SeGa4Biz: Model-Driven Framework for Developing Serious Games for Business Processes

Faezeh Khorram, Masoumeh Taromirad and Raman Ramsin

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Keywords: Model-Driven Development, Model Transformation, Serious Game, Business Process.

Abstract: Organizations look for effective ways to teach their business processes to their employees. The application of serious games for teaching business processes is getting attraction recently. However, existing works are by large business-specific and few of them aim at teaching business processes in general, besides that the development of such games inherently suffers lack of precise and clear development approaches. This paper presents SeGa4Biz, a model-driven framework for serious game development for teaching business processes. Modeling supports different levels of abstraction and hence, increases user involvement throughout the development. SeGa4Biz particularly provides metamodels for creating Educational Serious Games (ESG) and Game-Aware Process (GAP) models, and automates considerable parts of the modeling and development activities, via model transformation. The effectiveness and applicability of SeGa4Biz is examined through a serious game development project in a software development company.

1 INTRODUCTION

A serious game is an interactive computer application which has a challenging goal, is amusing to play, and conveys practical skill, knowledge, or attitude to the player (Cowan and Kapralos, 2017). Potential applicability of serious games for teaching business processes is getting attraction recently, since they can help understand and analyze business processes (Santorum, 2011). However, the relationship between the elements of these two contexts is not clear-cut.

Non-technical domain experts have a crucial role in the design of a serious game since they have knowledge of the target serious domain. Therefore, effective communication between them and the technical developers is required to ensure that all the serious objectives are realized by the game. This increases the complexity of the serious game development in comparison to general game development. Existing game development approaches are deficient in this regard as they typically do not balance game design with educational design (Matallanaoui et al., 2015).

Model-driven development (MDD) is recently used for serious game development as it, specifically, provides complexity management and effective user involvement by providing different levels of abstraction and good-level of automation (e.g., (Thillainathan and Leimeister, 2016)). Existing MDD approaches, particularly in the context of business processes, do not cover the essential aspects of modeling, such as precise definition of modeling levels and transformation rules (e.g., (Bancora et al., 2015; Tang and Hamegehn, 2010)). There are also few MDD approaches that are domain-specific (e.g., Van Broeckhoven and De Troyer (2013) in cyber bullying) and, hence, are not applicable to the business process domain.

This paper presents a model-driven approach, called SeGa4Biz, for serious game development for teaching business processes. Central to the proposal is the definition of the modeling levels, covering different features of the target domains; i.e., game development and business processes. SeGa4Biz also provides a set of transformation rules that supports automation and so, reduces human intervention. Moreover, it introduces a novel structured design method via specifying how the elements of the game and the business process domain would relate to each other.

SeGa4Biz is evaluated through a case study: its prototype implementation is used for developing a serious game in a company for teaching a business process to new recruits. The experiment and its outcome were assessed against a set of evaluation criteria, defined from the perspective of serious game development in the context of teaching business processes. The results show that, among other things,
SeGa4Biz is applicable in a practical context and addresses many of the current shortcomings hindering the use of serious games in this context.

The rest of this paper is structured as follows: Section 2 provides an overview of SeGa4Biz; modeling levels and transformation rules are explained in Section 3; Section 4 presents the evaluation results; an outline of the related research is presented in Section 5; and the paper concludes with a discussion of the limitations and future work in Section 6.

2 OVERVIEW OF SeGa4Biz

In general, a game development process involves three main phases: 1) Pre-production, focusing on the preliminary design of the game and high-level decisions, commonly documented in the Game Concept Document (GCD); 2) Production, covering the detailed design, development, and testing of the game, commonly resulting in the Game Design Document (GDD); and 3) Post-production, concerning the deployment and acceptance testing of the game. SeGa4Biz focuses on the detailed design of the game, so it covers the first and parts of the second phases.

At the heart of SeGa4Biz is a modeling framework that supports modeling the target business process and the desired serious game at four levels of abstraction. It also provides a set of model transformations that supports semi-automatic construction of the target models, describing the detailed design of the game. As depicted in Figure 1, it specifies the intra-and inter-level relationships between the models. In the next section, all the modeling levels are described thoroughly.

