第 49 卷第 S1 期
2021 年 12 月

同济大学学报(自然科学版)
JOURNAL OF TONGJI UNIVERSITY(NATURAL SCIENCE)

Vol. 49 No. S1
Dec. 2021

文章编号：0253-374X(2021)S1-0011-09 **DOI**：10. 11908/j. issn. 0253-374x. 22751

独立激励评估法在车身功能测试中的应用

FUCHS Julian[1]，STEINHAUSER Christian[1]，KING Christian[1]，
KAAG Kevin[2]，SAX Eric[1]

（1. FZI Research Center for Information Technology，卡尔斯鲁厄 76131，德国；2. Daimler AG，斯图加特 70372，德国）

摘要：车辆硬件在环仿真测试时,目前使用预先定义的测试结构,即测试结构将特定的测试激励与单独的测试步骤耦合到固定的、预先定义的测试体系中,测试后给出通过或不通过的结果。此方法中的每项测试都有其特定的功能测试重点,但无法适用于多项功能同时测试。未来的测试方法需要尽可能地完成全面性功能评估,并使评估资源得到有效利用,即应该在独立激励和现实环境中可同时对多项功能进行评估,因此,需要采用与之前不同的方法在独立激励的情况下进行测试。由于测试环境事先无法确定激励序列,因而也无法对每个测试步骤进行单独验证。激励的确切序列在测试运行开始时是未知的,从而可模拟出一个尽可能真实的现实环境,为此介绍一种独立激励的测试方法。该方法将组合法与基于模型法相结合,与相应功能测试要求联系起来,用于车身领域的系统性功能评估。该方法同样也支持现有的方法,并实现了比普通测试方法更广泛、更深入的评估覆盖面。该方法将在一家德国汽车制造商的车辆硬件在环舒适性功能测试中得到验证。

关键词：汽车测试;测试激励;独立激励;车身域

中图分类号：U467.4 **文献标志码**：A

Use of Stimulation-Independent Evaluation in the Context of Vehicle Body Domain

FUCHS Julian[1]，*STEINHAUSER Christian*[1]，
KING Christian[1]，*KAAG Kevin*[2]，*SAX Eric*[1]

（1. FZI Research Center for Information Technology，76131 Karlsruhe，Germany；2. Daimler AG，70372 Stuttgart，Germany）

Abstract： Currently established automotive test approaches in the field of vehicle hardware-in-the-loop （HiL） testing use predefined test structures. The test case structure couples a specific test stimulation with an individual test step to a fixed，predefined system validation with pass / fail result. Each test case has its own specific functional focus. If more than one functional aspect is to be tested，the current methods will not work anymore. The aim of future test methods is to evaluate the system as comprehensively and resource-efficiently as possible. If possible，several sub-functionalities should be evaluated at the same time in a randomly generated，realistic environment. A different validation approach is required for the evaluation of randomly generated stimulations sequences. Previously they are unknown to the evaluation environment. It is no longer possible to validate each test step individually. The exact sequence of the stimulation is not known at the beginning of the test run. This should simulate a behavior that is as realistic as possible. For this purpose，a methodology for the generation of system evaluation for randomly generated stimulations was introduced. Combinatorial and model-based approaches were combined to support the creation of system evaluation for the vehicle body domain （describes the passenger functions that can be experienced within a vehicle with its control units and functionalities，no direct influence on driving dynamics） and link them to the corresponding system requirements. The approach supports the existing methods and achieve a wider and deeper coverage of system assessments than a normal test catalogue implementation would give. This will be shown in a proof of concept with a vehicle comfort function on a HiL system at a German car manufacturer.

Key words： automotive testing； test stimulation； stimulation-independent； body domain

1 Introduction

Nowadays a vehicle consists of over 100 electrical， electronic and electromechanical components[1] which provide safety and comfort

收稿日期：2021-10-20
第一作者：FUCHS Julian (1995—)，男，汽车工程硕士，主要研究方向为系统工程。 Email：fuchs@fzi. de

functions[2-5]. The interaction of the various vehicle functions poses a challenge. According to reference [3], for example, the air conditioning system increases the engine speed in order to draw more power, or switches off the air supply when the reverse gear is engaged. In order to ensure the necessary functionality of the individual system, as well as their correct interaction in the vehicle, the individual electronic control units (ECU) must be tested[6]. To test the increasing functionalities of the ECU software, efficient test methods, which make optimal use of the available resources, are required[7-8]. At the same time, test methods should detect errors as early as possible and reliably in the development process. Model-based testing is already an established testingapproach for mobile devices[9-10]. Model-based testing is focusing the creation of test cases with many individual test steps. However, the large number of test cases cannot be handled in systems with real-time constrains, such as the hardware-in-the-loop (HiL) test instance. Each test case must be carried out and evaluated one after another.

