MEDITATE: Toward Model-based Generation and Optimization of ADAS Testing Scenarios in Co-simulation Environments

Luigi Libero Lucio Starace, Anna Corazza, Anna Rita Fasolino, Sergio Di Martino
University of Naples Federico II, Naples, Italy
{luigiliberolucio.starace, anna.corazza, fasolino, sergio.dimartino}@unina.it

Antonio Trovato, Dario Di Nucci, Carmine Gravino
University of Salerno, Fisciano (SA), Italy
{atrovato, ddinucci, gravino}@unisa.it

Daniele Di Pompeo, Vittorio Cortellessa
University of L'Aquila, L'Aquila, Italy
{daniele.dipompeo, vittorio.cortellessa}@univaq.it

***Abstract*—Advanced Driver Assistance Systems (ADAS) are fundamental in accelerating advances toward safer and more environmentally friendly mobility. Since validating ADAS on actual hardware is costly and traditional field testing is not feasible, the de-facto standard for ADAS validation consists of co-simulation solutions that integrate 3D virtual environments in the Model-in-the-Loop (MiL) paradigm. Unfortunately, co-simulation strategies are still in an embryonic stage since they lack a unified modeling notation for test scenario definition, effective approaches to automatically generating scenarios, and methods to optimize scenario suite executions. MEDITATE aims to enhance co-simulation-based validation of ADAS by developing an integrated framework that supports the representation of scenarios at different abstraction levels, the automated generation of test scenarios, and an effective optimization strategy that filters scenarios based on their critical features.**

***Index Terms*—Advanced Driver Assistance Systems, ADAS, Testing, Co-simulation Environments**

I. Motivations and Challenges

Advanced Driver Assistance Systems (ADAS) have the potential to overcome complex and dangerous situations on the road environment and represent one of the most challenging scenarios for AI-enabled systems due to their dynamic, continuous, and stochastic nature. Kalra and Paddock [1] estimated that to ensure ADAS are safer than human drivers, millions to billion kilometers of driving tests would be needed, and each time a modification to ADAS occurs, the entire process should be executed over again. That is where Model-in-the-Loop (MiL) comes in as the de-facto standard solution in ADAS validation. It is particularly well suited for AI-based ADAS, which require complex inputs. Recent solutions propose the integration of different simulation tools, i.e., *co-simulation* [2], to obtain much more effective validations. The idea behind co-simulation involves combining a simulator for the ADAS model (e.g., Matlab/Simulink), a 3D environment (e.g., an engine-based simulator like CARLA or BeamNG), and a traffic simulator (e.g., SUMO). Although research on this topic has increased in recent years, implementing effective co-simulation scenarios still presents numerous challenges that need to be addressed.

However, there is no unified notation for modeling ADAS test scenarios. Although various languages have been proposed for ADAS modeling, only preliminary or simulator-specific languages have been developed [3]. The unification of modeling languages would promote interoperability between methodologies and tools used throughout the entire ADAS verification process; furthermore, it could enable developers to highlight certain aspects of scenarios that might be obscured if only a simulator-specific language were used.

A significant challenge developers face when dealing with ADAS testing is creating scenarios that expose critical behaviors without incurring excessive manual effort. Recently, tools like TARGET [4], LeGEND [5], and ChatScene [6] leverage Large Language Models to translate incident reports or traffic rules into structured and executable scenarios. Other proposed tools, like DriveSceneGen [7], exploiting data-driven generation or reinforcement learning, can simulate realisting scenarios and dynamic environments. Optimization-based approaches, like MOSAT [8], focus on incouraging critical scenarios, while discouraging unrealistic situations. Despite recent improvements, challenges persist in state-of-the-art approaches, generally lacking of integration into testing workflows and often requiring complex setups. Improving this aspect in terms of both search space exploration and the time required for generation is essential.

Finally, the execution of scenario-based testing in 3D simulation environments is costly, and the entire execution of the suite is completely unfeasible. Wachenfeld and Winner [9] demonstrated that the effort required to test scenarios can be drastically reduced by executing only *relevant cases*. Although Test Case Minimization, Selection, and Prioritization (TCM, TCS, TCP) are widely employed meta-heuristic software engineering practices to increase efficiency without losing the effectiveness of the test suite, they have been applied to the context of ADAS scenario testing just a few times. Recently, SPECTRE [10] applied optimization techniques to reduce the effort required to test ADAS, focusing on multi-objective algorithms. Although the results are promising, TCP/TCS for ADAS still requires in-depth experimentation. Furthermore, there is a significant lack of empirical comparisons between different techniques to optimize ADAS scenarios.

