https://doi.org/10.2352/EI.2022.34.16.AVM-110

Paving the way for certified performance: Quality assessment and rating of simulation solutions for ADAS and autonomous driving

Marius Dupuis, M. Dupuis Engineering Services, Bayrischzell, Germany, marius.dupuis@simcert.org

Abstract

Simulation plays a key role in the development of Advanced Driver Assist Systems (ADAS) and Autonomous Driving (AD) stacks. A growing number of simulation solutions addresses development, test, and validation of these systems at unprecedented scale and with a large variety of features. Transparency with respect to the fitness of features for a given task is often hard to come by, and sorting marketing claims from product performance facts is a challenge. New players – on users' and vendors' side – will lead to further diversification.

Evolving standards, regulatory requirements, verification and validation practices etc. will add to the list of criteria that might be relevant for identifying the best-fit solution for a given task. There is a need to evaluate and measure a solution's compliance with these criteria on the basis of objective test scenarios in order to quantitatively compare different simulation solutions. The goal shall be a standardized catalog of tests which simulation solutions have to undergo before they can be considered fit (or certified) for a certain use case.

Here, we propose a novel evaluation framework and detailed testing procedure as a first step towards quantifying simulation quality. We will illustrate the use of this method with results from an initial implementation, thereby highlighting the top-level properties Determinism, Real-time Capability, and Standards Compliance. We hope to raise awareness that simulation quality is not a nice-to-have feature but rather a central aspect for the whole spectrum of stakeholders, and that it needs to be quantified for the development of safe autonomous driving.

Introduction

System and environment simulation has been involved in developing and testing Advanced Driver Assist Systems (ADAS) and Autonomous Driving (AD) stacks for the best part of the past ten to fifteen years [1, 2]. The system-under-test (SuT) has evolved from a dedicated function based on clearly stated algorithms (e.g., Adaptive Cruise Control – ACC) to complex, overarching control processes based on Artificial Intelligence (AI), trained by Machine Learning (ML) technology (e.g., execution of unprotected left turn maneuvers).

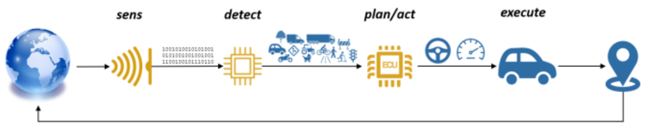


Figure 1: Generalized signal chain in ADAS/AD applications

The number of situations which have to be mastered by modern ADAS and AD systems can hardly be estimated. The range of miles required for proving that a system is operating safely has reached the order of billions [3] – although a distinction has to be made between the total number of miles which are statistically required for proving the safety of a system under normal operating conditions and the subset of relevant miles which comprise all situations a system is expected to handle safely.

Whatever the exact number of miles or situations, the order of magnitude alone and the potential risk involved in creating edge cases in the real world leave no other method but to supplement proving ground and road tests by simulation-based testing on a massive scale. Various research projects for enabling highly automated driving (HAD) have incorporated this principle into their methodologies [4, 5].

But whereas it is guaranteed that physics in real-world testing are correct per se, hardly any formal and commonly agreed processes exist today to prove that a stimulus originating from simulated components provides physically correct input to the SuT or that a specific simulation solution fulfills its intended purpose in general. Unless this problem is solved, an envisaged virtual homologation of vehicles equipped with ADAS and AD functions will not be possible [6].

The Goal

Simulation solutions come with a broad range of components, features, tooling, interfaces etc. For a function or system developer whose task is to ensure that the SuT performs as intended within its operational design domain (ODD), it might require in-depth knowledge of simulation technology in general and the implementation of individual features in particular to perform an assessment of a simulation solution's fitness for a given task.