In the domain of driver assistance systems (DAS) a randomly generated test drive was used to create long-term tests with low effort[9]. A driver model runs a "random" scenario in an environment model. The current approaches and methodologies of model-based and requirement-based testing no longer work properly[10]. For this purpose, a validation approach was introduced according to reference [11] and [7], which enables a continuous event-based evaluation of a randomly generated stimulation sequence. This concept is already being used on the driver assistant HiL at a German car manufacture[7-8, 12].

The system under test (SuT) consists of several functions. The behaviour of the SuT functionalities can be continuously monitored by using so called "Assessments". This process only focuses the validation of DAS and is not applicable to the domain of body functions directly.

In order to be able to use the existing test infrastructure more efficiently and the process of evaluation creation more effectively, this paper will introduce a novel approach, which reduces the effort of the evaluation creation and guarantees a deeper coverage of requirements at the same time. Furthermore, the existing approach should be converted to the domain of body functions. This method is demonstrated at a HiL system used by a German car manufacture. The mentioned points lead to the following research questions (RQ):

RQ1: How to increase test scope without increasing time on test bench?

RQ2: How to reduce the cost of test time by connecting isolated test cases in series to increase the efficiency of the whole test and evaluation process?

In summary, the contributions (C) of this paper can be presented as follows:

This paper is contributive because it increases the effective test time on HiL systems by continuous-event-based testing on a vehicle body domain functionality with the help of model-based testing, and it introduces a process to generate stimulation independent evaluations.

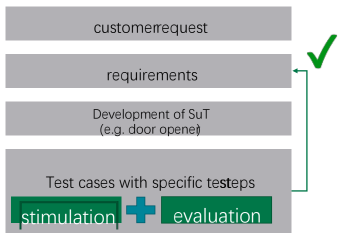
2 Related work—testing of automotive software

Testing and development of software functions in the automotive context is traditionally based on the V-model[13-15]. A test concept provides information about the used resources, the schedule of the test activities, delimitations and the exact procedure. Moreover it mentions the exact SuT and the test tasks[16]. The test concept defines the operational test environment (e. g., simulation, proving ground, and field tests) in which the logical scenarios with the associated parameter characteristics are tested. Requirements are set up for the test[17]. According to IEEE 829 "Standard of Software Test Documentation"[18], the test concept is a standardized documentation[19]. Two test methods for checking a SuT, which are the basis for the new approach, are described below.

2.1 Requirement-based approach

The verification of comfort and body functions in the automotive context takes place in the right part of the V-model classically[15]. Typically, the requirement-related verification is used as the main test concept. According to reference [14], the test creation is based on the principles that for each requirement , there is at least one test case and that there is no test without requirement.

Test gaps can be identified with the help of requirementsbased testing and known risks can be addressed[14]. The test coverage is used as a metric to determine the degree of maturity of the system under test. Requirement-based testing uses systematic test cases consisting of a precondition and test steps including evaluation conditions and stimulation (see Fig. 1)[19]. The approach presented in this paper uses system requirements to derive system evaluation conditions, too. However, the concrete evaluation criteria and stimulation of the SuT function are separated here (see Section 3).



One Test case consists of many test steps.Each test step couples a stimulation to an evaluation[7].

Fig.1 Classical test case structure

2.2 Model-based approach

Model-based testing is a software test concept, which creates concrete test runs (stimulation and evaluation). For the creation, a predictive system model is used. The model describes the behavior of the system and takes care of various combinations of input variables, events, transition conditions, and output states[9-10]. As a result, test cases are generated systematically.

