IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 8, 2021, accepted January 21, 2022, date of publication February 7, 2022, date of current version February 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3149542

A Taxonomy for Quality in Simulation-Based Development and Testing of Automated Driving Systems

BARBARA SCHÜTT[1], MARKUS STEIMLE[2], BIRTE KRAMER[3],
DANNY BEHNECKE[4], AND ERIC SAX[1]
[1]Embedded Systems and Sensors Engineering, FZI Research Center for Information Technology, 76131 Karlsruhe, Germany
[2]Institute of Control Engineering, Technische Universität Braunschweig, 38106 Braunschweig, Germany
[3]Institute of Systems Engineering for Future Mobility, German Aerospace Center, 26129 Oldenburg, Germany
[4]Institute of Transportation Systems, German Aerospace Center, 38108 Braunschweig, Germany

Corresponding author: Barbara Schütt (schuett@fzi.de)

This work was supported by the German Federal Ministry for Economic Affairs and Energy within the Project "Simulationsbasiertes Entwickeln und Testen von automatisiertem Fahren (SET Level)-Simulation-Based Development and Testing of Automated Driving," a Successor Project to the Project "Projekt zur Etablierung von generell akzeptierten Gütekriterien, Werkzeugen und Methoden sowie Szenarien und Situationen zur Freigabe hochautomatisierter Fahrfunktionen (PEGASUS)" and a Project in the PEGASUS Family.

ABSTRACT Ensuring the quality of automated driving systems is a major challenge the automotive industry faces. In this context, quality defines the degree to which an object meets expectations and requirements. Especially, highly automated and automated vehicles at SAE levels 4 and 5 will be expected to operate safely in various contexts and complex situations without misconduct. Thus, a systematic approach is needed to show their safe operation. A way to address this challenge is simulation-based testing, as pure physical testing is not feasible for automated driving at levels 4 and 5 since several billion kilometers of driving are necessary. During simulation-based testing, the data used to evaluate the actual quality of an automated driving system are generated using a simulation. However, to rely on these simulation data, the overall simulation, including its simulation models, must provide a certain quality level. This quality level depends on the intended purpose for which the generated simulation data should be used. Therefore, three categories of quality can be considered: simulation quality (e.g., reliable simulation tool), quality of the automated driving system (e.g., handling of dangerous situations), and scenario quality (e.g., scenario criticality). Quality must be determined and evaluated in various process steps in developing and testing automated driving systems, the overall simulation, and the simulation models used for the simulation. In this paper, we propose a conceptual taxonomy to better understand quality in the development and testing process to have a clear separation and insight where further testing is needed – both in terms of automated driving systems and simulation, including their simulation models and scenarios used for testing.

INDEX TERMS Automated driving, criticality, metrics, PEGASUS family, quality, scenario, scenario-based testing, simulation, taxonomy, validation, verification.

I. INTRODUCTION

Future mobility systems will face many challenges as the growing urbanization may bring transportation systems to their limits [1]. Thus, problems we are already facing today, e.g., traffic jams or accidents, will likely become worse. Assisted and automated driving systems have the potential to meet these challenges by promising more mobility for everyone, driving more efficiently, environmentally friendly, and safely. However, it also leads to growing system complexity and interconnection of automotive features [2]. Above all, safety will be a crucial issue for introducing and accepting these systems in society. Thus, ensuring and validating the vehicles' safe behavior is essential. For this purpose, the vehicle must be thoroughly tested at various test levels to assure requirements are met, the system has necessary capabilities in all intended use cases, and unreasonable risk is avoided [3, p. 6].

The associate editor coordinating the review of this manuscript and approving it for publication was Divanilson Rodrigo Campelo.

Presently, real-world test drives are used to validate the safe behavior of vehicles equipped with assisted driving systems. These test results state their quality, ''the degree of excellence'' [4]. According to Wachenfeld and Winner [5], the distance-based test approach is no longer feasible at a certain degree of automation, e.g., SAE level 4 (high driving automation and no reliable human fallback) and level 5 (full driving automation under all driver-manageable road conditions) [6], since more than six billion test kilometers would be necessary to ensure the vehicles' safe behavior unless a very simplistic operational design domain is considered. Additionally, if there are changes or variations in the automated driving system, all testing must be repeated.

Scenario-based test approaches promise an alternative or supplemental test method, especially combined with a simulation-based approach. Compared to the random scenarios emerging during a real-world test drive, in scenario-based testing, new and relevant scenarios are systematically derived and tested at different process steps during the development and test process [7], [8]. These approaches aim to create a collection of relevant scenarios, depending on the test object, the test objectives, and the test object's preceding requirements, and can be executed in a simulation-based or real-world test. Simulation-based tests are significantly cheaper than real-world test. It has to be ensured that the system under test (SUT) meets its specifications, but further, all used simulation models and tools must possess a certain quality level to generate sufficiently valid simulation results [9]. We propose three categories of quality that can be distinguished: the quality of the automated vehicle and the quality of the simulation, consisting of simulation models as the simulation environment and scenarios as the schedule of events. The system under test depends on the category and test level considered. Within the first category, the system under test can be, for example, the overall automated vehicle or a system, component, or a unit of the automated vehicle like the automated driving system or a path planning algorithm. Within the second and third categories, the system under test can be, for example, the overall simulation, coupled simulation models, a single simulation model, or a scenario. All categories have to be considered when simulation-based approaches are used.

While the proposed approach might be applicable for other areas of safety-critical simulation, e.g., flight simulation, it mainly focuses on automotive simulation and its evaluation. However, the generalized parts of our work, e.g. terminology, may apply to other domains. This comparison between domains was not the focus of this work but might be considered in future work.

A. NOVELTY AND MAIN CONTRIBUTION TO THE STATE OF THE ART

This paper's novelty and main contribution is a conceptional taxonomy for a systematical classification of quality during the simulation in the development and test process of automated driving systems. Therefore, we

- define of relevant quality-related terms to avoid ambiguity,
- divide the simulation quality into three main categories that may occur during scenario-based testing and simulation: the quality regarding the simulation model or tool, the quality of a vehicle or components of it, and the quality of scenarios for testing, and
- propose an expandable taxonomy and terminology for all three categories and levels of resolution, to serve a categorization of quality metrics and aspects and to make it easier to communicate the area of interest within a simulation setup. The taxonomy uses classification to help scientists imminently understand and organize the differences of all areas of quality in the field of automotive simulation.

B. STRUCTURE

In section II, scenario-based testing and different quality metrics are described. Section III defines relevant quality-related terms. Section IV introduces the proposed taxonomy for quality throughout simulation-based testing and gives a simulation example that shows that each category and resolution level plays a role during the complete development and test process. Additionally, examples from literature to show the taxonomy's role in the existing research were collected. Finally, section V gives a short conclusion.

II. RELATED WORK

In this section, we summarize scenario-based testing, give an introduction to traffic simulation abstraction levels and quality metrics.

A. SIMULATION- AND SCENARIO-BASED TESTING

Quality assurance is an essential part of the development process. Therefore, it shall be integrated into the development process as early as possible. The V-model visualizes an established development approval [10] and is shown in Fig. 1. The left part of the V describes the top-down design process after the concept phase. The right part describes a bottom-up test process that includes verification and validation activities. This process can be used to develop of a simulation model, the complete simulation tool, vehicle components, or the complete vehicle.

Depending on the proportion of simulated and real elements, the corresponding test method can be called, for example, Hardware-in-the-loop (HiL), Software-in-the-Loop (SiL), Vehicle-in-the-Loop (VeHiL) [5].

As stated by Wood *et al.* [3, pp. 83], scenario-based testing is a suitable approach to supplementing the distance-based approach of real-world driving, thus reducing the necessary mileage. The scenario-based approach includes the techniques and strategies during the test process listed below to gain information and make statements about the quality of a system under test:

- decomposing the system and individual testing of system elements,

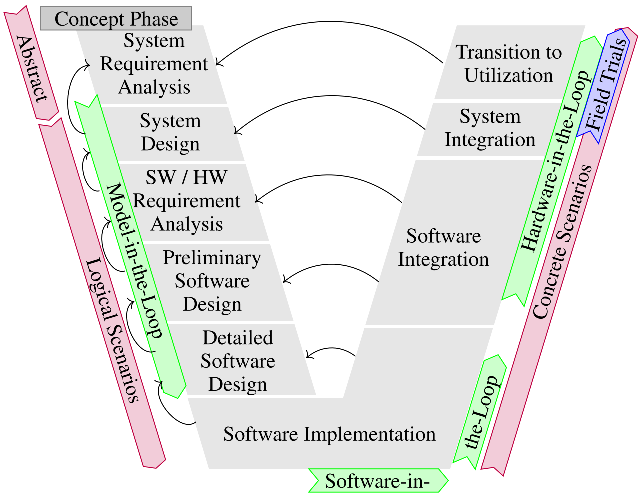


FIGURE 1. **V-model with different X-in-the-loop phases [10] combined with abstraction levels of scenarios at different development stages, adapted from Bock *et al.* [11].**

- combining different platforms and design techniques (e.g., variable variation or stochastic variation for increasing test coverage),
- defining scenarios by using expert knowledge about interesting situations or automatic extraction of traffic data, and
- defining surrogate metrics (e.g., crash potential of a given situation) and performance measures.

