Simulation of Software Architectures of Smart Ecosystems: Theory and Practice

Valdemar Vicente Graciano Neto[a], Wallace Manzano[b], Pablo Oliveira Antonino[c] and Elisa Yumi Nakagawa[b]

[a] *Federal University of Goiás, Brazil*
[b] *University of São Paulo, Brazil*
[c] *Fraunhofer IESE, Germany*

Abstract

Smart ecosystems combine various heterogeneous and independent software-intensive systems to enable complex functionalities for highly dynamic smart applications, such as Industry 4.0, smart cities, transportation, automotive, and many other critical domains. Due to their complex nature, such ecosystems, which are often referred as to Systems-of-Systems (SoS), should be completely reliable and work without interruption or failures that could cause serious losses and damages. During the smart-ecosystem architectural design, the impact of eventual failures or architectural changes should then be predicted to avoid potential losses or damages. This tutorial presents a simulation-based approach to support the prediction, at design time, of the structure and behavior of smart-ecosystem architectures (which are inherently dynamic at runtime), aiming to evaluate whether the smart ecosystems can sustain their operation. To do that, we present the foundations and concepts associated with smart ecosystems/SoS and simulation, as well as the results of our multiple studies. We also offer hands-on experience in the simulation of software architectures, using artifacts associated with a pre-specified smart-ecosystem architecture and a free commercial simulator. We share with the European Conference on Software Architecture (ECSA 2021) audience the theoretical knowledge and practical experience that could leverage the adoption of simulation approaches during the development of software-intensive systems, in particular, those so complex and dynamic as smart ecosystems.

Keywords

Smart Ecosystem, System-of-Systems, Software Architecture, Simulation, Architectural Evaluation, DEVS

1. Introduction

Software has been increasingly embedded into several types of systems, making them smarter and software-intensive, i.e., software has crosscut the entire system development life cycle. Such software-intensive and independent systems have been connected through communication technologies, raising alliances of highly interoperable constituent systems and forming what is known as smart ecosystems or Systems-of-Systems (SoS)[1] [1, 2, 3, 4]. Smart ecosystems combine heterogeneous and independent constituent systems to offer complex functionalities for several critical application domains. Such ecosystems have a considerably dynamic software architecture, i.e., the architecture has its structure changing over time due to constituents that join and leave the ecosystems or are replaced or reorganized at runtime. Due to the critical nature of the domains supported by them, smart ecosystems should be reliable and work without interruption or failures that could cause serious losses or damages. However, given the dynamic nature of smart-ecosystem architectures, assuring the feasibility of each architectural arrangement that a smart ecosystem can assume at runtime requires a prior analysis, still at design time, to assure that both the smart-ecosystem structure and behavior can be sustained when its architecture changes.

Over the past years, different initiatives have been proposed to assure the quality of the software architectures of smart ecosystems [5, 6, 7, 8, 9]. In the context of our research projects, we have explored the adoption of simulation and observed its capability to address the prediction of the structure and behavior of such architectures at runtime [10, 11, 12, 13, 14]. In particular, we can mention ASAS [11] and Dynamic-SoS [13]. The former comprises a simulation-based process-oriented approach for evaluating smart-ecosystem coalitions (i.e., each different architectural arrangement that a smart ecosystem can present at runtime), whereas the latter is a method for evaluating smart-ecosystem dynamic architectures, benchmarking them still at design time to predict the architecture properties.

Motivated by the results achieved in our research involving smart ecosystems/SoS and simulation, this tuto-

ECSA'21: European Conference on Software Architecture, 13-17 September 2021, Virtual from Växjö, Sweden

valdemarneto@ufg.br (V. V. Graciano Neto); wallace.manzano@usp.br (W. Manzano); pablo.antonino@iese.fraunhofer.de (P. O. Antonino); elisa@icmc.usp.br (E. Y. Nakagawa)
0000-0003-2190-5477 (V. V. Graciano Neto); 0000-0001-5602-3023 (W. Manzano); 0000-0002-9631-8771 (P. O. Antonino); 0000-0002-7754-4298 (E. Y. Nakagawa)

[1]For sake of simplicity, smart ecosystems and SoS are used interchangeably in the context of this text.

rial provides for the ECSA attendees the theoretical foundations and hands-on experience on using simulation models to evaluate, still at design time, the structure and behaviors of smart-ecosystem architectures at runtime. More specifically, we provide (i) theoretical foundation on smart ecosystems/SoS, dynamic software architectures, and simulation; (ii) hands-on experience with a free and well-known commercial simulator (MS4Me[2]) in which participants have the opportunity to run simulation models using a pre-conceived software architecture specified in a simulation formalism (i.e., DEVS [15]).