Today, potential users execute their own selection process for simulation tools along individually created criteria lists in order to identify available solutions for the intended use case(s). But, as indicated, this process may have various shortcomings:

- users may not be simulation experts and may, therefore, not be in a position to map tools' features correctly to their use case
- up-to-date knowledge of the range and categories of available tools may be missing
- the process to come from a long list of candidates to a short list may include only a basic set of common criteria but may consume a substantial part of the selection process itself due to the number of candidates
- the list of selection criteria may be incomplete, and important checkpoints for the intended use case(s) may be missing

- the use case definition may be incorrect or too narrow and may disqualify tools that might be a better choice in the long run

Therefore, a method is proposed where experts with a market and technology overview regularly perform a basic assessment of available solutions along criteria derived from the most common use cases and provide the results to interested parties. These results may be considered the short list of candidates which a potential user might want to investigate further. Therefore, the user, an expert on the own use case, will not have to be an expert on simulation technology itself, and will still be in a position to make an informed choice:

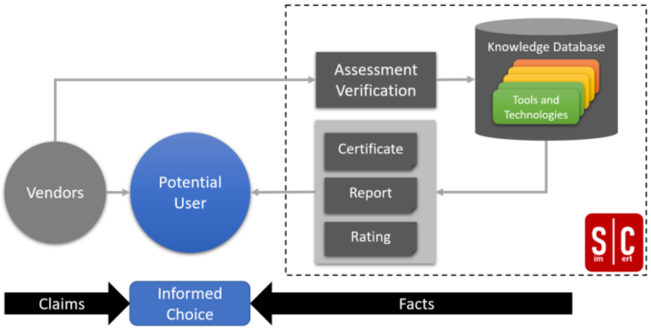


Figure 2: Enabling informed choices

Interacting With the System

In the process depicted in Fig. 2, the user shall be guided when querying the expert system. Starting with the ODD and use cases, adding solutions that are already in place on the user's premises, considering the user's preferences along an unbiased list of criteria, a short list of viable solutions shall be created. Commercial aspects, except for a general view on licensing terms, must not play a role in this process; they shall remain under the sole responsibility of bi-lateral negotiations between vendor and user.

The outcome of a tool assessment shall be easily understood by various persona (from management to developer) and shall provide a sufficient level of detail for a given purpose (broad market overview vs. specific task). As in many other industries, an entry-level five-star rating system complemented by further details for interested parties has proved to be a good way of communicating the findings:

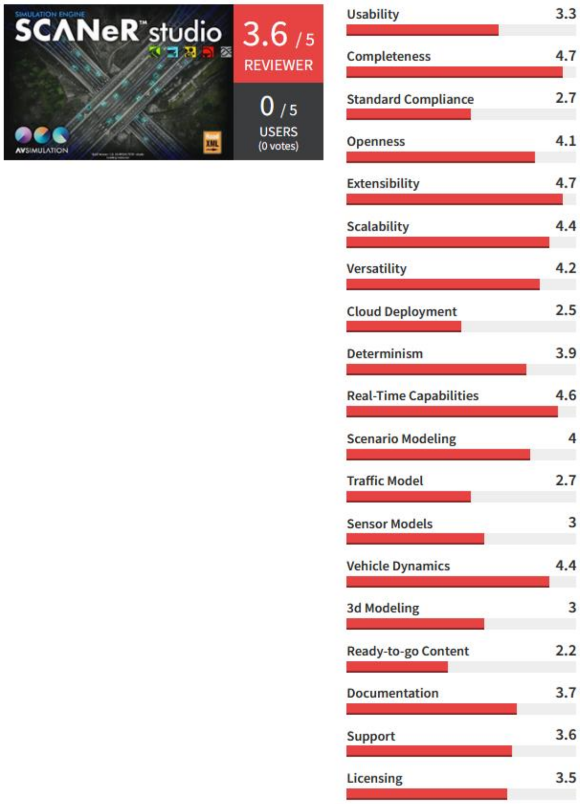


Figure 3: High-level rating results for an integrated solution [13]

It shall remain the user's responsibility to pick a tool that fulfills or comes close to fulfilling her/his individual requirements. The method presented here shall ensure that any decision "to build or buy" or "what to buy" is a well-informed decision; there shall be little room for surprise concerning "what to expect", and it shall also be clear which tools, combination of tools, or sequence of tools will pave the way along the user's development process.

