases caters for the fast production of medium to high fidelity 3D geo-specific and geo-typical terrain databases (Smelik et al., 2010). In particular, the latter are extremely valuable in the development for training in situated assessment.

SGs are currently being used to facilitate e.g. dedicated mission preparation by providing geo-specific representations of mission areas. For instance, many of the primary mission areas of the international coalition forces, Tarin Kowt, Deh Rawod, Kandahar, Helmand and Bagdad have been made available. Such geo-specific terrain databases are used in the international community to train mission-specific situated tactics, to enable troops to gain an understanding of the specific threats at critical locations (infrastructure, high value targets, overwatch locations) and learn to understand the situational tactics, techniques and procedures of the opposing forces (v.d. Hulst et al., 2011b). Also terrains critical to national safety have been modeled, such as the industrial port of Rotterdam, e.g. being used for the training of first responders in handling Chemical Hazards.

SGs will not fully replace FXs, but allow for tasks to be trained that are dangerous, hard, if not impossible, or just too expensive to train at live training sites. SG relieves the logistic burden of training and allows to reduce time on task and thus allow com-

manders to experience a great variety of settings within a limited time span. Games are also applied when the use of equipment is too expensive or just impossible, for example in training the tactics of large CAT/convoy operations (v.d. Hulst et al., 2011a). Finally, the nature of SG makes it relatively easy to offer multiple scenarios, either similar or very different, within a short time span. Henceforward, SGs are critical to facilitate the construction of a comprehensive repertoire of patterns needed for RPDm in a variety of circumstances and incidents.

SG do have quite some potential, yet they frequently fail to live up to the expectations as made clear by Hays (2005) who e.g. reviewed 48 empirical research articles on the effectiveness of “instructional” games. Hays is clear in his conclusions; SG Technology alone will not guarantee effective and efficient learning. For SG to be effective, it requires a dedicated training approach.

4 Training Approaches

From the NDM research, several approaches to training complex decision making have emerged. Examples are Decision Skills Training (Pliske et al., 2001) and Job Oriented Training (Stehouwer, 2005, v.d. Hulst, 2008). These approaches provide indications for the instructional design of training for complex decision making and for the implementation of such training.

Decision Skills Training (DST) has been applied extensively in military training for complex decision making, of which, for instance, the application of DST to Urban Operations training for junior leaders (Phillips, et al., 2001) is of particular interest to this work. DST aims at training military to ‘learn like experts’. DST strives to provide students with as much relevant experience as possible in the form of a series of increasingly complex scenarios relevant to (aspects) of the decision making. While working in these scenarios, students are trained to mentally simulate possible plans they come upon as a solution to the challenges of the scenario. Also, they are trained to extensively reflect upon their own decision making and to make reflection a habit in their professional life.

To students, DST provides a method for mentally simulating plans, a method for reflecting on the decision making in the scenario’s and a method to obtain feedback on the expression of intent.

Job Oriented Training (JOT) is also based on the principles of Naturalistic Decision Making and is a partial implementation and further elaboration of DST.

Personnel in the safety domain is not trained to simply reproduce knowledge, perform standard procedures or solve standard problems. The application of tactical measures must be tailored to best suit the specific mission to be accomplished in a military or safety environment. As a result, training for tactical events should focus on delivering professionals who can act in ever changing and unpredictable situations. In JOT, therefore intention is to target not only conceptual knowledge, but also the skills of independent and competent problem solving in entirely new situations as well as a

‘can do’ attitude, which includes tackling complex situations not previously encountered. The aim, therefore, is to integrate the acquisition of conceptual knowledge, skills and attitude and thus strive for development of rich and integrated competencies.

To target these rich competencies, JOT defines principles for the instructional design of exercises aiming at discovery learning of complex decision making, mostly in a military or first responders context. In JOT, students are confronted with a series of quite short –cyclic, increasingly complex and challenging exercises to allow them discover the essential principles of their job. Crucial is that no theory is provided in advance; theoretical insights are acquired while solving realistic cases. As such, the students do not need to have completed theory oriented training prior to the exercise, as they are expected to discover the essential tactical principles themselves during the JOT exercises. This simultaneously trains them in problem solving in situations entirely new to them and aims at developing a ‘can do’ attitude in tackling new situations. Also, self-reflection is deemed crucial to conceptualize experiences and to make the concepts stick.