The remainder of this text is structured as follows: Section 2 covers the tutorial learning aspects, Section 3 provides the technical aspects, and Section 4 introduces the presenters' background.

2. Tutorial Learning Aspects

This section provides the tutorial structure, the topics covered, learning objectives, key takeaways for the audience, and the relevance of the theme addressed in this tutorial for ECSA.

2.1. Tutorial structure

This tutorial is organized in two phases: *instructional phase* and *practical phase*. The first phase refers to an explanatory presentation on: (i) Fundamentals on smart ecosystems, smart-ecosystem architectures, and Dynamic-ASAS; and (ii) DEVS basics. The practical phase addresses a supervised MS4Me installation in the attendees' machines and a hands-on experience applying what was presented in the previous phase, as follows:

- **Step 1. Design of the Architecture**: Architects design the smart-ecosystem architecture from the specifications of the requirements and missions, which were established in the early phases of the engineering life cycle. **During the tutorial:** Considering the time constraints of this tutorial, we provide the artifacts of a pre-conceived architecture and explain and discuss the structure of this small-scale smart ecosystem.
- **Step 2. Evaluation Planning**: An evaluation plan prepared in this step is composed of one or more smart-ecosystem missions to be observed, a set of different coalitions to be analyzed, and a set of metrics related to a given quality attribute to be measured. We provide and discuss an evaluation plan with parameters to be measured during the simulation. These parameters include: (i) variables (metrics) to support the measurement during the simulation execution; and (ii) behaviors to be observed that are often defined at the requirements level as a set of missions to be evaluated through behaviors that emerge during the simulations. **During the tutorial:** We provide an artifact for the attendees with a pre-established evaluation plan, followed by an explanation and a discussion. The evaluation plan is composed of, for instance, a set of three missions of a smart ecosystem and a set of metrics associated with quality attributes such as functional suitability. An example of metrics is Functional Completeness (FCom), i.e., the degree to which the set of functions covers all specified tasks and user objectives. Considering the set of the three pre-established missions, the simulation verifies how many of them are effectively achieved [14].
- **Step 3. Specification of DEVS Simulation Models:** DEVS simulation models are built in conformance to the architectural design. **During the tutorial:** We provide a set of pre-built simulation artifacts such as the Dynamic Reconfiguration Controller (which is a mechanism to manage the SoS reconfigurations to exercise the multiple coalitions that a SoS can assume). We also provide the specification in DEVS of the smart-ecosystem architecture, followed by an explanation and discussion.
- **Step 4. Environment Installation and Simulation Deployment**: This step involves the management of the artifacts obtained in Step 3 and their deployment into MS4Me. **During the tutorial:** We supervise the attendees with the tool installation in their machines. Following, we present the way to accordingly deploy the DEVS models into MS4Me and support them.
- **Step 5. Simulation Execution and Architectural Analysis**: This step consists in the launching of the simulation in MS4Me, monitoring it through observation, possibly interacting with the simulation, and exercising multiple architectural configurations. Data and execution traces are logged during this process for further examination. **During the tutorial:** We stimulate attendees to run the simulations over MS4Me and accordingly evaluate the SoS architecture being simulated.
- **Step 6. Analysis Execution**: This step performs an inspection of the execution traces in log files. Conclusions are obtained according to the pre-established set of missions, the manifested behaviors of the simulation model, and corresponding metrics. **During the tutorial:** We provide for the attendees a guided tour on the analysis of the execution logs so that conclusions can be drawn on the properties analyzed regarding the SoS architecture being simulated.

[2]http://goo.gl/NmBBuu

2.2. Covered topics

The main topics covered in this tutorial are smart ecosystems, smart-ecosystem software architectures, and simulation. **Smart ecosystem** is the main topic and refers to an emerging topic in the software architecture research; so, they are presented herein under theoretical and applied perspectives. **Smart-ecosystem software architectures** have been mentioned in several studies over the past years [5, 8, 9], demonstrating the importance of this topic. We address it in this tutorial compilating the theoretical basis accumulated by our group and presenting a practical analysis of such architectures using simulation. **Simulation** is broadly recognized as one of the main techniques to evaluate software architectures [16]. Herein, we adopt DEVS simulation formalism for evaluating multiple smart-ecosystem coalitions.

