Generation of adapted learning scenarios in a serious game: lessons learnt

Pierre Laforcade[0000−0001−8498−2731]

LIUM (Computer Science laboratory), University of Le Mans, France
pierre.laforcade@univ-lemans.fr

Abstract. This article presents the lessons we learnt during the development of a generation component in the *Escape it!* learning game. They are presented according to the different development stages of the generator. They may be considered useful by designers or researchers sharing similar contexts and objectives.

Keywords: Serious Game · Adaptation · Design · Generation

1 Introduction

Adaptivity is an important success factor to enhance the learning facet of serious games [11]. In this article we are interested in adaptive serious games and more specifically adaptive *learning games*. Adaptations may refer to some learning side targets (learning resources, activities, etc.), game side targets (difficulty, gameplay components, etc.) or both at the same time. Our concern is about personalizing the levels presented to the learner/player by only considering the learning dimensions (progress in the skill tree). Such adaptations at runtime are generally called generations. The perimeter of our research is then about learning games proposing the generation of adapted scenarios (ordered activities) according to the available game elements and learners' progress.

Even if some researches have dealt with the generation of serious game components [9, 6, 12, 1], only a few research works [5] propose guides, generic frameworks or approaches to support the design and development of such generators. This led us to propose a dedicated 3x3x2 approach ([8, 7]) while developing a serious game in the *Escape it!* project. Our proposition is design-centered because it guides and supports the formalization of models and metamodels that capture three dimensions of the adaptation (context, game components, scenario) into three iterative perspectives (objectives, structures, features). Nevertheless, it impacts all stages of the development of the generator (analysis, design, implementation, and re-engineering). Because we met with different issues during the development of the generator, we intend to share the lessons we have learnt. Researchers and designers sharing similar objectives and context could benefit from this feedback information.

We propose the following organization. Section 2 presents the context of the *Escape it!* project and serious game. Section 3 gathers all lessons according to the development stages they refer to. Finally, Section 4 concludes this paper.

2 Context

2.1 The Escape it! project

The project aims to develop a mobile learning game dedicated to children with ASD (Autistic Syndrome Disorder). The purpose of the game is to support the learning of visual skills derived from a curriculum guide [10]: matching an object to another identical object, sorting objects into different categories, making seriation of objects, etc. It will be used to reinforce and generalize the learning skills [2].

The project involves autism experts and parents.The main challenge is about the proposition of a large variety of playable scenes to learners in order to support the visual skills generalization. Because the engineering costs of hard-coded scenes could not be considered, it has been decided to study the development of a dedicated generator that will dynamically build playable game scenes in relation to learners' profile. A Design-Based Research method has been conducted on account of its usefulness as an exploratory research adapted to study research issues while producing designed artifacts.

2.2 The Escape it! serious game

Overview The main gameplay consists, for players, in finding some relevant objects in the displayed scene, and dragging and dropping them to appropriate places in order to respect the association / match / sort / ... objective. When all required actions are performed, the door is unlocked, giving access to the next level. The game design relies on the best practices from the literature [3][16] and the recommendations/requirements expressed by the ASD experts involved in the design sessions. Our game fits the ASD requirements by its targeted skills from the ABBLS curriculum [10] and the use of ABA pedagogical key features (fading guidance by the pairing adult, positive rewards, no task failure, etc.)[2]. The main concerns are listed below:

- Targeted skills: a subset of the visual performance skills derived from [10] (those that can be adapted for a mobile gameplay).
- Variable game sessions: the game proposes from 3 to 5 levels at the convenience of the pairing adults or the children themselves.
- Scenes as meaningful living places: for example, the *bedroom*, the *kitchen* and the *living room*.
- Adapted difficulty: the difficulty level is set according to the current child's progress related to the targeted skills. Basically, three successful activities for a same skill (along one or several game sessions) raise the difficulty level for this skill.
- Generalizing the acquired skills: scenes have to be changed in accordance with previous difficulty levels. Hence, the game proposes non-identical challenges for the same skill. We quote variation examples: changing the background elements of a scene, adding background elements to disrupt visual reading; changing the objects to find and handle; adding other objects that are not useful for the resolution; hiding objects behind or into others.

The current prototype Various screenshots of the current prototype are illustrated in Figure 1. To present them briefly, the children with ASD, supervised by adults, ask for a new game session (1) and choose the number of levels composing the session (3), for example a 4-level length.The generator is then called to provide an adapted scenario (4). Children can visualize the current progress within the session (5). They successively solve the 4 levels (6-7-8-9). The progress screen (5) is updated and shown after each completed level. When the session is over, an end-game screen is shown (10), and a reinforcer element is won (11). The pairing adults can access the secured children' profile in order to see their progress (2 and 12), their history reports, etc.