The model should ideally be comprehensible, reusable and should contain a precise description of the SuT functionalities. Basically, there is a large number of models, which always describe a certain aspect of the system[5,20]. In automotive development, model-based methods play an increasingly important role, especially in the area of E/E system architecture and in software and control design[10]. An important consideration of this development is the increase of system functions, their distribution over several control units and the increasing networking of previously independently developed system components[9]. Models can be represented by different methods. The representation can, for example, be shown in natural language, images or graphics, mathematical equations, graphic notations (e. g. state diagram or unified modelling language (UML) diagram) or matrices (e. g. decision tables) or a combination of all of them[21]. The model can contain information about the timing behaviour, external ground truth signals, redundancy and functional signals[20]. The advantage of using model-based testing is an improved degree of automation in the test life cycle (planning to documentation). Furthermore information about the concrete test run can be generated more easily and systematically[10-20]. The test engineer usually derives test input variables from the model by hand and supplements the associated result. With the aid of test automation and a corresponding model, test cases can now be derived automatically[3]. The focus of current model-based testing approaches is to generate a whole test case with test steps (see Fig. 1). It is often used for software testing with no real-time restrictions or parallel execution.

3 Functional validation of previously unknown

Classic test cases consist of several test steps. Each test step links a stimulation to specific functional validation criteria (see Fig. 1). In the following, an approach is described which dissolves the fixed coupling between stimulation and validation in order to evaluate previously unknown stimulation sequences.

3.1 Generation of previously unknown stimulation sequences

In the body domain, many systems are based on a causeeffect principle. This means that there is a known system reaction to a defined trigger, signal change or timing behaviour which is written down in the requirements, too (e. g. When the air conditioning is switched on (cause), it must be displayed in the infotainment system (effect)).

Extended functionalities can often be described by a dedicated sequence of triggers. In conventional, classic test approaches, this procedure is fixed by the sequence of test steps and cannot be influenced during the test. However, if you try to simulate the real framework conditions in the vehicle, at the beginning of the journey the decision to switch on the air conditioning is not clear to the driver or front passenger. Rather, the driver decides on the basis of other influencing factors and external stimulation (e. g. external smells) or on the basis of other system functions (e. g. whether the window is currently open) to switch the air conditioning on or off. Dynamic and randomly driver and passenger behaviour cannot be mapped in the context of predefined test steps. According to references [11] [22] and [7], there are approaches in the DAS world where the ego behaviour is influenced by a driver, environmental and traffic model in its driving behaviour (digital test drive). A driver or passenger behaviour model can also be added in the area of body domain to generate system stimulation. The exact sequence of the stimulation is not known at the beginning of the test run. This should simulate a behaviour that is as realistic as possible.

3.2 Activation—and check conditions

A different validation approach is required for the evaluation of previously unknown sequences. It is no longer possible to evaluate each test step individually. In comparison to requirements-based testing, the validation logic has no knowledge of the exact sequence of the respective scenario. Instead, a continuous evaluation approach is required, which does not need any information about the exact order of the stimulation triggers. For this purpose, a test approach was introduced in references [11] and [7], which enables simultaneous and continuous system evaluation based on references [23]. So called "Assessments" are used for the creation of system validation criteria[8]. If you take a closer look to the use of the simultaneous continuous evaluation of system functions, it is mandatory to estimate which assessments need to be active in a certain situation. Therefore, a separation of the assessment into the activation and the actual check condition is proposed. According to references [7], Fig. 2 shows this division. If all activation conditions of an assessment are valid, the check condition is verified. If the check condition is not valid for a certain period of time, an entry is written to the test protocol. At the end of the evaluation, the protocol contains the concrete activation times of the assessment and the related evaluation. The activation conditions are checked in each model cycle[7].

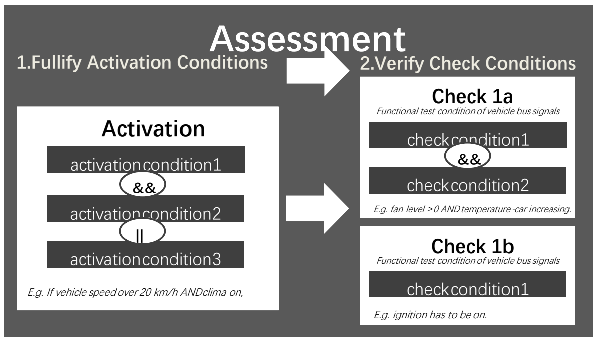

Assessment
1.Fullify Activation Conditions
2.Verify Check Conditions
Activation
activation condition1
&&
activation condition2
||
activation condition3
E.g. If vehicle speed over 20 km/h ANDclima on,
Check 1a
Functional test condition of vehicle bus signals
check condition1
&&
check condition2
E.g. fan level >0 AND temperature -car increasing.
Check 1b
Functional test condition of vehicle bus signals
check condition1
E.g. ignition has to be on.