2.3. Learning objectives

This tutorial provides practical experience in evaluating smart-ecosystem architectures. We rely on MS4Me and DEVS language, one of the main simulation formalisms used in software engineering empirical studies and a language prepared to simulate SoS architectures [15]. This tutorial copes with the following learning objectives:

- **Knowledge of fundamentals of smart-ecosystem architectures**: Before offering a practical experience for the attendees, we aim to consolidate a consensual understanding of what we consider (in the context of this tutorial) as smart ecosystems, their dynamic architectures, and simulation. Hence, we discuss the nature of smart ecosystems besides the reconfigurations that can take place over their architecture at runtime;
- **Introduction of DEVS for newcomers**: We introduce DEVS as a formalism suitable for smart-ecosystem architectures by presenting the basics of DEVS models (both atomic and coupled models), their canonical structure, and DEVSNL (DEVS Natural Language) used to specify DEVS simulation models in MS4Me;
- **Introduction of a simulation approach**: We explain the steps to specify and evaluate smart-ecosystem architectures according to Dynamic-ASAS; and
- **Practical experience of smart-ecosystem simulation**: We equip the attendees with pre-programmed DEVS codes so they can deploy them in the MS4Me platform and run simulations for assessing a smart-ecosystem architecture.

2.4. Key takeaways for the audience

The essential messages that we intend the audience retains are: (i) Smart-ecosystem software architectures are highly dynamic; (ii) The structure and behaviors of smart ecosystems must be planned and assessed at design time; and (iii) A simulation-based approach can evaluate smart-ecosystem architectures and predict their properties.

2.5. Relevance for the ECSA Audience

Smart ecosystems are one of the topics of interest of ECSA 2021 and an emerging relevant topic that represents cutting-edge technology, imposing important challenges, including for the software architecture area. They also support critical domains in which failures can cause damages, losses, and financial harm. Hence, the establishment of approaches to evaluate their architectures accordingly is imperative [8, 11]. Simulation supports software architecture assessment [16, 17] and allows architects to (i) prototype large-scale systems and test their structure and behaviors at design time, (ii) anticipate/predict the consequences of architectural changes on the overall systems, and (iii) offer a visual appeal to enable the architects to draw new architectural alternatives to accordingly conform to the pre-established requirements. Hence, the dissemination of knowledge on simulation, simulators, and languages becomes very relevant in the software architecture community. Finally, this tutorial not only copes with the conference scope but also updates the audience with findings, results recently published, and experience from the industry.

3. Tutorial Technical Aspects

The target audience comprises any ECSA attendees only requiring experience in basic programming. This half-day workshop is structured as follows: (i) **Instructional Phase:** Instructors presentation - 10 minutes; Fundamentals on smart ecosystems, their architectures, and Dynamic-ASAS - 50 minutes; DEVS basics - 30 minutes; and (ii) **Practical Phase:** MS4Me Installation - 20 minutes; Hands-on lab following the six steps presented in Section 2.1 - 60 to 120 minutes. Hence, this tutorial mixes expositive lecture and hands-on practical experience. This tutorial is conducted in a virtual mode, raising specific technical challenges. Hence, we send the MS4Me installation instructions before the workshop, and supplementary materials are available in https://ww2.inf.ufg.br/~insight/tutorialecsa2021/. We also record videos that show the main activities as well as the expected results of those activities.

4. Background of Presenters

VALDEMAR VICENTE GRACIANO-NETO received his Ph.D. degree from the University of São Paulo, Brazil and the Docteur degree from the Université Bretagne-Sud, France, in 2018. He is an Assistant Professor at the Federal University of Goiás, Brazil. He has co-authored more than 80 peer-reviewed papers, besides co-organizing scientific events, such as the Workshop on Modeling and Simulation of Software-Intensive Systems (MSSiS) and Workshop on Blockchain-Based Software Architectures (BlockArch at ICSA). He serves as a reviewer for important vehicles, such as IEEE SoSE, IST, and IEEE Computer. He is a member of the Brazilian Computer Society.

WALLACE MANZANO is a Masters' candidate at the University of São Paulo, Brazil. He has an Information Systems Bachelors' degree and a large experience on SoS, model-driven development, and software architecture. He is also an expert in DEVS language and simulators (including MS4Me) and has accumulated large experience in simulations models over the past five years.