The *Operational Design Domain (ODD)* for an automated driving function has to be stated to choose scenarios for testing. According to SAE [6], the ODD defines conditions under which an automated driving system or feature is intended to function. It determines *where* (e.g., environmental or geographical characteristics) and *when* (e.g., time-of-day restrictions) an automated driving system must act reliably.

Ulbrich *et al.* [12] defines a scenario as the temporal development of an initial scene and its participating static and dynamic actors, similar to a storyline. According to Bach *et al.* [13], scenarios can be divided into movie-related acts and use abstract propositional and temporal descriptions. Consistency checks can be utilized to generate derivations of these scenarios to create a database with a collection of scenarios. This movie-related storyboard approach is also taken up by OpenSCENARIO, an emerging scenario description standard [14].

Menzel *et al.* [8] suggest three abstraction levels for scenarios: functional, logical, and concrete. Scenarios are developed at an abstract level during the concept phase [15] and get detailed and concretized throughout the development and test process [8]. According to Menzel *et al.* [8], the most abstract level of scenario representation is called *functional* and describes a scenario via linguistic notation using natural, non-structured language terminology. The main goal for this level is to create scenarios that are easily understandable and open for discussion. Bock *et al.* [11] propose a supplementary abstraction level, called abstract scenarios, defined by a controlled natural language format that is machine-interpretable, and an exemplary assignment of scenario abstraction levels to the V-model is shown in Fig. 1. The next abstraction level is the *logical* level and refines the representation of functional scenarios with the help of parameters. The most detailed level is called *concrete*. It describes operating scenarios with concrete values for each parameter in the parameter space. These systematics means that one logical scenario can yield many concrete scenarios, depending on the number of variables, range size, and their step size.

Scenarios can be defined by scenario description languages, e.g., SceML [16] or OpenSCENARIO [14], which goes hand in hand with the map format OpenDRIVE [17] and the road surface description OpenCRG [18]. Another related term is *test case*. According to Steimle *et al.* [19], in scenario-based test approaches, a test case consists of at least a (concrete) scenario and evaluation criteria.

The automotive domain has brought forth a multitude of simulation software. The following tools are currently relevant for this work. However, we do not warrant its completeness. Commercial tools for automotive simulation are available from Vires VTD [20], dSPACE [21], and IPG [22]. All three simulation tools provide modules for map and scenario creation sensor and dynamic models, to name some examples. A further example tool is Carla, an open-source simulator with a growing community and based on the game engine Unreal [23]. It offers several additional modules, e.g., a scenario tool that includes its scenario format and support for OpenSCENARIO, a graphical tool for creating scenarios, a ROS-bridge, and SUMO support. SUMO is an open-source software tool for modeling microscopic traffic simulation from DLR [24]. It specializes in big-scale traffic simulation and can be used to evaluate traffic lights cycles, evaluation of emissions (noise, pollutants), traffic forecast, and many others. Other tools worth mentioning are: openPASS [25], PTV Vissim [26], or esmini [27], an OpenSCENARIO player.

B. SIMULATION PROCESS

The IEEE Std 1730-2010 *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)* [28] defines processes and procedures to develop and execute distributed simulations. It refers to distributed simulation engineering, and many automotive simulation tools and XiL-applications lie within its field. It states seven main steps for simulation engineering:

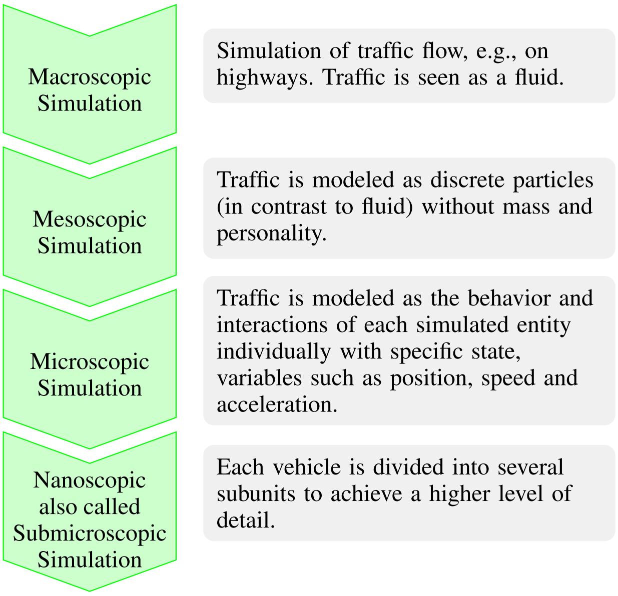
1 Define simulation environment objectives: Define and document the problem space addressed with the simulation environment.
2 Perform conceptual analysis: Developing a real-world representation regarding its problem space and scenario, requirement specification.
3 Design simulation environment: Select and design member applications and models.

4 Develop simulation environment: Implement member applications, models, and coupling methods.
5 Integrate and test simulation environment: Integrate member applications and test simulation environment.
6 Execute simulation: Execute planned simulation and document execution problems.
7 Analyze data and evaluate results: Analyze how well requirements are met and test criteria are fulfilled.

Furthermore, Durak *et al.* [29] propose a simulation qualification level (SQL) that states how strictly a simulation environment in the sense of reliability and credibility is evaluated. Their idea concerns flight simulation, however, translates to the field of automotive simulation and reminds of the tool confidence level of ISO 26262 [30].

C. ABSTRACTION LEVELS OF TRAFFIC SIMULATION

Traffic simulation can generally be divided into different abstraction levels regarding the depth of resolution: nanoscopic, microscopic, mesoscopic, and macroscopic [31]–[33], as shown in Fig. 2. In macroscopic traffic simulation, the traffic is modeled as fluid and is used to evaluate traffic flows or congestion in heavy traffic situations. The following resolution level is mesoscopic traffic simulation. Every participant is modeled as a discrete particle whose position lacks personality, such as mass or size. This lacking personal information is added at the microscopic level. At this level of resolution, each participant has its own modeled behavior with an individual state and variables, such as mass, speed, and acceleration. Additionally, individual maneuvers relevant for specific scenarios are modeled. An example for microscopic traffic simulation is the SUMO framework [24]. The next abstraction level in traffic simulation is nanoscopic (sometimes called submicroscopic) and views each vehicle as a composition of different subunits that need to be coupled to achieve a higher level of detail. Scenario-based testing often occurs in microscopic and nanoscopic simulations since the main goal is to evaluate (sub)units and their individual behavior in given scenarios.



Simulation of traffic flow, e.g., on highways. Traffic is seen as a fluid.

Traffic is modeled as discrete particles (in contrast to fluid) without mass and personality.

Traffic is modeled as the behavior and interactions of each simulated entity individually with specific state, variables such as position, speed and acceleration.

Each vehicle is divided into several subunits to achieve a higher level of detail.

FIGURE 2. Different levels of resolution in traffic simulation [31], [32].

D. QUALITY METRICS

To define suitable metrics for evaluation, it must first be determined what needs to be tested, i.e., what is the system under test and its requirements, and which aspects should be considered in the corresponding test cases [7]. In a less formal phase, e.g., prototyping a proof-of-concept for a simulation model, at least goals for the intended functionality have to be known. From these requirements or goals, quality metrics can be derived and are essential parts of a test case [19] to determine and quantify the quality of the intended functionality.

Before assessing the system under test using data generated by simulation, the quality of the overall simulation and its individual simulation models must be assessed and ensured, e.g., tool qualification [30]. Viehof and Winner [34] introduced a method for objective quality assessment of simulation models by statistical validation, where a simulation model and its parameterization are validated separately. This method has already been used successfully for vehicle dynamics simulation models and adapted for sensor perception simulation models by Rosenberger *et al.* [35]. Furthermore, Riedmaier *et al.* [36] present a unified framework and survey for simulation model verification, validation, and uncertainty quantification. More challenges have to be faced through the coupling and execution of the simulation models. To our knowledge, there are no established or widely recognized verification or validation methods for simulation models and their coupling mechanisms.

Metrics to evaluate driver behavior or driving functions are standard, and there exists a long list of different possibilities [37]. Well-known metrics are surrogate safety measures to analyze the conflict potential or severity of microscopic traffic scenarios [38]. Example metrics are the calculation of the time-to-collision (TTC) [39], post-encroachment time (PET) [40], and Gap Time (GT) [40]. These metrics are well-known criticality metrics and easy to compute. However, they are only recommended in certain use cases, e.g., TTC was developed for car-following scenarios and PET for intersection scenarios. A further evaluation of possible scenario types for these and other metrics is given by Westhofen *et al.* [37].

The evaluation of scenario quality depends on the aspects of a scenario that are important for further test cases and scenarios. Abeysirigoonawardena *et al.* [41] use a distance-based metric for two traffic participants to find new scenarios, whereas Hallerbach *et al.* [42] consider the current traffic situation, e.g., traffic density, to make a statement about a highway scenario. Junietz [43] proposes a metric to evaluate the macroscopic accident risk, the average risk of fatal accidents, and the microscopic risk, describing the accident risk in a single scenario.