Structured Assessment

As much as simulation tools for ADAS and AD applications might differ in the details, they all adhere to a basic structure:

> *A system under test (e.g., driving function) executes its commands on a vehicle dynamics model which carries the sensors perceiving a static environment complemented by dynamic entities (traffic participants) and time-variant conditions (weather).*

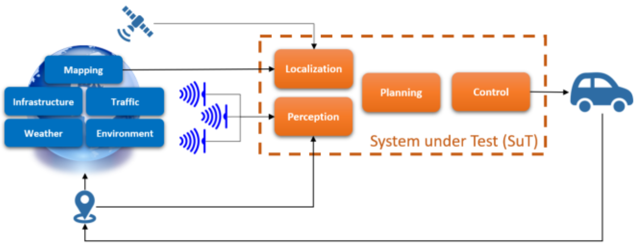


Figure 4: Basic structure of simulation solutions for ADAS/AD

This common structure allows for a classification of solutions along their integration level:

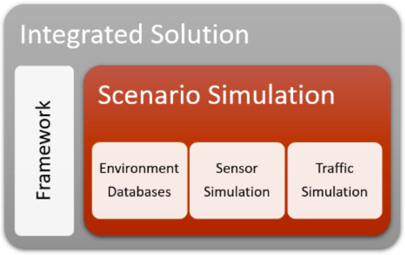


Figure 5: Tool classification

The classification paves the way to comparing solutions – either within the highest common level (e.g., Integrated Solution vs. Integrated Solution) or along a common denominator (e.g., Traffic Simulation of an Integrated Solution vs. a dedicated Traffic Simulation solution).

The actual assessment of a solution is performed along criteria which apply for a given solution class while also taking into account all classes it incorporates.

From a background of designing and integrating simulation solutions, we have compiled an initial, hierarchical list of criteria, currently encompassing 1063 check points on the lowest level, and aggregating into 19 top-level groups.

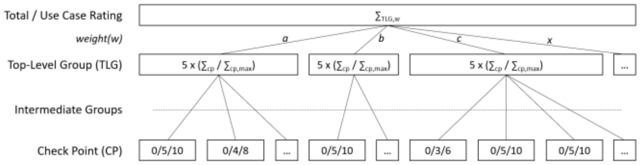
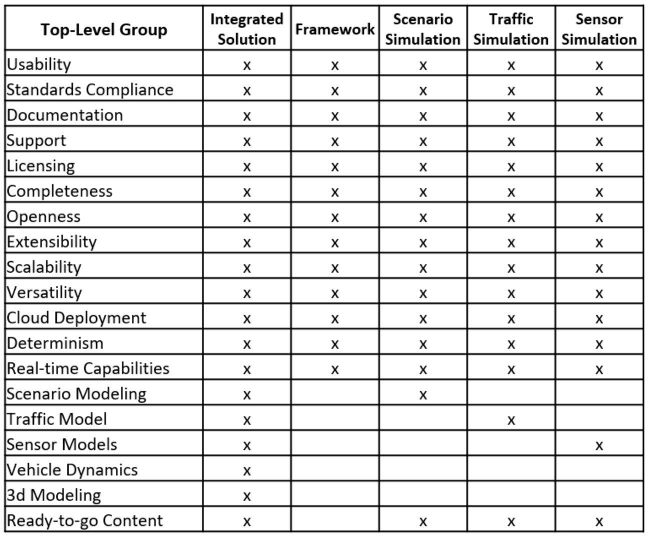


Figure 6: Rating hierarchy

Each single check point has a value between zero and up to ten (depending on its weight within its next-level group) and can be fulfilled either fully (maximum number of points), partially (half the number) or not at all (zero points). Points per top-level group are accumulated and normalized to a range between zero and five.

The total rating, for now, is the mean value of all top-level groups. This does not yet reflect the use-case centered approach described previously. For this to be implemented, an additional weighting of the top-level group results will be introduced in the future (see "Outlook" below).

