HAL
open science

Design of a Component-Based Mobile Learning Game Authoring Tool

Pierre-Yves Gicquel, Sébastien George, Pierre Laforcade, Iza Marfisi-Schottman-

To cite this version:

Pierre-Yves Gicquel, Sébastien George, Pierre Laforcade, Iza Marfisi-Schottman-. Design of a Component-Based Mobile Learning Game Authoring Tool. Games and Learning Alliance conference, Dec 2017, Lisbonne, Portugal. pp.208-217. hal-01635581

HAL Id: hal-01635581

https://hal.science/hal-01635581v1

Submitted on 15 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Design of a Component-Based Mobile Learning Game Authoring Tool

Pierre-Yves Gicquel, Sebastien George, Pierre Laforcade, and Iza Marfisi-Schottman

-
{pierre-yves.gicquel,sebastien.george,pierre.laforcade,iza.marfisi}@univ-lemans.fr

-

Abstract. Mobile learning games (MLGs) have great potential in education, especially in fields requiring outdoors activities such as botany or cultural heritage education. However, the number of mobile learning games actually used for outdoor education remains very low. The absence of dedicated applications allowing to build MLGs, without technical expertise, is certainly one of the most important factors. To overcome this limitation, we propose a design method for MLG authoring tools, based on reusable components. We describe in this paper the design and first results of Moggle-Designer, a full-web MLG authoring tool requiring no technical expertise. We detail how a model-based approach, combined with component-oriented programming, adequately allow the representation and manipulation of didactic expert knowledge. Finally, we present Moggle-Player, an application for running the designed MLGs on any mobile browser. Moggle-Designer was used to design several mobile games that have been tested during a pilot study in a botanical park.

Keywords: Mobile Learning, Serious Games, Authoring Tool, Situated Learning

1 Introduction

Mobile devices, such as mobile phones or tablets, have proven to be interesting mediums for computer supported learning. These devices are particularly interesting out of the classroom, for authentic learning activities. Mobile Learning Games (MLGs) present great pedagogical interest because they favor environment exploration and self-regulated mediation through conversation between learners [1]. These exploratory and conversational activities allow sustainable learning and appropriation through collaborative knowledge building [2].

However, the cost for producing usable MLGs remains prohibitive for most educational structures. The few available authoring tools are either very generic, and require a strong technical expertise, or only allow the design of overly simple and specific learning scenarios [3].

This raises the question of how to design a simple and generic MLG authoring tool. In this paper, we propose the use of a component-based approach. First, we define our research context and questions, then we present the methodology for collecting the needs of teachers in terms of games scenarios and for formalizing these scenarios through a model driven approach. We will then present the Moggle-Designer authoring tool, along with Moggle-Player, which allows users to play the authored MLGs on any mobile browser. Finally, we present the results of a first experimentation in real conditions, in which subjects played various gaming scenarios conceived with Moggle-Designer in a botanical park.

2 Research Context and Questions

The research context of this paper is the ReVeRIES project[1] (French acronym that translates to dreams and that stands for Educational, Interactive and Fun Plant Recognition on Smartphones). This project aims to introduce mobile applications in botanical education. It relies on the assumption that the urbanization of society has drastically reduced common botanical knowledge, leading to a gradual increase in so-called *plant blindness*, defined by [4] as *the inability to recognize the importance of plants in the biosphere and in human affairs*.

In order to motivate children and botanical neophytes to learn about plants and explore their natural environment, we intend to design MLGs that will help to learn about plant characteristics. We define a MLG as a mobile application, combining educational content with a playful scenario to favor learning. This kind of application relies on mobile devices' features (e.g. location, orientation and proximity sensors, media capturing and recording, augmented reality) to contextualize activities. One of the project goals is to propose an MLG authoring tool that does not require any programming knowledge. This authoring tool has a generic base but specific features could be added for enriching MLG with domain-specific activities. In the case of botany, we propose activities based on automatic image analysis to recognize plants from photos of their organs (e.g. leaves, barks, flowers). This aspect is detailed in [5].

Conceiving an authoring tool in the context of mobile learning games raises research questions we address in this paper:

- How to collect and formalize designers' needs, expressed in natural language, in an appropriate formalism for conception? (section 3)
- How to facilitate the creation of MLGs based on this formalism and deploy them on the players smartphones? (section 4)

3 Modeling Mobile Learning Games

This section is dedicated to the modeling issues related to the complexity of the MLG authoring tool. Indeed, the authoring tool should allow the design

[1] http://reveries-project.fr

of a large variety of MLGs. Such learning scenarios require a domain-specific modeling language, at least as a conceptual language to drive the design and development of the authoring-tool. Designing a generic but still botany-oriented modeling language relevant for any botanical and mobile learning scenario is not an achievable objective. Our modeling approach aims to capture, explicit and validate a broadly-usable instructional language. This language being then extended with botany specific concept for the design of botany MLGs scenarios.

