An Authoring Tool for Adaptive Digital Educational Games

Florian Mehm, Johannes Konert, Stefan Göbel, Ralf Steinmetz

Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany
{florian.mehm, johannes.konert, stefan.goebel, ralf.steinmetz}@kom.tu-darmstadt.de

Abstract. Digital educational games, especially those equipped with adaptive features for reacting to individual characteristics of players, require heterogeneous teams. This increases costs incurred by coordination and communication overhead. Simultaneously, typical educational games have smaller budgets than normal entertainment games. In order to address this challenge, we present an overview of game development processes and map these processes into a concept for an authoring tool that unifies the different workflows and facilitates close collaboration in development teams. Using the tool, authors can create the structure of a game and fill it with content without relying on game programmers. For adding adaptivity to the game, the authoring tool features specific user support measures that assist the authors in the relatively novel field of creating non-linear, adaptive educational experiences. Evaluations with users recruited from actual user groups involved in game development shows the applicability of this process.

1 Introduction

Digital Educational Games promise to combine the strengths of computer games (high acceptance especially among adolescents, high immersion, motivation and inherent learning by design) with the educational value of e-learning systems. It has long been suggested that this mixture can be beneficial to learning [24][25], and educational games have been on the market for a long time. A possible means for increasing the effectiveness and enjoyment of an educational game is the introduction of adaptivity, allowing a game to be customized for a specific player based on assessment of their state of learning or other characteristics. This approach has been used widely in e-learning tools (e.g. [6]) and can lead to a higher effectiveness of the resulting application [7]. However, in the context of games, it has seen only few adopters. Possible explanations for this are the increased efforts and the associated increase in production costs.

As will be detailed in section 3, the creation of an adaptive educational game requires special care during design, writing, and later on in the production because of the need for adaptable paths through the game which allow the game to be customized for a player. Also, since more content is required, production costs rise.

We propose that the major hurdles in the production of an adaptive educational game can be overcome by optimizing the production process. This is achieved by mapping the traditional roles and workflows to an authoring tool specialized in adaptive educational games, allowing close collaboration and minimizing overhead due to coordination in game production teams. Section 4 describes a concept for such an authoring tool, which is then realized prototypically in the Serious Game authoring tool StoryTec as shown in section 5.

2 State of the Art

As the focus of this paper lies on authoring tools for adaptive educational games, basic details of game-based learning cannot be expanded upon in detail here for the sake of brevity. For such basics, the reader is referred to Prensky's foundational text in [24] and a current account of developments in [10].

2.1 Adaptive Technology in e-Learning

In the area of e-Learning, adaptivity has been used in many commercial and academic projects [14]. Two areas where this approach has long been researched are adaptive hypermedia systems and intelligent tutoring systems (ITS). As specified in several publications, such systems typically feature several models used for adaptivity. Shute and Towle [26] name the following: A *content model*, indicating the learning domain and interdependencies between knowledge; a *learner model* summarizing characteristics of the learner; and an *instruction model* binding the two previously mentioned by assuring the learner is provided with the right information or assessment at the right time. The actual adaption is then handled by a software component referred to as an *adaptation engine*. Usually, the model of the learner is created by assessment using tests involving computer-readable exercise formats (such as clozes, multiple choice questions or drag & drop exercises). This user model is then used to select the content to present to the user.

2.2 Adaptive Digital Educational Games

In the field of adaptive educational games, fewer examples abound. For entertainment games, one of the main fields of work so far has been the work on dynamically adjusting difficulty e.g. [11]. In the field of procedural content generation, current ideas include the application of generation algorithms while the game is running, based on the current state of the game [27]. Lopes and Bidarra [16] provide an overview of challenges and methods in this field.

In addition to the models adaptive e-learning systems use, adaptive (educational) games can also account for play preferences, thereby building a *player model* and using this model for adaptation purposes. The PaSSAGE [28] project uses a model sorting players into one of 5 possible categories, other player modeling approaches use the model presented by Bartle [2].