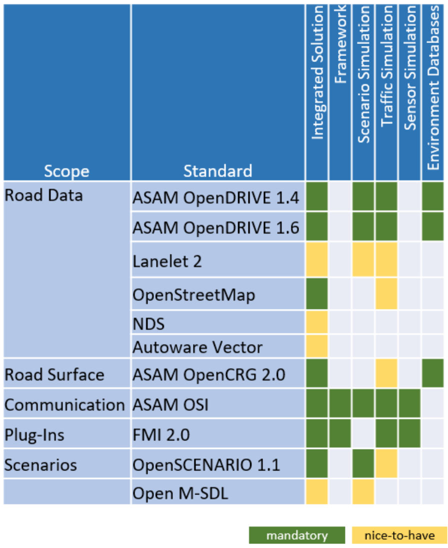
Top-level groups encompass the whole user experience, starting from rather soft aspects like "usability" via medium features like "openness" and "versatility" to hard facts like "real-time capability" and "determinism". Not all top-level groups are applicable to all solution classes:



Top-Level Group	Integrated Solution	Framework	Scenario Simulation	Traffic Simulation	Sensor Simulation
Usability	x	x	x	x	x
Standards Compliance	x	x	x	x	x
Documentation	x	x	x	x	x
Support	x	x	x	x	x
Licensing	x	x	x	x	x
Completeness	x	x	x	x	x
Openness	x	x	x	x	x
Extensibility	x	x	x	x	x
Scalability	x	x	x	x	x
Versatility	x	x	x	x	x
Cloud Deployment	x	x	x	x	x
Determinism	x	x	x	x	x
Real-time Capabilities	x	x	x	x	x
Scenario Modeling	x		x		
Traffic Model	x			x	
Sensor Models	x				x
Vehicle Dynamics	x				
3d Modeling	x				
Ready-to-go Content	x		x	x	x

Figure 7: Top-level groups vs. solution classes

Similar to the envisaged weighting of top-level groups in the rating of a solution's fitness for a certain use case, the low-level check points may also be weighted differently for different product classes. For the top-level group "Standards Compliance", for example, the detailed weighting is shown in the following figure:



Scope	Standard	Integrated Solution	Framework	Scenario Simulation	Traffic Simulation	Sensor Simulation	Environment Databases
Road Data	ASAM OpenDRIVE 1.4	🟩		🟩	🟩		🟩
	ASAM OpenDRIVE 1.6	🟩		🟩	🟩		🟩
	Lanelet 2	🟨		🟨	🟨		
	OpenStreetMap	🟩			🟨		
	NDS	🟨					
	Autoware Vector	🟨					
Road Surface	ASAM OpenCRG 2.0	🟩			🟨		🟩
Communication	ASAM OSI	🟩	🟩	🟩	🟩	🟩	
Plug-Ins	FMI 2.0	🟩	🟩		🟩	🟩	
Scenarios	OpenSCENARIO 1.1	🟩		🟩	🟨		
	Open M-SDL	🟨		🟨			

🟩 mandatory 🟨 nice-to-have

Figure 8: Standards relevance vs. solution classes

The Reference

It is crucial that solutions be benchmarked vs. an imaginary ideal solution whose characteristics allow it to fulfill a given use case with one hundred percent coverage of the applicable requirements. In the short and medium term this may lead to none of the existing solutions on the market scoring fully on each of the test criteria. But it avoids that the industry is tested vs. itself; instead, it is tested vs. what is needed from a user's and, at some point, regulator's point of view.

Quantifying Quality

The basis of an objective product assessment is that each criterion can be verified, and the level of fulfillment can be measured. The input to measurements across the whole spectrum of criteria may be retrieved from the following sources (in decreasing order of relevance):

- live test under real operating conditions
- product documentation
- feedback by the provider's product development team
- supporting material on public platforms
- marketing material

This may be complemented by reports of actual users who are given a means to provide their own, subjective ratings along the official rating.

The most reliable measurement method is the live test. What can be achieved here shall be illustrated for three top-level criteria groups: *determinism, real-time capability,* and *standards compliance*.

Determinism

Determinism is a key principle, especially in, but not limited to, software-in-the-loop (SiL) and model-in-the-loop (MiL) deployments. It requires, at minimum, that providing a system with an identical sequence of inputs leads to an identical sequence of output in each simulation run. Random behavior built into the system (e.g., for stochastic traffic around the SuT) is identical in each simulation run ("repeatable randomness").