Fig. 1. Overview of the different screens during a learning game session

3 Lessons learnt while developping the generator

3.1 Preliminary stage

Focus on the design of the generator When the serious game is already developed, the integration of an adaptation component can benefit from a lot of information already delimited about the game. Nevertheless, when both learning game and generator have to be considered from scratch, as it was for the *Escape it!* case-study, the design of the generator and the overall design of the serious game are hard to differentiate. Learning domain experts cannot focus on the adaptation part without considering overall aspects.

For example, during our participatory design sessions with experts and parents, the discussions were not limited to the runtime adaptation but also related to the game aesthetics and sound environment (adaptation at design-time to the children's sensory profiles), about the regulation of the children's activities (prompts, guidance, feedback, reinforcements), or about the tracking system that will be used to update the children's profiles after a game session. Other information also concern some design-time adaptations (the game uses some best practices design to be adapted to children with ASD), or runtime customizations to be made by the pairing adult (parameters adjustments in the profile screen).

Identify the generation characteristics Adaptations are generally characterized [4] by their intention (goal), the element to be adapted (target), the elements to be adapted with (sources) and their strategy [15]. We think that identifying these information is important to guide the design of adaptations as well as the generation of adapted elements.

We illustrate them for the *Escape it!* serious game. **Intention** (what for): individualization of learning sessions; **Trigger** (when): after the children choice about the session length (3, 4 or 5 levels); **Target** (what): a learning scenario as an ordered sequence of elements configurations declaring the initial setup of the game levels; **Sources** (according to what): current progress of the children (skills and difficulty levels); **Participating elements** (with the help of what): available scenes (kitchen, living room, etc.) and their components (objects, locations, hideouts...); **Level of automation**: full, no human intervention; **Feedback time of the result**: just after its generation; **Generation approach**: composition of existent elements within every scene, with random selections.

The generator as a black-box component The learning game should be considered as a complex object composed of inter-related components. The generation part can then be analyzed as a black-box software component with inputs and outputs, or as a service consuming and producing data. It is useful to identify the generator role, avoiding considering related but external tasks as under the responsibility of the generator. As inputs we can distinguish the sources and the participating elements (cf. previous characteristics). Sources inputs may vary for each generation whereas participating elements are more likely to stay invariant.

The *Escape it!* generation component produces as output a scenario composing of 3 to 5 levels configurations including the initial location of objects to find, objects solution, hideouts, etc. The source input is the child's profile (current progression for each skill) including the context-sensitive choice of the session length. The additional input is the description of the game in terms of skills, scenes and game objects. The generator is not in charge of updating the children profiles according to their results during the learning sessions (tracking system concern). It is also not responsible of setting up the levels presented to learners according to the scenarios descriptions (game engine concern). By analogy with some existent procedural context generation taxonomy [14] the Escape it! generation could be considered *online* (during the runtime), *necessary* (the content has to be correct), *parameterized* (the generator takes as an input the game description model), *stochastic* (randomness is used when several combinations are possible), *constructive* (the algorithm never produces broken content).

3.2 Specification stage

The main objective of the specification is to capture and model the elements (with their properties and relations) and the generation rules involved in the generation process. This specification will drive the implementation stage. It is important to identify these information, even more to avoid considering information not necessary to the generation.

An iterative and incremental specification centered on the target The generation design can be eased by decomposing the specification into successive but complementary perspectives. They are all centered on the specification of the target as a first class element.

The 3x3x2 approach [7] suggests to consider at first the *Objective* perspective. It refers to the selection of targeted learning objectives according to the user's profile. In the *Escape it!* project, the scenario to generate is then an ordered sequence of the visual performance skills that will be considered; these skills are selected in accordance with the number of levels to generate, the considered skills available in the game and the child's progression.

The second additional perspective is about the *Structural* part of the scenario. It consists in the selection of learning game exercises or large grain game components that are compliant with the previous selected skills. In our context, we focus on the various scenes that are compliant with the previously chosen skills. This scenario specifies correspondences between the selected pedagogical large-grained resources (i.e. scenes) and their targeted skills.

