

Enabling Control of 3D Visuals, Scenarios and Non-linear Gameplay in Serious Game Development Through Model-Driven Authoring

Sofie Van Hoecke¹(✉), Koen Samyn², Gaétan Deglorie¹, Olivier Janssens¹, Peter Lambert¹, and Rik Van de Walle¹

¹ Multimedia Lab, ELIS, Ghent University-iMinds, Ghent, Belgium
sofie.vanhoecke@ugent.be

² DAE, University College of West-Flanders, Kortrijk, Belgium
koen.samyn@howest.be

Abstract. Due to the absence of high-level authoring environments and support for non-technical domain experts to create custom serious games, a model-driven authoring framework is presented in this paper. Through model-driven authoring, non-technical people can manipulate the 3D visuals of their serious game, model the scenarios of the game, and even easily add non-linear narrative to the game. The different tools and methods have been implemented and are currently used to build a serious game for the Friendly ATTAC project in order to help youngsters who are confronted with cyberbullying. The presented model-driven authoring framework enables non-technical domain experts to produce serious games easily and quickly, at a lower cost, and therefore lowers the barriers that hinder the production of serious games.

Keywords: Serious game · Cyberbullying · Adaptivity · Scenario generation · Non-linear gameplay · Sandbox · Model-driven authoring

1 Introduction

Serious games are games considered to be both educational and fun. The need is raised for high-level authoring environments in order to (1) allow people without professional game design skills to create or modify serious games [1], and (2) enhance serious game development as development costs are one of the major drawbacks [2].

To solve this shortcoming, this paper presents a model-driven authoring framework that enables non-technical domain experts to produce serious games easily and quickly, at lower cost, and therefore lowers the barriers that hinder the production of serious games. The authoring framework is used within the Friendly ATTAC project, in which an innovative serious game is developed to help youngsters deal with cyberbullying (bullying via electronic communication tools) by focussing on the bystander behavior. Cyberbullying is a relatively recent phenomenon that especially occurs among early adolescents (12–15 year olds).

As it may have a serious impact on the mental (and physical) wellbeing of victims, many societal actors are currently involved in anti-cyberbullying initiatives. They strongly plea for evidence-based, appealing serious games that empower youngsters confronted with cyberbullying.

The remainder of this paper is as follows. Section 2 outlines the general concept of our model-driven authoring framework that semi-automatically translates theoretical and empirical knowledge into attractive virtual scenarios and allows to add non-linear narrative. The subsections describe the steps of the process in more detail. After presenting screenshots and video footage of the resulting game against cyberbullying in Sects. 3, 4 summarizes our work.

2 Model-Driven Authoring Framework

Figure 1 presents the general concept of our model-driven authoring framework. Several steps have to be followed in order to generate scenarios and levels for the serious game. Firstly, social scientists have to write scenarios which incorporate their theories and hypotheses they want to test or model. Secondly, via the authoring framework they can translate their scenarios into ATTAC-L scenarios and add non-linear narrative. The ATTAC-L language is a graphically illustrated modelling language for educational virtual scenarios, understandable by non-technical people. In parallel, they can model the visual aspects of the game levels using our own sandbox that offers features tailored to the development of serious games. The third step consists of the XML file generation based on the ATTAC-L scenarios on one hand and the sandbox configuration on the other hand. The fourth step takes the XML file and translates it into actual game level and moves.

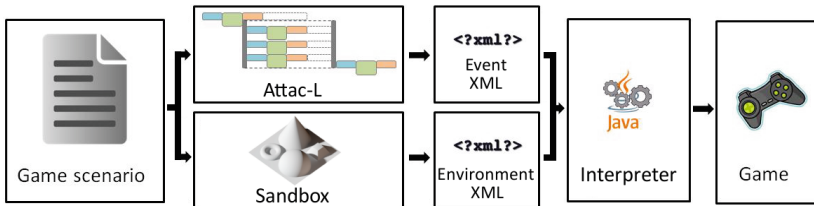


Fig. 1. General concept of process to generate game scenarios and levels

In the next subsections, the different steps of the process are described in more detail. First, we define how scenarios and game levels can be designed using ATTAC-L. Second, we describe how non-linear narrative can be generated in order to adapt the gameplay at runtime based on the player's progress. Third, we describe the sandbox to manipulate the 3D elements of the game. Finally, the produced XML files of these three steps are translated by the interpreter into actual game levels and moves.

2.1 Designing the Scenarios and Levels Using ATTAC-L

A domain-specific modeling language, called ATTAC-L, is developed within the Friendly ATTAC project [6] to accommodate domain experts in the design process of scenarios and allow the semi-automatically creation of serious games.

