Gamification of Software Modelling Learning

Alfa Yohannis

Department of Computer Science, University of York, York, United Kingdom
ary506@york.ac.uk

Abstract. The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be written using the *abstract* environment.

Keywords: software modelling, gamification, learning

1 Introduction

Software modelling is commonly perceived as an difficult subject since it requires a great deal of abstractions [1]. However, this subject has a fundamental and crucial role in software engineering. Failure to master this subject will affect the students abstraction capability in analysing and designing a real-world software. Less proficiency in software modelling will likely cause software engineering students facing difficulties completing their degrees, as most of the software engineering related subjects have a sense of intrinsic abstraction problems [2]. Their perception of software modelling will affect their attitude towards software engineering today and their career paths in the future.

For new software engineering students, software modelling is commonly perceived as subjects that are difficult to learn. Software, by its nature, is abstract and complex that demand loads of abstraction to model it. This problem is similar to problems in mathematics where much of the concepts can only be accessed through symbolical representations [3]. Abstraction also means requiring the students to perform information hiding, generalization, approximation or reformulation, leaving out the irrelevant aspects but keeping the relevant ones, or separation from the concrete reality [4]. In order to overcome these challenges, we need to put more efforts on delivering software modelling in a more concrete and motivating presentation which can engage and ease students to learn it.

On the other hand, the use of games or game elements for serious purposes other than leisure has drawn lots of attentions. Gamification [5] and Serious Games [6] have been viewed as solutions to solve motivational problems that emerge when users engage in boring, irrelevant, difficult activities, e.g. learning sorting algorithms [7] or C-programming [8].

Therefore, the purpose of this research is to investigate and develop a gamification design framework that will systematically direct game design to produce a more accurate software modelling gamification yielding better-designed software modelling games. More precisely, this research aims to answer the following research questions that are derived from the general research objective:

1. Which processes, aspects, and components of software modelling and their teaching and learning practices that are essential to take into account?
2. What types of game elements and their roles that can deliver software modelling at best?
3. What kind of framework that orchestrates, design the interaction between, software modelling and game elements, to produce a better software modelling gamification?
4. To what extend gamification of software modelling better engage, motivate, and improve learners performances?

2 Related Works

Several approaches have been conducted to bring software modelling into a more concrete presentation that can be easily understood by learners, ranging from didactic learning (Moisan & Rigault, 2010), modeling tools utilization (Akayama, et al., 2013), alternative communication channels and the use of modelling language (Brandsteidl, Wieland, & Huemer, 2011), immersive visual modelling through virtual environment (Neubauer & Harris, 2003), software design studio (Whittle, Bull, Lee, & Kotonya, 2014), project-based approach (Szmuro & miaek, 2007), to code generation investigation (Schmidt, Kimmig, Bittner, & Dickerhof, 2014).

However, most of the approaches have weaknesses in motivating learners to engage continuously, frequently, and actively to learn software modelling, which is the important aspect to impact greatly on learning (Naps, 2005). This condition then elevates game as one approach, a.k.a. game-based learning, to learn or teach software modelling. This approach provides students a new way of learning software modelling, which is not only interactive but also engaging enough to keep them learning continuously.

The use of game elements for a purpose other than leisure is called gamification (Deterding, Dixon, Khaled, & Nacke, 2011). Regardless gamification design is still an ongoing challenge (Deterding, Bjrk, Nacke, Dixon, & Lawley, 2013), it is an opportunity for research that up today there is still no gamification design framework that particularly address how to guide the design of software modelling gamification; a framework that guides how to integrate game specific domain into software modelling. Therefore, this research aims to develop a gamification design framework of software modelling.

Research publications on the application of games or game elements for software modelling are difficult to find. One that exists is the work of Groenewegen et al. that proposed a game dedicated to the learning and teaching of the validity of a model of software architecture (Groenewegen, Hoppenbrouwers, & Proper, 2010). However, their work is limited to the validity of software architecture and did not provide a deep comprehension on how game elements or mechanics should be integrated into software modelling in general. The gamification of software modelling was not discussed in depth.

This is in line with the question prompted by (Prensky, 2005) that the how of learning games design is essential to be addressed. Likewise, Perotta et al. proposed a research challenge that a more analytic approach need to be developed on how game elements contribute to learning (Perotta, Featherstone, Aston, & Houghton, 2012), including a better understanding on how games match to desired outcomes and how games integrate into users learning experience (Conolly, Boyle, MacArthur, Hainey, & Boyle, 2012). This condition also aligns to what Deterding et al. also prompted that much work needs to be done on gamification design, how to integrate existing design patterns and dynamics of games into non-game contexts (Deterding, Bjrk, Nacke, Dixon, & Lawley, 2013).

These challenges can be interpreted as a focus on design problems that seek how to organize game elements and mechanics to meet certain functionalities or desired results. This focus will open the underlying processes of how games contribute to learning unveiling particular principles on game elements and mechanics. The understanding of the processes and the role of game mechanics will lead to the construction of gamification design frameworks.

