A Process Mining Approach to Improving Defect Detection of SysML Models

Mounifah Alenazi*,†, Nan Niu*, and Juha Savolainen‡
* Department of Electrical Engineering and Computer Science, University of Cincinnati, USA
† College of Arts and Sciences, University of Hafr Albatin, SA
‡ Global Software and Control R&D, Danfoss Drives A/S, Denmark
Email: alenazmh@mail.uc.edu, nan.niu@uc.edu, juha.savolainen@danfoss.com

***Abstract*—The development of complex and dependable systems like autonomous vehicles relies increasingly on the use of systems modeling language (SysML). In fact, SysML has become a *de facto* standard for systems engineering. With model-driven engineering, a SysML model serves as a reference for the early defect detection of the system under design: the earlier the errors are detected, the less is the cost of handling the errors. Mutation testing is a fault-based technique that has recently seen its applications to SysML behavioral models (e.g., state machine diagrams). Specifically, a system's state-transition design can be fed to a model checker where mutants are automatically generated and then killed against the desired design specifications (e.g., safety properties). In this paper, we present a novel approach based on process mining to improve the effectiveness and efficiency of the SysML mutation testing based on model checking. In our approach, the mutation operators are applied directly to the state machine diagram. These mutants are then fed as traces into a process mining tool and checked according to the event logs. Our initial results indicate that the process mining approach kills more mutants faster than the model checking method.**

***Index Terms*—systems modeling language (SysML), mutation testing, model checking, process mining, defect detection, model-driven engineering**

I. Introduction

The development and maintenance of industrial systems like autonomous vehicles is becoming more challenging and costly with the increased size and complexity. To address the challenges, model-based systems engineering (MBSE) defines a methodology of modeling to support systems requirements, design, analysis, and verification and validation activities starting from the conceptual design phase and continuing through development and later lifecycle phases [1]. The overall objective is to replace a mere document-centric approach with a model-centric one, which enables domain experts and engineers to easily capture requirements, understand design change impacts, define traceability paths, and analyze system design before it is built [2]. Indeed, the development of complex systems such as cyber physical systems and automated production systems [3], [4] increasingly relies on models developed in Systems Modeling Language (SysML).

For systems whose developments involve different disciplines (e.g., software, mechanical, electrical, or pneumatic components are co-developed), SysML is considered as a standard modeling language to use [5]. SysML has been adopted by many companies (e.g., Boeing, Rolls Royce, SAP, Honeywell, and HSBC [6]), as it greatly helps to create object-oriented models of systems that incorporate not only software, but also hardware, devices, and other control components [7], expressing both structure and behavior for complex systems. Thus, current industry trends look for models that are not only generative and operationable, but also allow for defect detection early in the design phase [8].

Model checking is a formal verification technique for examining the behavior of software and hardware applications. Given a model of the system and a property, the model checking algorithm explores exhaustively and automatically the model state space in order to determine whether it satisfies the checked property or not. If a property is violated a counterexample is provided. However, model checking is often hindered by the state explosion problem. To mitigate this problem, engineers often attempt to prove properties on a system by simplifying the system's model first. However, the results are valid only if the model is an appropriate abstraction of the real system.

In this paper, we propose a novel approach based on process mining techniques and mutation testing to examine how changes in SysML design models affect safety properties. Some changes violate the properties under test and can be easily detected, and some changes may not create any violations. Compared to model checking, process mining takes an event log (instead of a model) and a property and verifies whether the property is satisfied or not. Mutation testing on the other hand uncovers faults that are not detected by property checking. Van der Aalst *et al.* [9] develop a process mining approach with an extension of Linear Temporal Logic (LTL) tailored towards event logs holding information on activities, cases (i.e., process instances), timestamps, originators, and related data. This language is specifically developed to formulate properties in the context of event logs. We adapt the same concept but on a SysML model using its XML Metadata Interchange (XMI) representation. Process mining is an emerging discipline based on data mining and business process management [10]. It is used to automatically construct a model (e.g., in a form of Petri nets) of an existing process by analyzing its event logs. The main aspect of process mining is to capture system