Which inputs are provided in a test for determinism, depends on the solution that is being tested, but for an integrated solution, for example, one would choose a set of increasingly complex scenarios and configurations:

- SuT on a simple road with specific driver input (-> determinism of vehicle dynamics and driver model)
- SuT on a simple road with additional scripted road users (-> determinism of scripted scenario entities)
- SuT on a simple road with additional scripted and random road users (-> determinism of complex scenario entities and randomness)
- SuT equipped with sensors (-> determinism of sensor models)
- etc.

Each configuration is initialized and executed identically at minimum twice but preferably dozens of times while recording the states of all participants and the outputs of the sensors, for example, after each simulation step. Only a deterministic solution will provide identical recordings within the same configuration.

Tests for determinism are also repeated with different workloads. If implemented by the simulation engine, the simulation will also be paused and resumed randomly and/or executed step-by-step by an external controller.

Typically, a basic up-front analysis of a solution's software architecture and its scheduler API will be a good indication whether deterministic behavior may be expected. Only if a general mechanism exists that ensures that modules / functions are always executed in the same order, that data flowing from one component to another is fully received before being processed, and that a notion of a central simulation frame or simulation time exists, will a solution be capable of deterministic behavior. The tests will show to what degree it actually is.

Solutions which are highly modularized (or federated) in order to optimize for flexibility and performance, tend to be increasingly susceptible to breaking deterministic behavior for a lack of a clear execution sequence and guaranteed message transfer.

Real-time Capability

Real-time capability is mandatory for simulation systems which are designed to interact with actual SuT hardware (in hardware-in-the-loop – HiL, or vehicle-in-the-loop – ViL environments) or with humans (in driver-in-the-loop – DiL environments).

Key to testing for real-time capability is to follow the vendor's recommendation for a target system when setting up the test system. Using less capable systems or deviating from a recommended module layout may render the test meaningless.

Two key measurements will provide a basic indication of real-time capability: *elapsed simulation time* and *latency*

In a real-time system, the *elapsed simulation time* must comply with the actual progress of real-world time in each simulation frame. Frame drops must be noticed by the system, must be reported, and must be compensated for. By attaching a test module to the simulation system and recording simulation time vs. a real-world clock, deviations over longer simulation periods can be detected. Tests will typically run from a couple of minutes to hours (depending on the use case) and may, at the same time, also reveal other adverse effects like memory leaks etc.

Minimizing *latency* is extremely crucial for real-time systems. Measuring the real-world time from an input to an output along relevant paths (e.g., from steering input to camera sensor output) will reveal a system's fitness for a given use case. For DiL systems, for example, total latency from vehicle input to screen output has to be kept well below a certain threshold in order to avoid driver-induced oscillations (the tolerable latency itself may vary with the dynamics of the driving maneuvers and the experience of the simulator drivers).

Latency may also be expressed in simulation frames. This reflects a potential to adjust latency in the time domain by adapting the time step of the simulation frames. It also indicates to the user which components need more than one simulation frame to execute (or how long they have to wait before they are executed) and, therefore, will always react with a given latency to an input. Properly federated systems will trigger all modules within the same simulation frame and delays will result from inter-dependencies of modules or internal processing pipelines only.

For interactive systems (i.e., with a human in the loop), real-time performance might be more important than determinism since the human is, by nature, already to be considered a non-deterministic component. In this case, the system layout will indicate whether components may even run asynchronously, processing the latest inputs and computing their own output in real time.

Standards Compliance

Standards compliance, finally, is harder to measure than the two previous criteria. One reason is that some standards (e.g., ASAM OpenDRIVE [7]) tend to leave room for interpretation. Another is that a solution may not need all elements of a standard in order to fulfill its purpose. Therefore, testing for standards compliance has to be carried out with respect to the intended functionality.

