Integrating Serious Games in Adaptive Hypermedia Applications for Personalised Learning Experiences

Maurice Hendrix, Sylvester Arnab, Ian Dunwell, Panagiotis Petridis, Petros Lameras, Sara de Freitas
Serious Games Institute
Coventry University
Coventry, UK
{MHendrix, SArnab, IDunwell, PPetridis, PLameras, SFreitas}@cad.coventry.ac.uk

Evgeny Knutov
Department of Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
e.knutov@tue.nl

Laurent Auneau
Succubus Interactive
Nantes, France
laurent.auneau@succubus.fr

***Abstract*—Game-based approaches to learning are increasingly recognized for their potential to stimulate intrinsic motivation amongst learners. While a range of examples of effective serious games exist, creating high-fidelity content with which to populate games is resource-intensive task. To reduce this resource requirement, research is increasingly exploring means to reuse and repurpose existing games. Education has proven a popular application area for Adaptive Hypermedia (AH), as adaptation can offer enriched learning experiences. Whilst content has mainly been in the form of rich text, various efforts have been made to integrate serious games into AH. However, there is little in the way of effective integrated authoring and user modeling support. This paper explores avenues for effectively integrating serious games into AH. In particular, we consider authoring and user modeling aspects in addition to integration into run-time adaptation engines, thereby enabling authors to create AH that includes an adaptive game, thus going beyond mere selection of a suitable game and towards an approach with the capability to adapt and respond to the needs of learners and educators.**

Keywords - Adaptive Hypermedia; Adaptation; Serious Games; Educational Games; Education; Personalization.

I. INTRODUCTION

In recent years computer games have increasingly been used for training purposes. Frequently cited benefits of so called serious games include increased learner motivation through increasing learner engagement, achieved by a combination of education and entertainment [1]. However, learning styles can prove diverse: for example some learners are happy to find solutions by trial and error, while others prefer to first learn about what the solutions are and why, before trying them out [2]. Games and Simulations may also be more or less suitable depending on the sub-topic. For example skills commonly improved through drill and rehearsal, such as emergency evacuation, are well suited to a game-based learning environment which can recreate an evacuation scenario whilst providing motivation for rehearsal through game play. By comparison, low-level cognitive transfer [3] may be more suited to other instruction methods. The success of a serious game is also directly related to the effectiveness of the interactive learning experience responding to the evolution of learners' needs and requirements. Although a major game design concern, focuses on learning experience often jeopardize the game developers' efforts to fulfill the intended serious goals. Serious games are often content-rich and can use high fidelity visual and audio learning objects with diverse pedagogic approaches. This means that development costs can be prohibitive compared to other media.

Reducing costs of designing and developing a serious game is essential. Sharing and reusing or re-purposing (re-using for a different purpose) is therefore particularly important. The mEditor [4] is a novel tool which allows re-purposing of serious games and offers the potential to significantly reduce the development cost of serious games.

Education is also a popular application area of adaptive hypermedia (AH). In order to adapt to different learning styles, it is important that the learning capabilities, styles and progress of users is captured. User modeling methods and AH have been widely used in Tutoring Systems [5], but, are not common in serious games.

Studies such as Pierce et al. [6] demonstrate that integrating serious games with Adaptive Learning Systems can be very effective, however while both authoring of AH and of serious games have been active research areas, work towards the integration the two in all aspects: authoring, user modeling and delivery remains limited. Addressing this gap requires developing techniques and tools that allow for games to be effectively adapted and pedagogically integrated whilst retaining their unique benefits in areas such as motivation and engagement.

The mEditor tool demonstrates a step towards addressing some of these issues around reuse and repurposing. In Section 2, a detailed examination of the mEditor tool through an illustrative scenario illustrates how it uses a graph of game dialogues with certain conditions attached to represent individual scenarios within the game. Both the dialogue and the conditions can be changed, allowing rapid and accessible refinement, or repurposing of content without requiring a high level of programming skill, and is thus more accessible than the bespoke development required for adapting most existing serious games.