Maciuszek [18] describes an architecture for educational games utilizing the strengths of the role-playing game genre that combines game-based learning with work from intelligent tutoring systems in order to create adaptive educational games. The 80Days project [13] created a game architecture allowing adaption of an educational game both on the local level (giving hints, changing difficulties, …) and the global level (different learning paths, …). Bellotti et al. [3] describe a refined architecture for adaptive serious games which treats adaptivity as an optimization problem and proposes to use genetic programming for solving this problem.

What unifies the cited examples is that assessment in the games is handled to be minimally disruptive of the gameplay. This is captured by the notion of *evidence* being collected from the game whenever a player completes or fails a task in the game [23].

2.3 Authoring Tools for Educational Games

In this section, we provide an overview of authoring tools which have been created specifically for the purpose of educational games. Tools in adaptive e-learning are often based on existing e-learning authoring tools which allow the creation of learning objects and add the possibility to control adaptive features. An example of this can be found in [5].

The major example of tools for educational games is the e-Adventure authoring tool [22], which is conceptualized as an authoring tool for adventure games. Using a simplified authoring language which user can program by selecting from a list of possible actions and conditions, non-programmers are addressed by this tool. By confining the tool to one genre, the realization of the authoring process can rely on a set of assumptions which limit the choice of authors and prohibits the creation of games from other genres than adventures. e-Adventure does not provide an automatic means of adaptation; however, adaptivity is possible by using the means of the authoring toolkit.

3 Adaptive Digital Educational Games

This section provides an overview how adaptation can be introduced into educational games, including the specialized requirements on game content resulting from this.

3.1 Narrative Game-Based Learning Objects

As pointed out in section 2, several possible axes along which adaptation in games can be carried out are available. We propose to choose narration (adaption of the game's story, play (using a player model as described above) and learning. This choice of adaptation axes is consistent with the previously presented concept of Narrative Game-Based Learning Objects (NGLOB), for in-depth information about this readers are referred to [9].

In order to structure the game for a game engine that can handle adaptation, a minimal unit of gameplay must be chosen. At a later stage, this allows authors to work on the game content in a similarly structured way, see section 4. In our work we propose the concept of a scene, similar to movies or stage plays. Thereby, a scene has a minimal context, involving a fixed set of characters, props, as well as logical objects necessary for the game engine, such as variables. Scenes can be hierarchically organized to allow better structuring of game content. For example, it is common to have the notion of a level in a computer game, in which graphical assets such as the level geometry or other features, such as background sounds, are shared. By hierarchically organizing scenes, these features are inherited by scenes lower in the hierarchy from those above them.

Each scene can then be seen as a Narrative Game-Based Learning Object and accordingly be annotated with relevant information about the scene for the adaptation algorithm. For storytelling purposes, this involves the narrative context of the scene, i.e. the function in the game's narrative this scene has. In order to formalize this, we make use of available narrative structures, such as the Hero's Journey (as mentioned in [9]). In order to adapt for gaming preferences, we utilize the notion of a player model capturing the different interest of gamers. As an example, the player model presented by Bartle [2] is used. However, the concept is flexible in this regard and allows other, similar models to be used or for authors to create their own player models customized for the game genre or content. Finally, for the purpose of learning, a learner model based on Competency-based Knowledge Space Theory (CbKST) [1] is used in which a scene can have different prerequisite competencies that are required to understand the educational content presented in the scene.

3.2 Adaptive algorithms

In a non-adaptive game, the unfolding of the game's story is controlled directly by the choices made by the player inside the space of options provided to him or her by the game's author. In the concept described here, this is modeled by transitions between scenes. After a transition has been triggered by an action of the player, the game switches from the old to the new scene.

For making the game adaptive, different possible paths through the game have to be created, each allowing adaptation by choosing a different variation based on the current state of the information about the player. In the presented model involving scenes and transitions as links between scenes, several methods for providing such paths are possible. One model is that of transitions which are marked as "free". Using this kind of transition, an author does not connect a given player input directly with a fixed transition, but rather with a set of possible transitions. Based on the possible scenes indicated by the free transitions and the models of the player, an adaptive algorithm chooses the most appropriate in the current context. This variation has the advantage that an author has a direct overview of the possible points the player can get to from a certain action and plan alternatives explicitly.