Tool Support. A prototype implementation of the proposed framework is implemented using ATL Development Tools (ADT), which are built on top of the Eclipse Modeling Framework (Jouault et al., 2006). All the code for SeGa4Biz, and also the case study’s artefacts, are available online (khorram, 2020).

3 MODELING LEVELS

This section introduces the four modeling levels of the proposed framework.

3.1 Level 1: Scope Modeling

During scope modeling, the goals and requirements of the game are identified and modeled. This level involves two main, yet totally different, domains: business processes and serious games. Scope model-

Figure 1: SeGa4Biz framework.
randomly. In the random case, a separate BPMN model must be created for each possible path to ensure that all the paths are covered in the game, and the learning is therefore complete. Whereas in other cases, the player or the system chooses the correct path by examining the conditions, and thus no additional process model is required. Accordingly, a single (partial) business process may be represented with a set of, so-called, prime models; each corresponding to only one simple path in that process.

**Complementary Information.** Certain information is required for designing the game, but is not covered by BPMN models. We have identified the following essential parameters, which should be specified for each (prime) process:

- **The process level** indicates the importance and complexity of the process in the organization, so determines the difficulty of the game level corresponding to the process.

- **The knowledge resources** specify the available or required resources for completing the process. The information is either asked from the players during the game to challenge their knowledge or given to them to perform their tasks properly.

- **The required knowledge and skills** explains the capabilities of each specific role to perform their tasks. We map roles to game characters, and tasks to game activities. Thus, this data is essential for the specification of the capabilities of game characters, the conditions of progress through game levels, and the rewarding mechanism.

- **The task weight** indicates the importance of a task in the process, and is required for calculating the task completion score at a game level.

- **The level of a player character** with respect to the others is essential in defining the rules for proceeding in the game and designing the rewards for the game levels. As mentioned above, each game player character corresponds to an organizational role, and thus, its level is defined based on its position in the organizations roles hierarchy.

### 3.1.2 Serious Game Modeling

Game development activities start by preparing game-specific documents. We work with textual documents in early development stages, and in particular, prescribe the use of a Game Concept Document (GCD) as well as a Game Design Document (GDD). The GCD specifies the high-level requirements and context of the target game, such as its premise, player motivation (win and lose conditions), target market, genre, target platform, license, risk analysis and goals (Minaei, 2017).

The GDD contains the information required for finalizing the pre-production phase and being ready for the game production. This study jointly uses the templates presented in (Miles, 2016) and (Minaei, 2017), and collects the following information: game story, game characters (player and non-player), game audio, game world, mechanics, UI, technologies, and production plan. We suggest two formats for representing the information in the GDD: 1) Descriptive, providing the information in natural language; and 2) Logical, associating each GDD item with one or more models (i.e., a model represents or complements the information).

**Game-aware Process Models (GAP).** are introduced to systematically specify the relationships between the elements of the business process (BPMN) and the serious game, such that a serious game can be designed for a given business process in a (semi) automatic and structured fashion. These models are later used as the basis for model transformations.

To define the GAP metamodel, we first introduced a metamodel for Educational Serious Games (ESG) (briefly presented in Figure 2), and then we linked its elements to those of BPMN, driven from OMG metamodel (OMG, 2014). For instance, for each prime process, a game level is defined, in which the level number and the needed score are determined based on the process’s complementary information. Each lane of the process model can be an organizational role, a system, or an organizational unit, that are assigned to different game elements (e.g., an organizational role is a game character which could be player or non-player). Tasks and gateways are respectively mapped to states and knowledge challenges; they are assigned to the player character defined for the related role. An excerpt of GAP metamodel is depicted in Figure 3.

We suggest a two-step process for creating a GAP model from a BPMN model: 1) Specifying the basic game elements that are directly linked to the BPMN elements in the GAP metamodel (e.g., game challenges for the process gateways) 2) Defining additional game elements linked to those of initially specified (e.g., schorning rules for a game challenge defined in the first step). While the two-step process facilitates modeling, it particularly provides iterative and incremental game design.

