Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

COMPLEXITY REDUCTION USING A STRUCTURED ASPECT-ORIENTED SIMULATION FRAMEWORK

Sebastian Bohlmann
Helena Szczerbicka

FG Simulation and Modelling
Leibniz Universität Hannover
Welfengarten 1 30167 Hannover, DE

ABSTRACT

Model complexity constantly increases in scientific and engineering applications. To be able to implement models and simulators efficiently and accuracy modularization and reusablitiy are key features. In this paper a combination of different approaches is combined to decrease modelling complexity. Inspired by paradigms used in professional software engineering a methodology is developed transport the benefits into the area of modelling and simulation. Furthermore a framework using standard open source components is presented. This generalized multi level framework is designed to be used for simulator or model construction. It is guiding the user to separate cross-cutting concerns. Likewise the framework presents a embedded method how to use runtime validation techniques to validate complex simulation systems.

1 INTRODUCTION

Modularization is a key strategy to reduce the complexity of complex dynamic models so they can be handled efficiently. This strategy is well known in the area of software engineering. Fine grade modularization increases the ability to reuse existing modules in different contexts or scenarios. Considered abstract the nowadays widely used object oriented programming paradigm is a form of fine graded modularization. But in software development it turned out that for highly complex problems a hierarchical modularization strategy is required. Software engineers know two main strategies to decrease the complexity of a software module. The first is called aspect oriented (Ionescu, Piater, Scheuermann, and Laurien 2010)(Kiczales, Lamping, Mendhekar, Maeda, Lopes, marc Loingtier, and Irwin 1997) programming. This paradigm is separating cross cutting concerns to different software units. This units are then combined (usually during runtime) to form a more complex software module. This could be done by a framework being aspect oriented. One positive aspect is that side effects are getting less common. The second strategy to reduce complexity is to make the modularization a dynamic component model. A framework being able to do this consists of multiple independent parts usually being linked by some sort of meta description.
The authors will give a short overview about a new generalized simulation framework prototype developed at the University of Hannover called JGSE. Although it is no classical simulator it is capable to be used to serve as a base to implement a custom simulator in a very comfortable fashion. It translates mechanism known from state of the art software engineering frameworks into the world of modelling and simulation software combining and extending well known open source standard software modules. Furthermore a efficient model (re)validation strategy is throughout designed form scratch into the framework.

2 GENERAL APPROCH

To satisfy all previously mentioned requirements a couple of basic techniques is required: Dependency injection mechanism, runtime code compiler/enhancer, runtime reflection, aspect oriented programming capabilities, full featured modularisation platform, object oriented programming language, artifact repository support. To meet all requirements Java (Version 8) was chosen as the designated programming language. The main aspects in this context are the bytecode intermediate opcodes, the wide dissemination of the language, the JVM (Java virtual machine) capability to change code while execution, a strong software design pattern support and simply the availability of tools to satisfy all following aspects. Java online code instrumentation support creates the possibilities to implement all kinds of runtime modifications required for aspect oriented framework design. The OSGI-Framework was chosen as the de facto standard in java modularisation. Equinox P2 provides all previously mentioned repository and update related mechanisms. Extended by some Apache foundation libraries and some prepacked Eclipse OSGI Packages the basic groundwork is complete. But without modifications and extensions there is no further capability enhancement to be used in modelling and simulation. Therefor the JGSE(Java Generalized Simulation Engine) is integrated into the basic components. It consists of multiple components. The core module is a context sensitive runtime compiler being implemented. This module therefore is capable to determine all modules providing a code refactoring component and passing aspect code while being loaded in some context to the corresponding component.

3 EXPERIENCES WITH THE EXECUTION FRAMEWORK

The basic idea of this system was to increase design efficiency when implementing a new type of model or simulation. In parallel with the development of the JGSE-Framework some types of behavioural models have been implemented. Most of them using less than 100 LOCs. In the end turns out, that the a previous implemented simulator (same author) was outperformed by the JGSE-Framework based simulation. If many components are beeing observed the selective and dynamic code instrumentation of JGSE increases the performance advantage even further.

4 CONCLUSION

Using modern software engineering concepts in modelling and simulation can be done. Although the developer is required of deep knowledge about the different components to create a best effort solution. For second stage developers usually working in the area of modelling and simulation using a generalized simulation framework can have huge efficiency impact. Every day tasks can be outsourced to a modular system increasing reusability and decreasing the complexity of the custom model implementation. Code distribution and teamwork are promoted by using standard modularisation systems. Even if out of the scope of this poster different modelling formalisms can interconnected and cosimulated. The most prominent benefit in modularized aspect oriented simulation though can be a huge complexity reduction by the separation of cross-cutting concerns.

ACKNOWLEDGMENTS

We acknowledge support of this project by Deutsche Forschungsgemeinschaft (DFG) within the project ”BauSim - Online Simulation in der Baustellenfertigung”.

REFERENCES

Ionescu, T. B., A. Piater, W. Scheuermann, and E. Laurien. 2010. “An Aspect-Oriented Approach for the Development of Complex Simulation Software.”. *Journal of Object Technology* 9 (1): 161–181.

Kiczales, G., J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. marc Loingtier, and J. Irwin. 1997. “Aspect-oriented programming”. In *ECOOP*: SpringerVerlag.