The second possible method for authors to indicate adaptive choices to the game engine is by providing pools of scenes. In this variation, instead of modeling each possible transition between scenes explicitly, all scenes are placed in a container, thereby implying a net of transitions linking all pairs of scenes. When the game gets to a section of the game modeled in this way, a sequence of free choices from the available scenes can be made before this section is left again. This allows a very modular approach to adaptive game authoring, since a large pool of scenes with different content, gameplay and parts of the narrative can mean that the game is assembled at runtime and can be adapted very specifically to the player in each play session. However, this model is at the same time more abstract for authors (especially those not used to creating non-linear experiences) since no fixed order of the scenes in the pool is apparent, making storytelling, the creation of clearly designed learning paths and a learning curve in the gameplay harder. Here, the use of a rapid prototyping tool as described in section 5.2 becomes paramount for authors to quickly test their choices.

3.3 Assessment

In order to update the models the adaptive engine uses at runtime, assessment has to take place. In the presented concept, the interactivity in the game (inputs to the game and outputs from the game) is modeled as sets of stimuli and responses. Each stimulus is an action carried out by the player, such as clicking a button in the game. Responses are sequences of actions, high-level instructions for changes in the game.

Each action by the player which can be interpreted to yield information about his or her current state is annotated with the corresponding information. For example, finishing a task requiring knowledge of certain facts indicates that the player has gained this knowledge while playing, while a choice between different story continuations with varying levels of action or social interaction can give information about the player's game preferences. In essence, this provides the adaptive engine with an interpretation of game evidence [23] as noted in section 2.2.

Each of the updates to the user models of the player should be balanced with previous information in order to lower the effects of errors of measurement and of concept drift [4], i.e. when a player initially prefers action-laden sequences and later grows more interested in social interaction. Therefore, different update strategies are possible. We propose a simple weighted update function, which takes into account the old value of a certain attribute of the player model with a weight alpha and the updated value with weight *1 – alpha*. The factor *alpha* then determines the importance of older information compared with newer updates.

4 Authoring Processes

The following section describes game development processes that are usually found in the creation of educational games and how these processes are mapped in the authoring concept described in this paper. The analysis of game development processes is based on various accounts, including [12] and a study carried out by the authors with a German educational game studio.

4.1 Educational Game Development Processes

The game development of an educational game in general is similar to a regular entertainment digital game, with the additional challenges of providing educational content. Traditional roles found in game development include *game designers*, who are tasked with setting up the game's story, world, characters and gameplay. *Technicians*, i.e. game programmers and associated roles, are then tasked with creating the technical infrastructure of the game and realizing the gameplay, while *artists* (graphical artists, sound artists, …) create the necessary assets such as 3D models or sounds for the game. Finally, the game's quality is assured by *testers* before being released.

In the creation of an educational game, the above user groups receive new tasks and simultaneously new groups are added. This already indicates the increased complexity compared to entertainment-only games. The development team is augmented with *domain experts*, who introduce specialized knowledge about the target domain, as well as *pedagogues* in order to establish an educational design of the game. Common tasks for these groups include the creation of exercises or exercise pools, whereby in practice commonly general purpose tools are used for the creation and dissemination of the created content to the rest of the team.

The core game development team, as mentioned above, also receives more tasks as compared to the development of an entertainment-only game. Since one major purpose of the educational game is the presentation of educational content, the game design has to be adjusted for this, either by providing possibilities for placing learning content in the game or by adapting the gameplay itself in such a way as to be educational. An example for the former could be an adventure game placing educational content in the dialogue with a character, while an example for the latter is a physics game involving actual simulated physics-based puzzles the player has to solve by simulation.

The necessity for close integration of educational content continues from the design to all aspects of the game, including the art production required to produce assets which conform to the educational content and the game programmers realizing educational features or adding mechanisms for adaptivity.

