Challenges in Modularization of Discrete Event Simulations

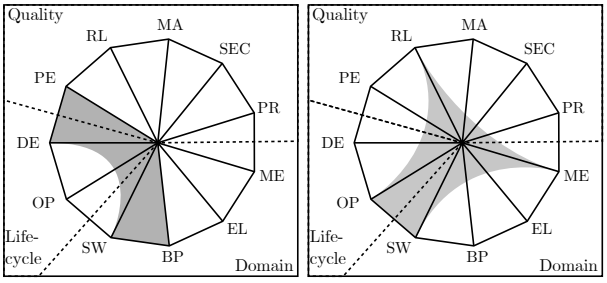
Sandro Koch
sandro.koch@kit.edu
Karlsruhe Institute of Technology

Abstract

Due to digitalization software systems in general and simulations, in particular, are becoming more and more complicated. This results in increased development cost and development time of simulations. Different approaches have demonstrated that to handle complexity, modularization of monolithic simulations enables the reuse of already developed modules and reduces the overall complexity. The High-Level Architecture, a widely used and researched approach, is the standard for distributed and modularized simulations. However, there are not many alternatives to the existing methods, and the modularization process of current simulations is not well documented and researched. This paper gives an overview of the existing approaches and shows possible research topics in the domain of modular discrete event simulations. Furthermore, we propose the idea of an Architecture Description Language and resulting from it, a template solution for analyzing existing discrete event simulations.

1 Introduction

Software is gaining increasing importance in all aspects of our daily lives. Therefore, the software quality has to be assured to secure the overall quality of information systems.As a system grows, especially a long-living system, it is difficult to implement and test all possible changes in advance (i.e., different change scenarios). The simulation of such systems enables the system architect to see the impact of possible changes or configurations without the concrete implementation. In the context of quality simulation, there are different kinds of quality aspects. Besides the performance simulation, reliability, security, privacy, and maintainability might be of some interest regarding quality assurance. During the lifetime of a project, not all possible simulation requirements can be foreseen and not all use cases can be considered during the design phase. Therefore a simulation should be extensible for further aspects and also combinable with other simulations to prevent duplications. The problem of simulating systems is not exclusive to the domain of software development. The type of used simulation depends on the desired information a simulation must deliver and in which life-cycle period, development- or runtime, the simulation runs.



(a) Software and business process simulation consisting of four modules at design time

(b) Reliability simulation of a mechanical system consisting of four modules at runtime

Figure 1: Performance (PE), Reliability (RL), Maintainability (MA), Security (SEC), Privacy (PR), Mechanics (ME), Electronics (EL), Business Processes (BP), Software (SW), Operations (OP), Development (DE)

Our focus lies in the quality simulation of software systems. Performance is the central aspect which we simulate in the context of software systems. Fig. 1 shows an excerpt of the combination of simulation modules which are divided into three dimensions: quality, domain, and life-cycle. Based on a distinct set of simulation modules, different types of simulations are created. Part (a) shows a maintainability simulation (MA) of a mechanical system (ME) during the operational life-cycle (OP). Part (b) shows a reliability (RL) and security (SE) simulation of an electrical- (EL) mechanical (ME) system with business processes (BP) during the development life-cycle (DE). Although an analytical approach solves security and reliability, the impact on other quality aspects of the system (e.g., an effect on the performance of an offline CPU core) can be simulated.

2 Challenges in modularizing discrete event simulations

We have identified four problems to the current state of discrete event simulation development. This section presents the occurring challenges while designing and

implementing modularized discrete event simulations.

