Summary of the 1st Workshop on Modeling and Simulation of Software-Intensive Systems (MSSiS 2019)*

Valdemar Vicente Graciano Neto†, Elisa Yumi Nakagawa‡, Bernard Zeigler§

October 30, 2019

Abstract

The Workshop on Modeling and Simulation of Software-Intensive Systems (MSSiS) is a forum for researchers and practitioners from both communities modeling and simulation. The goal of the workshop is to provide an environment for discussions on how to propose and/or adapt notations, techniques, and methods to foster the adoption of dynamic models in Software Engineering (SE), besides exchanging ideas, challenges, practices, experiences, and results. The long-term aim is to build a community and a body of knowledge within the intersection of modeling and simulation. The first edition of the workshop was structured in technical sessions and two invited talks. Papers were presented, including short and full research papers that reported results on the adoption of simulation in SE projects and also the use and proposition of dynamic models for the engineering of software-intensive systems. The workshop attracted more than 30 participants (from 20 different institutions) during the full day and the positive feedback encouraged us to organize the workshop again in the next year.

*Co-located with the 10th Brazilian Congress on Software (CBSoft 2019)

†Federal University of Goiás, Goiânia, Brazil

‡University of São Paulo, São Carlos, Brazil

§University of Arizona, Tucson, United States

1 Introduction

Complex systems (such as smart cities) are emerging and imposing novel challenges for software engineering. Such systems can be composed of millions of constituent systems, such as mobile phones, autonomous cars, lighting systems, and smart traffic, buildings, and houses. Under this perspective, a sustainable planning of the structure and behaviors that can be drawn within the context of this type of systems intrinsically demands for *modeling*, i.e., the adoption of artifacts that capture the essential properties of that system and make use of abstraction to deal with the their complexity. UML and SysML are often used to predict properties of software-based systems; but in most cases the understanding and planning of the intended behaviors is manuall performed, what can be tiring, error prone, and very difficult to debug, particularly when considering systems that could potentially contain thousands or millions of constituent parts (systems).

This is the scenario of the 1st Workshop on Modeling and Simulation of Software Intensive Systems (MSSiS 2019), which occured in September 25th, 2019, and was co-located with the 10th Brazilian Congress on Software (CBSoft 2019) and was organized by Valdemar Vicente Graciano Neto (Universidade Federal de Goiás), Elisa Yumi Nakagawa (University of São Paulo), and Bernard Zeigler (University of Arizona).

The purposes of MSSiS 2019 were: (i) providing an environment for discussion of possible synergies between the existing software modeling techniques and the use of dynamic models, i.e., models that can be executed so that their structure and behaviors can be visualized and predicted at design time; (ii) outlining a community to work on and advance the state of the art about modeling of complex software-based systems; (iii) discussing how simulation models and other types of dynamic models (such as models@runtime) can be adopted in the software engineering of complex systems; and (iv) advancing on model-driven software engineering to also benefit the engineering of those types of models.

The workshop attracted more than 30 active participants during a full day and was structured in three technical sessions, besides two keynotes: Prof. Bernard Zeigler (University of Arizona, USA) and Prof. Breno Bernard Nicolau de França (State University of Campinas - UNICAMP, Brazil). Proceedings of the workshop are available online[1].

[1] https://sol.sbc.org.br/index.php/mssis/issue/view/438

2 Workshop Program Summary

MSSiS 2019 accepted technical papers (lenght of 10 pages) presenting research results or industrial practices and experiences related to the proposition and adoption of modeling and simulation (M&S) and correlated topics on the engineering of software for complex systems. Position papers (length of 5 pages) presenting position or preliminary results within the scope of the workshop were also accepted. MSSiS 2019 accepted eight technical papers, two position papers, and one invited paper.

Authors had 15 minutes to present full papers and 5 minutes to the short papers. After the presentations of all the papers in that technical session, a discussion moment with all the presenters was conducted by: (i) answering the chair questions; (ii) answering questions from the workshop participants; and (iii) also discussing potential intertwining among their research topics.

The program of MSSiS 2019 comprised of three thematic blocks. A summary of the papers presented are discussed in the following sections, which are named with the technical sessions titles.

2.1 Simulation Adoption and Dynamic Models

This thematic session gathered papers whose purpose was to report the adoption of simulation models at design time during software development process for real projects of a smart city. The papers focused on the following topics:

- The use of simulation models to evaluate, at design time, two architectural alternatives for IoT-based home monitoring systems to detect falls. Authors concluded that the addition of a new hardware component (and its corresponding software functionality) could increase the degree of falls detection in 8.2% in regards to a precedent result that was significantly high in precision;

- The adoption of DEVS simulation models to the use of an algorithm to validate the constructs of control access policies of a specification language for shared resources in a ubiquitous computing scenarios. Authors could conclude that both the language and the algorithm are valid approaches to handle concurrent access to smart objects resulting from overlapping smart spaces;

- Authors used simulation models to ensure a Software Assurance Case, i.e., an auditing instrument used to ensure that software for critical systems is safe in accordance with standards established by the responsible entities. The case was applied in aeronautical domain;
- Adoption of simulation for evaluating architectural alternatives of a Smart Street Lighting System of a Smart city.