To consider the mEditor tool alongside a range of approaches, the rest of this paper is organized as follows: Section II expands upon the concepts of AH and serious games, and examines relevant developments. Section III contrasts the different components of authoring game scenarios with those used by existing authoring tools for AH. Section IV outlines how an authoring tool for adaptive games, integrated into an AH could be achieved; finally Section V highlights the future challenges.

II. Background

Educational authoring tools allow educators and domain experts to prepare courses and presentations, often relying on the concept of learning objects (LOs) [7]. A LO is a unit of instruction for e-learning and should be auto-consistent and modular making them reusable. Learning Management Systems (LMS) are software platforms that provide didactic materials assembling LOs. Discovering and reusing LOs is facilitated by the existence of a standardized description format the IEEE Learning Object Metadata (LOM) [8] and the existence of repositories for sharing LOs, such as ARIADNE [9]. The Sharable Content Object Reference Model (SCORM) [10] is a collection of standards and specifications for web-based e-learning. However, these specifications offer some support for personalization, mainly based on adapting the notion of sequencing of content (i.e. the order in which content is presented to the student). IMS-Learning Design (IMS-LD) is another set of standards, which describes what it calls the learning design. Its aim is to be able to represent all major pedagogies and it models roles and activities within an environment that consists of LOs.

In recent years, serious games have been recognized for their educational potential. In particular they can increase learner motivation due to increased levels of engagement. However developing games is costly and time consuming. Various efforts have been made to apply the concepts of adaptivity and personalization to serious games. RETAIN is a serious game design paradigm aimed at applying instructional strategies concurrent to game development [11]. It highlights the importance of the presentation and feedback of the didactic choices to the player and their linkage to reinforce the lesson and test the transfer of knowledge. Riedl et al. [12] presents a framework for creating interactive narratives for entertainment, educational, and training purposes based on an experience manager agent, which is a generalization of an automatic drama manager. Bellotti et al. [13] present an Experience Engine (EE), which exploits computational intelligence algorithms to schedule tasks matching the requirements of a teaching strategy that can be expressed by an instructor, the needs estimated by profiling the user performance and with the aim of keeping the flow. Based on the user feedback, the EE learns a strategy that aims to maintain the performance of learners in a “narrow zone” between too easy and too difficult, maintaining “flow” [14]. This relies on a model of the user that the EE builds and of the instructional tasks features that keep into account both of the entertainment and educational aspects (e.g., learning styles [15]) typical of serious gaming. Moreover, the teaching strategy itself must be modeled, so to allow educators to express their educational line. The EE uses machine learning algorithms to adapt its strategy based on the user profile that is updated from the user feedback. Dynamic assembly is important also to support long-term playability, since missions will be different, without repetitions (if not required for learning purpose set by the learning strategy).

Repurposing of learning resources refers to changing the learning resource to suit a new educational context [16] rather than reuse which merely refers to using the learning resource in its original context without any changes [17]. The changes made to the content can be for various reasons such as the use of different pedagogies, different technologies, or different contexts and learners. Protopsaltis et al. [18] propose a methodology for serious game repurposing games and introduce a practical tool [4], the mEdictor tool, for repurposing serious games, integrated with a commercial game engine. Dunwell et all. [19] suggest that serious games are especially suited to a type of learner called intuitive learners [2], although it indicates that serious games could be useful for other learners as well, especially if the teaching approach is adapted to suit the different learners individually. Part of this is deciding when to present the game to students, but adapting the game itself, for example by changing the dialogues, difficulty level, and language used within the game are all important parts of this adaptation. Games authoring tools such as the mEditor tool, see Figure 2. allow this sort of adaptation, in order to use the serious game with different types of learners or in different contexts. The approach relies upon an educator to actively repurpose the game. The tool uses a graph-based paradigm, in which educators can change game scenarios by changing connection, conditions and nodes in a graph, requiring little or no programming knowledge (though some technical insight may still be needed).

In recent years various efforts have already been made to create authoring tools for serious games, however a full integration with AH is still lacking. eAdventure [20] aims to facilitating the integration of serious games into educational processes and LMSs in particular. While it focuses mostly on LMSs such as moodle [21], its aims are very similar to our aims of integrating serious games and AH. eAdventure contains a graphical authoring tool for authoring adventure games. Its main focus is point & click games and it offers customizable menus and interfaces, artwork and scenarios. Just like the mEditor it uses a XML based notation for describing games that are deployed to a java based games engine. It allows editing of the main elements that make up a game such as scenes, characters, dialogues and navigation. It has built in assessment mechanisms and some support for adaptive learning scenarios. Editing dialogues is done by creating flow diagrams using a graph-based editor.