### 3.2 Level 2: Structural and Behavioral Modeling

The second modeling level of SeGa4Biz provides the structural and behavioral aspects of the whole
game, through UML class and sequence diagrams, respectively. Both diagrams are generated semi-automatically by applying vertical model transformation to GAP models. The transformation rules implemented for generating the sequence diagram, respect those for generating class diagram, since system behavior indicates the run-time interaction between the system’s structural elements. Transformation rules are grouped with respect to the granularity and importance of the target element and hence, are not applied all at once: a transformation is carried out in a series of consecutive steps: each step is applied on the output of the previous step and augments the target diagram. For example for the class diagram, assuming that the target product follows the three-tier architecture, the first group of transformation rules produces Game Logic, Game Object, and Database components, followed by rules that generate the related classes under these components. Then, the game elements of the GAP model (i.e., the input model), are transformed into classes within the Game Logic and Game Object components, while a new component is generated that contains the classes corresponding to the BPMN elements. The connections between the generated classes are determined according to the relations between their matching elements in the input GAP model. It has to be noted that, this work involves the use of combined fragments, the most complex parts in sequence diagrams, which results in complex and challenging transformation rules.

3.3 Level 3: State-based Modeling

The behavior of individual objects is modeled in the third modeling level of SeGa4Biz using UML state machines. State machines are generated by applying a vertical transformation on the previously generated sequence diagrams. Several research efforts have focused on this area. The earliest work introduces an algorithm using OCL to resolve the conflicts and the similarities between sequence diagrams (Whittle and Schumann, 2000). Using this algorithm, an approach is presented in (Graaf, 2007) to establish compatible behavioral models, and thereby generate a set of state machines from a set of sequence diagrams using ATL. However, the main shortcoming is the lack of support for combined fragments and the extra complexity caused by OCL constraints. In (Grønmo and Møller-Pedersen, 2010), an algebraic graph transformation method is used to transform sequence diagrams into state machines; this method supports combined fragments, but its implementation is incomplete and only partial automation is provided. Aiming to address these shortcomings, our proposed model transformation particularly supports combined fragments without using OCL. For each lifeline of the sequence diagram, a distinct state machine is generated. Transformation rules are prioritized and then applied, from the highest to the lowest, on the elements of each lifeline. Doing so, the sequence of the states is derived from the source sequence diagram.

3.4 Level 4: Game Engine-based Modeling

At the last level of modeling, the platform-specific models are generated. In the context of this study, the game engine is the most important (if not the only) part of the platform. Due to Unity’s popularity, we chose it along with its Playmaker (Miles, 2016) plugin, which provides a powerful editor, debugging tool, and run-time library. Playmaker automatically generates the game logic code from a set of custom Finite State Machines (FSMs).
Hence, the last modeling activity aims at generating Playmaker FSMs. Playmaker FSMs and UML state machines are similar in principle, the distinction is in a set of executable actions and events, provided by Playmaker, to facilitate the implementation of the game object’s behaviors.

The models that are automatically generated in this level are closer to the required format, yet are not immediately supported by Playmaker. Mapping the actions and events of the UML state machines to the system actions and events in Playmaker requires considering their semantics. Therefore, the generated models are completed by the developers, w.r.t. the available models and documents, to be imported into Playmaker and used for the next development tasks.

4 EVALUATION

To assess the applicability and effectiveness of SeGa4Biz, we conducted an evaluation that addresses the following research questions: RQ1. How logical and accurate are the modeling levels and model transformations? RQ2. Does SeGa4Biz facilitate/ease the serious game design and development tasks? RQ3. Are the game-related concerns well covered?

4.1 Case Study

To answer the research questions, SeGa4Biz was applied to a case study, designed based on the guidelines provided in (Runeson et al., 2012), which are specific to software engineering experiments. The case was a real-world project in a software development company that specializes in using and tailoring MDD methodologies and developing software solutions for medium to large businesses. The production manager of their software product line was actively involved in the study, both as the business expert and the customer. SeGa4Biz was used to develop a serious game for teaching the Leave-of-Absence Request process to new recruits, so only those parts of the process that relate them (i.e., employees) were considered.