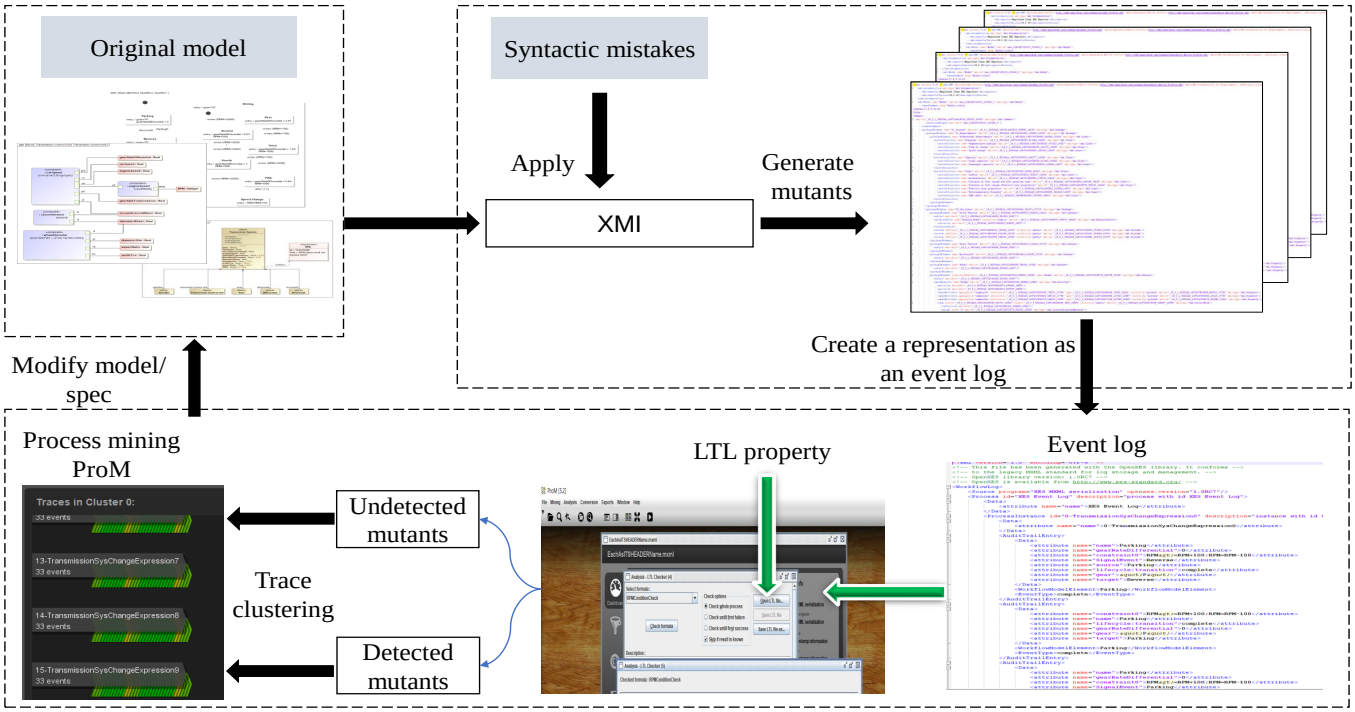

Fig. 1. General overview of our process mining-based approach to verify properties based on SysML models.

behavior by relying on sequentially ordered events recorded by an information system (i.e., event logs). There are three main classes of process mining techniques: discovery where a model is automatically constructed from event logs, conformance checking where an existing model is available and compared with an event log for checking purposes, and enhancement where an existing model is also available but used to improve the performance of existing model [10]. Process mining covers different aspects such as the control aspect, which captures the order in which tasks (e.g., SysML elements) are executed (i.e., the control-flow), and the information aspect, which captures the data, documents, and information needed and produced by a task (e.g., constraints, conditions, and timing).

The goal of this work is to develop an automated approach based on process mining and mutation testing to improve the defect detection of SysML design models. To the best of our knowledge, no existing research achieves the same goal. Specifically, this paper makes the following contributions: (1) defining the mutation operators by identifying common SysML modeling mistakes, (2) a tool that automatically simulates these mistakes (i.e., changes) and generates mutants of the original model, (3) creating a representation of these models as variants into a process mining tool, and (4) checking properties with an LTL-checker based on event logs of these mutants.