Gamification has been implemented in different fields but the results are predicted will not always as expected and some will end in failures (Gartner, 2012). Therefore, the presence of a gamification design framework for software modelling is imperative since it will direct gameful design to produce a more accurate software modelling gamification yielding better-designed gameful experiences of software modelling.

Existing gamification frameworks are not adequate as they come from marketing or management backgrounds, not from software engineering discipline, and they provide guidance more to wise-step implementation approach rather than structural and dynamic points of view (Werbach & Hunter, 2012) (Huang & Soman, 2013) (Kumar, 2013). The relations between software modelling components and relevant game mechanics and elements are important to be studied, as it will provide a basis for constructing a gamification design framework for software modelling.

3 Research Methods

Since the output of this work is design artefacts, we decided to utilise the Design Science Research Methodology (DSRM) [9] as our umbrella methodology. DSRM is selected because it provides a comprehensive high-level conceptual framework how to undergo a full-cycle research process. It also provides six-activity guidelines for understanding, developing, executing, and evaluating design artifacts. The six activities are (1) problem identification and motivation activity, (2) define objectives for a solution activity, (3) define objectives for a solution activity, (4) design and development activity, (5) demonstration and evaluation activities, and (6) communication.

The high-level characteristic means that we can employ other research methods as the sub methods in each activity. For examples, we employ interviews, literature reviews, and discussion with experts as our methods in problem iden-

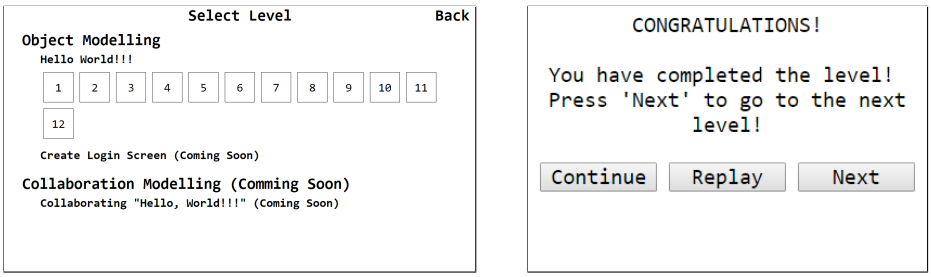
tification and motivation activity as well as utilise Deterding's lens of intrinsic skill atoms [10] to produce a gameful design in the design and development activity. As information, our work is still in the design and development activity and there are three activities remaining to work on. We use the first three activities of DSRM as our guidelines to report our in-progress accomplishment.

4 Problem Identification and Motivation Activity

5 Define Objectives and Evaluation Activity

6 Design and Development Activity

6.1 Game Design



(a) Level Selection (b) Positive Reinforcement

Fig. 1: Game and learning elements embedded into the artifact.

6.2 Architecture

6.3 Modelling Editor

7 Discussion

In order to develop a gamification design framework of software modelling, the software modelling itself has to be understood at first. By doing so, we can capture the underlying principles that are essential in delivering software modelling and therefore we can develop the framework more accurate. There are many approaches have been conducted so far to learn and to teach software modelling. From those approaches, we can identify important processes, aspects, and components, which without them will bring impairment to the software modelling, and use them as the consideration in developing the framework.

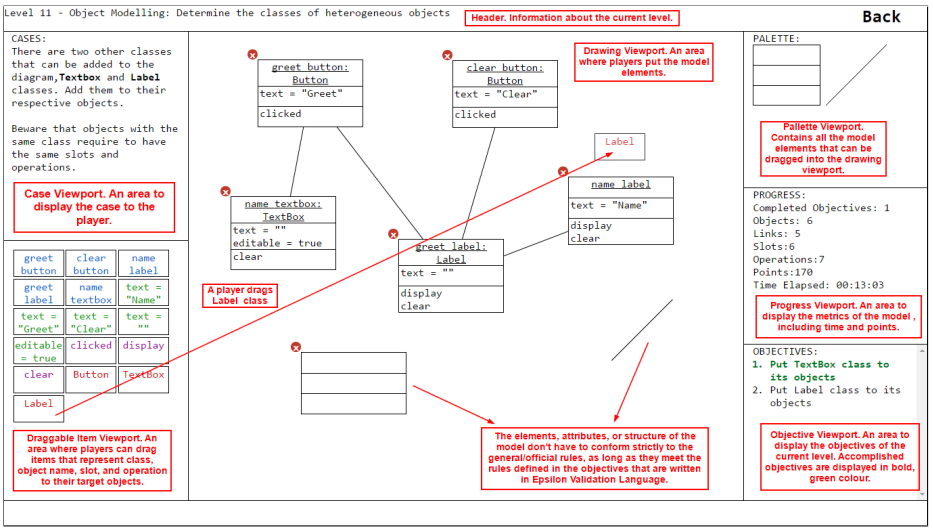


Fig. 2: The game's display.