Fig.2 Continuous event triggered validation concept[7]

Concrete conditions from raw ECU signals or from derived variables, which are determined from the simulation with the help of an observer, can be set up. The test scripts are independent. They do not change or influence the state of the SuT functions[7-11]. Test scripts only monitor the current status of the SuT function[11]. The test concept from references [11] and [7] will increase the effective test time. For each model cycle or test step, several evaluations are executed in parallel. This is one way of testing a system continuously on a real-time constrained test bench[11].

4 Automatic functional validation of SuT

According to reference [7], driving functions are already evaluated with an continuous event-triggered test concept on a HiL system. The creation of evaluation criteria is difficult and requires a lot of knowledge of the existing code framework. Cross relationships between the assessments are difficult to establish. In order to simplify this process, parts of the modelbased testing were applied to this evaluation concept as a new achievement.

4.1 Use and benefit

The desired target state of the new model-based assessment generation (simulated by the model) was compared with the actual state of the SuT functionality. It is possible to evaluate the system behaviour against the data from the created model. The model accesses the real control unit signals and to extended system information, which is provided by an observer. This extended system information provides additional referencing data for the simulation environment. In order to keep the intelligibility of the model as low as possible, the entire functionality does not have to be mapped in a single system model. Several models can be executed in parallel or can be divided into sub-states to structure the system model. This ensures clarity and maintainability. The requirement for the system model is not to reproduce the test system exactly, but to find a simplified representation of the system in order to quickly find relevant and potentially critical points. The structure of the model should always be kept as simple as possible and the concerns should be separated into several simpler sub-models. The focus should be placed on the simplification and derivation of system behaviour models (see Section 6).

4.2 Systematic assessment generation

As a first step, a system and behaviour model for the SuT function must be created. Existing models, which have already been created in the product development cycle, can be used or new models can be built with the help of requirements (step 1). The system model should map the basic functionalities of SuT and show simple logical relationships. When modeling, the system should be divided into basic states. The evaluations are mainly used to find errors on the SuT. The creation of the system model is described in Section 5).

In the next step, the system model is converted into a uniform machine-readable data format (step 2). This guarantees the use of every code generation tool regardless of the source of the model (e. g. Simulink/Stateflow). Existing standards of representation such as UML can be used here[24-25]. Assessments are generated from the system model in a standardized form. To create the assessments, abstract assessments (if-then-structure) are generated first (step 3). The states and transitions of the created model are used to derive assessments. For this purpose, the states serve as activation conditions ("if"-part) on the one hand and as check conditions ("then"-part) on the other [26]. The following assessments are created for a system with two states (state 1 and state 2) and two transitions (transition 1-2 and transition 2-1).

• Activation condition from state:

- If state 1 is active, then the condition of transition 2-1 must be checked;

- If state 2 is active, then the condition of transition 1-2 must be checked.

• Activation condition from transition:

- If transition 2-1 is active, then the condition of state 1 must be checked;

- If transition 1-2 is active, then the condition of state 2 must be checked.

Many interdependencies in the vehicle body domain consist of two system states with the corresponding transition conditions. That is why a bus idle state system is used as an example later in Section 5. There are two states bus active and bus inactive, between which a change is made depending on the current system state (transitions). Every transition and every state lead to an assessment. The system model never represent the real SuT functionality exactly. There are always differences between the output of the system model for the test creation and the implemented SuT functionality, because of simplifications or unknown thresholds. The exact timing behaviour can never be reproduced on test infrastructures with real time conditions (e. g. HiL), because of missing signals (internal signals of the control unit), unknown specifications and other sensor information. For this purpose intermediate assessments are introduced, which map the transition between two defined states.

In the next step "if-then clauses" are translated into the corresponding code for the required test automation tool or framework (step 4). In the last step, a library was generated, which was derived from the code and was easily integrated into the existing tool environment (step 5). In this environment assessments were performed, evaluated, and logged. The test generation process is shown in Fig. 3.