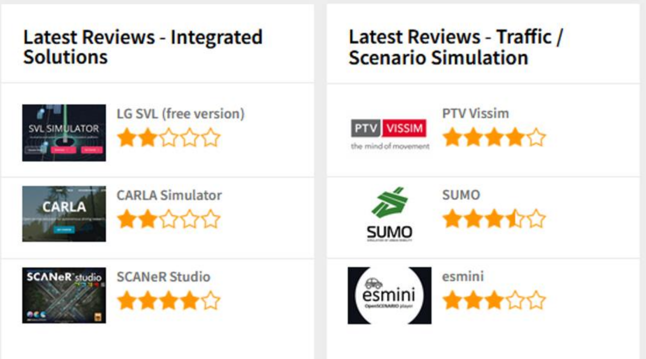
One key challenge when testing for standards compliance is a frequent lack of reference data or implementations that have been officially approved by the standardization body. Therefore, tests have to be carried out with publicly available data sets from trustworthy sources that have proved to perform as intended in at least one implementation which may be considered a reference. For data in ASAM OpenSCENARIO [8], for example, the open-source application esmini [9] is one such reference implementation.

Today, tests for standards compliance are carried out with a set of reference data and applications compiled by the testing team. Approved data sets and test tools by standardization bodies would be a great help, though, and would increase confidence in the respective test results.

One very straightforward and, sometimes, quite revealing test of a solution which claims compliance with a standard on its data import and export paths is to feed the result of an export back as an import. This is easy to achieve and, as said, very effective.

Results

Several tools have already undergone testing along the process laid out above. The high-level results have been published [10] and are available to everyone. Early tests involved open-source solutions, but also fully commercial solutions have already come on board.



Latest Reviews - Integrated Solutions		Latest Reviews - Traffic / Scenario Simulation	
LG SVL (free version)	★★☆☆☆	PTV Vissim	★★★★☆
CARLA Simulator	★★☆☆☆	SUMO	★★★⯪☆
SCANeR Studio	★★★★☆	esmini	★★★☆☆

Figure 9: Published test results [10]

Each product's results are also differentiated along the top-level groups laid out above and accompanied by the tester's remarks concerning pros and cons of a solution as well as a summary of the findings.

Below these high-level results, there are, of course, the details. Let's briefly illustrate some of them without referring to any specific solution that has undergone testing:

- Tests for determinism: these tend to come out consistently for the mere entities involved (e.g., vehicle dynamics, traffic participants) but they fail in many cases when it comes to sensor simulation. Setting up a camera sensor and pausing the simulation will, in most cases, result in rain continuing to fall or reflections of water surfaces continuing to move. This is a clear indication that, although entities are frozen, the graphics engines do not fully adhere to the notion of a central simulation time.
- Tests for documentation quality: small inconsistencies between the actual user interface and the documentation may happen at some point; but snapshots of user interfaces in the documentation that do no longer reflect the layout of the solution result in zero points for the applicable criterion.
- Tests for sensor modeling: the current catalog contains around 300 test criteria concerning sensor modeling. It is crucial, for example, that sensor field-of-view (FOV) definitions allow for the parameterization of the characteristics of the actual sensor technology. Hardly any tested tool provides, for example, means for defining the sensor frustum of an ultrasonic sensor which is not a simple cone shape. An implementation of cross echoes (a very significant property of ultrasonic sensors for parking applications) is also hardly found.

Summary and Outlook

Assessments carried out so far have shown that solutions may well be compared along the classification laid out above.

Explicit endorsement of test results by commercial providers (e.g., AVSimulation for SCANeR studio and PTV Group for PTV Vissim), help create trust in the testing methods and in the Sim|Cert initiative itself.

Even if tools come out with similar total scores, their individual ratings along the criteria groups may differ considerably.

By having laid out test criteria along an imaginary "ideal" tool, achieving a full score across all criteria groups is hardly feasible for a single product today. This leaves enough room to distinguish tools along criteria groups and, thus, their fitness for a specific use case.

The assessment system itself is "work in progress" and further rating mechanisms are in the making. A weighting of criteria along applications for ADAS and AD (e.g., planning and control, perception, data for machine learning) will lead to further diversification of the results.