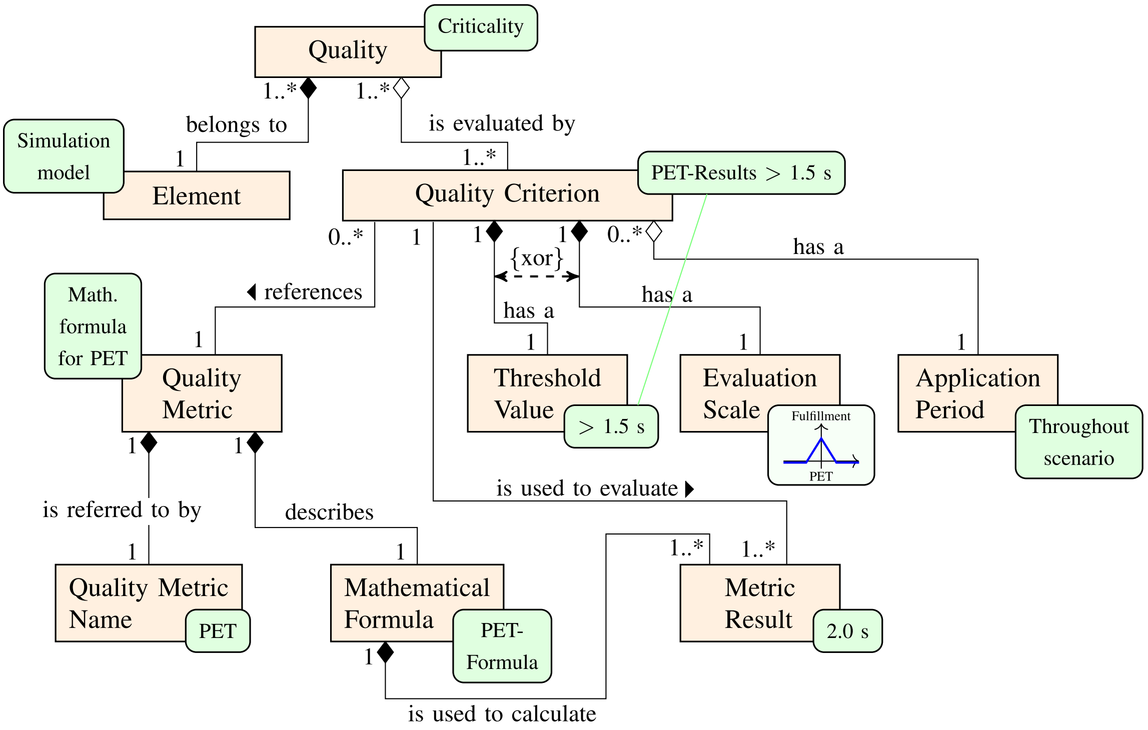


FIGURE 3. Relationship between relevant terms related to quality aspects. An example for each term is given in the assigned green boxes (PET: post-encroachment-time), based on [19].

III. TERMS AND DEFINITIONS

In this section, terms related to quality aspects that are relevant for this paper are explained. Fig. 3 shows these terms and their relationships as a UML diagram. Additionally, an example is given for each term (green boxes). In the following paragraphs these terms are described.

According to the Cambridge Dictionary [4], **Quality** is (1) "the degree of excellence of something, often a high degree of it." Hence, quality always belongs to something we call element, e.g., simulation model or system under test. An **element** may be, e.g., a simulation or an automated driving function. This quantified quality can be set in relation to other quantified qualities. Quality is evaluated by one or more quality criteria. This work distinguishes between different quality categories within the domain of automated driving, which are further explained in section IV.

The following terms and their descriptions are cited from Steimle *et al.* [19]. In the original text, Steimle *et al.* [19] use the more general term "evaluation." Since these descriptions refers to a quality, the term "evaluation" is replaced by "quality." Additionally, we highlighted (in bold) the terms shown in Fig. 3.

"[A **quality**] **criterion** is used to evaluate one or more metric results in relation to a threshold value or an evaluation scale within a specified application period. These metric results are calculated using a mathematical formula (described by [a quality] metric) and data generated during test case execution [. . .]. Thus, [a quality] criterion references [a quality] metric, it has a threshold value or an evaluation scale, and it has an application period." [19] An example quality criterion is the evaluation of post-encroachment-time calculated for a scenario execution and comparing to a threshold, e.g., a scenario is considered non-critical if post-encroachment-time exceeds 1.5 s [40].

"[A **quality**] **metric** is referred to by [a quality] metric name and describes a mathematical formula. This formula is used to calculate one or more metric results based on data generated during test case execution [. . .]. Examples of [quality] metrics related to automated driving are the metric named time-to-collision (TTC) and the metric named post-encroachment-time (PET) (each including the associated mathematical formula)." [19]

"[A **quality**] **metric name** (e.g., time-to-collision (TTC) or post-encroachment-time (PET)) refers to a specific [quality] metric used to calculate one or more associated metric results." [19]

"A **mathematical formula** (described by [a quality] metric) is a calculation rule used to convert input values (generated during test case execution) at a specific point in time into an output value (called metric result) that can be used for test case evaluation." [19] Post-encroachment-time uses the formula $PET = t_2 - t_1$, where t_1 denotes the time when actor 1 leaves the designated encroachment area and t_2 the time where actor 2 enters this area. Thus, post-encroachment-time is the distance in time between two actors passing the same area of interest.

"A **metric result** is calculated using a mathematical formula (described by [a quality] metric) and data [...] generated during test case execution [...]. A metric result is calculated at a certain point in time and consists of a number and a unit. The calculated metric results are evaluated according to the corresponding [quality] criteria." [19] A possible PET result could be 3 s, which means actor 1 leaves the area of interest 3 s before actor 2 enters it.

The metric results can be evaluated using two different methods, which usually exclude each other: using a threshold value or an evaluation scale.

"A **threshold value** is a fixed number (with a unit) used to test the compliance of calculated metric results with this fixed number according to the [quality] criterion. Therefore, only a statement is possible regarding whether the [quality] criterion is fulfilled or not." [19]

Allen *et al.* [40] propose 1.5 s as threshold for post-encroachment-time, where the result can be seen as critical if it falls below 1.5 s.

"An **evaluation scale** is a scale used to evaluate the adherence of calculated metric results with this scale according to the [quality] criterion. Therefore, it is also possible to make a statement about how well the [quality] criterion is fulfilled." [19] Another interpretation of post-encroachment-time could include the severity of the potential critical situation that can be coupled to the distance in time and results in more severe situations when the time interval between two actors gets smaller.

"An **application period** defines the periods in which the corresponding [quality] criterion is applied. The application period is defined by one or more conditions that are linked with AND and/or OR operators. When the linked conditions are fulfilled, the application of the [quality] criterion starts. Its application continues until the linked conditions are no longer fulfilled, a specified time has elapsed, or a specified event has occurred." [19] A possible application period for post-encroachment-time is throughout the complete scenario.

A wide spread term in the automotive area is criticality or criticality metric. From our point of view, (low) criticality is a subcategory of quality. Therefore, when we mention quality criteria and quality metrics, they also include criticality criteria and criticality metrics.

IV. TAXONOMY FOR SIMULATION QUALITY

This section explains our taxonomy with its domains of interest and resolution levels in simulation quality. In addition to traffic resolution, we propose three domains of interest for simulation quality (columns): simulation quality, system under test quality, and scenario quality. All domains can be examined in different levels of resolution, as it is possible for traffic simulation (rows): nanoscopic, microscopic, and macroscopic. New rows, columns, or single entries can be added to our taxonomy if another resolution level or domain of interest is needed. Further, this section states the role of all domains and their different resolutions during the development and test process. Fig. 4 shows all proposed combinations of domains of interest and resolution levels in a table, whereas Fig. 5 depicts the base structure of quality interaction. The bracketed numbers in the text match those in Fig. 4 and Fig. 5 and show which entry they belong to. The last column in Fig. 4 also shows the established resolution levels of traffic simulation as an entry point.

A. SIMULATION EXAMPLE

In the following description of the taxonomy, we use a simplified example for describing each domain of interest and its resolution levels:

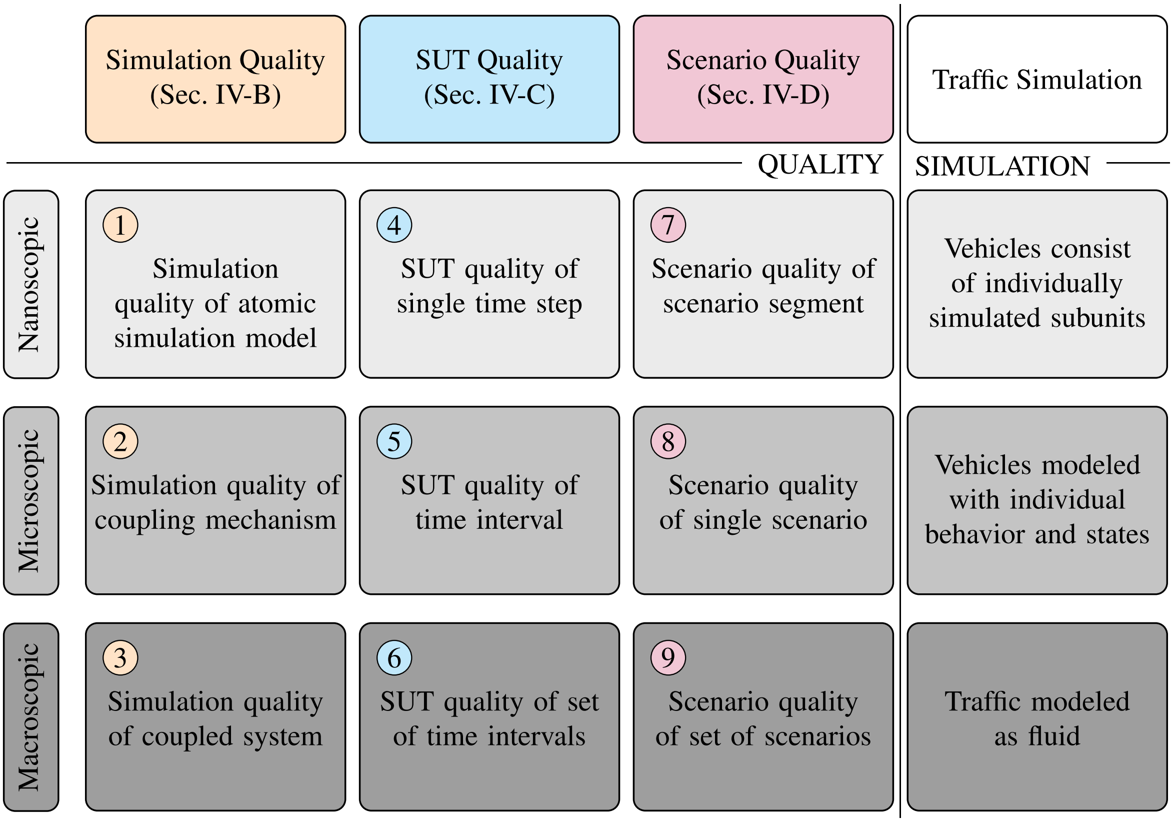
> A highly automated driving function to avoid collisions at urban intersections without traffic signs or lights shall be tested with simulation-based methods. Therefore, the driving function is the system under test. Its input is an object list that is available after sensor fusion. The output is a target trajectory with additional information about target speed and acceleration for several time steps in the future. A simulation environment shall be used for evaluation.