4.2 Challenges related to Adaptivity

When adding adaptivity to a Digital Educational Game, another layer of complexity is introduced in the processes. The design (especially concerning the narrative and the gameplay) has to be adjusted towards allowing adaptivity by providing several different ways to play the game or by different variations for the paths the players can take through the game.

Another challenge for designers and storywriters lies in the dynamic and algorithmic nature of adaptivity. The effect of the narrative paradox captures this to a certain degree [17]: while in a classical, linear medium such as a movie the consumer of the entertainment product has no freedom in choosing how the experience continues, a player in a game can influence the continuation of the game. This leads to the author of the game not being completely in control of the narrative, but having to foresee the possible actions of the player and providing the gameplay and story for each action. On top of this, adaptivity adds an element of uncertainty for the author since the actual continuation of the game depends on the current state of the user models, thereby potentially differing in each play session. Authors, especially those trained mainly with classical media, can find it difficult to retain an overview of the whole game and the flow of events in the game with the player and the adaptive algorithm influencing it continually. This effect can similarly be observed in e-Learning [8].

4.3 Mapping Authoring Processes into Authoring Tools

The challenges addressed in this section have been mapped into the authoring tool concept described in this paper and realized in the authoring tool StoryTec. In the following, we provide an overview how processes have been mapped into a unified authoring tool. Foss and Cristea [8] present a set of imperatives how e-learning authoring tools should support authors of adaptive content. These imperatives are in line with our concept.

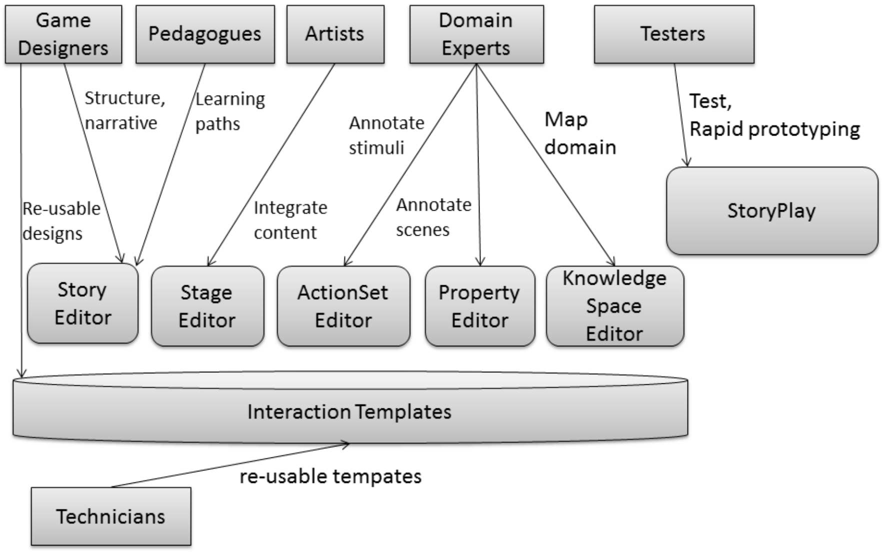

Fig. 1. Some of the possible mappings between user groups and the components of the described authoring tool. Note that not all combinations are shown, for example, game designers could also use StoryTec and StoryPlay in conjunction for storyboarding and rapid prototyping.

Figure 1 shows some of the mappings between users in the educational game development process and the components of the authoring tool. In general, user groups are separated in the way they interface with the authoring tool in order to contribute work towards a game. On a technical level, game programmers are tasked with building a basic framework on which the game can be run. This includes an environment in which the game can be executed (a game engine) as well as templates encapsulating various types of gameplay found in the game. Of course, the selection and specifics of these templates are governed by the initial game design carried out by game designers. These templates are programmed and integrated into StoryTec to be used in the creation of a specific game [21].

On the other hand, most users found in the game development process collaborate by using the authoring tool directly via its normal interface. Game designers set up the structure of the game and the individual levels/rooms by the visual interface. Artists add to these structures the finished assets, for example background graphics for the rooms the designers created. By providing a visual programming approach, game designers can directly manage the high-level flow of the game, while lower-level details are then handled by the interaction templates provided by the programming team.