Fig. 2. Self – reflection, squad infantry.

Amongst other things, JOT prescribes the nature of instructor support, debriefing and feedback as was deemed crucial by Hays (2005). It also defines requirements with regard to the design of the virtual environments.

JOT was developed for the Dutch Ministry of Defence and has been applied respectively to Serious Gaming for the training of Floodcontrol, Crowd and Riot control (Buiel et al., 2012), Virtual Tactical Trainer for Counter-Improvised Explosive Devices (VTT-C-IED) (v.d. Hulst et al., 2011b), Urban Operations (v.d. Hulst, 2011a), Minewarfare (Stubbe, et al. 2011), Infantry and Cavalry operations (v.d. Hulst, et al., 2008), Naval tactical- and operational tasks (Stehouwer et al., 2006) and Air Defense (Stehouwer et al., 2005). From the above listed efforts in practical applications of the initial JOT concept, many design issues emerged and only few have been solved as of yet.

5 Lessons learned and issues to be solved

The hypothesis underlying JOT was that to obtain optimal effects, the use of SGs for SA demands for 1) a dedicated learning approach as well as a 2) dedicated design of the virtual environment (v.d. Hulst et al., 2008a, 2008b). JOT, therefore, provides high level conceptual prescription for the design of the didactic setting and the design of the virtual environment. Yet, when implementing JOT in realistic settings, many lower level design issues emerge. Those design choices still are found to have substantial impact on the effects of training (v.d. Hulst, et al. 2011a).

Below, we'll list the predominant design issues yet unsolved.

5.1 Fidelity

JOT prescribes that a virtual environment has to provide a 'relevant reality', i.e. a virtual environment that provides the cueing needed to enable adequate SA. The easiest solution to create a relevant reality is to always demand for a high fidelity environment. However, high fidelity models of environments and human behaviour are extremely costly and experience is generally that the costs of such models will exceed available budgets. Also, fidelity studies like those reported in Hays and Singer (1989) provide evidence that, when aiming at tactics, low physical fidelity frequently still yields good learning results. Hence, from a cost perspective, one should aim at defining the minimum level possible that still provides sufficient cueing for SA. In doing so, one must be very careful not to create an environment that leads to negative transfer (Hays & Singer, 1989).

Until now, no generic heuristics to define cueing for SA have been found. Phillips et al. suggest defining cues on the basis of a so called Cognitive Task Analysis, that is, interviewing experts on the cues they use for their SA. Our experience is that it is both hard to find a sufficient number of real experts and time consuming to do a sufficient number of CTAs. Still after defining and implementing the cues, it demands a fair amount of testing to get the cueing right.

Minimum requirements with regard to cueing can well be defined. Visschedijk et al., for instance, used a comparative approach to define the minimum level of cueing needed for proper recognition of emotions in a Crowd and Riot Control training. The authors compared various settings with avatars having either a posture, voice or facial expression representing emotions or combinations of the former. Recognition of those emotions with and without context information was tested with about 20 subjects. This experimental approach, however, is too time consuming to apply at a larger scale. Within a single virtual environment, many different types of cues need to be defined, and such controlled experimentation generally is too expensive during a design trajectory.

Where SA is at the heart of complex decision making, we'll need to find good methods to define the proper level of cueing in Serious Games and adequately test that cueing. Such methods shouldn't be so time consuming that they will not be applied properly.

5.2 Scenario progression

JOT defines that throughout the progression of the training, the complexity of the environment in which the decision making takes place is controlled. The tasks gradually increase in complexity (see e.g. White and Frederiksen, 1990), while performance requirements increase. SA training, therefore, requires a series of scenarios where the initial scenarios are challenging, but do-able and where each subsequent scenario increases in complexity, builds upon the insights acquired in the prior scenarios and introduces sufficient new challenges. In practice, this statement leaves too much to the designer. One needs a good insight which cases are easy and which ones are hard and which ones are really challenging. In designing the military SGs (e.g. the above mentioned VTT C-IED and Air Defence Tactical Training), in the initial phases we had no clue as to the difficulty of cases. We observed that some cases were generally assumed trivial while others were, without exception, found to be really hard. Yet, our Subject Matter Experts had real trouble explaining why cases were easy or hard.