The assessment system will incorporate additional commonly accepted quantifiers of simulation quality as they get identified by the respective interested parties. A permanent review of the criteria catalog shall ensure that the assessment stays ahead of the implemented technology.

Wherever other initiatives or institutions carry out trustworthy and commonly accepted tests of sub-components of the solutions under investigation here, passing these tests may even become a criterion which needs to be fulfilled in the proposed method. And it will definitely help to avoid reinventing the wheel.

With solutions incorporating more relevant standards, testing will be facilitated since unified data sets and test tools may be used.

One clear request in this respect is that standardization organization not only provide documents but also publicly available examples and tooling to certify that a data set or data stream complies with their requirements. Clarity on standards implementation and application is the key to quantifiable quality.

The Remaining Question

Finally, the question remains whether systems designed, trained, and tested with simulated data are at least as safe as systems that used only real data. This question reaches far beyond what can be achieved by assessing simulation tools. But several assumptions can be made:

- the principle of coverage-driven verification [11], mandates that tests be carried out in simulation in order to cover the event space by automated testing

- the event space does not only relate to entities participating actively in a scenario but also to infrastructure, weather, communication etc. [12] and the respective permutations
- real-world footage of edge cases (e.g., critical or fatal incidents) is not complete and/or may not be created per request

"Nothing beats reality" and many simulation tools still struggle to create synthetic data that can be processed correctly. The initiative and methods presented here are aiming for narrowing the gap between the real and the simulated world by asking the right questions and quantifying simulation solutions' answers.

References

[1] Bock, T.; Maurer, M.; Färber, G.: Vehicle in the Loop (VIL) — A New Simulator Set-up for Testing Advanced Driving Assistance Systems. Driving Simulation Conference, Iowa City, IA (USA), 2007

[2] von Neumann-Cosel, Kilian, „Simulation umfeldbasierter Fahrzeugfunktionen", PhD Thesis, 2014, http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20140206-1126934-0-2

[3] Nidhi Kalra, S.M.P.: Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability? RAND Corporation (2016). http://www.jstor.org/stable/10.7249/j.ctt1btc0xw

[4] PEGASUS Project Office, „The PEGASUS Method", available at https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf, 2019

[5] ENABLE-S3 research project, „Testing & Validation of Highly Automated Systems – Summary of Results", 2019, https://drive.google.com/file/d/15c1Oe69dpvW5dma8-uS8hev17x-6V3zU/view?usp=sharing

[6] H. Abdellatif, C. Gnandt, "Use of Simulation for the Homologation of Automated Driving Functions", in: ATZ electronics worldwide, (2019), No. 12, p. 60-63

[7] ASAM e.V., ASAM OpenDRIVE, at: https://www.asam.net/standards/detail/opendrive, 2021

[8] ASAM e.V., ASAM OpenSCENARIO, at: https://www.asam.net/standards/detail/openscenario, 2021

[9] E. Knabe e.a., esmini – a basic OpenSCENARIO player, at: https://github.com/esmini/esmini, 2021

[10] M. Dupuis, Sim|Cert website, at: https://www.simcert.org, 2021

[11] Coverage-Driven Verification. In: Functional Verification Coverage Measurement and Analysis. Springer, Boston, MA. https://doi.org/10.1007/978-1-4020-8026-5_7, 2008

[12] M. Scholtes, L. Westhofen, L. Turner, K. Lotto, M. Schuldes, H. Weber, N. Wagener, C. Neurohr, M. Bollmann, F. Kortke, J. Hiller, M. Hoss, J. Bock, L. Eckstein, "6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment". 2021. IEEE Access. PP. 1-1. https://ieeexplore.ieee.org/abstract/document/9400833

[13] Sim|Cert rating, at: https://www.simcert.org/integrated-solutions/#SCANeR, 2021

Author Biography

Marius Dupuis received his Dipl.-Ing. degree in aerospace engineering from the University of Stuttgart (1994). After a first engagement in helicopter flight simulation, he co-founded VIRES Simulationstechnologie GmbH and worked for it until early 2020. Since then, he is working as independent consultant and started the Sim|Cert initiative in early 2021.