1st Workshop on Game Development and Model-Driven Software Development

Robert Walter¹, Maic Masuch¹, and Ed Merks²

¹ Entertainment Computing Group
Department of Computer Science and Cognitive Science
University of Duisburg-Essen
47057 Duisburg, Germany

{robert.walter, maic.masuch}@uni-due.de

² 402 – 1168 Richards Street
Vancouver, BC, Canada
ed@macromodeling.com

Abstract. This workshop brings together game development professionals and experts of model-driven software development (MDSD). MDSD can improve the way software is developed by providing a higher level of abstraction and model-to-model transformation respectively model-to-text generation by automated tools. With this workshop, we want to identify how game development can benefit from MDSD. Thereby, the workshop focuses on a strong collaboration of both expert groups to determine the chances, challenges, and boundaries of introducing MDSD to the game domain.

Keywords: Game Development, Model-Driven Software Development, Model-Driven Game Development.

1 Relevance of Workshop Topic

Games are highly interactive media applications within a hard to define common scope. Developed in multidisciplinary teams, they combine the artistic challenges of multimedia productions with the engineering challenges of IT-productions [4]. Hence, we face ambitious demands regarding the overall development process. However, the growing complexity and scale were not encountered by refined game development methods for many years [5]. Only recently, agile project management methods like Scrum were applied successfully in several productions [7, 8, 9].

In the last decade, MDSD has successfully made the transition from the academic sphere to industrial-grade software development and it provides many advantages [6] game development could benefit from. Key features of a system are formally described on a higher level of abstraction (the problem domain), omitting distracting details like the technical realization on a distinct platform (the solution domain). This allows for a better integration of domain experts (e.g. game designers, game writers, concept artists) in the development process. Using model-to-model transformation or model-to-text generation, the transition from problem space to solution space can be automated, which is more efficient and less error-prone than the manual implementation in a third generation

language. Moreover, often grounding on a common game engine, games feature characteristics of software product lines, while product-line engineering is an explicit application field of MDSD [6]. That is why we think that game development and MDSD should get in touch.

2 Workshop Objective

The workshop's main objective is to bring together researchers and industry professionals of both fields to identify if and how the game development process could benefit from MDSD. Thereby, we want to initialize a dialogue that holds on and evolves far beyond the boundaries of the workshop, creating an information transfer both communities benefit from.

2.1 Proposed Form and Schedule

In order to accomplish the described objective, we plan to provide a full day workshop that is divided into four sections, with a distinct sub-goal each.

Why Should We Care? During the first section, two keynotes introduce MDSD respectively the game development process. With the keynote speakers moderating the discussion of this section, the goal is to create a common understanding of the advantages and boundaries of MDSD as well as the existing and forthcoming challenges in game development that can be addressed by MDSD.

Position Papers: In the second section, selected researchers and industry professionals present their experiences and opinions on model-driven game development, covering theories, techniques, tools, infrastructure, and boundaries of MDSD. Based on these presentations as well as the results of the first section, the participants should identify one or more exemplary sub-domains and describe requirements for corresponding DSLs in different working groups.

Finding the Right Abstraction: The central aspect of defining a DSL is to find the right abstraction that completely describes the domain from the target group's perspective. Hence, in the third section of the workshop, we introduce a framework that describes how a DSL can be defined cooperatively by domain experts (game development professionals) and language engineers (MDSD experts). Afterwards, the participants are asked to apply this framework, designing DSLs for the former identified sub-domains.

Open Discussion: In the final section, the different DSL designs of the working groups are presented and optimized. We believe that the DSL results can be used as a foundation of upcoming project and research cooperation between the participants. We want to close the workshop with an organized summary of lessons learned.

2.2 Expected Workshop Outcome

First, as our list of potential participants with confirmed interest also points out, there is a strong request in the game development community to get in touch with MDSD. Therefore, the website of the workshop will feature a list with contact information of all participants. The position papers will be published in the proceedings of the workshop. The achieved results of the workshop are going to be published in form of a technical report on the website. This report will also feature all open issues for future work. Participants with similar research and development interests can connect and collaborate on solving these issues.