2.2 Variability and Architecture with Dynamic Models

At that session, authors presented studies and results on how variability can be modeled so that it can support dynamics in software. A context-aware exception handling (CAEH) domain-specific language, called CatchML, was proposed in one of the studies. A case study was conducted on a sample system called UbiParking with nine volunteers. The results showed that the CatchML language is easy to model the context-aware exception handling and also allowed the participants to quickly locate the injected design faults. In turn, another study focused on modeling of variability in a Software Ecosystem (SECO) named SOLAR. SOLAR is a Virtual Learning Environment (VLE) that allows courses publication and interaction with them among its various users. Dynamic variability aspects of the SOLAR VLE were discussed using a features model. Finally, a study that provided visualization of commits of software repositories based on multiagent models (another important type of dynamic model used to reason about systems) was presented.

2.3 Model-Driven Development and Engineering

The last technical session focused on papers that reported results of Model-Driven Engineering/Development (MDE/MDD) projects, both contributing to (i) MDD as the target, and also (ii) as a means for supporting elegant solutions.

In regards to MDD as a means, one study presented a summary of results of a systematic mapping on Model-Driven Game Development. Another study focused on the adoption of MDD as a means to automatically obtain test cases generation and execution, enabling the tester to track execution

results and points in which software could exhibit problems. The tool was supported by graphical interface.

In regards to MDD as a target, two studies focused an experience report on how to automatically provide technology transfer in MDE as a Service, and implementations to support the technology transfer.

3 Summary of the Keynotes

Two important keynotes were provided in MSSiS 2019.

The international keynote speaker, Prof. Dr. Bernard Zeigler, is the inventor of DEVS simulation formalism. He is an emeritus professor of Electrical and Computer Engineering at the University of Arizona, USA. In his talk entitled *Model Based Systems Engineering (MBSE) with/out Simulation: State of the Art and Way Forward*, Prof. Zeigler showed the problems raised by MBSE taken as a modeling activity without the support of full strength integrated simulation capability and the potential for a closer integration between the two streams.

In turn, Prof. Dr. Breno França (UNICAMP) established the foundations for the conduction of experimental studies in software engineering using dynamic simulation models. In his talk, entitled *The Role of Simulation-based Studies in Software Engineering Research*, Prof. França enlightened us with the results of his research, showing how simulations are interesting and necessary in software development.

4 The Two Lessons Learned

Some important lessons were learned during the workshop. Answers for questions such as *Why should I use simulation?* were very clarifying.

Dr. Zeigler remarked that ***Lesson #1. Simulation models are important prototyping tools***. By using simulation models, we can, for instance, test how a software that drives a missile could work, since we can not explode a missile every time we want to assess this type of system.

Lesson #2. Static models are not good to provide feedback. By using dynamic models, i.e., models that can be run and that offer a corresponding animation can provide valuable feedbacks such as the effect of implementation decisions on a system, both on its structure and behaviors.

5 Research Agenda

Over the day, we collected challenges that the community reported as important and that should be tackled in the forthcoming years. Besides, in the end of the day, we opened to the participants to report us a list of research challenges and gaps, which we compiled as follows.

Reuse of dynamic models. Particularly regarding to simulation models, the cost to specify them at design time can be high. Techniques and methods on how to reuse dynamic models for software engineering purposes should be investigated.

Education and training on M&S. There are still practitioners in software engineering teams that have not expertise on the specification dynamic models, such as simulation models (DEVS or Systems Dynamics), Multiagent-based simulation, and models@runtime. Hence, training and education on M&S and all sort of dynamic modeling are still required.

Empirical evaluation of specification effort for dynamic models, particularly between simulation and ontologies. Studies on IoT were presented and participants argued that ontologies are a type of model frequently used to develop IoT systems.

Integration of modeling tools with simulation tools. Participants pointed out the need to not only offering a production environment that supports simulation, but also integrating environments that support models specification, such as ontologies and other static (but useful models) such as UML and SysML. Integration with domain-specific languages and their respective environments was also highlighted. Mechanisms to support visual modeling using those tools and an automatic transformation to provide operational semantics and visualization are also welcome.