StoryTec [22] is a digital storytelling platform that features a comprehensive authoring tool.It can be used for creating serious games and features a story editor, stage editor, action set editor, property editor and asset manager. The story editor is based on the use of the Unified Modeling Language (UML) [23] a popular modeling language among software engineers. UML uses a standardized set of graphic notations resulting in graph structures. Hence this approach is somewhat similar to the approach taken by the mEditor and eAdventure. While other authoring tools exist the use of these graph-based structures is a clear trend, hence using the mEditor as an example for this paper is justified.

AH systems build a model of their users and use this to adapt the hypermedia corpus to the user's knowledge, needs or goals [5]. The dimensions of adaptation are well known and various models have been defined such as AHAM [24], based on the Dexter [25] Hypertext model, Munich [26] a UML extension to AHAM, GAHM [27], and more recently GAF [28] in addition to the more traditional hypermedia (text, images and videos etc.). Adaptive games such as the e-Game have been integrated into AH effectively. Simulations [6] have also been integrated into personalized learning environments. A possible integration of a game and a Learning Management system has also been shown [19].

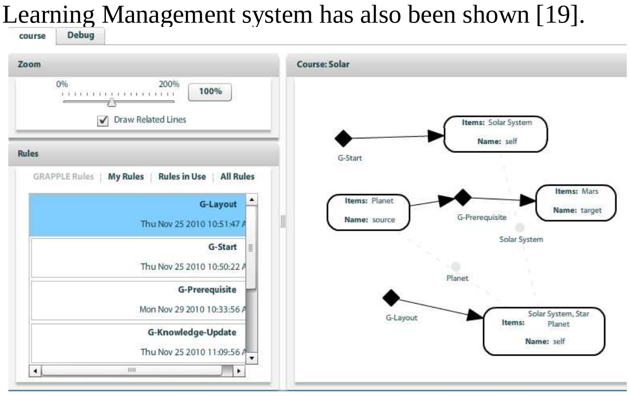

Figure 1. The GRAPPLE authoring tool is graph-based [29]

Authoring of AH is an active research area. Various different models have been proposed such as [24, 29–31]. Authoring of AH consists not only of content creation but also of specification of adaptation strategies that dictate when to show what content and in which way to show it. Various methods have been tried for this part of the authoring process, from a pre-defined selection of strategies [32], to domain specific programming languages [33]. Another method that has been tried with some success is one based on a graph structures [24, 29], see also Figure 1. , where authors can edit strategies by changing the connections, conditions and nodes in a graph interface, with limited programming knowledge. While many graph based editors exist the strength of the GRAPPLE authoring tool lies in the use of a library of adaptive strategies (or Pedagogical Relationship Types) containing adaptation code. This is quite similar to the way the mEditor uses containers and functions.

However while games and simulations have been integrated into personalized learning systems with some success, little work has been done on the integration of games into the authoring and user modeling process. Gaffney et all. [31] propose a simulation authoring tool, but the simulations authored consist mainly of hypermedia elements, rather than being integrated into a game engine as is often the case with serious games or game-based simulation. The use of such a game engine is often necessary to achieve the representation required, for example when designing 3D games or virtual environments. De Troyer at all. [34] have designed an authoring environment for Virtual Reality. This gives the opportunity to include 3D virtual objects in an AH; however the focus is on integrating these objects in the web-based environment. Our work does not integrate objects into the web pages, but rather allows launching a personalized version of a full serious game, with the possibility for in-game adaptation.

A. *Motivating example*

In this section we showcase the repurposing process for serious games using the mEditor tool, and contrast it with the authoring process of an AH object. As a representative of AH we use the CAM model [29]. As an example we use a serious game devised for healthcare, developed by Succubus within the mEducator project. The game allows a medical student to rehearse a session with a patient. The student takes on the role of the doctor and can move around the office, ask the patient to sit down, describe his symptoms, undress lie on the bed, or administer drugs from a selection of available drugs. He can also ask a nurse, to take the blood pressure or make an Electrocardiography (ECG). There is a beginner mode where the player will be corrected when making mistakes and an advanced mode where feedback is only received at the end, allowing mistakes to be made ranging from misdiagnoses through to patient mortality.

