PhD topic: Towards an operational semantics for digital twins models based on SysML v2 and intelligent assistants for formal verification and validation

PhD advisors: Iulian OBER (ISAE-SUPAERO), Sébastien GERARD (CEA-List), Jérémie TATIBOUET (CEA-List)

To apply, please send a motivation letter to iulian.ober@isae-supaero.fr accompanied by:

- A detailed CV
- Transcripts of records with grades during the Bachelor's and Master's degree
- If possible, letters of recommendation or reference contacts from past academic advisors
- Any other information that supplements your CV: links to past publications, code repositories, etc.

Digital twins are at the heart of industrial concerns for the control and analysis of complex systems, from their design to their maintenance in operation. Digital twins are built with a given use in mind (e.g., understanding a complex phenomenon, optimizing a production line or controlling a biological process), and once deployed they are at the heart of decision processes. It is therefore important that their design is rigorous, and that we can verify that they do what we expect of them.

Systems Modeling Language (SysML) version 2.0 [4] has been recently published in beta version by the OMG and constitutes a major departure from SysML 1.x. While the previous versions were defined as extensions of the Unified Modeling Language (UML) [5] and carried with them a lot of software engineering legacy, the version 2 adopts a fresh start and takes a systems engineering perspective from the outset. The new language definition is based on a simpler and smaller language, the Kernel Modeling Language (KerML) [6], specifically designed as a foundation for SysML v2. KerML includes general constructs for structuring models, such as elements, relationships, annotations and namespaces, constructs based on classification (types, specialization), and additional constructs for commonly needed modeling capabilities, such as associations and behaviors. The KerML standard includes a declarative semantics formalized in first order logic, but which cannot be used directly to derive an executable semantics for KerML models.

The first objective of this thesis is to propose a SysMLV2 method for modeling digital twins. Digital twins models built thanks to the methodology will conform to the standard and support verification and validation through simulation and formal verification. To make this possible, the operational semantics of SysML v2 will be defined.

Recently, modern semantic definition formalisms have been introduced by the programming language theory community, which are at the same time formal, readable and practical to use for large and complex languages, e.g., [3,7]. Tools like [2] allow to obtain with little effort a simulator or debugger for any modeling language with an executable semantics, including languages that support concurrency [8]. Such formalisms and tools have not yet been used beyond programming languages to systems engineering models, but they are deemed well suited for the definition of the semantics of most SysML v2 constructs.

Formal verification has always been a discipline reserved for a handful of experts who have mastered technologies that are often confidential, thus limiting their application at industrial level to a select few. With the advance of generative AI and intelligent assistants, it now seems conceivable to create virtual assistants for formal verification. The second objective of the thesis will be to define and implement a technological framework for augmenting the modeling tool with formal verification assistants for the SysML V2 digital twin model.

The plan for this thesis is as follows:

- Stage 1: Conduct a comprehensive study of the state of the art on semantics definition frameworks, with focus on tool support for automatically deriving simulation and verification engines from semantic definitions. In parallel, an in-depth study of the semantic definition clauses of the KerML and SysML v2 standards is to be conducted. Select a simplified yet realistic system, such as the one from [1], to be used as running example and produce a prototype SysML v2 model for it.

- Stage 2: Produce an incremental definition of the operational semantics of the most used constructs from SysML v2, starting with those defined in KerML. At each iteration, to the extent supported by the semantic framework, automatically generate a simulation/verification platform for the defined language fragment and test it by applying it to the running example model. Prototypes will be produced in *Rust*[1] and integrated into the SysON open-source modeler (https://mbse-syson.org/). In parallel, the conformity of the operational semantics to the declarative semantics prescribed by the standard will be studied, and a method for proving the conformity between the two semantics will be proposed.

- Stage 3: Study the state of the art of intelligent agent technological frameworks in order to propose and deploy a verification assistance solution that will be integrated into the digital twin modeling tool, in order to make formal verification of SysML v2 conformant digital twin models accessible.

Required qualifications

Candidates must be motivated to work in the field of formal methods, model-based approaches, multi-agent systems, and GenAI for systems engineering. A Master’s degree or equivalent is required, with a good background in systems engineering, computer science, formal methods and closely related topics, as well as excellent abstraction and reasoning capabilities. Good experience in programming is required, and experience in Rust is a plus. A solid background experience in cyber-physical systems is a plus. Proficiency in English as well as good writing skills are required.

[1] https://www.rust-lang.org/

References

[1] F. Boniol and V. Wiels. The landing gear system case study. In Frédéric Boniol, Virginie Wiels, Yamine Ait Ameur, and Klaus-Dieter Schewe, editors, ABZ 2014: The Landing Gear Case Study, pages 1–18, Cham, 2014. Springer International Publishing.

[2] Erwan Bousse, Dorian Leroy, Benoît Combemale, Manuel Wimmer, and Benoit Baudry. Omniscient debugging for executable DSLs. J. Syst. Softw., 137:261–288, 2018.

[3] Xiaohong Chen and Grigore Rosu. The K vision for the future of programming language design and analysis. In Ezio Bartocci, Yliès Falcone, and Martin Leucker, editors, Formal Methods in Outer Space - Essays Dedicated to Klaus Havelund on the Occasion of His 65th Birthday, volume 13065 of Lecture Notes in Computer Science, pages 3–9. Springer, 2021.

[4] OMG. OMG Systems Modeling Language (SysML) – Part 1: Language Specification. https://www.omg.org/spec/SysML/2.0/Beta2/Language/PDF, 2024

[5] OMG. Unified Modeling Language (UML). https://www.omg.org/spec/UML/2.5.1/PDF, 2017

[6] OMG. Kernel Modeling Language (KerML). https://www.omg.org/spec/KerML/1.0/Beta2/PDF, 2024

[7] Grigore Rosu. K: A semantic framework for programming languages and formal analysis tools. In Dependable Software Systems Engineering, volume 50 of NATO Science for Peace and Security Series - D: Information and Communication Security, pages 186–206. IOS Press, 2017.

[8] Steffen Zschaler, Erwan Bousse, Julien Deantoni, and Benoît Combemale. A generic framework for representing and analyzing model concurrency. Softw. Syst. Model., 22(4):1319–1340, 2023.