Dealing with and processing large simulation (dynamic) models. BigMDE is a community that investigates how to specify and maintain ultra-large models. Since a smart city can have millions of constituent systems, large models are needed, as well as underlying supporting tools and methods. A simulation with more than, for instance, 200 parts requires more powerful processors and even distributed simulation (and corresponding processing

units).

Dynamic models validation and testing. This research gap is aligned with the classic testing problem: how to be sure that all possible combinations of states were tested? How to validate those models?

Datasets with real data. Simulations (and other dynamic models) require stimuli to be delivered so that they can trigger the simulation execution. Results can be considered more reliable depending on the dataset used to feed the models executions. Hence, it is imperative to create repositories with real datasets.

Adoption of dynamic models to simulate integration of toolchains. An investigation is needed to evaluate how dynamic models could be used to predict the compatibility of tools in MDE toolchains. Moreover, productivity could also be evaluated.

Tool support. The adoption of dynamic models and MDE is only feasible with underlying tool support, since they materialize methods and could contribute to the advance of the state of the art, enabling agility and productivity. Tools are also important for technology transfer, while academic tools are majorly not enough mature.

Generic modeling formalism to support reuse. One of the participants argued that model reuse is interesting, but in practice it does not work because different tools and DSL are used. Hence, a research on abstract formalisms is needed what could abstract several different DSL and tools so that a unified operational semantics could be provided to execute any of the DSL models.

Finally, during the workshop, we collected data from the participants so we could establish contact and a research group to involve them. Besides that, participants suggested to add other topics of interest in the next call for papers, particularly: (i) generation of models from metamodels; (ii) modeling for specific domains, such as smart farm, educacional, health, and robotics; and (iii) self-adaptive systems.

6 Future

The first edition of the MSSiS workshop was well-received and attracted several participants from many different institutions, what showed the potential of the workshop and the importance of the targeted main topic: the adoption of dynamic models in software engineering. The increasing complexity and dimensions of software-intensive systems pressure for the presence of these models in software development practice. Hence, the number of participants and the importance of the topic motivated us to submit a new edition of the workshop in the next year. We already invite researchers, students, academics, and practitioners to submit their papers to this next edition.

7 Acknowledgements

We thank all participants of the workshop and all the authors who submitted papers. We thank the IEEE Section Bahia (in the people of the professors Betânia Filha, Fabrício Braga, and Thiago Barbosa) and the IEEE Section Centro-Oeste Norte (in the people of the professors Alba Cristina Melo and José Oniram Limaverde Filho) for making possible the participation of Prof. Bernard Zeigler in MSSiS.

We thank the ADUFG (Sindicato dos Docentes das Universidades Federais de Goiás) for the sponsorship that was crucial for the realization of this event. We would like to extend our sincere thanks for all Organization Committee of CBSoft 2019, which was solicitous and willing about the demands of MSSiS 2019. We also thank the editors of the Journal of the Brazilian Computer Society (JBCS), Prof. Mike Hinchey (Lero/Irlanda) and Lúcia Drummond (UFF), by the waivers granted to the authors of the three best papers awarded in MSSiS 2019. In addition, we would like to sincerelly thank the members of the Program Committee:

Adair Rohling, UTFPR, Brazil
Breno França, UNICAMP, Brazil
Carlos Westphall, UFSC, Brazil
Claudio Gomes, University of Antwerp, Belgium
Davi Viana, UFMA, Brazil
Diana Adamatti, FURG, Brazil
Doohwan Kim, RTSync Corp, United States

Everton Cavalcante, UFRN, Brazil
Fabio Basso, UNIPAMPA, Brazil
Fabio Moreira, UFG, Brazil
Fernando Barros, Universidade de Coimbra, Portugal
Flavia Delicato, UFRJ, Brazil
Flavio Horita, UFABC, Brazil
Guilherme Travassos, UFRJ, Brazil
Gerd Wagner, Brandenburg University of Technology, Germany
Luis G. Nardin, Brandenburg University of Technology, Germany
Lina Garcés, ICMC/USP, Brazil
Marco Antonio Araujo, UFJF/IF Sudeste de Minas, Brazil
Mohamad Kassab, Pennsylvania State University, United States
Pablo Antonino, Fraunhofer IESE, Germany
Paulo Maciel, UFPE, Brazil
Paulo Pires, UFRJ, Brazil
Rafael Capilla, Universidad Rey Juan Carlos, Spain
Rita Maciel, UFBA, Brazil
Rodrigo Santos, UNIRIO, Brazil
Rosana Braga, USP, Brazil
Santiago Matalonga, Universidade ORT, Uruguai
Sérgio Teixeira, UFG, Brazil
Sofia Paiva, UFSJDelRei, Brazil
Toacy Oliveira, UFRJ, Brazil
Vânia Neves, UFF, Brazil