An operational design domain for the example function under the definition of operational design domain from SAE [6] is defined: *The driving function is designed to operate at an urban intersection at daylight and at speeds not to exceed* $58\,\mathrm{km\,h^{-1}}$. Additionally to the ego vehicle, other traffic participants, e.g., pedestrians or opponent vehicles, are needed to test collision avoidance. We used dSPACE Motion Desk [21] as a simulation tool for our example scenarios. Although we used a simplified example to understand the proposed concept better, it does not mean the taxonomy concept cannot be applied to other scenarios and use cases or more complex situations. All quality metrics in the following text were used as examples. It is possible to use quality metrics in more than one category. Additionally, thresholds can be changed, and additional metrics might be required to evaluate the use case's requirements properly.

B. SIMULATION QUALITY

Simulation quality refers to a simulation model's quality or is presumed to have. The simulation model or their couplings are the elements from which the qualities are quantified. Before a model can be used, it must be ensured that it approximates its real-world equivalent or functionality sufficiently in the relevant aspects.

According to Balci [44], the process of ascertaining the simulation model quality consists of two parts: model verification and model validation. Model verification evaluates the accuracy of transforming a problem formulation into a model, i.e., building the simulation model *right*. Model validation checks whether the simulation model is sufficiently valid for its intended purpose, i.e., building the *right* simulation model. Regarding verification and validation, model testing is the process of finding errors or inaccuracies within a simulation model. Hence, concerning verification and validation, the



	Simulation Quality (Sec. IV-B)	SUT Quality (Sec. IV-C)	Scenario Quality (Sec. IV-D)	Traffic Simulation
	QUALITY			SIMULATION
Nanoscopic	① Simulation quality of atomic simulation model	④ SUT quality of single time step	⑦ Scenario quality of scenario segment	Vehicles consist of individually simulated subunits
Microscopic	② Simulation quality of coupling mechanism	⑤ SUT quality of time interval	⑧ Scenario quality of single scenario	Vehicles modeled with individual behavior and states
Macroscopic	③ Simulation quality of coupled system	⑥ SUT quality of set of time intervals	⑨ Scenario quality of set of scenarios	Traffic modeled as fluid

FIGURE 4. Quality matrix with three proposed quality categories and resolution levels: simulation quality, system under test (SUT) quality, scenario quality. The category of traffic simulation serves as entry point for this presentation.

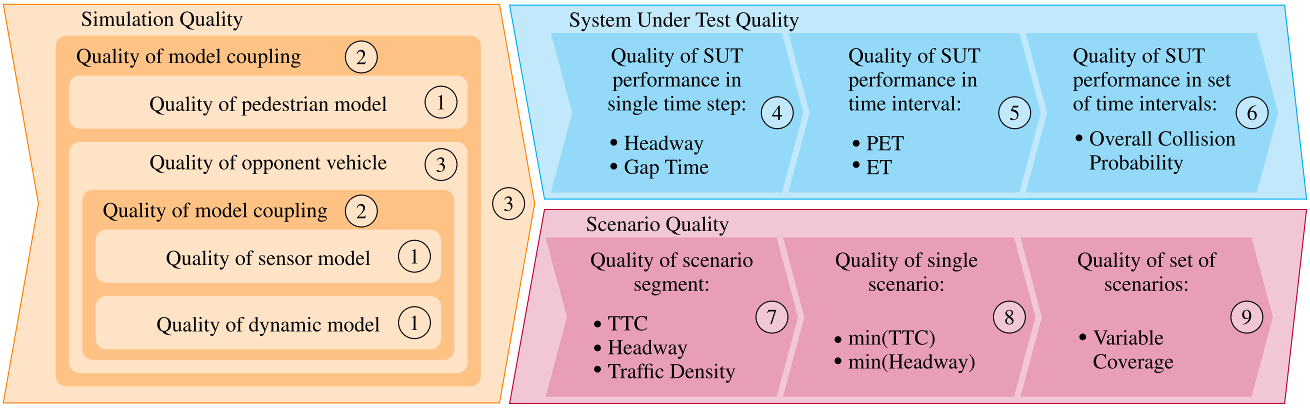


FIGURE 5. Basic structure of quality interaction described by section IV with examples. Abbreviated metrics are post-encroachment time (PET), encroachment-time (ET), time-to-collision (TTC).

simulation quality describes the degree to which simulation model(s) and their coupling methods fulfill both aspects during model development and application.

In the urban intersection example described in section IV-A, simulation quality includes the following parts, which are also shown on the left side of Fig. 5:

1: Nanoscopic: quality of single atomic simulation models, e.g., pedestrian model, subunits of opponent vehicles,
2: Microscopic: quality of coupling, e.g., synchronization, message format, actor availability,
3: Macroscopic: quality of coupled system, e.g., opponent vehicle, simulation environment module.

There are several methods for the quality assessment of simulation models. An overview can be found in Riedmaier *et al.* [36]: the focus is on a single atomic or coupled simulation model (1) and (3), and several quality criteria for validation are proposed, depending on the kind of simulation model (deterministic vs. non-deterministic) as well as the output characteristics (boolean, probabilistic, real-valued). Unit tests for simulation models, partial simulations,

TABLE 1. Simulation quality in literature.

Level	Description	Publication
Nanoscopic	simulation model V&V methods survey	[36]
	steering system evaluation	[45]
	scenario-based model evaluation	[48]
Microscopic	accuracy, effort, efficiency	[49]
Macroscopic	simulation model V&V methods survey	[36]
	ISO 26262 Standard: tool qualification	[30]

comparison with real-world data, or fault-injections fall into categories (1) and (3). Another example is given by Frerichs and Borsdorf [45], where the simulation model of a steering system is tested. In simulation quality, the quality of atomic simulation models or units is called **nanoscopic**, and the quality of coupled simulation models is called **macroscopic**. Between atomic and coupled simulation model quality falls the quality of the coupling mechanism (2), called **microscopic** simulation quality. (2) and (3) can lead to several iterations until the quality of the coupled units is adequately assessed. If (atomic) simulation models are coupled, we would ideally expect that by using validated coupling mechanisms, we could automatically assume that the coupled simulation models are of high quality. For most applications, however, this is unfortunately not the case. On its own, the coupling quality can be rated as good, but in an overall system with multiple simulation models, errors can still occur. Examples of coupling quality are checking actor availability, synchronization between actors, or data availability in message protocols. To our knowledge, there are no established standards yet.

Moreover, we assume that a general statement on the simulation quality can never be made and that measured quality is only an approximation of the actual quality. The description of simulation quality might lead to the impression that tool qualification is done by a bottom-up approach. However, it does not contradict a top-down approach and merely states what levels of tool qualification can take place. Table 1 shows a few examples for simulation model testing. There are many methods for verification and validation of simulation models, and the ISO 26262 provides a standard for tool qualification. However, it is hard to come by methods for coupling strategies of simulation models.

The example intersection scenario from section IV-A is used in different simulation tools for a better understanding. To state the reliability of simulation outcome of different simulation environment setups and, therefore, a macroscopic quality, we compared execution results from each setup among several executions of the same scenario. Our goal is to find critical parameter sets within a logical scenario and define a possible requirement regarding the used simulation setup: *If a concrete scenario is executed twice or more, the deviation of traffic participants' trajectories measured with dynamic time warping shall not deviate further than* 10.0 m *for a trajectory with 2500 data points.* For the first run of experiments to find the parameter sets, this is sufficient since it is known that the used driving function is non-deterministic. Two simulation environments were used: Carla 0.9.11 and CarMaker 8.0.2 from IPG combined with two different driving functions, called F1 and F2, respectively. F1 is an external driving function connected via the open simulation interface (OSI) [46] and F2 the BasicAgent from Carla's PythonAPI [23]. The same intersection scenario was executed ten times for each setup, where the first execution was used as a reference point for the following results. The trajectories of the ego vehicle and an adversary vehicle were compared with the reference scenario's trajectories via dynamic time warping (DTW), a metric to get the distance between two time series [47]. Ideally, and to indicate deterministic results, the distance should be 0.0 m or close to 0.0 m for nearly identical simulation results. We used 10.0 m as a threshold, indicated by the red lines in Fig. 6 a) and b); in c) no threshold is shown since all values lie beyond this threshold. If a simulation consists of 2000 steps, this means that the average trajectory deviation per step should be smaller than $10/2000\,\text{m} = 0.005\,\text{m}$.