The more SGs we built, we observed some indications that might help to define a good progression of scenarios. We e.g. found out that the easy ones were scenario's that were assumed to be prototypical situations, e.g. situations where all well known factors indicating a threat were present. In contrast, the really challenging cases were those where students had to combine less known indicators with information from several different and sometimes unreliable sources. Also, assumed difficult were the situations very uncommon to the students, e.g. students just back from Afghanistan had trouble imagining the potential threats in a high tech terrorist scenario in a modern city. They were basically looking for the well known threat indicators as known from their Afghanistan experience and didn't look beyond that experience. Still, we observed that there must be many more heuristics underlying the complexity of scenario's, yet still unknown to us.

Without an understanding of factors in complexity of cases, it is impossible to design a good progression. A proper scenario progression is essential to building experience and thus to learning and the field badly needs design heuristics for such progression.

5.3 Flow

The SG world is in a fierce debate about the 'fun' and flow factor (e.g. in Ritterfeld et al., 2009). In our view, Serious Gaming is absolutely not about fun as is, but it is all about flow. That is, the flow that emerges from being fully immersed in a process that is perceived relevant to the job.

If the SG based training is designed well, we see a tremendous flow. In our observations, a major indication for flow is that, while actively engaged on virtual missions, students don't mind extra-curricular hours. When training in a well designed game setting, our students voice no objections to the issuance of orders past 10 p.m. for missions that will start the following day at 7:40 am. and they indicate in their evaluations that this is pretty cool. Their positive attitude towards the training is reflected in their performance, which is generally above standards, as confirmed in the instructor evaluation. Besides effects from using virtual environments for

experiential learning, at least time on task is boosted, which is positive in itself since time on task is one of the predominant predictors for learning (Carroll, 1963).



Fig. 3. Flow- Air Defense Tactical Training.

Design for flow is far from easy. Small inadequacies in the design of the game or the context are observed to make students ‘fight the system’ rather than be immersed.

For instance, we use VBS2 for the creation of several of above mentioned military training environments. When first confronted with VBS2, students automatically try to use it as a first person shooter and start shooting every human being around, just for the fun of it. Only if we design the context (the ‘big game’) right, they refrain from going into ‘entertainment modus’. That is, if we provide a proper briefing, assign roles and provide adequate time for analysis and planning, only then they will use the system to truly train for military operations.

Also, a good design for one target groups isn’t necessarily a good design for another group. We notice in training for the ‘comprehensive approach’ that novices really get into flow and start learning when using the SG Go4it (Hulst, v.d. et al. 2012). In contrast, more experienced students that had been trained to be competitive, however, soon after the initial rounds, just tried to win by playing interventions that were likely to have optimal outcomes irrespective of the validity of those actions at that point in time. Those groups generally do express afterwards that they had fun, but we do observe neither flow nor learning and they quit gaming way before the novices as well as non-competitive experts. We only found out during the trials that we had to provide such a competitive target group with a completely different organisation of the training for it to be effective.

For us, design for flow is to a large extent a trial and error process and indeed we do experience a fair bit of error. Certainly, the field lacks good SG mechanics and design rationales.

Training for Situational Awareness demands for extensive practice and virtual environments allow for such practice and inherently support accelerated learning. SGs hold a great promise for experiential learning especially when the learner experiences the virtual world as emotionally involving and mentally stimulating (Green, 2006).

However, SGs frequently reveal too much error in the design and consequently many games fail to provide such an involving and mentally stimulating environment. Certainly, the design for such environments should become less of an art and more of a science.