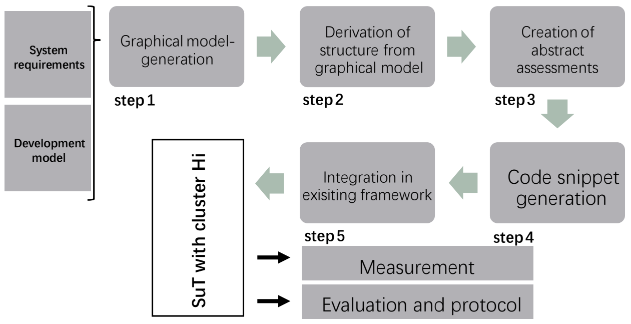


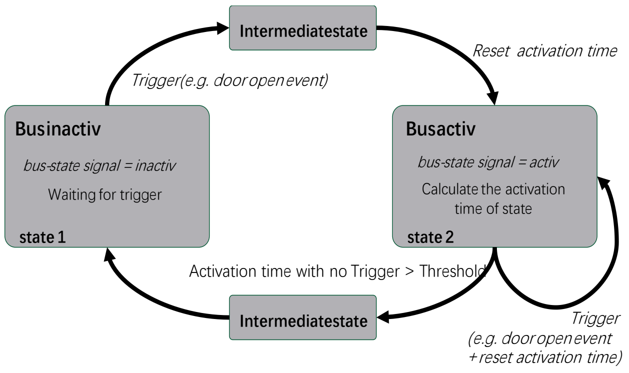
Fig.3 Assessment generation process

Due to the model-based creation, a link between assessments and model was achieved. This increases the consistency. In Section 5, the process is applied to a body function on a HiL system at a German car manufacture as an example.

5 Evaluation and application

The new methodology was applied to system function in the vehicle body domain for a proof-of-concept. In order to reduce the power consumption in the vehicle, the control units fall asleep after an inactive period without a trigger signal. If all control units have fallen asleep, this is indicated by a signal on the vehicle bus. If a trigger occurs, the control unit wakes up and the data signals are sent again via the data bus. The corresponding bus signal changes its value and indicates the current state of the bus. It was decided to use the "bus-sleep" function, because it is a widely used function in the vehicle body domain. The model of the "bus-sleep" system consists of two main states with its transitions. The bus wakes up by the hazard warning lights or a door trigger event only. The system and behaviour model of the SuT function is seen in Fig. 4.

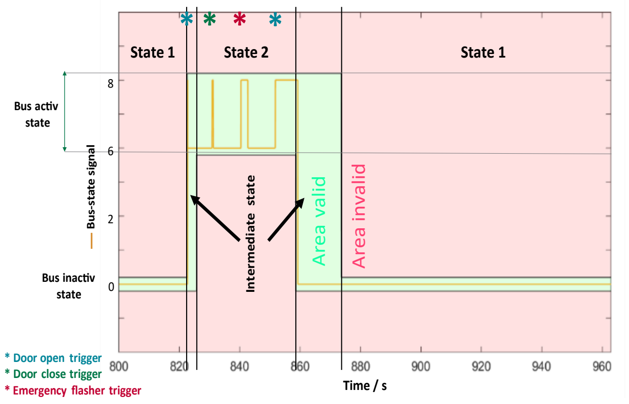
This system model is used for the automated



The vehicle SuT function is stimulated by a random generated trigger sequence.

Fig. 4 Proof-of-concept system and behaviour model bus-sleep

derivation of assessments. These assessments can be imported into the existing test automation tool framework on the HiL system at a German car manufacture. The system assessments are mapped in the corresponding real-time code and check the system function continuously. The system is stimulated using previously unknown triggers, which are intended to simulate real driver behavior. These triggers (door open trigger, door close trigger and emergency flasher trigger) occur in any order and sequence. The parallel evaluation makes it possible to evaluate the system over a longer period of time and to collect further information about the correct bus sleep behaviour (see Fig. 5).



The Astrix shows the trigger events on the bus, which stimulates a bus wake up.

Fig. 5 Signal state-time-plot with invalid and valid areas derived from system and behaviour model and subdivision in different model states

One check per transition and one check per status were created (see Section 4). The intermediate status was omitted from the code generation. It was generated for the model shown in Fig. 4. The implementation of time behavior and tolerance times is particularly useful in order to map transitions and intermediate states. As an example the following abstract assessment are generated:

• If *activation time with no Trigger is greater than Threshold*, then the *bus—state should be inactive*;

• If *a trigger occurs*, then *the bus—state should become active*.