2.1 Simulation Segregation (C1)

To allow composable simulations as shown in Fig. 1 a separation into modules is necessary. Segregation points in existing simulations must be identified, a process or guideline does not exist. If a new simulation is developed, reasonable modules concerning reusability and maintainability have to be defined. The High-Level Architecture (HLA) [3] standard allows to break down a simulation into different modules. Thus, simulations can be developed in a modular way. The intention of the HLA approach is, to have multiple simulations separated distinct modules. An architectural view of an HLA simulation shown in Fig.2. Besides the simulation modules, it is possible to add monitoring and live participants to the overall simulation. It was the purpose to allow distributed simulations,

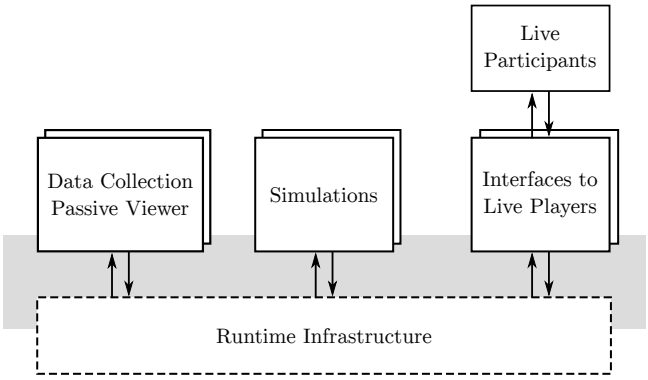


Figure 2: Functional view of a HLA simulation [4]

in order to combine the computing power of multiple systems. But HLA only enables the use simulation modules, the process of how to modularize or design a modular simulation is not defined. Also, best practices or patterns are not existent.

2.2 Simulation Coupling (C2)

A simulation which consists of multiple modules must be coupled to allow communication between this distinct simulation modules. Simulations must be coupled to enable communication and synchronization in between simulations. The HLA standard specifies how to handle the exchange of data between simulations and how to design individual communication packages. Simulations are managed by the Run-Time-Infrastructure (RTI) [3]. The RTI monitors the simulations and manages the data exchange between the simulation modules. Therefore, the HLA standard describes interfaces to connect a simulation with the RTI. Also, the data exchange between simulations and the RTI is encapsulated in a distinct model [11].

To add a simulation to the RTI The intra-communication of a simulation module or the communication between modules of one simulation is not given. This is necessary to allow the combination of the existing simulations available which results in reduced costs and maintenance effort. A general approach for simulation coupling to integrate simulations which are not explicitly developed with the HLA standard is missing.

2.3 Simulation Interoperability (C3)

The interoperability of simulation modules is another problem we have identified. It should be possible to use a simulation module in a different domain without making adjustments to the module itself. Apart from different domains, a simulation module should be able to communicate with another simulation module without modifying either of the modules. The standardization of a modular and distributed simulation is defined in the HLA standard. But an architectural formalization is missing, and structural analysis is not possible.

2.4 Simulation Behavior Preservation (C4)

A further problem arises when a simulation is designed in a modular way. It has to be ensured that the results of a modularly designed simulation are the same as the results of a monolithic simulation which runs the same kind of simulation. Besides the results of an individual simulation module compared to non-modularized simulation, the expectation value of a composed simulation with multiple modules has to be the same as if the simulation was monolithic. Current approaches have no way of determining or guarantee behavior preservation if a simulation is modularized.

In the "Related Work" section (4), simulation modeling and modularization approaches are presented. Also approaches for creating compose- and interoperable simulations and simulation modules are introduced. By looking at different domains, it might be possible to exchange individual building blocks within the simulation not only within a domain but also interdisciplinary.

3 Approaches in simulation modularization

Considering the evolution and maintenance of software-aided simulations a distinct language in combination with model-driven engineering approaches would be the next step in developing simulations. Different existing simulations should then be realized based on this language to extract a template solution for modular simulations.

3.1 Creating an ADL for Simulations

This approach addresses the challenges C1 and C2. Different approaches to simulation coupling and modularization must be analyzed to create an Architecture Description Language (ADL) for modular simulations. Therefore, it is a good idea to use an existing simulation of business processes and software systems as a

basis. Since this type of simulation already exists in Palladio [15] with IntBIIS [17], it is advisable that the assessments test this case precisely.