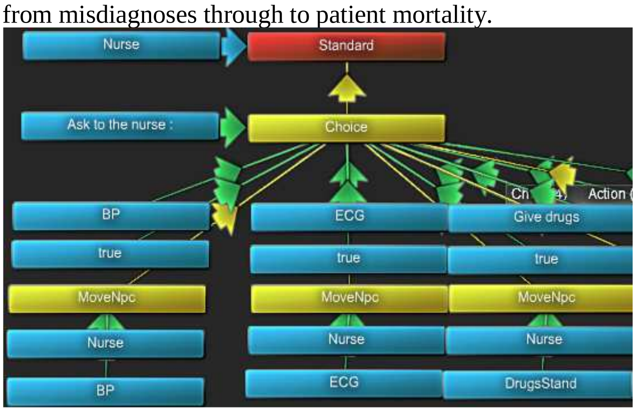

Figure 2. The mEditor [4]for scenario-based repurposing is graph-based

Figure 2. illustrates a part of the scenario as visualized by the mEditor tool, showing the main elements that are used in building scenarios. The example shown describes the interaction with the doctor and the nurse present in the session. The resulting game is show in Figure 3. . There are three characters, the patient doctor and nurse and the player can click on either of them to communicate or on some of the objects in the room, such as the chair and the bed.

In the context of repurposing this game, consider a tutor, who wishes to use the game's multimedia resources, but rather than for cardiac conditions which it is currently aimed at, wants to use it to support teaching on lung conditions. Prior to having the mEditor tool, the tutor would have been required to gain access to and edit the games source code, engaging in extensive bespoke development, or commission a new game with its associated costs.

In all these cases, technical development would incur substantial costs in time or outsourcing, and could result in a game-based approach being disregarded due to these prohibitive factors. With the mEditor however, the tutor can repurpose the game so as to use it in a scenario where the patient has a different condition without as much need for technical development or significant investment. For more extensive repurposing some understanding of boolean logic and functions is still needed.

Figure 3. The doctor training game by Succubus Interactive

Imagine now that the tutor also wants the game to automatically adapt the difficulty level to learner's background and knowledge. Let's assume we are working in the context of an advanced LMS, which builds a learner profile and allows the tutor to author an AH [35]. The mEditor tool can be used to quickly develop a range of adapted games to suit various learner needs and ability levels. The tutor may also feedback the students' performance to the learning environment and update the students' level of understanding.

III. COMPARING AUTHORING LANGUAGES

Figure 1. shows the main elements the mEditor uses to create game scenarios. Below we discuss them and particularly how they can be represented in AH authoring frameworks. As mEditor uses a graph based authoring paradigm, it seems reasonable to focus on graph based AH authoring frameworks.

A. Events

mEditor responds to the user via events, similarly to AH systems. Many models rely on so called event condition action rules (ECAs). These connect events and actions via certain conditions. Within the mEditor the author adds events and connects these to actions. In most scenarios there will be containers in between, especially the IfThenElse container, effectively constructing ECAs.

1) Start

Executed at the start of the game, the start event handles the initialization of variables, and showing and hiding of assets. It has an action or series of actions attached to it. AH models such as the Concept Adaptation Model (CAM) [29] and Layers of Adaptation Granularity (LAG) [30] model (via the Initialization part) have similar start states.

2) Standard

Triggered from within the game, when the user clicks on the appropriate object, and comes with both an action, and an item. This item, identified by name, indicates which item triggered the action. In AH the events are usually page clicks or accesses. Actions specified based on a particular page access are possible in LAG, CAM and the AHA! Graph editor [36]. One of their strengths is the ability to generalize and respond to page access for particular types.

B. Actions

1) Container

A grouping and selection mechanism for actions closely resembling programming constructs. The following containers can be used. Containers can also be combined.