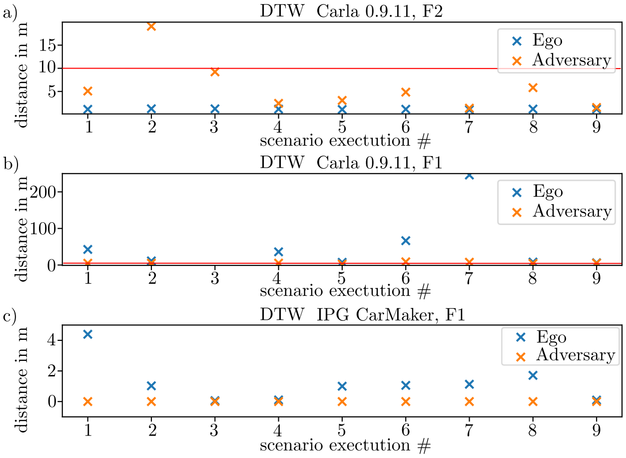


FIGURE 6. Dynamic Time Warping (DTW) comparison of different ego driving functions (F1, F2) and adversary vehicles in different simulation environments (Carla, CarMaker) for several executions of the same scenario description.

Fig. 6 shows the results for all executions. The only actor with completely deterministic behavior is the adversary vehicle in Fig. 6 c), a trajectory follower and part of the CarMaker software. The biggest DTW distances can be observed for F1 in Fig. 6 b), where scenario 3 has a value over 5000 for the ego and over 9000 for the adversary vehicle (both values are not shown since the scale would make it even more difficult to read other results). In this scenario, the ego vehicle came off the course and drove a circle at the intersection, which results in a very different trajectory than the reference trajectory. Additionally, it influences the opponent's behavior, e.g., an adversary vehicle might need to wait longer until it can pass the intersection if it is occupied by the ego vehicle. There are several possible reasons why these distances are as high as in Fig. 6 and indicate a poor macroscopic simulation quality: the used driving functions are not reliable enough (low system under test quality, which is explained further in section IV-C, coupling methods are of bad quality (low microscopic simulation quality), or non-deterministic physics model (if a deterministic model is needed for simulation a non-deterministic model indicates low nanoscopic simulation quality). Fig. 6 a) shows a DTW value around 1.0 for each F2 trajectory and since Carla 0.9.11. uses the non-deterministic physics model from its Unreal engine, it indicates that in the best case, there are minor differences, but overall results are still close to each other. The adversary vehicle has higher DTW values since the adversary vehicle reacts to the ego vehicle and, therefore, the differences from the ego actor are added and result in even higher trajectory deviations mostly in time. Furthermore, Fig. 6 c) demonstrates that CarMaker's adversary vehicle is capable of deterministic behavior. However, F1 still shows non-deterministic results, but all values are below the given threshold of 10.0 m. From these observations and the knowledge we have about the simulators, the results indicate further experiments are needed to evaluate the microscopic simulation quality (coupling of the driving function), where problems might come from the used PID controller or the message interface (OSI) as well as evaluation of the system under test (driving function F1) quality in all three levels of resolution.

This example shows that the proposed taxonomy offers terms and terminology for a faster and more precise understanding and communication of areas of interest in evaluating simulation results.

C. SYSTEM UNDER TEST QUALITY

The System under test quality evaluates the observable behavior and performance of a system under test with respect to the desired or intended functionality according to predefined requirements. A remarkable and widely used sub-class of the system under test quality is safety quality, which evaluates how *safe* a system under test can handle certain situations. Safety can be functional safety according to ISO 26262 [30], where it is described as "absence of unreasonable risk due to hazards caused by malfunctioning behavior of Electrical/Electronic systems". A further approach is to assess safety in critical scenarios derived from an initial criticality analysis [50]. A typical example is a near-collision situation evaluated with metrics like the time-to-collision (TTC) metric [39]. Assessing safety, e.g., functional safety needs to follow defined development and test processes and tool qualification rules. In the early stages of the development process, where proof of concepts and ideas are tested, performance might play a more significant role for developers than safety. Additionally, it is essential to note that different quality metrics can contradict each other: improving comfortable braking might also lead to more collisions in critical situations.

In the urban intersection example described in section IV-A, the system under test quality shall investigate how well the driving function can avoid collisions in critical situations and might include the following quality metrics, which are also shown on the top right side of Fig. 5:

4: Nanoscopic: quality of system under test for one time step, e.g., headway or gap time,

5: Microscopic: quality of system under test for a time interval, e.g., post-encroachment-time or encroachment-time,

6: Macroscopic: quality of system under test for a set of time intervals, e.g., overall collision probability for functional scenario or ODD.

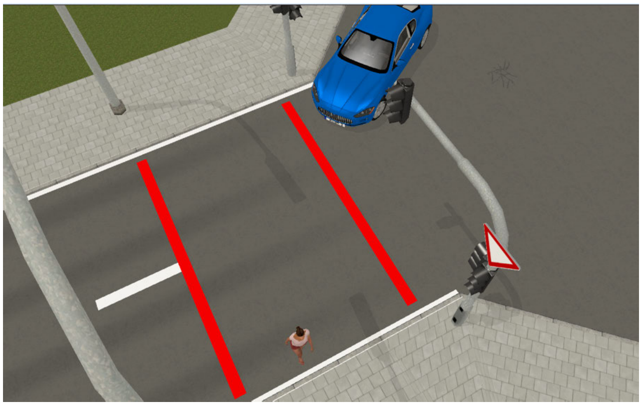


FIGURE 7. Example scenario with ego vehicle and pedestrian at an intersection as it is shown in dSPACE [21].

A scenario-based test approach is used, and functional and logical scenarios and their pass/fail criteria can be derived [8]. We propose one possible functional scenario: at an urban intersection, the ego vehicle shall turn right; by entering the right arm of the intersection, a pedestrian crosses the street. To keep the logical scenario simple, only three variables were introduced that can vary throughout the derived concrete scenarios: the maximum speed v_{max} allowed for the ego vehicle, the time t_{cross} the pedestrian needs to cross the street, and the starting distance d_{start} between ego vehicle and pedestrian, that has to be reached for the pedestrian to start crossing the intersection. Concrete scenarios can then be obtained and executed when all variable ranges are defined, and Fig. 7 shows an example scene from a simulation of a possible concrete scenario. In our example, the derived concrete scenarios consist of all possible combinations of the three variables, where possible values for v_{max} are from $30\,\mathrm{km\,h^{-1}}$ to $58\,\mathrm{km\,h^{-1}}$ with step size $2\,\mathrm{km\,h^{-1}}$, t_{cross} from 5 s to 9 s with step size 1 s, and d_{start} from 10 m to 24 m with step size 2 m. After deriving and concretizing scenarios, these concrete scenarios can be executed in a simulator, and gained information can be summarized and combined to assess the system under test quality.

At first, quality at the time step level can be evaluated. This step is associated with the matrix entry (4) in Fig. 4 and Fig. 5. Information on this level can differ throughout time series, e.g., the distance between two traffic participants. Fig. 8 a) shows the metric results calculated for four different quality metrics at the time step level of the intersection scenario with

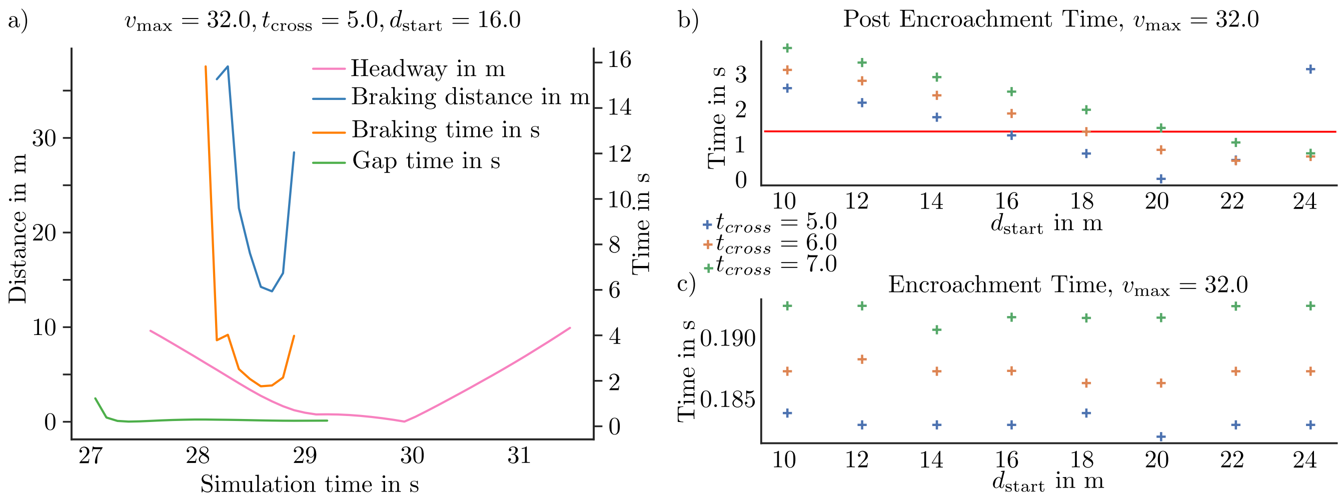


FIGURE 8. a) Example for nanoscopic metrics. b) + c) Examples for microscopic metrics.