Finally the third *Feature* perspective completes the scenario by specifying additional inner-resources/fine-grained elements along those which are compliant with the previous chosen skills and large-grain components. In the *Escape it!* project, this concerns the initial configuration of each scene in terms of additional background elements, objects to find, hideouts, etc.

Specification into 3 inter-related dimensions Each of the three previous perspectives can be specified through 3 inter-related dimensions. The first one to consider is the scenario to generate. It will drive the other dimensions. The specification consists in modeling the generic domain concepts, properties and relations, required for the generation of scenarios. In our work we use metamodels representations to this aim but other formalisms, ontologies for example, could be considered.

In order to avoid repetitions we propose that the *Scenario description metamodel* elements that are known before the generation are captured into the *Game description metamodel* and referenced by the *Scenario metamodel*. That is a debatable choice because it complexifies metamodels and implies cross-references into the models. Nevertheless we consider it plays a valuable part in supporting the complex and subjective modeling work, improving the overall quality of the design.

The *Game description metamodel* is then the abstract syntax declaration of the additional elements required by the generation. These elements are considered as a model conformed to this metamodel. As a model they explicitly declare the game elements according to the 3 perspectives (objectives, structure and features): skills, resources or exercisers, in-game objects, etc.

The third metamodel to consider captures the description of the source context: the learner model and other context elements. Similarly to the game dimension, the metamodel specifies the abstract syntax describing all children's profiles, but only the elements that will be useful for the generation (the ones involved in the generation rules). Children's concrete profiles will be part of models in conformance with this metamodel. The 3x3x2 approach does not suggest to specify 9 different metamodels (3 perspectives and 3 dimensions) but to consider the iterative and incremental completion of only 3 metamodels (scenario, game and context).

As an illustration, we propose in Figure 2 a visual representation of metamodels and models for the *objective* perspective in the *Escape it!* case-study. To explain briefly, the Scenario metamodel expresses that the Objective part is composed of various *TargetedSkill*. Each one of them has a *difficulty level* and references a concrete *Bxskill* among those defined in the description of the game. These *BxSkill* are part of the *Domain* section of the *GameDescription*. Each *Bxskill* has a textual description and can refer to another *Bxskill* as a *prerequisite* or refer to several ones as available *Bxskill* when the former will be achieved. On the *Profile* part, the desired number of levels is specified. Several *Skill2Consider* indicate the current *difficulty level* and *progress* for the referenced *Bskill*. The game description model concretely describes 7 *Bxskills* with their *prerequisite/unlockedBx* relations. The learner profile is for a fictive "Tom" who would like a 5-level length new game session according to his current difficulty levels and progress for the 7 considered skills. Finally, the scenario model is an example of generated model in conformance with its metamodel. It is composed of 5 targeted skills. We can notice that the first one concerns the B19 skill at an "advanced" difficulty level.

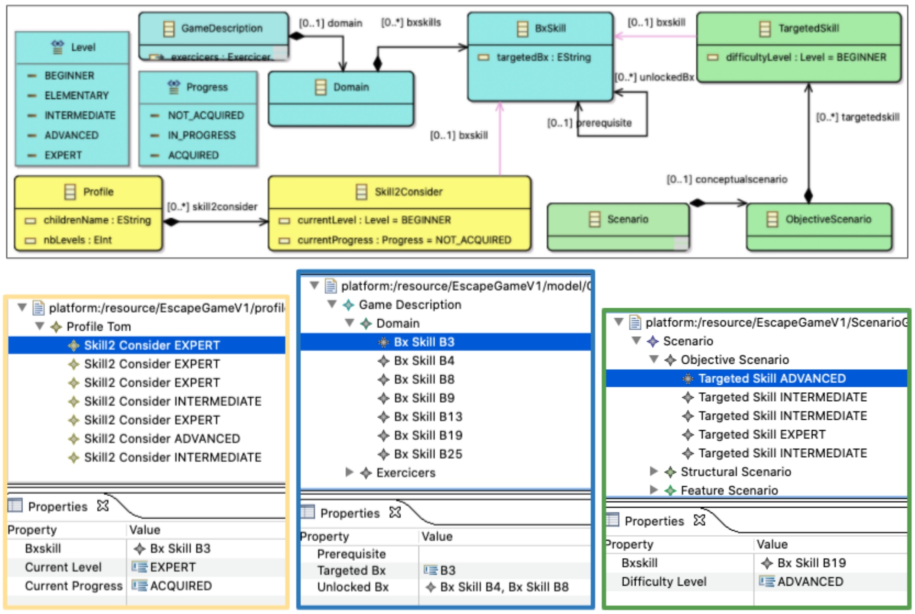

Fig. 2. Visual representations of the 3 inter-related metamodels (top), source input model (bottom left), game input model (bottom center), and output scenario model (bottom right), for the *Objective* perspective.