The next sub-sections detail our global approach, the meta-model we propose, and some feedback about its validation for designing botanical MLG scenarios.

3.1 Overview of the Modeling Process

Our modeling process follows several steps:

1. **Collecting informal (textual, drawings...) scenarios of botanical MLGs**: collaborative sessions with various actors involved in the ReVeRIES project (including botanical park managers, botanical scientists, and teachers) were conducted. Participants were split into groups and had to propose a botanical MLG for the *Echologia* natural park in Laval, France (a map was available about the tree species present in the park). The main objective was to identify and collect a large range of MLG activities. Six scenarios were produced and discussed during a collective retrospective [6].
2. **Describing a conceptual model for botanical MLG scenarios**: it offers high-level concepts, similar to the natural concepts used by teachers, and is rich enough to adapt to various types of MLGs [7]. Without going into detail, the model formalizes a MLG as a series of *SituatedGameUnits* each related to a *POI (Point Of Interest)*, i.e. a geographical zone involving zero or several *objects of interest* (mainly plants although not restricted to them). Each *SituatedGameUnits* can be described in terms of optional elements: *triggers* (to start the unit), *clues* (to help the player find the POI), *guidance* features to the POI (marker on a map, GPS, vibrations, etc.), *validation* actions (to validate the presence of the player on the POI), *OnSiteActivities* (activities to do around the POI), *rewards* (points/items/... in relation to the arrival on the POI and/or according to the success of the OnSiteActivities), *pedagogical information* (to debrief with the player the knowledge acquired in relation to this unit).
3. **Formalizing a specific meta-model for botanical MLG scenarios**.

Because the first two points have already been detailed in previous publications ([6] and [7]), we detail here the third point: the botanical MLG meta-model.

3.2 Meta-modeling Approach

The meta-model we propose is a formal and explicit description of all the concepts, properties and relations required to model a botanical MLG scenario. MLG scenarios are the models, and they are conform to this meta-model. This meta-model has several objectives:

- to verify that the MLG scenarios can be formally specified (i.e. machine-readable with no ambiguity) in order to be exploited by the MLG player;
- to drive the design of the MLG authoring tool: we can check the consistency between the scenario elements (i.e. the visual information depicted in the mock-up screens and the interactions between screens);
- to drive the specifications and the development of the components-based MLG player engine.

One of the main issues to tackle is to ensure the expressiveness, completeness, consistency and the non-reducibility of the meta-model. This will help reduce the design and development complexity of the authoring and player tools. In order to obtain such a meta-model, we progressively merged all the models of MLGs it needs to cover. Indeed, during the creativity session with the ReVeRIES members, we realized that the designed MLGs actually had three different types of scenarios:

- linear scenario (e.g. treasure hunt): game units are chained up (use of guidances to drive the player to the POI);
- emergent scenario (e.g. interactive walk): game units start when players are physically close to a POI;
- hub scenario (e.g. geocaching): all game units and clues are available and depicted on a map, players decide in which order they want to play them.

For each of these scenario types, we therefore proposed a dedicated meta-model, and merged it to the meta-model. The merge relies on a (meta-)model composition in the meaning of the Model-Driven Engineering (MDE) principles. Briefly, we concretely merge the meta-concepts sharing a same semantics, gathering their potential different meta-attributes and meta-relations. Concepts that cannot be aligned are added, while elements that can be mapped to existent ones are removed. We also integrated game ingredients (e.g. rewards, points, inventories and virtual items) in order to cover all the elements of the designed MLGs.

Concretely speaking, this meta-model was specified using the Eclipse Modeling Framework [8] included in the *Obeo designer community* IDE. Specifying the MLG meta-model allowed to describe formally the cardinality and restriction between the elements of the meta-model (such as *OnSiteActivities*).

4 Moggle-Designer

4.1 Presentation

Moggle-Designer (MObile Geolocated Games for Learning) allows non-programmers to create MLGs with a bottom-up design process: learning and gaming components, such as media (e.g. images, videos, formatted text) or point of interest, are first defined as basic building blocks.

These blocks can then be aggregated to construct more complex components. For instance, a guidance activity is composed of a point of interest (the destination) and a medium (feedback) that is displayed when the user reaches this destination. Finally, these building blocks are linked together to create a game unit, which can be seen as a self-contained, independent and complete MLG [7].