$v_{\max} = 32\,\text{km}\,\text{h}^{-1}$, $t_{\text{cross}} = 5\,\text{s}$, and $d_{\text{start}} = 16\,\text{m}$. The used quality metrics are braking time, braking distance, gap time, and headway (distance) between the ego vehicle and the pedestrian during the simulation. Gap time describes the predicted distance in time between vehicle and pedestrian passing the intersection of their trajectories. The fact that it is going towards 0 s means there is a near-collision situation or even a collision. Gap time cannot be measured anymore when one actor passes the intersection of both trajectories and, therefore, the green gap time graph stops after about 29.2 s when the application period of gap time has passed. The gaps in the graphs of braking time and distance show the system under test either stood still or tried to accelerate (where braking time and distance approach infinity) in between braking phases since the application period condition for both metrics is a negative acceleration. Additionally, threshold values can be defined for the considered quality metrics, e.g., Gap Time $> 2\,\text{s}$. In Fig. 8 a), the gap time and headway approach a value of 0.0 s and 0.0 m, respectively. Both metrics indicate that the ego vehicle gets dangerously close to the pedestrian. We conclude that the ego vehicle is already standing at the time of the smallest headway since both metrics reach their minimum at different times. The ego vehicle fulfilled its goals in terms of not hitting the pedestrian. However, regarding the small gap time, other goals, e.g., the ego vehicle shall not frighten vulnerable road participants, might not be achieved and demand further software development phases. Another performance quality example could be to evaluate comfortable braking behavior. Simulation results, as depicted in Fig. 8 a) are called **nanoscopic** system under test quality.

The following entry is information evaluation on a time interval level (5), where quality criteria and their results can be used to gain more information on a time interval, e.g., (partial) scenario. A time interval can also be defined through an application period, e.g. if two traffic participants are within a certain distance and only if this condition is true other metrics are used. Fig. 8 b) and c) show the metric results of post-encroachment-time (PET) and encroachment-time (ET), respectively, and each value belongs to a concrete scenario derived from the logical scenario in the urban intersection example. According to Allen *et al.* [40], post-encroachment-time is defined as the actual time gap between two traffic participants passing the intersection point or area of their trajectories. Encroachment-time is the time an actor is occupying the intersection point or area and, therefore, describes the time it is exposed to a possible accident. As shown in Fig. 8 b), encroachment-time increases slightly with the time the pedestrian needs to cross the street (t_{cross}), but, as expected, the ego vehicle's starting distance and ego vehicle's speed have no impact as they are not related to the pedestrian's movement. Therefore, in the urban intersection example, the quality metrics are post-encroachment-time and encroachment-time. The metric results can be evaluated with the example threshold values PET $> 1.5\,\text{s}$ [40] and ET $> 2.0\,\text{s}$. In Fig. 8 b) it is shown that the ego vehicle falls below the given threshold in several scenarios, which indicates critical scenarios. In an exemplary use case, the goal of the simulations is to find out if the ego vehicle needs additional development time for algorithm refinement to avoid critical situations. Encountering a set of critical scenarios shows that further development is needed. Additionally, these scenarios can serve as a performance baseline for testing following software revisions. Unfortunately, encroachment-time does not deliver new insights, apart from the trivial fact that the slower the pedestrian moves, the longer they occupy the road. For both metrics, application time is controlled by the time the actors pass the trajectory intersection. Another possible quality metric is the smallest measured distance between two actors during one scenario. The smallest measured distance is the microscopic version of the nanoscopic headway. Simulation results of time intervals as depicted in Fig. 8 are called **microscopic** system under test quality.

TABLE 2. **System under test quality in literature.**

Level	Description	Publication
Nanoscopic	Time-to-collision	[39], [51]
	Gap Time	[40]
Microscopic	PET, ET	[40]
	min. Time-to-collision	[51]
Macroscopic	collision probability	[51]

Quality evaluation on the next abstraction level is called **macroscopic** since it combines microscopic quality criteria of different time intervals, test series, or aggregated scenarios regarding a system under test. This step is associated with the entry (6). A functional scenario can lead to different logical scenarios, e.g., similar situations on different maps, different types of pedestrians, e.g., children or handicapped with walking aids. Additionally, one logical scenario can be implemented differently: actors can follow predefined trajectories or only be given goal positions they have to reach. In the urban intersection example, results from scenarios derived from the functional scenario can be combined with other near-collision scenarios and evaluated, e.g., calculate overall collision probability in a functional scenario or ODD. Table 2 lists a few examples for the system under test quality evaluation. Elements for which system under test quality is measured can be highly automated driving functions, assisted driving functions, or even simulation models. For instance, the system under test is a simulation model, then nanoscopic, microscopic, and macroscopic system under test quality can be used to get nanoscopic simulation quality for this simulation model.

D. SCENARIO QUALITY

In some phases during the development and test process, it is necessary to evaluate the quality of a scenario segment, a single scenario, or a set of scenarios, e.g., to determine the criticality of a situation regarding the criticality analysis of a system under test [50]. In this case, the quantified element is the scenario itself. The field of scenario quality needs further research since it is mainly equated with the system under test quality [52], [53]. Often the scenario is assessed from the point of view of the ego vehicle, which is an objective point of view. A more subjective point of view would be to assess the overall situation independent of the ego vehicle. For introducing scenario-based testing as a standard for testing procedures, scenarios have to represent realistic situations and traffic participants. To our knowledge, there is no research regarding subjective points of view within scenarios or the degree of how realistic a scenario represents a situation. The latter also overlaps to some degree with simulation model quality.

A widely used objective scenario quality metric, the distance between two cars, can be used to evaluate an adaptive cruise control system [54] and find new scenarios [32]. Another scenario quality worth mentioning is scenario coverage, which plays an essential role in overall test coverage. Scenario coverage can be viewed on different levels, e.g., does a set of concrete scenarios sufficiently represent a logical scenario, or does the scenario set or database sufficiently represent all situations within an operational design domain. Reducing mileage is one of the goals of scenario-based testing, and for that, it is crucial to make a statement about the coverage. Scenario quality metrics can give evidence about a scenario's criticality or relevance for a test series. The difference between the system under test quality and scenario quality is the focus of the evaluation. In the system under test quality, all metrics pertain to the system under test, whereas scenario quality focuses on various properties of a scenario. However, the system under test quality can be part of it. Quality criteria for scenarios can be a composition of different quality metrics or the same metrics taken from different traffic participants and evaluated together.

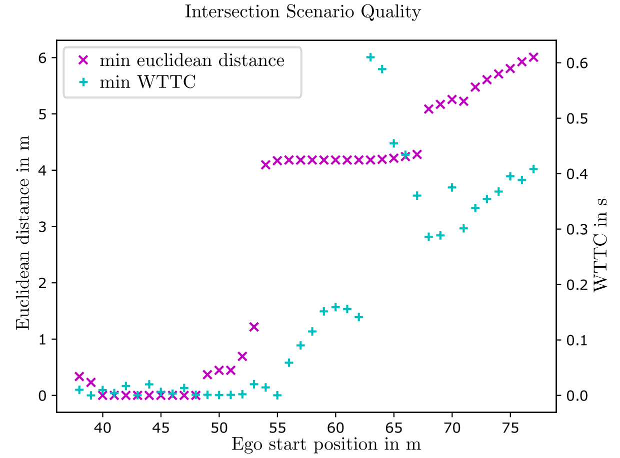


FIGURE 9. **Euclidean distance and worst time-to-collision (WTTC) between ego vehicle and pedestrian as microscopic scenario quality.**

In the urban intersection example described in section IV-A, two goals could be finding scenarios with near-collision situations or scenarios with insufficient environmental information due to covert road users for further testing and monitoring the quality of collision avoidance during different software revisions. Fig. 9 shows the minimum euclidean distance and worst time-to-collision (WTTC) [52] between ego vehicle and the road crossing pedestrian. In this case, the focus does not lie on the performance of the ego vehicle but the microscopic scenario quality, the end value for an executed scenario. The nanoscopic qualities for this scenario are the euclidean distances and worst time-to-collision that were measured for every time step during the simulation with their microscopic version of choosing the minimum value for a completely simulated scenario, respectively.

This example shows that the proposed taxonomy helps easily distinguish and communicate the difference in the area of interest. In this case, the performance of the ego vehicle is not the focus of the evaluated scenarios but the scenario itself.

TABLE 3. Scenario quality in literature.

Level	Description	Publication
Nanoscopic	TTC	[53]
	traffic quality	[42]
	WTTC	[52]
Microscopic	scenario uniqueness	[55]
	search-based techniques	[32] (min. distance), [53] (min. TTC)
Macroscopic	coverage	[56]

For instance, a set of scenarios gained this way can be used to evaluate changes made to the used driving function in a later revision.