By using the ATTAC-L control structures such as communication, sequence, choice and concurrency structures, it is possible to create entire scenarios and game levels in ATTAC-L. For more information on the scenario generation using this ATTAC-L, the reader is referred to [7]. Once the scenarios and levels are constructed in ATTAC-L, they are translated to an XML file in the back-end of the authoring framework.

2.2 Adding Non-linear Narrative with Scenario Templates

Non-linear gameplay is needed to provide an optimal learning experience and teach certain behaviour or knowledge to the users of the serious game by adapting the gameplay based on the real-time player input or post-game feedback. Therefore, a story engine is created that selects a scenario template from a scenario database based on the current state of the player model and provides the template with contextual information from the game. The story engine is also responsible for maintaining the integrity of the storyline, defined by additional metadata in the templates. This way, the player's learning state can be tracked in real-time and used to adapt the gameplay and occurrence of learning challenges in order to better suit the problem areas of the player. An overview of the story engine is presented in Fig. 2.

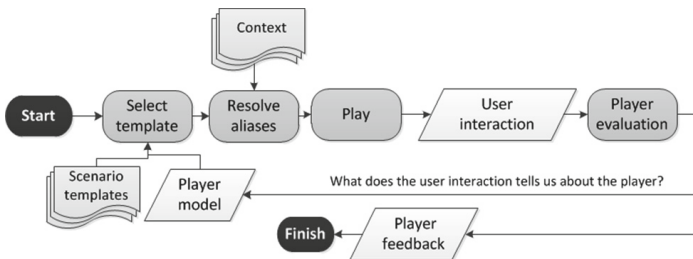


Fig. 2. Story engine overview

When the game requires the start of a new scenario, a scenario is selected from a database of scenario templates. The selection process takes the current state of the player model (i.e. the relative skill of the player in each of the specific learning challenges) into account, including the previous actions and the current location of the player. The scenario might also require the proximity or availability of non-player characters (NPC's). Furthermore the scenario might define the characteristics (traits and facets) of these NPC's. Once a scenario

is selected, the aliases in the template are replaced with the appropriate and currently available NPC's and/or items. The scenario template can require an NPC, with a predefined set of personality characteristics (e.g. bully trait vs victim of bullying), and/or a specific state from the player model. Next the actions in the scenario are played or executed, and possible options are presented to the player. The decision of the player will trigger an evaluation and leads to an update of the player model. The player model is used to adapt the probability that a given scenario template will be selected, given the performance objective to which it belongs. For more information on generating non-linear narrative of serious games using these scenario templates, the reader is referred to [8].

2.3 Sandbox

Most game engines provide developers with a sandbox, i.e. an environment in which the 3D elements of a game can be manipulated. The sandbox is typically tightly coupled with the features of the game engine and this introduces two major problems for serious game development. The first problem is that extending the game engine to make it suitable for serious game development is not trivial. Another problem is that efforts are mostly non portable. As a result, a game developed within sandbox A for game engine A cannot easily be converted to run on another game engine.

Within our model-driven authoring framework, we also want to describe the visual aspects of the game in a model driven manner in order to solve the two mentioned problems. We therefore introduce our own sandbox that offers features tailored to the development of serious games. The sandbox exports the serious game levels as XML files, but it also possible to extend the sandbox by writing a plug-in that exports the sandbox in another desired file format.

The properties and actions that an object offers are described in an XML file. An example of such a descriptor is given in Listing 1. This descriptor makes it possible to create UI customizers for every supported object, but also to add new features to existing objects, or to add an entirely new object.

Listing 1. Object descriptor

```
<object label="Sound" class="dae.prefabs.standard.SoundEntity">
  <parameters section="Transform">
    <parameter type="string" id="name" />
    <parameter type="float3" id="localTranslation" />
    <parameter type="sound" id="soundFile" />
    <parameter type="range" id="volume" min="0" max="10" step="0.1" />
    <parameter type="boolean" id="looping" />
    <parameter type="boolean" id="positional" />
    <parameter type="action" id="playSound" method="playSound" />
  </parameters>
</object>
```

Aside from the basic features that every sandbox must possess, our solution focuses on users with limited or no skill with regards to the creation of a 3D virtual world. Therefore care has been taken to provide the users with sufficient tools to place and manipulate objects. An example of such a tool is

the *insert object tool* which uses ray casting to make it easier to quickly “drop” an object on the scene. The tool can also take the surface of the current scene into account to create a perfect alignment (e.g. placing an object on a ramp). Another example is the lighting tool which shows the changes to lighting in real time. A demonstration of these features and more can be seen on <http://youtu.be/F37eClz8Xyg>.

An additional feature of the sandbox is the layer and level system, allowing users to quickly switch from one context to another. Items like lights, way points, trigger boxes and paths are automatically put in separate layers. Switching a layer’s visibility on and off makes it easier to manage an entire level.