II. Project Objectives and Methodologies

The final aim of MEDITATE is to develop a framework in which domain expert developers need to define, through a high-level *Meta-Model-Based Visual Editor*, *functional/logical test scenarios* to validate ADAS. At this level, the output of the Meta-Model is fed to the *Automated Generation Module*, which generates the suite of *concrete* scenarios for a specific supported simulator (previously chosen and configured by the developer) or integrates the already existing suite with the new scenarios. Finally, each time a developer makes a change to the ADAS under test, MEDITATE will apply regression testing to ensure that the new version of the system does not introduce regression. In particular, since running the entire suite would be unfeasible, the *Test Suite Optimization Module*, using the results of previous suite executions as feedback, aims to identify the most effective subset of scenarios to run. The following paragraphs explain how each project objective will be addressed to achieve the final goal of the project.

a) Modeling Languages and Techniques for ADAS Testing Scenarios: To enable the specification of flexible and interoperable test scenarios, MEDITATE aims to define a comprehensive model-based notation tailored for AI-based ADAS validation. Rather than creating a new language from scratch, the approach builds upon existing standards (e.g., OpenSCENARIO), selectively extracting and extending essential constructs. Using state-of-the-art model-driven engineering tools, such as the Eclipse Modeling Framework, the notation will support both static elements (e.g., roads, infrastructure) and dynamic actors (e.g., vehicles, pedestrians). The resulting meta-model will enable scenario specification at multiple abstraction levels and facilitate transformations across notations, thereby fostering both horizontal and vertical interoperability in ADAS testing.

b) Automated Generation of Test Scenarios for ADAS: To enhance the effectiveness of co-simulation-based ADAS testing, MEDITATE develops advanced test scenario generation strategies that can produce diverse and safety-critical situations. Building on existing literature and cross-domain insights, the project explores both search-based techniques and Machine Learning. The generation process will rely on metrics tailored to assess criticality and ensure scenario diversity. The resulting methods will enable the automated generation of fault-revealing scenarios and be benchmarked against state-of-the-art approaches.

c) Optimization of Testing Scenarios for ADAS: To cope with time and resource constraints in ADAS testing, MEDITATE introduces optimization strategies to reduce the size of the test suite and prioritize the execution of scenarios. The approach investigates established techniques, such as greedy algorithms and meta-heuristics, adapting them to the co-simulation context. The emphasis is on selecting the most relevant tests and ordering them to reveal faults early. The project also explores similarity metrics to guide selection and enable both single- and multi-objective optimization. The results will include validated methods for efficient test execution and metrics tailored to assess the relevance and diversity of the scenario.

d) Integration and Validation of the Developed Solutions: MEDITATE aims to deliver an integrated framework that unifies the modeling, generation, and optimization of ADAS test scenarios. The framework will be validated on real ADAS models within state-of-the-art co-simulation environments. The outcome will be a unified methodology and toolset representing one of the first comprehensive solutions for automated and scalable ADAS scenario testing across multiple platforms.

Acknowledgment

This work has been partially supported by the European Union - Next Generation EU, Mission 4 Component 1 CUP E53C24002440006 and by the Italian PNRR MUR project PE0000013-FAIR.

References

[1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," *Transportation research part A: policy and practice*, vol. 94, pp. 182–193, 2016.

[2] X. Zhao, X. Liao, Z. Wang, G. Wu, M. Barth, K. Han, and P. Tiwari, "Co-simulation platform for modeling and evaluating connected and automated vehicles and human behavior in mixed traffic," *SAE International Journal of Connected and Automated Vehicles*, vol. 5, no. 12-05-04-0025, pp. 313–326, 2022.

[3] B. Schütt, T. Braun, S. Otten, and E. Sax, "Sceml: A graphical modeling framework for scenario-based testing of autonomous vehicles.," in *23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, pp. 114–120, 2020.

[4] Y. Deng, J. Yao, Z. Tu, X. Zheng, M. Zhang, and T. Zhang, "Target: Automated scenario generation from traffic rules for testing autonomous vehicles via validated llm-guided knowledge extraction," *arXiv*, 2023.

[5] S. Tang, Z. Zhang, J. Zhou, L. Lei, Y. Zhou, and Y. Xue, "Legend: A top-down approach to scenario generation of autonomous driving systems assisted by large language models.," *arXiv*, 2024.

[6] J. Zhang, C. Xu, and B. Li, "Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles," *arXiv*, 2024.

[7] S. S. et al., "DriveSceneGen: Generating diverse and realistic driving scenarios from scratch," *arXiv*, 2023.

[8] H. T. et al., "MOSAT: finding safety violations of autonomous driving systems using multi-objective genetic algorithm," in *30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 94–106, 2022.

[9] W. Wachenfeld and H. Winner, "The release of autonomous vehicles," in *Autonomous Driving: Technical, Legal and Social Aspects*, pp. 425–449, Springer, 2016.

[10] C. Lu, H. Zhang, T. Yue, and S. Ali, "Search-based selection and prioritization of test scenarios for autonomous driving systems," in *International Symposium on Search Based Software Engineering*, pp. 41–55, Springer, 2021.