All scenarios compared in Fig. 9 were the same, and only the x coordinate of the ego vehicle's starting position varied between 38 m and 78 m. Throughout of these scenarios, it is possible to see that the worst time-to-collision starts with scenarios close to 0 s but increases the further away the ego vehicle is set. Regarding worst time-to-collision, it could be interesting to do more simulations with a starting point between position 63 m and 64 m since there is a gap between the results, which often indicates a change in the order and course of the scenario and its participants. The same holds for starting positions 53 m and 54 m regarding the minimum euclidean distance.

Possible metrics for scenario quality, which are also shown on the bottom right side of Fig. 5 are:

7: Nanoscopic: quality of scenario segment, e.g., time-to-collision (time step), headway (time step), or traffic density (time step or time interval),
8: Microscopic: quality of single scenario, e.g., min(time-to-collision) or min(headway),
9: Macroscopic: quality of set of scenarios, e.g., variable coverage within this set.

Example metrics as used in section IV-C are possible on a **nanoscopic** (7) level and quantify the scenario quality for a scenario segment, i.e., time step, time interval, or application time depending on conditions that are not fulfilled during the complete scenario. The next level of scenario quality is the **microscopic** (8) level, where the quality of a complete scenario is evaluated. The last abstraction level is **macroscopic** scenario quality, where the aggregation of a set of scenarios is evaluated (9). A typical example for macroscopic scenario evaluation is coverage [56]. Table 3 lists literature about scenario quality evaluation.

V. CONCLUSION AND FUTURE WORK

In this paper, we defined and delineated the quality and important terms relevant during the simulation process. We explained all relevant terms in section III and gave an overview of their relationships to each other. Based on these terms, we analyzed different domains of interest and simulation resolution types and proposed a classification to assess quality for each aspect. For this purpose, terms regarding resolution and established in other domains were introduced: macroscopic, microscopic, and nanoscopic. Although this work is highly conceptional, this approach eases the evaluation process since it defines comparable aspects during the verification and validation process and clearly defines what level of information a quality criterion provides. A taxonomy for three different quality classes was introduced, represented in a two-dimensional matrix. This taxonomy shows that the classification is a way to help scientists imminently understand and communicate the differences of areas of quality in automotive simulation.

As long as simulation model quality is not good enough, simulation models have to be developed further until they can be used for testing, which belongs to (1)-(3) in Fig. 4. Scenario quality decides if scenarios are interesting and relevant for a function at hand, can give a coverage statement, or help to find new scenarios or better fitting ones (7)-(9). Finally, the system under test quality shows *how good* a function is and if it needs more revisions during development or performs compared to others (4)-(6). If other domains of interest arise, they can be easily added to the taxonomy.

Additionally, quality metrics for simulation models and driving functions can be used (among scenario-specific metrics) to evaluate scenario quality, and scenarios can be used to test simulation models and driving functions. However, this subjective method might lead to a cyclical dependence: system under test/simulation model quality metrics are used to find critical scenarios, and these, in turn, are then used to test the system under test/simulation model. In our opinion, it is essential to distinguish between scenario and system under test/simulation model quality to allow more objective metrics. Metrics that do not center around the performance of an ego vehicle, but on other features, e.g., assess the start set up of a scenario, can also give insight to the criticality within a scenario and could be used to define universal baseline sets of scenarios to ensure specific quality standards independent of driving functions or simulation models.

In the future, quality assessment and metrics can be related to this taxonomy for an easier understanding and classification, and if needed, new domains of interest or resolution levels can be added, or single categories can be studied more in-depth. Moreover, new tools and standards are important to assess and compare quality throughout the development and test process. However, simulation model verification and validation need further systematic approaches for a better quality evaluation in general. In particular, the entries (2) and (7)-(9) in Fig. 4 can be researched further to establish useful, well-defined, and safe methods to ensure overall simulation and testing quality. Proposed future research includes quality metrics for coupling methods and metrics to evaluate the credibility of a scenario, i.e., how realistic is the testing scenario or scenario coverage if a set of scenarios is used for testing. In addition, scenario quality needs more objective evaluation metrics combined with existing ones, e.g., to define baseline scenario sets for testing. Therefore, we propose the following list of future work as an anchor how to proceed further with this work as a basis:

- **Simulation Quality**: methods and metrics for evaluating coupling mechanisms,
- **System under Test Quality**: methods to summarize metrics from a nanoscopic and microscopic level on a macroscopic level,
- **Scenario Quality**: subjective metrics to evaluate scenario segments and scenarios, either the degree to which they correspond to reality or their quality regarding test requirements, e.g., criticality; metrics for scenario coverage.

REFERENCES

[1] K. K. Goldewijk, A. Beusen, and P. Janssen, "Long-term dynamic modeling of global population and built-up area in a spatially explicit way: HYDE 3.1," *Holocene*, vol. 20, no. 4, pp. 565–573, Jun. 2010, doi: 10.1177/0959683609356587.

[2] J. Bach, S. Otten, and E. Sax, "A taxonomy and systematic approach for automotive system architectures-from functional chains to functional networks," in *Proc. 3rd Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, 2017, pp. 90–101, doi: 10.5220/0006307600900101.

[3] M. Wood. (2019). *Safety First for Automated Driving*. Accessed: Jan. 26, 2022. [Online]. Available: https://www.daimler.com/documents/innovation/other/safety-first-for-aut%omated-driving.pdf

[4] Cambridge University Press. (2020). *Cambridge Dictionary*. Accessed: Oct. 21, 2020. [Online]. Available: https://dictionary.cambridge.org/

[5] W. Wachenfeld and H. Winner, "The release of autonomous vehicles," in *Autonomous Driving*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds. Berlin, Germany: Springer, 2016, pp. 425–449.

[6] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor Vehicles*, Standard J3016, SAE Int., 2018.

[7] Pegasus Project Consortium. (2019). *Pegasus Method: An Overview*. Accessed: Apr. 16, 2020. [Online]. Available: https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung%/PEGASUS-Gesamtmethode.pdf

[8] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1821–1827, doi: 10.1109/IVS.2018.8500406.

[9] E. Böde, M. Büker, U. Eberle, M. Fränzle, S. Gerwinn, and B. Kramer, "Efficient splitting of test and simulation cases for the verification of highly automated driving functions," in *Computer Safety, Reliability, and Security*, B. Gallina, A. Skavhaug, and F. Bitsch, Eds. Cham, Germany: Springer, 2018, pp. 139–153, doi: 10.1007/978-3-319-99130-6_10.

[10] C. King, L. Ries, C. Kober, C. Wohlfahrt, and E. Sax, "Automated function assessment in driving scenarios," in *Proc. 12th IEEE Conf. Softw. Test., Validation Verification (ICST)*, Apr. 2019, pp. 414–419, doi: 10.1109/ICST.2019.00050.

[11] F. Bock, C. Sippl, A. Heinz, C. Lauer, and R. German, "Advantageous usage of textual domain-specific languages for scenario-driven development of automated driving functions," in *Proc. IEEE Int. Syst. Conf. (SysCon)*, Apr. 2019, pp. 1–8, doi: 10.1109/SYSCON.2019.8836912.

[12] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Sep. 2015, pp. 982–988, doi: 10.1109/ITSC.2015.164.

[13] J. Bach, S. Otten, and E. Sax, "Model based scenario specification for development and test of automated driving functions," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 1149–1155, doi: 10.1109/IVS.2016.7535534.

[14] ASAM E.V. *ASAM OpenSCENARIO*. Accessed: Apr. 21, 2021. [Online]. Available: https://www.asam.net/standards/detail/openscenario/

[15] B. Kramer, C. Neurohr, M. Büker, E. Böde, M. Fränzle, and W. Damm, "Identification and quantification of hazardous scenarios for automated driving," in *Model-Based Safety and Assessment*, M. Zeller and K. Höfig, Eds. Cham, Germany: Springer, 2020, pp. 163–178, doi: 10.1007/978-3-030-58920-2_11.

[16] B. Schütt, T. Braun, S. Otten, and E. Sax, "SceML: A graphical modeling framework for scenario-based testing of autonomous vehicles," in *Proc. 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst.*, Oct. 2020, pp. 114–120, doi: 10.1145/3365438.3410933.

[17] ASAM e.V. *ASAM OpenDRIVE*. Accessed: Dec. 6, 2021. [Online]. Available: https://www.asam.net/standards/detail/opendrive/

[18] ASAM e.V. *ASAM OpenCRG*. Accessed: Dec. 6, 2021. [Online]. Available: https://www.asam.net/standards/detail/opencrg/

[19] M. Steimle, T. Menzel, and M. Maurer, "Toward a consistent taxonomy for scenario-based development and test approaches for automated vehicles: A proposal for a structuring framework, a basic vocabulary, and its application," *IEEE Access*, vol. 9, pp. 147828–147854, 2021, doi: 10.1109/ACCESS.2021.3123504.

[20] MSC Software. *Virtual Test Drive (VTD) Complete Tool-Chain for Driving Simulation*. Accessed: Oct. 8, 2021. [Online]. Available: https://www.mscsoftware.com/product/virtual-test-drive

[21] dSPACE GmbH. *dSPACE—Automotive Simulation Models (ASM)*. Accessed: Sep. 6, 2021. [Online]. Available: https://www.dspace.com/en/inc/home/products/sw/automotive_simulation%_models.cfm

[22] IPG Automotive GmbH. *Everything About Virtual Test Driving*. Accessed: Oct. 8, 2021. [Online]. Available: https://ipg-automotive.com/

[23] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[24] P. A. Lopez, E. Wiessner, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flotterod, R. Hilbrich, L. Lucken, J. Rummel, and P. Wagner, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582, doi: 10.1109/ITSC.2018.8569938.