6 References

1. Buiel, E., Hulst, van der, A.H. Veldhuis, G. (2012). Inzetmogelijkheden van Serious Gaming voor Public Order Management Training. Soesterberg, TNO Rapport.
2. Carroll, J. (1963). A model of school learning. Teachers College Record, 64, 723-733.
3. Green, M. (2006). Narratives and cancer communication. Journal of communication, 56, 163-183.
4. Hays, R. T. & Singer, M. J. (1989). Simulation fidelity in training system design. New York: Springer Verlag.
5. Hays, R. T. (2005). The effectiveness of instructional games: a literature review and discussion. Technical Report 2005–2004 for the Naval Air Center Training Systems Division: Orlando, FL.
6. Hulst, van der, A.H, Muller, T.J., Besselink, S., Coetsier, D & Roos, C.L. (2008a). Bloody Serious Gaming – Experiences with Job Oriented Training. Proceedings: Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2008.
7. Hulst, van der & A.H., Muller, T.J. (2008b) Job Oriented Training: Onderwijskundige grondslag en onderbouwing. TNO Rapport A194.
8. Hulst, van der, A.H., Muller, T.J., Besselink, S, Smeenk, R. Vink, N. & Visschedijk, G. (2011a). Serious Gaming in Urban Operations: Potential for Tactics Training. TNO Report B073.
9. Hulst, van der, A.H., Amade, C.A.R. & Sain, de, G. (2011b). VTT-C-IED: Threat assessment and planning for search using virtual environments. Nato Research and Technology Organisation - MP-SET-175.
10. Hulst, van der, A.H., Christoph, N., May, A. & Aldershoff, F. (2012, accepted). GCAM: A modeling approach to the Comprehensive Approach. Nato Modelling Simulation and Gaming workshop, Stockholm, October 2012.
11. Klein, G. (1998). Sources of Power. How People Make Decisions. Cambridge, Massachusetts: The MIT Press.
12. Klein, G. (2003). The power of intuition. New York: Random House, Inc.
13. Klein, G. (2008). Naturalistic Decision Making. Human Factors: The Journal of the Human Factors and Ergonomics Society June 2008 50: 456-460.
14. Orasanu, J., & Connolly, T. (1993). The reinvention of decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsombok (Eds.), *Decision making in action: Models and methods* (pp. 3–20). Norwood, NJ: Ablex.
15. Phillips, J, McCloskey, M. McDermott, P. Wiggins, S., Battaglia D., Thorsden, M.L. & Klein, G. (2001). Decision-centered MOU training for small unit leaders. Technical Report 1776, U.S. Army Research Institute for Behavioral and Social Sciences, 2001.
16. Pliske, R. M., McCloskey, M. J., & Klein, G. (2001). Decision skills training: Facilitating learning from experience. In E. Salas & G. Klein (Eds.), *Linking expertise and naturalistic decision making* (pp. 37-53). Mahwah, NJ: Lawrence Erlbaum Associates.
17. Ritterfeld, U., Cody, M. & Vorderer, P. (2009). Serious Games. Mechanisms and effects. New York: Routledge.

18. Schraagen, J.M.C. (1997). Damage Control Decision Support System. In Zsombok, C.E. and Klein, G (1997) Naturalistic Decision Making. Lawrence Erlbaum Associates, Mahwah, NJ.
19. Schraagen, J.M.C. (2008). Naturalistic decision making and macrocognition. Ashgate Publishing, Ltd.,
20. Smelik, R.M., Tutenel, T., Kraker, K.J.D. & Bidarra, R. (2010). Declarative Terrain Modeling for Military Training Games. Int. J. Computer Games Technology. Volume 2010, January 2010.
21. Stehouwer, M., Serné, M. & Nielke, C. (2005). A tactical trainer for air defense platoon commanders. In: Proceedings IITSEC 2005.
22. Stehouwer, M., Stricker, J. & Gemeren, van W. (2006). Training Design for Professional Development. In: Proceedings IITSEC 2006, Orlando, FL.
23. Stubbé-Alberts, H.E. & Oprins, ET AL.P.B. (2011). Eindrapport JOT-KM: Over de implementatie van competentiegebaseerd leren. TNO-DV 2011 A143.
24. Visschedijk, G.C., Lazonder, A.W., Hulst, van der A.H., Vink, N. & Leemkuil, H. (2012, accepted). Emotions for tactical decision games. Modeling human emotions for tactical decision making games. British Journal of Educational Technology.
25. White, B. & Frederiksen, J. (1990). Causal model progression as a foundation for intelligent learning environments. Artificial Intelligence, 42:99-157.
26. Zsombok, C.E. & Klein, G (1997) Naturalistic Decision Making. Lawrence Erlbaum Associates, Mahwah, NJ.