One of the authors was involved in the project and carried out the activities prescribed by SeGa4Biz, in close collaboration with the product manager. The development took about 30 working days, and all the models and documents were delivered to the product manager (as the business expert and customer), to acquire his feedback and confirmation. Data gathering, analysis, and evaluation were performed iteratively to achieve a flexible design reflecting the experience gained during the study.

4.2 Evaluation Criteria

In order to evaluate SeGa4Biz in a structured way, particularly with respect to the research questions, we introduced a set of evaluation criteria focusing on the MDD features (Asadi and Ramsin, 2008), and support for serious game development in the context of business processes (Roungas and Dalpiaz, 2015; Thillainathan, 2013); they are shown in Tables 1 and 2. Accordingly, the participants involved in the case study were interviewed using these criteria, and the outcome is shown in the last column of the Tables. Also, the strengths and limitations of the proposed framework were discussed with the participants for future improvement.

4.3 Analysis of Results

The results are presented in order of the research questions. Answering RQ1- The accuracy level of the framework from the MDD perspective: The values given by business expert to the evaluation criteria of Table 1 showed that the modeling levels are defined accurately and distinguishably, and all the required abstraction levels are provided. The main limitations are the medium automation support and low automatic code generation (i.e., particularly in generating Playmaker FSMs). SeGa4Biz has low support for features that are general to any MDD approach (e.g., round-trip engineering), herein, we have focused on domain-specific features and have planned to work on rest as future work.

Answering RQ2- SeGa4Biz ease of use: The case study demonstrated that SeGa4Biz improves the understandability and quality, from the design and modeling perspectives. These features along with the provided automation, significantly increases the ease of use compared to similar approaches, which ultimately, motivates companies to move towards using serious games for teaching their business processes. However, we were once again acknowledged that the medium level of automation, especially at the last modeling level, was the main concern.

Answering RQ3- The coverage level of gamelated concerns: It is evident from the values given to the criteria of Table 2, that the game-related concerns, especially those required for the context of business process education, are well-satisfied by SeGa4Biz. The main deficiency is the minimal attention given to artistic features, that is however out of our scope.
Table 1: Evaluation criteria for assessing support for Model-Driven Development.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Description of possible values</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transparency between modeling levels</td>
<td>Boundary between levels: A: is accurately detectable, B: has relative transparency, C: cannot be distinguished</td>
<td>A</td>
</tr>
<tr>
<td>Classification of the modeling level's data</td>
<td>A: Precise classification, B: Relative classification, C: Lack of classification</td>
<td>A</td>
</tr>
<tr>
<td>Support for abstraction levels (CIM, PIM, PSM)</td>
<td>A: Full support for abstraction levels and transitions, B: All abstraction levels are defined, but transition between them is not supported, C: Some abstraction levels are not supported.</td>
<td>A</td>
</tr>
<tr>
<td>Structural, Behavioral, Functional modeling</td>
<td>A: All the system's aspects are modeled, B: Some aspects of system are not modeled.</td>
<td>A</td>
</tr>
<tr>
<td>Model Transformation type</td>
<td>Vertical: Abstraction levels of the source and target models are different. Horizontal: Source and target models are at the same level of abstraction.</td>
<td>Vertical</td>
</tr>
<tr>
<td>Automation level of the transformations</td>
<td>Low: Manual, Medium: Semi-automated, High: Fully-automated</td>
<td>Medium</td>
</tr>
<tr>
<td>Automatic code generation</td>
<td>A: All parts of the code are generated automatically, B: Most parts of the code, C: Some parts of the code.</td>
<td>C</td>
</tr>
<tr>
<td>Tool support (for model validation, metadata management, automatic test, traceability between models)</td>
<td>A: A complete toolset is provided, or precise guidelines are defined to select alternative tools. B: A complete toolset is not provided, but general guidelines are defined to select alternative tools. C: A specific tool, or guidelines to select an appropriate one is not defined.</td>
<td>C</td>
</tr>
<tr>
<td>Round-trip engineering</td>
<td>A: Detailed procedures are specified for the task in the methodology. B: Only general guidelines are provided for the task. C: The task is not covered by the methodology.</td>
<td>B</td>
</tr>
</tbody>
</table>

5 RELATED WORK

Little research is done on serious games, whereas gamification is considered relatively more, especially in the context of business processes. Despite the differences between these two domains, we will provide a discussion herein on the related efforts in both areas.