[25] OpenPASS Working Group. *openPASS Target Objectives*. Accessed: Oct. 20, 2021. [Online]. Available: https://openpass.eclipse.org/target_objectives/

[26] PTV Group. *Traffic Simulation Software—PTV Vissim*. Accessed: Oct. 8, 2021. [Online]. Available: https://www.ptvgroup.com/en/solutions/products/ptv-vissim/

[27] Esmini. *Esmini: A Basic Openscenario Player*. Accessed: Oct. 8, 2021. [Online]. Available: https://github.com/esmini/esmini

[28] *Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)*, Standard 1730–2010, IEEE Standards Association, 2011.

[29] U. Durak, A. D'Ambrogio, and P. Bocciarelli, "Safety-critical simulation engineering," in *Proc. Summer Simulation Conf.*, 2020, pp. 1–12.

[30] *Road Vehicles—Functional Safety*, Standard ISO26262, International Organization for Standardization, 2018.

[31] D. Ni, "A framework for new generation transportation simulation," in *Proc. Winter Simulation Conf.*, Dec. 2006, pp. 1508–1514, doi: 10.1109/WSC.2006.322920.

[32] M. Schiller, M. Dupius, D. Krajzewicz, A. Kern, and A. Knoll, "Multi-resolution traffic simulation for large-scale high-fidelity evaluation of vanet applications," in *Simulating Urban Traffic Scenarios*, M. Behrisch and M. Weber, Eds. Berlin, Germany: Springer, 2019, pp. 17–36, doi: 10.1007/978-3-319-33616-9_2.

[33] N. G. Eissfeldt, "Vehicle-based modelling of traffic—Theory and application to environmental impact modelling," Ph.D. dissertation, Dept. Math. Natural Sci., Univ. zu Köln, Köln, Germany, 2004.

[34] M. Viehof and H. Winner, "Research methodology for a new validation concept in vehicle dynamics," *Automot. Engine Technol.*, vol. 3, nos. 1–2, pp. 21–27, Aug. 2018, doi: 10.1007/s41104-018-0024-1.

[35] P. Rosenberger, J. T. Wendler, M. F. Holder, C. Linnhoff, M. Berghöfer, H. Winner, and M. Maurer, "Towards a generally accepted validation methodology for sensor models-challenges, metrics, and first results," in *Proc. Grazer Symp. Virtuelles Fahrzeug*, Graz, Austria, 2019, pp. 1–13. [Online]. Available: https://tuprints.ulb.tu-darmstadt.de/8653/

[36] S. Riedmaier, B. Danquah, B. Schick, and F. Diermeyer, "Unified framework and survey for model verification, validation and uncertainty quantification," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 2655–2688, Jun. 2021, doi: 10.1007/s11831-020-09473-7.

[37] L. Westhofen, C. Neurohr, T. Koopmann, M. Butz, B. Schütt, F. Utesch, B. Kramer, C. Gutenkunst, and E. Böde, "Criticality metrics for automated driving: A review and suitability analysis of the state of the art," 2021, *arXiv:2108.02403*.

[38] D. Gettman and L. Head, "Surrogate safety measures from traffic simulation models," *Transp. Res. Rec.*, vol. 1840, no. 1, pp. 104–115, 2003, doi: 10.3141/1840-12.

[39] J. C. Hayward, "Near miss determination through use of a scale of danger," in *Proc. 51st Annu. Meeting Highway Res. Board*, 1972, pp. 1–12.

[40] B. L. Allen, B. T. Shin, and P. J. Cooper, "Analysis of traffic conflicts and collisions," *Transp. Res. Rec.*, vol. 667, pp. 67–74, 1978.

[41] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8271–8277, doi: 10.1109/ICRA.2019.8793740.

[42] S. Hallerbach, Y. Xia, U. Eberle, and F. Koester, "Simulation-based identification of critical scenarios for cooperative and automated vehicles," *SAE Int. J. Connected Automated Vehicles*, vol. 1, no. 2, pp. 93–106, Apr. 2018, doi: 10.4271/2018-01-1066.

[43] P. M. Junietz, "Microscopic and macroscopic risk metrics for the safety validation of automated driving," Ph.D. dissertation, Inst. Automot. Eng., Techn. Univ. Darmstadt, Darmstadt, Germany, 2019.

[44] O. Balci, "Verification, validation, and testing," *Handbook Simul.*, vol. 10, no. 8, pp. 335–393, 1998.

[45] D. Frerichs, and M. Borsdorf, "Quality for vehicle system simulation," in *Proc. SIMVEC-Simul. Test. Vehicle Develop., VDI-Kongress*, Baden-Baden, Germany, vol. 19, 2018.

[46] ASAM E.V. *Welcome to Open Simulation Interface's Documentation*. Accessed: Oct. 8, 2021. [Online]. Available: https://opensimulationinterface.github.io/osi-documentation/index.html

[47] L. Ries, P. Rigoll, T. Braun, T. Schulik, J. Daube, and E. Sax, "Trajectory-based clustering of real-world urban driving sequences with multiple traffic objects," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1251–1258, doi: 10.1109/ITSC48978.2021.9564636.

[48] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 204–211, doi: 10.1109/IVS.2017.7995721.

[49] F. Günther, "Beitrag zur Co-simulation in der Gesamtsystementwicklung des Kraftfahrzeugs," Ph.D. dissertation, Dept. Mech. Eng., Techn. Univ. München, München, Germany, 2017.

[50] C. Neurohr, L. Westhofen, M. Butz, M. H. Bollmann, U. Eberle, and R. Galbas, "Criticality analysis for the verification and validation of automated vehicles," *IEEE Access*, vol. 9, pp. 18016–18041, 2021, doi: 10.1109/ACCESS.2021.3053159.

[51] E. de Gelder and J.-P. Paardekooper, "Assessment of automated driving systems using real-life scenarios," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 589–594, doi: 10.1109/IVS.2017.7995782.

[52] W. Wachenfeld, P. Junietz, R. Wenzel, and H. Winner, "The worst-time-to-collision metric for situation identification," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 729–734, doi: 10.1109/IVS.2016.7535468.

[53] F. Klück, M. Zimmermann, F. Wotawa, and M. Nica, "Performance comparison of two search-based testing strategies for ADAS system validation," in *Proc. IFIP Int. Conf. Test. Softw. Syst.*, Oct. 2019, pp. 140–156, doi: 10.1007/978-3-030-31280-0_9.

[54] A. Basantis, Z. Doerzaph, L. Harwood, and L. Neurauter, "Developing a standardized performance evaluation of vehicles with automated driving features," *SAE Int. J. Connected Automated Vehicles*, vol. 2, no. 3, pp. 157–172, Aug. 2019, doi: 10.4271/12-02-03-0011.

[55] J. Langner, J. Bach, L. Ries, S. Otten, M. Holzapfel, and E. Sax, "Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1860–1866, doi: 10.1109/IVS.2018.8500464.

[56] F. Hauer, T. Schmidt, B. Holzmuller, and A. Pretschner, "Did we test all scenarios for automated and autonomous driving systems?" in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2950–2955, doi: 10.1109/ITSC.2019.8917326.

BARBARA SCHÜTT received the B.Sc. and M.Sc. degrees in computer science from Technische Universität Berlin, in 2012 and 2015, respectively. She is currently pursuing the Ph.D. degree with the FZI Research Center for Information Technology. She was a Researcher at the Department of Robotics and Automation, Fraunhofer Institute for Production Systems and Design Technology. She worked as a Software Developer in the series development at Carmeq/Volkswagen in the field of situation analysis for ADAS. Since 2019, she has been a Researcher with the FZI Research Center of Information Technology. Her research interests include quality metrics and scenario evaluation.

MARKUS STEIMLE received the B.Eng. degree in electrical engineering and information technology from the Landshut University of Applied Sciences, Landshut, Germany, in 2014, and the M.Sc. degree in automotive software engineering from the Technical University of Munich, Munich, Germany, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Control Engineering, Technische Universität Braunschweig, Braunschweig, Germany. His research interests include scenario-based verification and validation of automated vehicles, focusing on the use of simulative testing methods.

BIRTE KRAMER received the B.Sc. and M.Sc. degrees in mathematics from Carl von Ossietzky University, Oldenburg, in 2015 and 2017, respectively. She is currently pursuing the Ph.D. degree with the Institute of Systems Engineering for Future Mobility, German Aerospace Center. From 2017 to 2020, she was a Researcher at OFFIS e.V., where she was working in the area of scenario-based verification and validation of automated vehicles. Since 2021, she has been as a Researcher with the Institute of Systems Engineering for Future Mobility, German Aerospace Center. Her research interest includes validity and quality criteria for simulations.

DANNY BEHNECKE received the M.Sc. degree in computer and communications systems engineering from Technische Universität Braunschweig, in 2017. Since 2017, he has been a Research Assistant with the Institute of Transportation Systems, German Aerospace Center. His research interests include simulation system architecture and validation of simulation execution.

ERIC SAX (Prof. Dr.-Ing.) is currently the Head of the Institute of Information Processing Technology (www.itiv.kit.edu), Karlsruher Institute of Technology. In addition, he is a Director of FZI Forschungszentrum Informatik (www.fzi.de) and the so called Hector School, the Technology Business School, KIT. His main topics of research, together with currently more than 50 Ph.D. employees, are processes, methods and tools in systems engineering, focusing the automotive industry.