Finally, the sandbox design is extensible. The object system makes it possible to define new components that can be attached to existing objects. For example, for the Friendly ATTAC project, it is necessary to add trait and personality information to NPC’s. Listing 2 shows the method to define the UI customizers for such a ‘Personality’ component. A demonstration of adding this ‘Personality’ component, a path, physics and animation to characters is presented in <http://youtu.be/u7jBSJ7c4nY>.

Listing 2. Define component

```
<component id="Personality" class="dae.characters.Personality">
  <!-- add customizers for personality component -->
  <parameters section="Personality">
    <parameter type="range" id="bully" min="0" max="10" step="0.1"/>
    <parameter type="range" id="friendly" min="0" max="10" step="0.1"/>
  </parameters>
</component>
```

The addition or removal of components from an object will affect all the existing objects of the same type. This way, useful components can be reused without the need to redefine the parameters.

Listing 3. Add component

```
<object id="NPC" class="dae.NPC">
  <components>
    <!-- add Personality component to every NPC object -->
    <component id="Personality" key="personality"/>
  </components>
  <parameters section="Transform">
    <parameter type="string" id="name"/>
    <parameter type="string" id="animation"/>
    <parameter type="string" id="actorName"/>
    <parameter type="string" id="actorId"/>
    <parameter type="float3" id="translation"/>
    <parameter type="float3" id="rotation"
      converter="QuaternionConverter"/>
  </parameters>
</object>
```

The resulting visual aspects of the game levels are exported by the sandbox as XML files, and given to the code generator.

2.4 Interpreter

The constructed XML files of, on one hand, the scenarios, levels and scenario templates, and, on the other hand, the visual aspects of the game levels, are then

used by our interpreter to translate them into actual game levels and moves using the in-engine format. This interpreter consists of hard-coded bridges between the ATTAC-L XML and the in-engine format of the game engine, and ensures that the correct scenarios and game moves are activated or executed based on the input and actions of the player. This way, the interpreter allows to hide the implementation details of the virtual world from the scenario writers.

3 Resulting Game Against Cyberbullying

As the Friendly ATTAC serious game focuses on helping youngsters deal with cyberbullying, a school context was chosen. This way, on one hand, the likelihood of impacting the social behavior of even poorly motivated youngsters increases, and, on the other hand, a nice game can be developed to draw their attention. The school context and characters of the Friendly ATTAC game are presented in Fig. 3. The resulting game is an online, single player, adaptive and personalised role-playing (RPG) detective game, called ‘The Messenger’. The scenarios and interactions within the game are based on theoretical and empirical knowledge regarding personal and contextual determinants of cyberbullying, obtained by performing a well-established intervention method from the field of health psychology, i.e. Intervention Mapping (IM) [3]. The game attempts to modify behavior patterns of bullies, bystanders and victims by allowing youngsters, through the use of virtual scenarios, experience different roles (bully, victim, or bystander) in cyber bullying incidents, and to react to those experiences in order to get adjusted feedback based on their individual reactions. This way, we hope to make youngsters more aware of the consequences of certain behaviour (as a bully or as bystander) and help them to prevent becoming a victim. Bullies will be encouraged to cease bullying by increasing their empathy, victims will be taught adequate coping strategies and bystanders could be sensibilized to intervene. The model-driven authoring framework has been implemented and is currently used to build the different levels of the Friendly ATTAC game. Figure 4 shows some screenshots of the resulting game. Video footage can be seen at <http://youtu.be/qwmLJzMdFzw>.