- DoAllNow: Do all connected actions in parallel.
- DoAllSequentially: Perform the connected three actions in sequential order.
- DoLoop: Repeat the connected actions in a loop.
- DoOneByOne: Perform the connected actions one at a time, without specified order.
- DoRandomly: Perform the actions in random order.
- DoWhile: Repeat the connected actions in an infinite loop, so long as the connected condition is satisfied.
- IfThenElse: Two (sets of) actions can be attached and a condition determines which one will be performed.

AH models contain the possibility for conditional execution and loops, either using programming language constructs or formal specification languages. Conditional loops are also possible. Finally explicit control of execution order is not always possible in many models directly. LAG, the AHA! Graph editor and CAM make no distinction between the available pages. They can be made available in different order, but the user will only notice this if that is done step by step, rather than in one step.

2) Standard

The standard actions contain actions for explicitly assigning vlues to variables (ChangeBoolValue, ChangeFloatValue, ChangeIntValue, ChangePointValue), an Empty action, Aborting the game, Tracing a certain action and waiting for a specified amount time.

Assigning values is well defined in AH models, time delays are often not explicitly modeled but it would be technically possible to embed an external object that tracks time. End states for most models are implicit.

3) Specific

Some specific choices have an equivalent in AH. Choice exists both implicitly, via navigating to a specific page, as well as in (multiple choice) tests. Then hideInstance and showInstance are very similar to showing and hiding of pages and links, one of the most used adaptation features in AH. The following specific actions are available in mEditor:

- ChangeScene: Change the current game scene.
- Choice: Allow the user to pick from a list of options.
- Dialog: Display an interaction dialogue.

- DoAnim: Play an animation of a particular character.
- HideInstance: Hide the game asses from view.
- Infos: Shows textual information about something, it has a target location.
- MoveNpc: Move the non-player character.
- ShowInstance: Review the (hidden) game asset.
- CreateNPC: Create a non player character at a position.

4) Engine

The engine actions currently predefined are playing a sound, enabling or disabling the mouse and moving ('teleporting') non player characters. In AH models, as the delivery is usually achieved via a web browser, there is no possibility to explicitly disable the mouse, though this is possible via embedded objects or JavaScripts. Sounds and videos can be embedded as regular hypermedia objects.

C. Variables

Variables of the following basic types: boolean, floating point, integer, string, point (an x,y location) can be used in and updated by functions and can either be global, i.e. available through the game scenario or temporary, i.e. available only for one container. Variables are basic building blocks of programming languages and indeed prevalent in all AH authoring models. I.e. LAG is a domain specific programming language and allows user defined variables, although a type definition is not required. CAM uses a formal specification language called GAL [37] and this uses variables of different types, just like the AHA! Graph Editor.

D. Functions

A function can access global variables and those linked to its container. Functions are available for each types of variable split into two groups: functions, working on a number of variables and operators working on only one variable. Available are predefined conversion functions (e.g. valueToBool) and conditional functions (e.g. conditional float with connected values and a condition determining which value to select).

Functions are handled by the different AH models in different ways. LAG does not support functions but allows procedural programming constructs, achieving achieves the same effect. CAM was built around the idea of packaging adaptation patterns, for use as complex functions. GAL and the AHA! Graph editor rely upon defining functions.

IV. ARCHITECTURE

Authoring adaptive serious games can be achieved in various ways. Adaptation with regards to selecting when to present the game can be done entirely in existing AH systems by generating a number of alternative games at authoring time. This would be very time consuming and be limited to the alternative game configurations that were compiled by the author. A tighter integration could be relatively easily achieved. Instead of creating a number of alternative games, the author would use variables and conditions inside the game scenario that refer to the learner profile. This would then result in a personalized scenario, generated at run time by the adaptive delivery engine, just before the system presents the game to the learner [19]. This approach would require a slightly closer integration and careful consideration of cold start issues.

However it is possible to go even further. In this case the game engine would need to read and write the user profile. This could be achieved in different ways, such as directly communicating with a database, or via the use of an intermediary communication mechanism such as a web-service. This is an attractive approach especially as flash-based game engines like Succubus' engine and some popular 3D engines like Unity 3D already support web services.