The first option is to implement HLA for business processes in the Palladio context. The conceptual part consists of the analysis of existing HLA implementations, concepts and interactions already in use, as well as the design for the structure of the future simulation. The conceptual part is followed by the implementation of the designed structure with a final evaluation. The evaluation focuses on preserving the behavior of the newly developed simulation in comparison to the already existing monolithic implementation.

In addition to the implementation of the HLA approach, at least one further approach should be implemented in the context of the business processes. The knowledge gained from this should enable a common structure to be extracted. If this is not possible, choose an approach by which the ADL is designed. The successful creation of an ADL for simulations is of particular importance for the extraction of a template solution.

3.2 Extracting a Template Solution

This approach also addresses the challenges C1 and C2. The generated ADL is only the basis for the application of different simulation templates. A template is to be used to determine how the structure of simulation coordination is realized. For example, a template defines whether the time is managed centrally in one component or each simulation component got its time management. Since there are different approaches how simulations can be coordinated and how the division of individual simulation roles is handled, multiple template solutions are possible. Therefore approaches are to be implemented with the help of the ADL so that they can be compared based on various quality criteria.

Because of the experience with the HLA approach in the ADL designing process, implementing HLA with the ADL is the next step towards a template solution. As described, at least one other approach has to be implemented with the ADL to be able to compare the approaches.

3.3 Designing an Analysis and Optimization Framework

This approach will help to segregate or design simulations (C1) or to specify how behavior preservation (C4) can be realized. Based on a common base (the proposed idea of an ADL) for modular simulations, existing simulations can be modularized. If the simulation is already modularized, it can be adapted to the developed ADL. A survey of simulation experts can help us to identify and avoid pitfalls in the simulation design Therefore, the ADL has to be introduced to simulation experts, to allow these experts to express their current simulation design process with the ADL. Thus it is possible to identify common practices among the present adapted simulations. Common practices can lead to a guideline of best practices and ideally to design patterns for modular simulations.

4 Related Work

Modularization of software is an already established topic within the software engineering community. The creation of components as in the Palladio Component Model [15] on an architectural level or the modularization of code generation transformations [14] on code level are some examples. But a modularization scheme for discrete event simulations is missing.

4.1 Simulation Modeling

To analyze simulations, an ontological analysis can lay the foundation of a robust knowledge-based system. The work of Perakath et al. [8] shows that an ontology facilitates the modeling of simulation. Cetinkaya et al. [10] utilizes model-driven development approaches to support the development of simulations. They propose a framework to aid with the modeling process for simulations. Law's reference book "Simulation Modeling and Analysis" [1] does not use an ontological or meta-modeling approach but gives an overview of general modeling approaches in the domain of simulations. All approaches have in common that the composition and interoperability aspect is not considered.

4.2 Architecture Description Languages

Using a meta model centric approach requires a maintainable and evolvable structure. Strittmatter et al. [13] proposed modular reference structure which meets these criteria. This approach can be used to design an ADL regarding the structural criteria, but it provides no solution for the modularization of simulations. Medvidovic et al. [5] evaluate the benefits of ADL's in the domain of software development. The work of McKenzie et al. [7] analyzes the utility and effectiveness of ADL's in the context of simulations. But their analysis is restricted to the federation structure of the HLA standard.

4.3 Composability and Interoperability of Simulations

To make simulation modules reusable they must be compose- and inter-operable. Petty et al. [6] published a composability lexicon to clear the connotation of composability in the context of simulations. The Distributed Interactive Simulation (DIS) [2] standard realized the composability of simulations on the protocol level. Based on the DIS standard, the High-Level Architecture (HLA) [4] evolved. The CODES approach of Teo et al. [9] utilizes an ontology approach to model a discrete event simulation. And Topçu et al. [16] use the HLA standard in combination with

modeling approaches. These proposed approaches are very restrictive regarding the standards they use, and none have an approach for generic simulations.

4.4 Decoupling of Monolithic Simulations

If an existing, monolithic simulation has to be modularized the modeling of simulations should be used and compose- and interoperability has to be integrated. Papadopoulos et al. [12] proposed an approach of automated decoupling of simulations.