Consider generation rules at a *meta* level Generation rules describe how to generate the scenario elements, properties and relations according to the information from a given child profile and the game description model. These rules can be specified based on the model information or the metamodel ones. They can be directed by the selection of input information to generate output ones or, on the contrary, be centered on the generation of target elements with respect to input conditions. In our case study, the generation rules are target-centered, at the meta level and expressed in a textual format with references to the named elements from the metamodels. These informal specifications are a reformulation of the information given by experts using the natural language.

In addition to the *Objective* perspective, the following rules have been considered: there are as many *TargetedSkill* as the *nbLevels* value; each of these *TargetedSkill* refers to an "eligible" Bxskill, i.e. 1/ that is referenced by a Skill2Consider, in the learner profile, which has an "in progress" *currentProgress*, and 2/ the potential *BxSkill* as *prerequisite*, if it exists, being also referenced by a *Skill2Consider* having at least an "intermediate" difficulty level. All the referenced *BkSkill* from a *TargetedSkill* must be different if possible. Finally, the difficulty level of the *TargetedSkill* has the same value as the one from the *Skill2Consider currentLevel* pointing at the shared *BxSkill*. These rules are not understandable by experts because of their expression close to the metamodels syntax. Nevertheless they are useful to drive their implementation. Experts do not refer to the metamodel elements obviously. For example, they stated that *the scenario to generate can only refer to skills that are currently in progress and for which their skill prerequisite has at least reached the 'intermediate' level*. That is simpler to express than the equivalent translation we specified.

3.3 Implementation stage

A specified generation can then be implemented to be effective. The usual approach consists in implementing it as a part of the serious game, using the same frameworks, languages or architectures. But another solution is possible.

Machine-readable specification models Considering that the previous specifications have been modeled thanks to a practical *Model Driven Engineering* tooling, they can be used to generate code to load, save, and handle the previous metamodels and models. Only the generation rules still require to be coded. Other formal or semi-formal modeling frameworks, like ontologies tooling, can ease or support the use of the specified models as machine-readable models to be used at runtime by the generator.

In the *Escape it!* case study we used the *Eclipse Modeling Framework* [13]. The metamodels are *ecore* file, and the models are XMI-formatted files. The generation of scenarios is implemented as a model transformation written in Java/EMF mainly based on a Java code generated from the *ecore* (meta)models by the EMF tooling. The generation rules are then hard-coded into the generator code.

Integration of an independent generator Having an independent generator from the serious game is helpful to develop and maintain it without any consideration for the game engine, mechanics, aesthetics, as well as the tracks collecting, the updating of the profile, etc. Nevertheless it is an integration challenge because of the potential gap between the required (scenario) and provided (learner profile, context...) formatting from the serious game perspective, and the ones required (profile...) and provided (scenario) from the generator perspective.

In our case study, the generator is deployed on a Web server and accessible through a Web service. The learner profile from the Unity-based serious game is an XML file that required some transformations, on the server side, to be in conformance with our metamodels. Similarly, the generated scenario is transformed before being sent to the serious game, in order to be compliant with the expected format.

3.4 Re-engineering

Similarly to the other parts of the serious game development, the generator cannot be designed and implemented without considering the evolving stakeholders' needs, the changes in the generation choices, the evolution of the game, etc.

Experts of the learning game domain are not experts in adaptation Serious game stakeholders are not adaptation or generation experts. They do not have a precise perception of the generation rules to consider. They need some feedback about their expressed rules and strategies to make some adjustments, or to remove / add some new rules.

In our project, some initial rules have been finally rejected ("80% of the considered skills in the scenario must concern mastered skills whereas 20% focus on new skills). The rule was not relevant with a 3-to-5-level length for the scenario. Experts finally propose 5 difficulty levels to always generate appropriate difficulties. Some other rules have been adjusted like the one stating the minimum and maximum range for the different categories of objects (backgrounds, hideouts, objects to find, useless objects, etc.) to add in the scene, according to the current level difficulty.