For domain experts and pedagogues, special components are provided. For example, the creation of a knowledge structure for the learner model on which adaptation is based can be carried out in a graphical editor which visualizes the structure of the game's learning domain. Apart from this, they work in the same environment as game designers and therefore both groups are able to see the results of each other's work and collaborate. This helps in creating a common basis for communication about the tasks at hand.

Adaptivity is a central part of the authoring tool and therefore visible to all user groups. By means of the Story Editor providing a visual overview of the whole story and all paths through the game as well as the adaptive parts, users are supported in retaining an overview of the adaptive features of the game. The added effort for the creation of different, adaptable paths through the game is mitigated by variations being quickly creatable using the interaction templates and the possibility of copying and varying existing structures. Finally, a rapid prototyping tool with specialized visualization for adaptive algorithms and user models assists authors in understanding how the game will typically react during a game session by quickly testing out variations, with a prototype version of the game.

5 StoryTec

5.1 Authoring Tool

In this section, the authoring tool StoryTec[1] (cf. [19] among others) is described as a realization of the concept shown above, incorporating the workflows and processes as detailed in the last section. Special focus will be laid on the support of the creation of adaptive educational games.

The main user interface of StoryTec as seen in figure 1 is the principal interface for all user groups collaborating directly in the authoring tool, including game designers, artists and domain experts. Therefore, all important information is provided visually in the interface, and all functions for editing the game rely on simple concepts instead of programming languages or other more technical systems.

---

1 Available at http://www.storytec.de

The main overview of the whole project is the Story Editor, in which scene hierarchies, objects and transitions are created and visualized. Authors create the structure of the game in this editor or re-use a provided structural template. Furthermore, the adaptive systems of free transitions and scene pools are available directly in the Story Editor, allowing all collaborating users to see and manipulate them.

The process of defining scene contents is carried out in the Stage Editor and relies on the interaction templates included in StoryTec or added by game programmers for a specific game (genre). Therefore, this allows quick editing of the content of the game, since each scene is equipped with an interaction template handling the details of gameplay implementation in the game, only requiring the input of the necessary content and settings. Since this approach of encapsulating gameplay allows the creation of several games in the same genre, it can lower the costs of production by fostering re-use and more rapid development cycles.

Fig. 2. The main components of the user interface of StoryTec (from top left in clockwise direction): Stage Editor, Objects Browser, Property Editor, Story Editor.

The two other user interface elements in the standard configuration of StoryTec are the Objects Browser and Property Editor. The former is used for providing an overview of all available content objects and for adding them to any scene via drag & drop, while the latter is used to change parameters for objects and scenes.

Interactivity on a low level (graphics rendering, sound playback, camera control in 3D games) is intended to be handled by the game engine and the interaction templates due to their inherent complexity. Authors are empowered to configure high-level rules by using the ActionSet Editor (cf. figure 3), which connects each Stimulus (cf. section

3) with a set of Actions that should be applied in the game at runtime. Boolean conditions allow branching, thereby reacting to the current state of the game. Additional Actions are provided for assessment purposes, i.e. to update the user models for adaptivity. These actions are to be used whenever a Stimulus can be interpreted to indicate a change in a user model (e.g. a player solving a task that requires understanding a certain piece of knowledge). Since this system again is available to all collaborating users and does not require previous knowledge in game programming, it can increase collaboration and support rapid prototyping by allowing designers to quickly test game prototypes without waiting for a programmed prototype.