Successfully identifying the underlying principles, processes, aspects, and components of software modelling is worthless without mapping them to representative game elements. Consequently, identifying representative types of game elements is a crucial effort to represent the software modelling within the gamification. As a result, more options are available in balancing the presence of the seriousness of software modelling and the fun delivered by the game elements in the gamification design.

Mapping software modelling and game elements is not a trivial task. The interaction between them needs to be identified. It is a kind of relationship that both software modelling and game elements can present at the same time while the user is interacting with a representative design artefact. Understanding the relationship will enable us to develop a framework that can orchestrate, design the interaction between, software modelling and game elements to produce a better software modelling gamification.

A working artifact, as the embodiment of the framework, has to be constructed, applied, and then evaluated in several case studies. It is the way we validate the framework since the artifact is the indirect representation of the framework. Measuring the artifacts outcomes will also measure the impact of the framework. The results of the validation will be the inputs to improve the artifact and mainly to revise and refine the gamification design framework of software modelling.

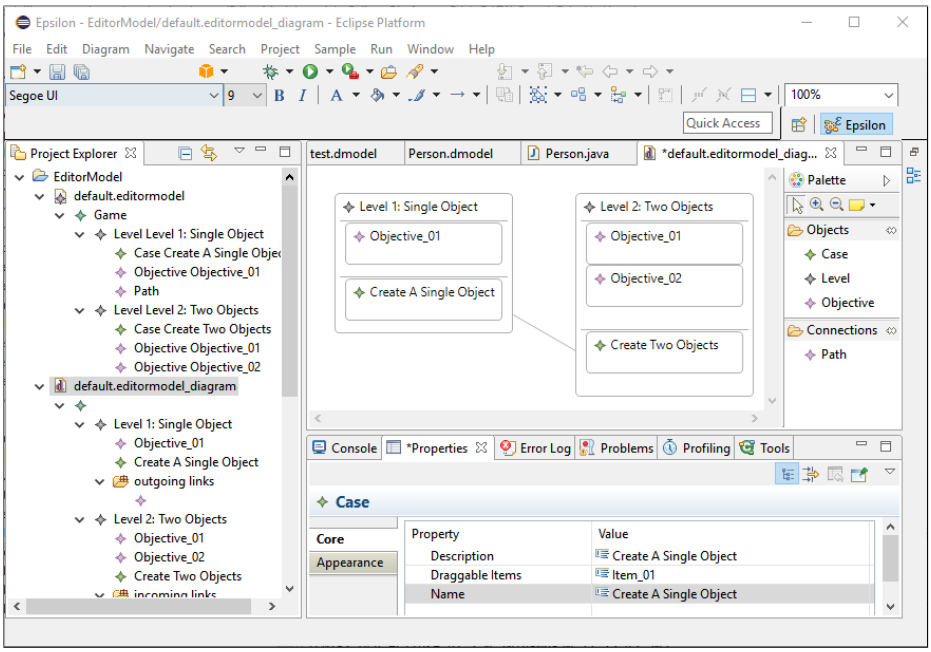


Fig. 3: Game editor to automatically generate the game.

8 Conclusion

Acknowledgments. We would like to thank our respondents that participated in our preliminary interview. This research is supported by *Lembaga Pengelola Dana Pendidikan Indonesia* (Indonesia Endowment Fund for Education).

Test Citing [5]

References

1. J. Börstler, L. Kuzniarz, C. Alphonce, W. B. Sanders, and M. Smialek, "Teaching software modeling in computing curricula," in *Proceedings of the final reports on Innovation and technology in computer science education 2012 working groups - ITiCSE-WGR '12*, (New York, New York, USA), p. 39, ACM Press, jul 2012.
2. J. Kramer, "Is abstraction the key to computing?," *Communications of the ACM*, vol. 50, no. 4, pp. 36–42, 2007.
3. R. Duval, "A cognitive analysis of problems of comprehension in a learning of mathematics," 2006.
4. L. Saitta and J.-D. Zucker, "02 - Abstraction in Different Disciplines," in *Abstraction in Artificial Intelligence and Complex Systems*, ch. 2, pp. 11–47, Springer New York, 2013.
5. S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proceedings of the 15th international*

academic MindTrek conference: Envisioning future media environments, pp. 9–15, ACM, 2011.
6. D. R. Michael and S. L. Chen, "Serious Games: Games That Educate, Train, and Inform," *Education*, vol. October 31, pp. 1–95, 2005.
7. A. Yohannis and Y. Prabowo, "Sort Attack: Visualization and Gamification of Sorting Algorithm Learning," in *VS-Games 2015 - 7th International Conference on Games and Virtual Worlds for Serious Applications*, 2015.
8. M.-B. Ibanez, A. Di-Serio, and C. Delgado-Kloos, "Gamification for Engaging Computer Science Students in Learning Activities: A Case Study," *IEEE Transactions on Learning Technologies*, vol. 7, no. 3, pp. 291–301, 2014.
9. K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.
10. S. Deterding, "The lens of intrinsic skill atoms: A method for gameful design," *Human–Computer Interaction*, vol. 30, no. 3-4, pp. 294–335, 2015.

Appendix A: Interview Questions