5.1 MDD Approaches for Serious Game Development

In (Pflanzl and Vossen, 2018), a descriptive gamification modeling language called GaML is proposed, based on which a model-driven architecture for the design and development of serious games is introduced in (Löffler et al., 2018). In (Roungas and Dalpiaz, 2015), a web-based model-driven knowledge management environment is introduced based on a conceptual model. In (Thillainathan and Leimeister, 2016), an MDD framework consisting of a visual programming environment (VIPER), a domain-specific modeling language (GLiSMo), and an MDD toolchain is introduced to enable non-technical people to get involved in serious games development. In (Tang and Hanneweg, 2010) an MDD framework for development of learning serious games is provided, introducing three models: Game Technology Model, Game Content Mode, and Game Software Model. In (Van Hoeye et al., 2015), an MDD framework is defined to build serious game production environments for non-technical users. In (Van Broeckhoven and De Troyer, 2013), a graphical modeling language, ATTAC-L, and an MDD framework are proposed for building cyber-bullying games. In (Calderon et al., 2018) an approach and a graphical tool (MEdit4CEP-Gam) is introduced that can be used by non-technical users to design and model gamification strategies that can be automatically transformed into code.

Most of the above approaches are domain-specific and are not appropriate for our intended domain. Some are dependent on technical knowledge, whereas in SeGa4Biz, different techniques are used to address this shortcoming.

5.2 Approaches for Serious Game Development and Gamification for Business Processes

In (Herzig et al., 2013), a serious game-based method for business process management and a role-playing game simulation tool are proposed to display, improve, and evolve existing business processes. In (Matallaoui et al., 2015) a business process gamification model (GameLog) for connecting game elements to a business process is introduced. To improve process sustainability, a gamification tool is presented in (Mancebo et al., 2017) that analyzes the events of business process management systems and encourages users to work more sustainably by using gamification mechanisms. Kaleidoscope of Effective Gamification is a gamification design model and an analysis tool (Kappen and Nacke, 2013), that present guidelines for designing gamified commercial applications based on the layers of the design model. A novel conceptual framework for gamification design in collaborative and online work environments is defined in (Rosmansyah et al., 2016). PierSim is a 3D business process simulation environment for teaching the basics of BPM to students in a gamified manner (Craven, 2015). The game ideas that are applicable to business process modeling are discussed in (Santorum, 2011) addressing issues like low quality of models and low motivation of modelers. In (Klevers et al., 2016) a framework for analysis and classification of requirements for the devel-
development of a simulation game is proposed, aiming at training business process changes in digital transformations. A combination of MDD, BPM techniques, and gamification mechanisms is introduced in (Bancora et al., 2015) focusing on the impacts on individual and social work management. In (Zribi et al., 2016), a gamification model for interactive learning is described and applied on the PAd business process simulator, an online model-based learning environment.

Most of the above research efforts focus on gamification of business processes and hence, are not directly useful for serious game development. Whereas our approach focuses on serious game development, and GAP models providing precise mapping between game elements and business processes.

6 CONCLUSION

In this paper, we introduced SeGa4Biz, a MDD framework for developing serious games for teaching business processes. It includes models at four levels of abstraction. GAP metamodel is proposed in the first level to connect two different domains, business process and serious game. For providing smooth transition between modeling levels as well as supporting automation, we implemented transformation rules to generate models demonstrating the structural and behavioral aspects of the final product. Experimental results showed that SeGa4Biz is effective and easy-to-use by non-technical customers. As future work, we plan to improve the medium degree of automation at the last modeling level by supporting Playmaker FSM semantics. We also aim to increase SeGa4Biz’s scalability by adding more complex game elements, such as AI and networking features.

ACKNOWLEDGEMENTS

This project was partially funded by the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No 813884.

REFERENCES


Rosmansyah, Y. et al. (2016). Gamification framework for designing online training and collaborative working system in statistics indoensia. In 2016 International Conference on Information Technology Systems and Innovation (ICITSI), pages 1–6. IEEE.