5 Conclusion

There are not so many solutions for modular simulations. The DIS and HLA standards are one of the few solution approaches. But a common base for modular, composable and distributed simulation is not available. A generic architecture driven approach is also not available. The architectural aspects would allow structural analysis regarding existing simulations. Decoupling approaches are rare and need further research. If structural analyses are available, then an aided or automated process of modularizing simulations can be developed. Design patterns and bad smells in simulations design which is based on structural analyses could be possible. Overall would this base make simulations more accessible and comparable. Therefore this research topic might create a new field of view for simulations.

Acknowledgement

This work was partially supported by the MWK (Ministry of Science, Research and the Arts Baden-Württemberg) in the funding line Research Seed Capital (RiSC).

References

[1] A. M. Law, W. D. Kelton, and W. D. Kelton. *Simulation modeling and analysis*. Vol. 2. McGraw-Hill New York, 1991.

[2] D. S. Committee et al. "The DIS vision: A map to the future of distributed simulation". In: *Institute for Simulation and Training* (1994).

[3] J. S. Dahmann. "High level architecture for simulation". In: *Distributed Interactive Simulation and Real Time Applications, 1997., First International Workshop on*. IEEE, 1997, pp. 9–14.

[4] J. S. Dahmann, R. M. Fujimoto, and R. M. Weatherly. "The department of defense high level architecture". In: *29th conference on Winter simulation*. IEEE Computer Society, 1997, pp. 142–149.

[5] N. Medvidovic and R. N. Taylor. "A classification and comparison framework for software architecture description languages". In: *IEEE Transactions on software engineering* 26.1 (2000), pp. 70–93.

[6] M. D. Petty and E. W. Weisel. "A composability lexicon". In: *Spring Simulation Interoperability Workshop*. Vol. 2003. 2003, pp. 181–187.

[7] F. D. Mckenzie, M. D. Petty, and Q. Xu. "Usefulness of software architecture description languages for modeling and analysis of federates and federation architectures". In: *Simulation* 80.11 (2004), pp. 559–576.

[8] P. Benjamin, M. Patki, and R. Mayer. "Using ontologies for simulation modeling". In: *Winter Simulation Conference*. IEEE, 2006, pp. 1151–1159.

[9] Y. M. Teo and C. Szabo. "CoDES: An integrated approach to composable modeling and simulation". In: *Simulation Symposium, 2008. ANSS 2008. 41st Annual*. IEEE, 2008, pp. 103–110.

[10] D. Cetinkaya, A. Verbraeck, and M. D. Seck. "MDD4MS: a model driven development framework for modeling and simulation". In: *Proceedings of the 2011 summer computer simulation conference*. Society for Modeling & Simulation International, 2011, pp. 113–121.

[11] B. Möller. "The HLA tutorial v1. 0". In: *Pitch Technologies, Sweden* (2013).

[12] A. V. Papadopoulos and A. Leva. "Automating dynamic decoupling in object-oriented modelling and simulation tools". In: *5th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools UK*. Linköping University Electronic Press, 2013, pp. 37–44.

[13] M. Strittmatter et al. "A Modular Reference Structure for Component-based Architecture Description Languages". In: *2nd International Workshop on Model-Driven Engineering for Component-Based Software Systems*. 2015, pp. 36–41.

[14] R. Jung, R. Heinrich, and W. Hasselbring. "GECO: A Generator Composition Approach for Aspect-Oriented DSLs". In: Cham: Springer International Publishing, 2016, pp. 141–156.

[15] R. H. Reussner et al. *Modeling and simulating software architectures: The Palladio approach*. MIT Press, 2016.

[16] O. Topçu et al. *Distributed Simulation: A Model Driven Engineering Approach*. Springer, 2016.

[17] R. Heinrich et al. "Integrating business process simulation and information system simulation for performance prediction". In: *Software & Systems Modeling* 16.1 (2017), pp. 257–277.