Fig. 3. School context and characters of the Friendly ATTAC serious game

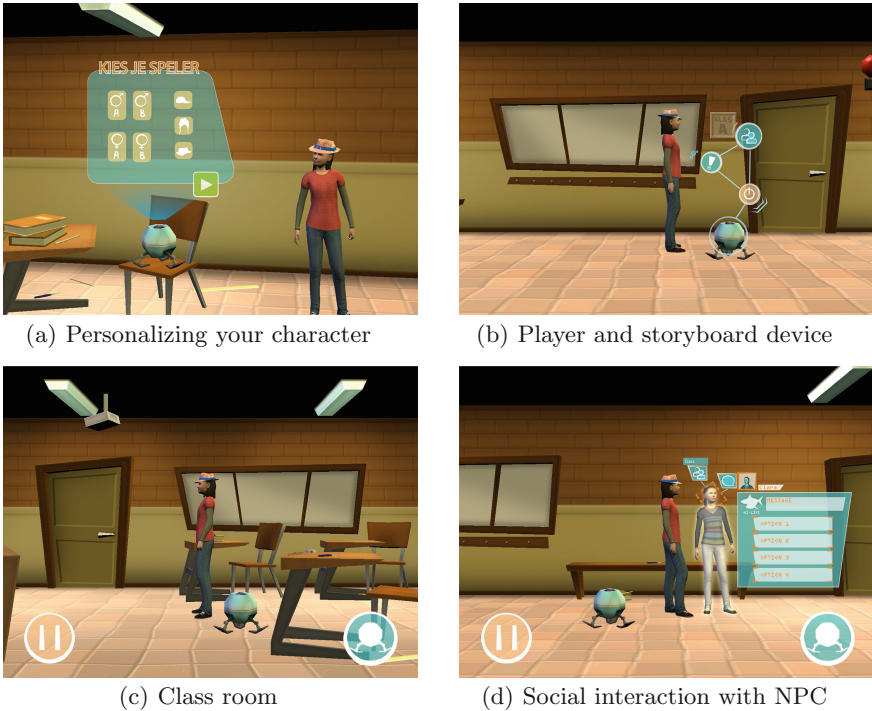


Fig. 4. Screenshots of Friendly ATTAC serious game

4 Conclusions and Future Work

We presented a model-driven authoring framework that allows non-technical people to manipulate the 3D visuals, model the scenarios, and easily add non-linear narrative to the game. The different tools and methods have been implemented and are used to build a serious game for the Friendly ATTAC project in order to help youngsters confronted with cyberbullying.

By using a virtual game scenario generation process, it is possible for non-technical people to model virtual scenarios. Technical details are provided on how the scenarios can be modelled in ATTAC-L and afterwards be translated in computer interpretable XML. This XML is then used to automatically build the scenario and game moves within the game engine so that it can be played.

Non-linear narrative is introduced using scenario templates and aliasing. The aliasing approach facilitates the design process by using scenario templates as a way to create a dynamic experience with an individualized storyline inside the game. Specific to the topic of serious games and learning is the creation of a player-driven narrative that tests the player on his/her skills. This is achieved through the mapping of performance objectives to each scenario template using metadata and by creating a model of a player that is adapted towards.

Finally, by using the presented sandbox, the visual aspects of the game can be manipulated and described using XML, making our sandbox better tailored to the development of serious games.

The presented model-driven authoring framework enables non-technical domain experts to produce serious games easily and quickly, at a lower cost, and therefore lowers the barriers that hinder the production of serious games.

As for future work, improvements in several areas are envisioned. First, the vocabulary of ATTAC-L verbs will be expanded, allowing more actions to become possible in the game. Second, linked templates with smart selection will be implemented by populating the pool of available event templates only by those events that can follow the previous one. Third, our work focuses on the aliasing aspects of characters. However, a story may also require geographic bifurcations. E.g. the main character could choose a different path to reach the destination. In this case a scenario would not only contain aliases for characters but also for geographic locations (e.g. hidden caves, treasures, or bridges).

Acknowledgment. This work was funded by the IWT SBO Friendly ATTAC project, a project of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT): <http://www.friendlyattac.be>

References

1. Djaouti, D., Alvarez, J., Jessel, J.P.: Can gaming 2.0 help design serious games? a comparative study. In: SIGGRAPH, Los Angeles, CA, USA (2010)
2. Bellotti, F., Berta, R., De Gloria, A.: Designing effective serious games: opportunities and challenges for research. *Int. J. Emerg. Technol. Learn.* **5**, 22–35 (2010)
3. Bartholomew, L.K., Parcel, G.S., Kok, G., Gottlieb, N.H., Fernandez, M.E.: *Planning Health Promotion Programs: An Intervention Mapping Approach*. Wiley, San Francisco (2011)
4. Aylett, R.S., Louchart, S., Dias, J., Paiva, A.C.R., Vala, M.: FearNot! - an experiment in emergent narrative. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) *IVA 2005. LNCS (LNAI)*, vol. 3661, pp. 305–316. Springer, Heidelberg (2005)
5. Riedl, M.O., Stern, A., Dini, D., Alderman, J.: Dynamic experience management in virtual worlds for entertainment, education, and training. *Int. Trans. Syst. Sci. Appl., Special Issue on Agent Based Systems for Human Learning* **4**(2), 23–42 (2008)
6. Van Broeckhoven, F., De Troyer, O.: ATTAC-L: a modeling language for educational virtual scenarios in the context of preventing cyber bullying. In: *IEEE 2nd International Conference on Serious Games and Applications for Health* (2013)
7. Janssens, O., Samyn, K., Van de Walle, R., Van Hoecke, S.: Educational virtual game scenario generation for serious games. In: *IEEE 3rd International Conference on Serious Games and Applications for Health* (2014)
8. Samyn, K., Deglorie, G., Lambert, P., Van de Walle, R., Van Hoecke, S.: Generating non-linear narrative for serious games with scenario templates. In: *Proceedings of GRAPP, Berlin, Germany* (2015)