An example of such integration is the approach taken within the ALICE project [19], integrating a LMS and a serious game to create a solution which can respond to input from LMS and adapt the game dynamically, an architecture is implemented which allows direct method invocation from LMS in-game and vice versa using XML log files. This allows assessment engines to use input from the game, and communicate feedback to the player, e.g. through a virtual companion's dialogue. Such methods provide a means for rich data capture on player interactions, and support blending and dynamic in game-based learning resources.

V. CONCLUSIONS AND FUTURE WORK

As we have seen in this paper, integrating serious games into a personalized learning environment has the potential educational benefits of combining a personalized delivery with increased learner motivation. The paper has shown how an integration of authoring tools and a serious game editing tool could concretely be achieved and can lead to an authoring environment for AH that include adaptive games and goes beyond the current state of the art in integration of AH and serious games. A logical next step is to build and test the proposed authoring environment. However, at this point another question is raised: what are the exact elements in the game that can be adapted and how might these impact different types of learners? For AH a comprehensive taxonomy exists [5] and there is a clear need for such taxonomy for adaptive games and an overview of how different techniques impact different learners.

ACKNOWLEDGMENT

This work is fully supported by the European Commission under mEducator (Grant ECP 2008 EDU 418006) and ALICE (Theme ICT-2009.4.2, Grant 257639)

REFERENCES

[1] L. A. Annetta, M. R. Murray, S. G. Laird, S. C. Bohr, and J. C. Park, Serious Games: Incorporating Video Games in the Classroom. Educause Quarterly. 2006.

[2] D.A. Kolb, Experiential learning: Experience as the source of learning and development. Prentice-Hall Englewood Cliffs. 1984.

[3] B.S. Bloom, Krathwohl, and D. R., Taxonomy of Educational Objectives: By a Committee of College and University Examiners. Longmans, Green 1956.

[4] A. Protopsaltis, L. Auneau, I. Dunwell, S. de Freitas, P. Petridis, S. Arnab, S. Scarle, and M. Hendrix, Scenario-based serious games repurposing. Proceedings of the 29th ACM