3 Bio Information of Workshop Leaders

Robert Walter received a diploma in Software Engineering and a Master of Science in Computer Science. In his Ph.D. studies, he examines the game development process with a focus on the creation and processing of narrative game content, develops concepts for domain-specific languages (DSLs)¹, and tools for an improved integration of all stakeholders—from game writers over artists and voice actors to localization studios—into the game development process [3]. The goal is to find patterns that reduce development time, effort, complexity, and costs.

Maic Masuch holds the chair of Media Technology and Entertainment Computing at the Faculty of Engineering, Department of Computer Science and Cognitive Science at the University of Duisburg-Essen. He received his Ph.D. with a dissertation on computer animation at the University of Magdeburg and, in 2002, became Germany's first Professor for Computer Games. He founded two companies, researches in the field of game design and game development for over twelve years, and is one of the pioneers of German video game research.

Ed Merks leads the Eclipse Modeling Framework project as well as the top-level Eclipse Modeling project. He is a coauthor of the authoritative book “EMF: Eclipse Modeling Framework” [1] which is published as a second expanded edition. He is an elected member of the Eclipse Foundation Board of Directors and has been recognized by the Eclipse Community Awards as Top Ambassador, Top Committer, and Top Newcomer Evangelist. Ed is currently interested in all aspects of Eclipse modeling and its application and is well recognized for his dedication to the Eclipse community, posting literally thousands of newsgroup answers each year. He spent 16 years at IBM, achieving the level of Senior Technical Staff Member after completing his Ph.D. at Simon Fraser University. He is a partner of itemis AG. His experience in modeling technology spans 25 years.

4 Potential Workshop Participants

We received confirmed expression of interest from the following experts: **Andrew Brownsword** (Former Chief Architect at EA BlackBox), **Paul Martin** (Director of

¹ “[A DSL is] a computer programming language of limited expressiveness focused on a particular domain.” [2].

Technology at Slant Six Games), **Markus Völter** (independent researcher, consultant and coach, itemis AG), **Karsten Thoms** (Software Architect and Coach, itemis AG), **José A. Carsí** and **Emanuel Montero** (Associate Professor and Ph.D. student, Software Engineering and Information Systems Research Group, Universidad Politécnica de Valencia), **Daniel Volk** (Ravensburger Digital GmbH), **Sonja Maier** (Bundeswehr University Munich).

Amongst others, we are also in contact with Richard Dansky (Central Clancy Writer, Ubisoft), Mike Acton (Engine Director, Insomniac Games), and Jeff Ward (Toolsmith/Programmer Fire Hose Games). In the moment, we suggest a total number of 20 people joining the workshop.

References

1. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework, 2nd edn. Addison-Wesley Longman, Amsterdam (2008)
2. Fowler, M., Pearson, R.: Domain-Specific Languages. Addison-Wesley Professional, Reading (2010)
3. Walter, R., Masuch, M.: Finding the Right Abstraction in Game Development – An Engineering Workflow for Domain-Specific Languages. In: Submitted at the 6th International Conference on the Foundation of Digital Games, Bordeaux, France (2011) (review pending)
4. Hight, J., Novak, J.: Game Development Essentials: Game Project Management. Thomson Delmar Learning, Clifton Park (2008)
5. Petrillo, F., Pimenta, M., Trindade, F., Dietrich, C.: What Went Wrong? A Survey of Problems in Game Development. In: Computers in Entertainment (CIE) - SPECIAL ISSUE: Media Arts and Games, New York, NY, USA, vol. 7(1) (2009)
6. Stahl, T., Völter, M.: Model-Driven Software Development: Technology, Engineering, Management. Wiley, Southern Gate, Chichester, England (2006)
7. Esmurdoc, C.: Postmortem: Behind The Scenes Of Brutal Legend (2009), http://www.gamasutra.com/view/news/25799/Postmortem_Behind_The_Scenes_Of_Brutal_Legend.php (last access: April 21, 2011)
8. Graft, K.: Racing Evolution: Forza 3 and the Changing Driving Sim (2009), http://www.gamasutra.com/view/feature/4144/racing_evolution_forza_3_and_the_.php (last access: April 21, 2011)
9. Nutt, C.: Q&A: Valve's Swift On Left 4 Dead 2's Production, AI Boost, http://www.gamasutra.com/view/news/25701/QA_Valves_Swift_On_Left_4_Dead_2s_Production_AI_Boost.php (last access: April 21, 2011)