In the time series shown in Fig. 5, the following triggers lead to the corresponding system states (model), which activates the corresponding tests.

Specifically, the random sequence of trigger signals is generated on the HiL system. The system is supposed to check SuT functionality in different initial states. For unambiguousness, the system model was included in the protocol with additional descriptions for evaluation reasons (see Tab. 1).

Tab.1　Timing sequence for related test sequence

Sim. time/s	Trigger event	State transition	Rel. ass.
822	door open	state 1 to intermediate	2
830	door close	stay in state 2	2
840	emergency flasher	stay in state 2	2
855	door open	stay in state 2	2
875	no trigger	change to state 1	1

This increases the traceability and creates a link between the specific assessment and the system and behaviour model. In the proof-of-concept, the system was monitored for one hour using the presented approach. 30 activations were carried out. Each signal was evaluated continuously. An external system triggers the individual signals in a random manner. To test the assessment generation tool, randomly three types of triggerevents were intentionally injected into the vehicle body domain system function, which are evaluated with the help of the created assessments. In traditional requirements-based testing, only five test steps with five evaluations are executed in a similar test case at the same time. The effective test time was used better on the HiL test bench. At the same time, the continuous evaluation concept offers the possibility to evaluate several systems in parallel. Therefore, the bus sleep assessments are also adopted on other system test cases in order to monitor the bus sleep behaviour of the control units in normal test operation.

Changes are quickly incorporated into the existing model during the assessment design. For example, adding further triggers or adapting the timing behaviour only requires small parameter adjustments in the model and does not have to be carried out on many different assessments or test cases as before. By using the system and behaviour model and the automated generation of assessments, conclusions about the system behaviour are generated quickly and easily. The methodology is particularly well suited for evaluating endurance tests, long-term tests and stimulation with a predefined unknown sequence.

If the manual requirement-based generation is compared with the automatic model-based creation, the following advantages are mentioned for the automated creation process:

• The effective test-time on test platforms can be increased with real-time coupling, because of parallel testing of more than on functionality.

• System function assessments can be transferred to other tests easily. It is possible to execute them in parallel to other generated stimulations or already existing test cases (reusability).

If the full potential of the process should be used, this approach has to be used during development, too. For this purpose, a system model can be derived from a development model, which can be transferred to assessments and executed on different test instances.

6　Conclusion and outlook

In software development, model-based approaches are a practiced and widely used test method. With the help of an automatic creation of assessments, the effort and the susceptibility to evaluation irregularities are reduced. With a continuous event-based system evaluation, a simultaneous evaluation of several SuT functions can only be carried out with great effort over a longer period of time. In combination with the simultaneous assessment execution, the large number of assessments generated by the model-based approach is not problematic anymore (RQ1). The methodology presented in this paper makes systematic and comprehensive system evaluation easier and more effective than the manual creation of evaluations. The test process presented in Section 3 represents a new type of automation process for body domain functions (RQ2). It has been shown that the presented process works with simple models. In further work this process is to be transferred to more connected and encapsulated models.

After creating an initial version, the tool chain is used to check the model with recorded measurements initially. Relevant signals can be saved in a measurement and loaded into a resimulation environment. It is possible to adapt the system model and optimize it. This is used to map influences that have not been taken into the model so far. The resimulation environment is not bound to the real-time limitation of the test instance (e. g. HiL) and can therefore be carried out in a resource-saving manner. This means that new versions of the model can be developed quickly. Specifically, a resimulation environment is to be created in which tests and stimulations can be derived automatically on the basis of a behaviour model - according to the method presented in this paper. In addition, a process should be created which transfers the test concept to further development stages. It should be evaluated to what extent this method can also be used on Software-in-the-loop or real-world test platforms.

References:

[1] JAKOBSON O. Core based service oriented architecture[EB/OL]. (2019-11-01) [2021-05-01]. https://dspace. com/shared/data/pdf/dwc2019/Volvo-Ola Jakobson.pdf.

[2] GUISSOUMA H, KLARE H, SAX E, *et al*. An empirical study on the current and future challenges of automotive software release and configuration management [C]// IEEE 2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). [S. l.]: IEEE, 2018: 298.