international conference on Design of communication. 11, 87–110 2001.
[5] P. Brusilovsky, Adaptive hypermedia. User modeling and user-adapted interaction. 11, 2001.
[6] N. Peirce, O. Conlan, and V. Wade, Adaptive educational games: Providing non-invasive personalised learning experiences. Digital Games and Intelligent Toys Based Education. 2008
[7] R. McGreal, Learning objects: A practical definition. International Journal of Instructional Technology and Distance Learning (IJITDL). 9, 2004.
[8] IEEE Learning Technology Standardization Committee, Final Draft Standard for Learning Object Metadata. 2002, http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf , viewed 11/02/2013.
[9] E. Duval, E. Forte, K. Cardinaels, B. Verhoeven, R. Van Durm, K. Hendrikx, M.W. Forte, N. Ebel, M. Macowicz, and K. Warkentyne, others, The Ariadne knowledge pool system. Communications of the ACM. 44, 72–78 2001.
[10] Advanced Distributed Learning Network, SCORM 2004, 4th edition, 2009, http://www.adlnet.gov/capabilities/scorm/scorm-2004-4th, viewed 11/02/2013.
[11] G. Gunter, R. F. Kenny, and E. H. Vick, A case for a formal design paradigm for serious games. The Journal of the International Digital Media and Arts Association. 3, 93–105 2006.
[12] M. O. Riedl, A. Stern, D. Dini, and J. Alderman, 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, 23–42 2008.
[13] F. Bellotti, R. Berta, A. De Gloria, and L. Primavera, Adaptive experience engine for serious games. IEEE Transactions on Computational Intelligence and AI in Games. 1, 264–280 2009.
[14] M. Csikszentmihalyi, Flow: The psychology of optimal experience. Harper Perennial 1991.
[15] N. Stash, Incorporating cognitive/learning styles in a general-purpose adaptive hypermedia system, PhD thesis. 2007.
[16] N. Dovrolis, S.T. Konstantinidis, P.D. Bamidis, and E. Kaldoudi, Depicting educational content re-purposing context and inheritance. Information Technology and Applications in Biomedicine. 1–4 2009.
[17] M. Meyer, T. Hildebrandt, C. Rensing, and R. Steinmetz, Requirements and an architecture for a multimedia content re-purposing framework. Innovative Approaches for Learning and Knowledge Sharing. 500–505 2006.
[18] A. Protopsaltis, D. Panzoli, I. Dunwell, and S. Freitas, Repurposing Serious Games in Health Care Education. XII Mediterranean Conference on Medical and Biological Engineering and Computing. 963–966 2010.
[19] I. Dunwell, P. Petridis, S. Arnab, A. Protopsaltis, M. Hendrix, and S. de Freitas, Blended Game-Based Learning Environments: Extending a Serious Game into a Learning Content Management System. Intelligent Networking and Collaborative Systems (INCoS). 830–835 2011.
[20] J. Torrente, P. Moreno-Ger, B. Fernández-Manjón, and J. L. Sierra, Instructor-oriented authoring tools for educational videogames. International Conference on Advanced Learning Technologies. 516–518 2008.
[21] M. Dougiamas, P. Taylor, Moodle: Using learning communities to create an open source course management system, World conference on educational multimedia, hypermedia and telecommunications. 2003.
[22] S. Gobel, L. Salvatore, and R. Konrad, StoryTec: A digital storytelling platform for the authoring and experiencing of interactive and non-linear stories. Automated solutions for Cross Media Content and Multi-channel Distribution. 103–110 2008.
[23] M. Fowler, K. Scott, UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Longman Publishing Co., Inc. 2000.
[24] P. De Bra, G.J. Houben, and H. Wu, AHAM: a Dexter-based reference model for adaptive hypermedia. Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots. 147–156 1999.
[25] F. Halasz, M. Schwartz, The Dexter hypertext reference model. Communications of the ACM. 37. 30–39 1994.
[26] N. Koch, M. Wirsing, The Munich reference model for adaptive hypermedia applications. Adaptive Hypermedia and Adaptive Web-Based Systems. 213–222 2006.
[27] J. Ohene-Djan, A Formal Approach to Personalisable, Adaptive Hyperlink-Based Interaction, PhD thesis, Department of Mathematical and Computing Sciences, Goldsmiths College, University of London. 2000.
[28] E. Knutov, GAF: Generic adaptation framework. Adaptive Hypermedia and Adaptive Web-Based Systems. 400–404 2008.
[29] M. Hendrix, P. De Bra, M. Pechenizkiy, D. Smits, and A. Cristea, Defining adaptation in a generic multi layer model: CAM: The GRAPPLE Conceptual Adaptation Model. Times of Convergence. Technologies Across Learning Contexts. 132–143 2008.
[30] A. Cristea, L. Calvi, The three layers of adaptation granularity. User Modeling. 145–145 2003.
[31] C. Gaffney, D. Dagger, and V. Wade, Authoring and Delivering Personalised Simulations—an Innovative Approach to Adaptive eLearning for Soft Skills. Journal of Universal Computer Science. 16, 2780–2800 2010.
[32] D. Dagger, V. Wade, and O. Conlan, Personalisation for all: Making adaptive course composition easy. Journal of Educational Technology and Society. 8, 9 2005.
[33] A. Cristea, M. Verschoor, The LAG grammar for authoring the adaptive web. Information Technology: Coding and Computing. 382–386 2004.
[34] O. De Troyer, F. Kleinermann, B. Pellens, and A. Ewais, Supporting virtual reality in an adaptive web-based learning environment. Learning in the Synergy of Multiple Disciplines. 627–632 2009.
[35] L. Oneto, F. Abel, E. Herder, and D. Smits, Making today's Learning Management Systems adaptive. Learning Management Systems meet Adaptive Learning Environments, Workshop at European Conference on Technology Enhanced Learning (ECTEL). 2009.
[36] P. De Bra, A. Aerts, and B. Rousseau, Concept relationship Types for AHA! 2.0. Proceedings of the AACE ELearn'2002 conference. 1386–1389 2002.
[37] K. Van Der Sluijs, J. Hidders, E. Leonardi, and G.J. Houben, Gal: A generic adaptation language for describing adaptive hypermedia. 1st International Workshop on Dynamic and Adaptive Hypertext: Generic Frameworks, Approaches and Techniques. 13–24 2009.