Game units Game units follows a similar pattern, organized in three phases. This first phase is guiding the player to the place of interest (POI). The designer can chose between two guidance modes, either the POI and user position are displayed on a classic map on the player phone, or a more playful mode where a circle grow as the user get closer to the POI.

Second phase is triggered by the player reaching the POI. The player is then proposed a learning task where she has either to answer questions related to the plant at this POI, or to correctly identify a specie by scanning the correct QR-code (various QR-code being placed on three at the POI). When the player correctly complete a learning task, she is rewarded by a bonus item stored in a personal inventory. For instance the bonus recipe of rozenbottel jam is acquired when the user correctly identify a Rosa Canina.

The third phase consist of proposing a feedback media that gives complementary information about the learning task (e.g information such as an easy way of identifying Rosa Canina from its fruit color).

4.2 Design Principles

Reusable and Shareable Building Blocks One common limitation of authoring tools in Educational Game Authoring tools, is the lack of non-trivial working examples that teachers can use/reuse to guide them [9]. To reduce this issue and favor appropriation of Moggle-Designer by instructors, it encourage, through its design, sharing and reusing game components at different levels of granularity (from simple media component to game unit and complete scenario).

Figure 1 illustrates the relation of composition between shareable building blocks. When an instructor uploads an image file, it appears as a shareable component. The instructor can then create a document containing the image accompanied by informative text. This newly created document, is then used as part of a Multiple Choice Question.

Ownership and Security of Building Block In order to favor sharing while keeping a personal space for the user, we designed the system such as unidentified users could not access any building blocks. Moreover, users have the liberty to easily share or unshare each of their own contributions. Each block has a 1-1 relation with its creators user-id: the blocks being stored in a document-oriented database, are stored, as a whole, with each of their properties. This allows to easily create unique hash from blocks and track their use.

Terminology and Usability [10] underlines the critical importance of clear terminology and guidance in a user interface which should *speak the user's language*. In the case of an authoring tool using building blocks, these block must be easily apprehended by the user.

By proposing different granularities in the building blocks, with complex blocks being created from more simple blocks, we aim at easing the learning curve. An unexperienced user will easily be able to apprehend the principles of

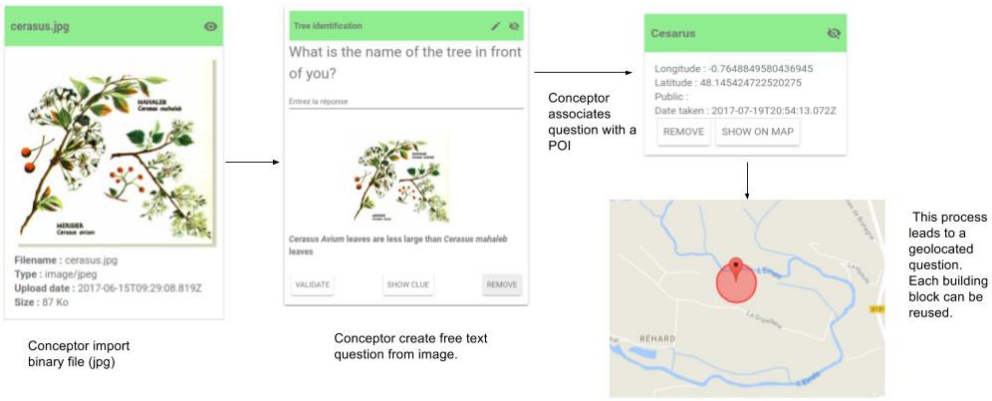

Fig. 1. Creating a situated question by combining building blocks

simple building blocks (e.g. image and text). Having acquired confidence and effectiveness in the conception of simple building block, the process of creating more complex building block by composition of simple ones will be more intuitive. Moreover, in order to encourage designers to explore and experiment, a direct visual feedback (WYSIWYG) is displayed during the block creation.

Advantages and Technical Challenges of Web Applications We decided to build Moggle-Designer as a web application. Various reasons guided this choice: first, web applications are platform independent. Moreover, the distribution of a web application is straightforward, there is no installation process for the users and the URL always point to the last stable version. From a technical point of view, we used the Web Component paradigm[2], a recent evolution in Web Application promoted by the W3C. The fundamental idea behind Web Components is to create easily reusable pieces of HTML and JavaScript code. Web Components provide a way to encapsulate the HTML/CSS and encapsulate the JavaScript within the context of execution of the component. This means one can use someone else's components with the complete guarantee that there will be not be any side effects on his/her own components.