Evolution of the serious game There are many reasons leading to reconsider the current metamodels and models involved in the generation. The serious game can evolve on both learning and gaming facets. It can imply to update the game description model, if it only concerns content changes, or also the game description metamodel for deeper changes. The learner profile can also be impacted by these changes. Differently, new learner information or other context elements can be available to be used into the generation rules; the context model and metamodel can then be updated. Generation rules can also have to be updated in case of input metamodels changes. Finally, the scenario model and metamodels can also be updated according to deep changes in the serious game.

We only encountered the situation of adding new scenes into the *Escape it!* learning game. It implied to declare them appropriately in the game description model to be directly taken into account by the current generator.

4 Conclusion

This paper gives us the opportunity to present the lessons we learnt during the development of a generation component of the *Escape it!* learning game. We present them according to the different stages of the development. This may be considered useful by other designers or researchers sharing similar contexts and objectives: proposing the generation of adapted scenarios that take into account context information, including learners' profiles, and the different components provided by the learning game. This adaptation is necessary in order to propose a large variety of situations to learners. The generation can contribute to reducing the cost of hard-coded situations.

Nevertheless, our results also point out the difficulty to express, specify and validate the generation rules. It is an important obstacle to overcome in the future. Our current research works are focusing on this point.

References

1. Belahbib, A., Lotfi, E., Bouhorma, M., Bakkali Yedri, O., Slimani, A., Fatiha, E.: Serious games adaptation according to the learner's performances. International Journal of Electrical and Computer Engineering **7**, 451–459 (02 2017)
2. Burton, L.R., McEachin, J.: A Work in Progress: Behavior Management Strategies and a Curriculum for Intensive Behavioral Treatment of Autism. New York: DRL Books (1999)
3. Ern, A.: The use of gamification and serious games within interventions for children with autism spectrum disorder (January 2014)
4. Grubišić, A., Stankov, S., Žitko, B.: Adaptive courseware: A literature review. Journal of universal computer science **21**, 1168–1209 (10 2015)
5. Hussaan, A., Sehaba, K.: Consistency verification of learner profiles in adaptive serious games. In: 11th European Conference on Technology Enhanced Learning. pp. 384–389. Springer International Publishing (2016)
6. Janssens, O., Samyn, K., Van de Walle, R., Van Hoecke, S.: Educational virtual game scenario generation for serious games. In: Proceedings of the IEEE 3rd International Conference on Serious Games and Applications for Health (SeGAH'14) (06 2014)
7. Laforcade, P., Laghouaouta, Y.: Supporting the Adaptive Generation of Learning Game Scenarios with a Model-Driven Engineering Framework. In: European Conference on Technology Enhanced Learning (ECTEL'18). pp. 151–165. Leeds, United Kingdom (Sep 2018)
8. Laforcade, P., Loiseau, E., Kacem, R.: A Model-Driven Engineering Process to Support the Adaptive Generation of Learning Game Scenarios. In: CSEDU. vol. 1. INSTICC, Funchal, Madeira, Portugal (Mar 2018)
9. Melero, J., El-Kechai, N., Yessad, A., Labat, J.M.: Adapting Learning Paths in Serious Games: An Approach Based on Teachers' Requirements. In: Computer Supported Education - 7th International Conference, CSEDU 2015, Lisbon, Portugal, May 23-25, 2015, Revised Selected Papers, Communications in Computer and Information Science, vol. 583, pp. 376–394. Springer (2016)
10. Partington, J., Analysts, P.B.: The Assessment of Basic Language and Learning Skills-revised (the ABLLS-R) (2010)
11. Ravyse, W., Blignaut, S., Leendertz, V., Woolner, A.: Success factors for serious games to enhance learning: a systematic review. Virtual Reality (09 2016)
12. Sina, S., Rosenfeld, A., Kraus, S.: Generating content for scenario-based serious-games using crowdsourcing. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. pp. 522–529. AAAI Press (2014)
13. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework 2.0. Addison-Wesley Professional, 2nd edn. (2009)
14. Togelius, J., Yannakakis, G.N., Stanley, K.O., Browne, C.: Search-based procedural content generation: A taxonomy and survey. IEEE Transactions on Computational Intelligence and AI in Games **3**(3), 172–186 (2011)
15. Vandewaetere, M., Desmet, P., Clarebout, G.: The contribution of learner characteristics in the development of computer-based adaptive learning environments. Computers in Human Behavior **27**, 118–130 (01 2011)
16. Zakari, H.M., Ma, M., Simmons, D.: A review of serious games for children with autism spectrum disorders (asd). In: International Conference on Serious Games Development and Applications. pp. 93–106. "Springer International Publishing" (2014)