PABLO OLIVEIRA ANTONINO is Head of the Embedded Software Engineering department of the Fraunhofer IESE, Germany. He holds a PhD in Computer Science from Technische Universität Kaiserslautern, and has experience with the design, evaluation, and integration of dependable embedded systems from various domains, such as automotive, avionics, agricultural and construction machines, medical devices, and smart industries. The Industry 4.0 middleware BaSyx is mainly developed by employees in the department managed by Dr. Antonino.

ELISA YUMI NAKAGAWA is an associate professor at the University of São Paulo - USP, Brazil. She was a visiting researcher in 2020 at the Fraunhofer IESE, Germany, conducted her post-doctoral research at the University of South Brittany, France, in 2015, and Fraunhofer IESE, in 2012. She received her Ph.D. degree from USP in 2016. She has coordinated several international research projects, has organized international conferences, and has served as a program committee member at many conferences and as a reviewer of various journals. She published more than 180 papers in selective journals and conferences, in addition to books and book chapters. She is a CNPq fellow and a member of IEEE and Brazilian Computer Society.

References

[1] M. Goncalves, E. Cavalcante, T. Batista, F. Oquendo, E. Y. Nakagawa, Towards a conceptual model for software-intensive system-of-systems, in: SMC, 2014, pp. 1605–1610.

[2] B. Boehm, A view of 20th and 21st century software engineering, in: ICSE, 2006, pp. 12–29.

[3] M. Jamshidi, System of systems engineering - new challenges for the 21st century, IEEE Aerospace and Electronic Systems Magazine 23 (2008) 4–19.

[4] M. Maier, Architecting principles for systems-of-systems, Systems Engineering 1 (1998) 267–284.

[5] P. Baumann, R. Samlaus, L. Mikelsons, T. Kuhn, J. Jahic, Towards virtual validation of distributed functions, in: SummerSim, 2019, pp. 1–12.

[6] M. Guessi, F. Oquendo, E. Y. Nakagawa, Checking the architectural feasibility of systems-of-systems using formal descriptions, in: SoSE, 2016, pp. 1–6.

[7] M. Guessi, F. Oquendo, E. Y. Nakagawa, Ark: a constraint-based method for architectural synthesis of smart systems, Software System Modeling 19 (2020) 741–762.

[8] H. Cadavid, V. Andrikopoulos, P. Avgeriou, Architecting systems of systems: A tertiary study, Information and Software Technology 118 (2020).

[9] H. Cadavid, V. Andrikopoulos, P. Avgeriou, P. Broekema, System- and software-level architecting harmonization practices for systems-of-systems - an exploratory case study on a long-running large-scale scientific instrument, in: ICSA, 2021, pp. 13–24.

[10] V. Graciano Neto, C. Paes, L. Garcés, M. Guessi, F. Oquendo, E. Y. Nakagawa, Stimuli-SoS: A model-based approach to derive stimuli generators in simulations of software architectures of systems-of-systems, Journal of the Brazilian Computer Society 23 (2017) 13:1–13:22.

[11] V. V. Graciano Neto, L. Garcés, M. Guessi, C. Paes, W. Manzano, F. Oquendo, E. Y. Nakagawa, ASAS: An approach to support simulation of smart systems, in: HICSS, 2018, pp. 5777–5786.

[12] V. Graciano Neto, C. Paes, A. Rohling, W. Manzano, E. Y. Nakagawa, Modeling & simulation of software architectures of systems-of-systems: An industrial report on the Brazilian space system, in: SpringSim, 2019, pp. 1–12.

[13] W. Manzano, V. Graciano Neto, E. Y. Nakagawa, Dynamic-SoS: An approach for the simulation of systems-of-systems dynamic architectures, Computer Journal 63 (2020) 709–731.

[14] V. Graciano Neto, F. Horita, R. Santos, D. Viana, M. Kassab, W. Manzano, E. Y. Nakagawa, S.O.B (Save Our Budget) - A Simulation-Based Method for Prediction of Acquisition Costs of Constituents of a System-of-Systems, iSys - Brazilian Journal on Information Systems 12 (2019) 6–35.

[15] B. Zeigler, H. S. Sarjoughian, R. Duboz, J.-C. Souli, Guide to Modeling and Simulation of Systems of Systems, Springer-Verlag London, 2012.

[16] L. Dobrica, E. Niemele, A survey on software architecture analysis methods, IEEE Transactions on Software Engineering 28 (2002) 638–653.

[17] V. Bogado, S. Gonnet, H. Leone, Modeling and simulation of software architecture in discrete event system specification for quality evaluation, Simulation 90 (2014) 290–319.