4.3 Moggle-Player

Moggle-Player is a tool that allows users to play scenarios designed with Moggle-Designer. It is also a full web application, thus it is usable on any mobile device with a browser (Iphone, Android, Windows phone...). The game player makes full use of mobile HTML5 capacities (Figure 2): use of GPS for guiding the

[2] https://www.webcomponents.org/specs

user to the different games steps, use of the phone camera for activities based on scanning a QR-code to identify a tree and use of websockets to permit chat between users.

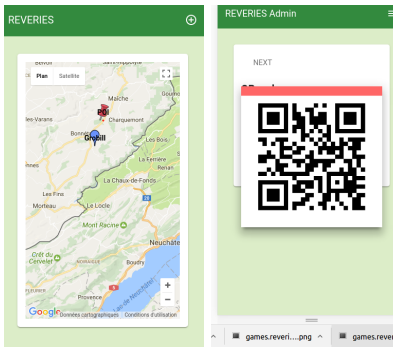

Fig. 2. Use of GPS and phone camera

One particular challenge we faced was to keep the web application running in absence of Internet connection. We set up a cache system (a JavaScript service worker) that preloads everything resources needed to keep the current MLG running when there is a disconnection and synchronizes back with the server when the network connection is restored.

5 Qualitative Pilot Study

We conducted a pilot study in a botanical park to evaluate how MLGs designed with Moggle-Designer were used and how the activities were perceived. We used three of the MLG scenarios designed during the ReVeRIES design sessions.

5.1 Participants and Apparatus

Twelve users took part in the experimentation (four on each MLG scenario) but only nine completed the feedback form, because some users actually shared the same smartphone. The nine that answered the feedback form were aged between 18 and 38 years old (mean: 28, SD: 8), and included four women and five men. Users where from three different backgrounds: four where fellow researchers from our laboratory, five were students in first year of computer science studies and three were working at the botanical park. User's personal smartphones where used for the experiment (these included iPhones and Android devices). All the devices were connected to a wireless mobile network, with variable connectivity.

5.2 Organization

We voluntarily gave little information to the players in order to assess the clarity of the MLGs. Three animators from the project team were present in case of technical problems, but they were actually solicited very rarely. The main reasons of solicitations were problem concerning geolocation, some smartphones having poor GPS accuracy.

The sessions lasted approximately 45 minutes. The MLGs were composed of six situated unit games each, implying a short walk through the park to reach the points of interest and progress in the game.

5.3 Main Results

We collected direct verbal feedback during and after the activities and also asked users to complete a questionnaire to assess their satisfaction and their interest in MLGs in the context of a botanical park. The verbal feedback was very positive: users enjoyed the fun aspect of having to reach a specific zone and having to observe the plants in the surroundings in order to answer questions.

The analysis of the questionnaire confirmed the verbal feedback: user found the MLGs motivating for the visit and useful to acquire knowledge in that botany. The questionnaire consisted mostly of Likert based unique choices questions with four possible items. A synthesis of the answers is presented in the figure 3.

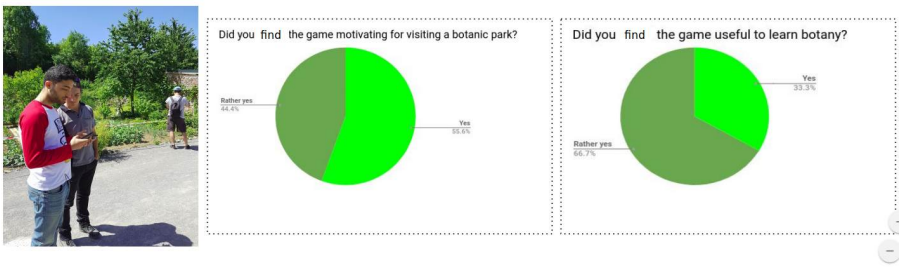

Fig. 3. Results of feedback form

While encouraging, these results need to be confirmed in a larger scale study. Nevertheless, this study demonstrates that Moggle-Designer can produce complete and usable MLGs.

6 Related Works

In this section, we compare two well established authoring tools for mobile learning applications to Moggle-Designer. A complete state of the art can be found in [11].

6.1 ARIS

ARIS is a Web based authoring tool that generates IOS Mobile Learning Games. While generating native mobile applications has some benefits, in terms of performance for instance, it greatly limits the possible users. Nevertheless, ARIS is particularly powerful, it allows the creation of different kinds of MLGs, from treasure hunts to location-based storytelling [12]. However, this expressive power comes with a steep learning curve. [13] argues about ARIS that the *challenge is learning the user interface and game logic [...] many who admire the possibilities of this platform find creating and understanding the logic of requirements for objects in a game scenario challenging.* Finally, one key difference between ARIS authoring tool and Moggle-Designer is the inability to share resources in ARIS. This design strongly limits re-usability of learning objects.