II. Process Mining-Based Approach to Detect Defects on SysML Models

The idea of integrating model checking with mutation testing was initially introduced by Ammann *et al.* [11]. The tenet is to automatically generate test cases from killed/detected mutants. From the perspective of model-driven engineering, a translation from state machine diagram into a formal model (i.e., input language) of some model checker is a prerequisite step before applying model checking approach [12] on the mutated models. In mutation testing, the most critical activity is to assess the quality of the mutants so that they reflect the typical or common mistakes. This is often performed at the code level. In our work, we identify the common mistakes at the modeling level [13]. Specifically, we identify the common mistakes modelers perform in SysML state machine and then generate mutants according to these mistakes. Our approach comprises the steps displayed in Figure 1:

- Automatically generate model mutants based on the identified syntactic mistakes using the XMI representation of the original model. As a result, our approach takes as input a state machine in XMI format, which is exported from an existing SysML modeling tool.
- Create a representation of these models as variants into a process mining tool, where each mutant represents a complete trace. This can be done automatically by parsing the XMI files of the model mutants. In particular, each mutant is defined with a case ID in the event log, and each state is defined as an event.
- Import the created event log into a process mining tool with proper configuration.
- Run a safety property check using the LTL checker on the event log. The safety property is written in the language described in [14]. The traces are then clustered to analyze

the results.

We compare our proposed approach with SysML mutation testing based on NuSMV model checker. Our preliminary results indicate that the process mining approach kills more mutants faster than the model checking method.

REFERENCES

[1] S. Friedenthal, R. Griego, and M. Sampson, "INCOSE model based systems engineering (MBSE) initiative," in *INCOSE 2007 Symposium*, vol. 11, 2007.

[2] L. E. Hart, "Introduction to model-based system engineering (MBSE) and SysML," in *Delaware Valley INCOSE Chapter Meeting, Ramblewood Country Club, Mount Laurel, New Jersey*, 2015.

[3] M. Alenazi, N. Niu, W. Wang, and B. Vogel-Heuser, "Traceability for evolving automated production systems," in *Grand Challenges of Traceability: The Next Ten Years*, Kentucky, USA, April 2017, pp. 28–30.

[4] W. Wang, N. Niu, M. Alenazi, and L. D. Xu, "In-place traceability for automated production systems: A survey of PLC and SysML tools," *IEEE Trans. Industrial Informatics*, vol. 15, no. 6, pp. 3155–3162, 2019.

[5] W. Schäfer and H. Wehrheim, "The challenges of building advanced mechatronic systems," in *International Conference on the Future of Software Engineering (FOSE)*, Minneapolis, MN, USA, May 2007, pp. 72–84.

[6] No Magic, "Clients of No Magic," Last accessed:May 2019. [Online]. Available: {https://www.nomagic.com/company/our-clients}

[7] E. Huang, R. Ramamurthy, and L. F. McGinnis, "System and simulation modeling using SysML," in *2007 Winter Simulation Conference*, Dec 2007, pp. 796–803.

[8] M. Hernandez, I. Diaz, C. Corbato, R. Bravo, and J. Bermejo, "Process systems lifecycle management using a model based engineering approach," *Computing and Systems Technology Division 2016 - Core Programming Area at the 2016 AIChE Annual Meeting*, pp. 242–244, 2016.

[9] W. M. van der Aalst, H. De Beer, and B. F. van Dongen, "Process mining and verification of properties: An approach based on temporal logic," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2005, pp. 130–147.

[10] W. M. P. van der Aalst, *Process Mining Discovery, Conformance and Enhancement of Business Processes*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[11] P. E. Ammann, P. E. Black, and W. Majurski, "Using model checking to generate tests from specifications," in *Proceedings Second International Conference on Formal Engineering Methods*. IEEE, Dec 1998, pp. 46–54.

[12] D. Latella, I. Majzik, and M. Massink, "Automatic verification of a behavioural subset of UML statechart diagrams using the SPIN model-checker," *Formal Aspects of Computing*, vol. 11, no. 6, pp. 637–664, 1999.

[13] M. Alenazi, N. Niu, and J. Savolainen, "SysML modeling mistakes and their impacts on requirements," in *International Model-Driven Requirements Engineering Workshop (MoDRE)*. Jeju Island, South Korea: IEEE, September 2019 (accepted).

[14] H. De Beer and P. van den Brand, "The LTL checker plugins: A reference manual," *Eindhoven University of Technology, Eindhoven*, vol. 132, 2004.