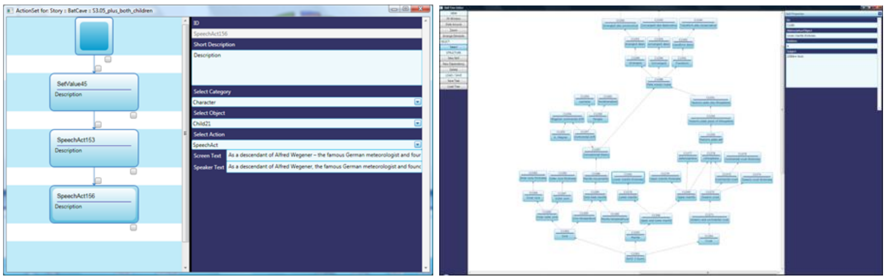

Fig. 3. Left: The ActionSet Editor of StoryTec, enabling non-programmers to structure the interactive flow of the application. **Right:** The Knowledge Space Editor. Boxes indicate facts or competencies, arrows indicate dependencies between them.

Specific support for the creation of educational games is offered in the Knowledge Space Editor (cf. figure 3). This editor assists in modeling the knowledge domain the game is based on by visualizing competencies and facts and the dependencies between them as a graph of boxes and arrows. A dependency here indicates that a certain competency A has to be understood before competency B can be addressed, a notion found in Competency-Based Knowledge Space Theory.

5.2 Runtime Environment

The complete StoryTec prototype also includes a full runtime environment for playing games created with the authoring tool. This includes, on the one hand, a player application intended to be used for evaluation purposes and as a rapid prototyping tool, and a multi-platform player application on the other hand.

As a basis for all provided player applications, several components are important to mention. The projects which are created by StoryTec are interpreted by a component referred to as Story Engine, which is linked to the game engine. The Story Engine acts as a high-level command instance, dispatching commands to the game engine and other components based on the parameters and actions the authors have set up in StoryTec. Concretely, it relays all gameplay commands to the game engine and the interaction template implementations included in the game engine and receives stimuli back from the game engine. For adaptivity purposes it includes the user models of the players whose updates it carries out based on information from the game as well as the algorithms for choosing how to continue in the game. Whenever an update in the game calls for an adaptive choice, all possible variations are considered (all free transitions or all scenes in a scene pool) and assigned a numerical value indicating their appropriateness when seen from a narrative, learning or play perspective. Depending on the overall goal of the play session, these values are then weighted in order to result in a choice of next scene. The chosen scene is that which yields the highest weighted sum of all values and conforms to all further constraints (e.g. not visiting the same scene again in a scene pool). For details of this process, see [9].

As described above, this basic architecture is realized in two player applications. The "StoryPublish" player is intended for cross-platform publishing a finished game. The second provided player application, "StoryPlay" [20], features a two-part user interface. One part is reserved for the gameplay and is therefore similar to StoryPublish. The second part visualizes current information such as the state of the user models, the history of previous choices by the adaptation algorithms as well as the state of variables. This tool can therefore aid authors in evaluating games concerning the effects of adaptivity by allowing checking the results of annotations and user models early during development. A slider allows quick tuning of the weights associated with the adaptive choices along the narrative, educational and play adaptivity axes.

6 Evaluation

Several evaluations of the presented approach have been undertaken (see also [19]). In the following, results of the evaluation studies will be highlighted. The subjects of the first evaluation were students involved in a course on Serious Games without previous exposure to StoryTec, with the study's focus being the general usability of StoryTec. It was carried out with 26 participants (1 female, 25 male, m = 25.2 years, SD = 3.71 years).

Basic principle	Mean value	Standard deviation
Suitability to the task	4.74	0.88
Self-description	3.51	0.93
Controllability	5.48	0.77
Conformity with user expectations	4.55	1.06
Error tolerance	3.42	0.80
Suitability for individualization	4.42	0.72
Suitability for learning	5.14	0.78

Table 1. Results of the usability questionnaire (Values range from 1 to 7)

The test was carried out with a variation of the "Thinking Aloud" method. The participants were assigned one of three roles: one participant read out the tasks aloud, the executing participant was given control of the computer running StoryTec, and an observer was asked to watch closely and give comments to the other two participants. In this way, the participants were encouraged to have conversations about the tasks at hand and how they could be solved in StoryTec. Afterwards, the participants were asked to rate StoryTec in a questionnaire based on the usability standard ISO 9241-10.