6.2 MLearn4Web

MLearn4Web [14] is a full web authoring tool allowing teacher to create web learning application. The tool is very easy to apprehend, applications are built as a serie of pages, each page being constructed visually by drag-and-dropping items on the page from a component pallet. First study shows a very good degree of acceptance among teachers.

However, MLearn4Web presents limitations in terms of design possibilities. Applications can only be designed and shared as a whole, that is to say its not possible to design subpart of an application or to reuse part of someone else application. Moogle design principles in contrast allows sharing at different level of granularity, which makes produced games modular and easily reusable. Moreover, MLearn4Web do not allow to implement game mechanics, such as challenges or rewards for instance. This limit the learner experience to be more passive than active. In contrast, the application designed with Moogle incorporate game mechanism, requiring the learner to actively investigate her environment to answer questions.

7 Conclusion and Future Work

We present, in this paper, a complete design process for MLG (Mobile Learning Game) authoring tools, from the collection of users' needs to implementation details. The MLG scenarios were formalized through a model-based approach while web-components structurally and semantically conform to the model elements were implemented.

This process led to **Moggle-Designer**, an authoring tool powerful enough to design the types of scenarios expressed by future users, while needing no technical expertise to build these scenarios, and **Moggle-Player**, capable of playing these MLGs on any mobile device. An experimental validation in the real setting of a botanical park allowed us to validate the expressiveness of the authoring tool. We were able to implement scenarios that were proposed during the collection

of users needs phase. We validated that these scenarios were playable and first results suggested a strong interest and engagement of volunteers that played the three proposed MLGs.

The next step is now the opening of the authoring tool to the staff of the botanical part. This step will be eased by having complete MLG scenarios corresponding to their need available as working examples.

References

1. Lee, H., Parsons, D., Kwon, G., Kim, J., Petrova, K., Jeong, E., Ryu, H.: Cooperation begins: Encouraging critical thinking skills through cooperative reciprocity using a mobile learning game. Computers & Education **97** (2016) 97–115
2. Sharples, M., Arnedillo-Sánchez, I., Milrad, M., Vavoula, G. In: Mobile Learning. Springer Netherlands, Dordrecht (2009) 233–249
3. Rosyid, H.A., Palmerlee, M., Chen, K.: Deploying learning materials to game content for serious education game development: A case study. CoRR **abs/1608.01611** (2016)
4. Wandersee, J.H., Schussler, E.E.: Preventing plant blindness. The American Biology Teacher **61**(2) (1999) 82–86
5. Bertrand, S., Cerutti, G., Tougne, L.: Bark Recognition to Improve Leaf-based Classification in Didactic Tree Species Identification. In: VISAPP 2017 - 12th International Conference on Computer Vision Theory and Applications. (February 2017)
6. Marfisi-Schottman, I., Gicquel, P.Y., George, S.: Meta Serious Game: Supporting Creativity Sessions for Mobile Serious Games. Proceedings of the European Conference on Game Based Learning ECGBL, Paisley, United Kingdom (October 2016) 407–415
7. Marfisi-Schottman, I., Gicquel, P.Y., Karoui, A., George, S.: From Idea to Reality: Extensive and Executable Modeling Language for Mobile Learning Games. Proceedings of the European Conference on Technology Enhanced Learning,, Lyon, France (September 2016) 428 – 433
8. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework. 2 edn. Eclipse Series. Addison-Wesley, Upper Saddle River, NJ (2009)
9. Torrente, J., Moreno-Ger, P., Fernández-Manjón, B., Sierra, J.L.: Instructor-oriented authoring tools for educational videogames. In: ICALT 2008, IEEE (2008) 516–518
10. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
11. Karoui, A., Marfisi-Schottman, I., George, S.: Mobile learning game authoring tools: Assessment, synthesis and proposals. In: Games and Learning Alliance. Springer (2016) 281–291
12. Holden, C.: Aris: augmented reality for interactive storytelling. In: Mobile Media Learning, ETC Press (2015) 68–83
13. Perry, B.: Gamifying french language learning: a case study examining a quest-based, augmented reality mobile learning-tool. Procedia-Social and Behavioral Sciences **174** (2015) 2308–2315
14. Zbick, J., Nake, I., Jansen, M., Milrad, M.: mlearn4web: a web-based framework to design and deploy cross-platform mobile applications. In: Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia, ACM (2014) 252–255