The results of the questionnaire, aggregated to the seven basic usability principles of the standard, are shown in table 1. The examination of the questionnaire results shows that there is a tendency of the study participants to rate the ergonomics of StoryTec positively.

In the second study, three professional game developers (aged 31, 37, 46; 2 male, 1 female) were first asked to complete a short series of tasks in StoryTec and give feedback by thinking aloud and commenting on their experience. After this first stage, the participants were lead through a guided interview during which they were questioned on the usability of StoryTec in their individual domains (including game design and game programming of educational games) as well as their assessment of the effects of using StoryTec for the creation of an educational adventure game. All participants commented that they were very interested in the approach of StoryTec and that they could imagine a finished version of the software being used in actual development of educational games. In the evaluated state the participants were able to imagine the tool being best suitable for storyboarding and prototyping. For the use in game production, they would require more detailed parameters than the prototype version offered.

Apart from tests concerning only the usability of StoryTec when seen in isolation and a focus group evaluation, a larger comparison study with the goal of comparing StoryTec with e-Adventure was carried out. A set of $N = 47$ test subjects were recruited from a university course on serious games (8 male, 39 female, age range from 21 to 32 years (m= 24.79; SD= 2.62;)). The experiment set-up consisted of a task that was phrased to be equally accomplishable in both tools which consisted of three tasks moving from simple to more complex interaction with the respective authoring tool. During each evaluation session, a group of up to 8 participants (each with an individual PC) was instructed to work for 25 minutes on the tasks in the first authoring tool and then for 25 minutes in the other authoring tool. The order in which the tools were evaluated was randomized per group of participants, with $n(1) = 25$ participants starting with StoryTec and $n(2) = 22$ starting with e-Adventure. After this, the participants were asked to fill out a questionnaire with individual sections for each authoring tool, again based on the areas of the ISO 9241-10 standard. Additionally, some background information, an assessment of the perceived level of mastery of the respective authoring tool and a comparative question between the authoring tools and demographical data was asked for.

Initial results indicate that StoryTec (m= 4.58; SD=1.17;) was preferred compared to e-Adventure (m= 4.21; SD=0.78;) by the participants ($p= .084$). Male participants were observed to rate StoryTec higher ($p=.023$) while female participants did not see a significant difference between the two tools ($p > .20$), which could be due to the low ratio of female participants. This interaction between gender (male, female) and tool (StoryTec, e-Adventure) borders significant ($p=.072$).

This evaluation also included performance data, including the time participants required for solving each task and the resulting project files, which are reviewed based on an objective set of rules for completeness and correctness. The result of this analysis has not yet been fully compiled and is therefore excluded here.

7 Conclusion

In this paper, we have presented an approach that maps the processes commonly found in the development of adaptive educational games into a unified authoring tool which allows structured and transparent collaboration between the involved user groups. It addresses the major problems found in the development of educational games, namely the higher costs of production due to differing tools and operation methods of different groups and the increased need for communication in order to collaborate effectively. Furthermore, the challenges faced when creating an adaptive game, including the need for authors to retain an overview of the game in its adaptive form and to create additional content for adaptive variations was included in the concept.

The concept has been realized as the authoring tool StoryTec and the associated player applications: StoryPublish for cross-platform publishing and for actual players; StoryPlay as a rapid prototyping and evaluation tool for authors. Evaluations of StoryTec have shown that users from the actual involved user groups (game developers, domain experts) have assessed the usability and usefulness of StoryTec in educational settings positively.

References

1. Albert, D., Lukas, J.: Knowledge spaces: theories, empirical research, and applications. Routledge (1999).
2. Bartle, R.: Hearts, clubs, diamonds, spades: Players who suit MUDs. Journal of MUD research. 1, 1, 19 (1996).
3. Bellotti, F. et al.: Adaptive Experience Engine for Serious Games. IEEE Transactions on Computational Intelligence and AI in Games. 1, 4, 264-280 (2009).
4. Black, M., Hickey, R.J.: Maintaining the performance of a learned classifier under concept drift. Intelligent Data Analysis. 3, 6, 453-474 (1999).
5. Bontchev, B., Vassileva, D.: Courseware Authoring for Adaptive E-learning. 2009 International Conference on Education Technology and Computer. 176-180 (2009).
6. Brusilovsky, P.: Developing adaptive educational hypermedia systems: From design models to authoring tools. Information Sciences. 377-409 (2003).
7. Conlan, O., Wade, V.: Evaluation of APeLS - An Adaptive eLearning Service based on the Multi-model, Metadata-driven Approach. In: De Bra, P. and Nejdl, W. (eds.) Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2004). pp. 291-295 Springer, Berlin / Heidelberg (2004).
8. Foss, J.G.K., Cristea, A.I.: The Next Generation Authoring Adaptive Hypermedia : Using and Evaluating the MOT3 . 0 and PEAL Tools. Complexity. 83-92 (2010).
9. Göbel, S. et al.: Personalized, Adaptive Digital Educational Games using Narrative, Game-based Learning Objects. Entertainment for Education. Digital Techniques and Systems 5th

International Conference on E-learning and Games, Edutainment 2010, Changchun, China, August 16-18, 2010. Proceedings. pp. 438-445 Springer, Berlin/Heidelberg (2010).
10. Harteveld, C.: Triadic Game Design. Springer (2011).
11. Hunicke, R., Chapman, V.: AI for Dynamic Difficulty Adjustment in Games. Assessment. 91–96 (2003).
12. Kelly, H. et al.: How to build serious games. Communications of the ACM. (2007).
13. Kickmeier-Rust, M.D., Albert, D. eds: An Alien's Guide to Multi-Adaptive Educational Computer Games. Informing Science Press, Santa Rosa, USA (2012).
14. Knutov, E. et al.: AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. New Review Of Hypermedia And Multimedia. 15, 1, 5-38 (2009).
15. Kunin, T.: The construction of a new type of attitude measure. Personnel Psychology. 51, 4, 823-824 (1955).
16. 16. Lopes, R., Bidarra, R.: Adaptivity Challenges in Games and Simulations: A Survey. IEEE (2011).
17. Louchart, S., Aylett, R.: Solving the narrative paradox in VEs – lessons from RPGs. Intelligent Virtual Agents. 2792, 244–248 (2003).
18. Maciuszek, D., Martens, A.: A Reference Architecture for Game-based Intelligent Tutoring. In: Felicia, P. (ed.) Handbook of Research on Improving Learning and Motivation through Educational Games Multidisciplinary Approaches. IGI Global (2011).
19. Mehm, F. et al.: Authoring Environment for Story-based Digital Educational Games. In: Kickmeier-Rust, M.D. (ed.) Proceedings of the 1st International Open Workshop on Intelligent Personalization and Adaptation in Digital Educational Games. pp. 113-124 (2009).
20. Mehm, F. et al.: Bat Cave: A Testing and Evaluation Platform for Digital Educational Games. Proceedings of the 3rd European Conference on Games Based Learning. Academic Conferences International, Reading, UK (2010).
21. Mehm, F. et al.: Introducing Component-Based Templates into a Game Authoring Tool. In: Dimitris Gouscos, M.M. (ed.) 5th European Conference on Games Based Learning. pp. 395-403 Academic Conferences Limited, Reading, UK (2011).
22. Moreno-Ger, P. et al.: Adaptive Units of Learning and Educational Videogames. Journal of Interactive Media in Education. 2007, 05, 1-15 (2007).
23. Peirce, N. et al.: Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences. 2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning. 28-35 (2008).
24. Prensky, M.: Digital Game-Based Learning. Paragon House (2007).
25. Rieber, L.P.: 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).
26. Shute, V., Towle, B.: Adaptive E-Learning. Educational Psychologist. 38, 2, 105-114 (2003).
27. Smith, G. et al.: PCG-based game design: enabling new play experiences through procedural content generation. Proceedings of the 2nd International Workshop on Procedural Content Generation in Games PCGames 11. (2011).
28. Thue, D. et al.: Learning Player Preferences to Inform Delayed Authoring. Psychology. pp. 158-161 (2007).