[3] POFAHL E, WIESSALLA J, HOFMANN O, *et al*. Modellbasierte erzeugung von testfallen¨ mit integrierter modellbasierte erzeugung von testfallen¨ mit integrierter fehleranalyse [BE/OL]. [2021-05-03]. https://www. semanticscholar. org/paper/Modellbasierte-Erzeugung-von-Testf%C3%.

[4] SCHNITTENHELM H. Testing of level-3 systems [Z]. PEGASUS Projekt, 2017: 470.

[5] TUNCALI C E, FAINEKOS G, PROKHOROV D, *et al*. Requirements-driven test generation for autonomous vehicles with machine learning components[J]. IEEE Transactions on Intelligent Vehicles, 2020, 5(2): 265.

[6] SCHONEMANN V, WINNER H, GLOCK T, *et al*. Scenario-based functional safety for automated driving on the example of valet parking [C]// ARAI K, KAPOOR S, BHATIA R. Advances in Information and Communication. [S. l.]: Springer International Publishing, 2019: 53.

[7] King C, Ries L, Kober C, *et al*. Automated function assessment in driving scenarios[C]// 12th IEEE Conference on Software Testing, Validation and Verification. [S. l.]: IEEE, 2019: 414.

[8] OTTEN S, BACH J, WOHLFAHRT C, *et al*. Automated assessment and evaluation of digital test drives [C]// ZACHAUS C, M¨ ULLER B, MEYER G. Advanced Microsystems for Automotive Applications 2017. [S. l.] : Springer International Publishing, 2017: 189.

[9] BROY J. Modellbasierte entwicklung und optimierung flexibler zeitgesteuerter architekturen im fahrzeugserienbereich[D/OL]. Karlsruhe: Universitat Karlsruhe, 2010. https:// publikationen.bibliothek.kit.edu/1000021274.

[10] SHOKRY H, HINCHEY M. Model-based verification of embedded software[J]. Computer, 2009, 42(4): 53.

[11] GUSTAFSSON T, SKOGLUND M, KOBETSKI A, *et al*. Automotive system testing by independent guarded assertions [C]// 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation workshops (ICSTW 2015). Piscataway, NJ: IEEE, 2015: 1. https://www.diva-portal.org/ smash/get/diva2:1043558/FULLTEXT01.pdf.

[12] Kober C. Stochastische verkehrsflusssimulation auf basis von fahrerverhaltensmodellen zur absicherung automatisierter fahrfunktionen [M]. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.

[13] International Organization for Standardization. Road vehicles—functional safety: ISO 26262-1: 2011[S/OL]. [2021-05-04]. https://www.iso.org/standard/43464.html.

[14] SAX E. Automatisiertes Testen eingebetteter Systeme in der Automobilindustrie[R/OL]. Munchen: Hanser, 2008. http:// www. hanser-elibrary. com/action/showBook? doi=10.3139/9783446419018.

[15] STARON M. Automotive software development [C]// STARON M. Automotive Software Architectures. [S. l.] : Springer International Publishing, 2017: 51.

[16] CAMPERO W. Testkonzept IEEE 829 testmanagement nach istqb standard[J]. Qytera, 2019.

[17] PEGASUS. PEGASUS method [EB/OL]. 2020. [2021-05-20]. https://www.pegasusprojekt.de/.

[18] IEEE Computer Cociety. IEEE standard for software and system test documentation: IEEE. 829–2008[S]. 2008. [2021-05-20]. https://ieeexplore.ieee.org/document/5983353.

[19] WITTE F. Testmanagement und softwaretest: Theoretische grundlagen und praktische umsetzung [M]. Wiesbaden: Springer Vieweg, 2016. http://dx.doi.org/10.1007/978-3-658-09964-0.

[20] Tech Pvt Ltd. Model based testing tutorial: What is, tools & example[G]// Model Based Testing Tutorial. [S. l.]: Tech Pvt Ltd, 2020.

[21] BACH J. Methoden und ansatze fur die entwicklung und den test pradiktiver fahrzeugregelungs fun ktionen[Z]. KIT Bib, 2018.

[22] WOHLFAHRT C. Von systematischer absicherung zur digitalen erprobungsfahrt[Z]. Stuttgart, 2016-10-27.

[23] TRIOU E, ABBAS Z, KOTHAPALLE S. Declarative testing: A paradigm for testing software applications [C]// Information Technology: New Generations, 2009. ITNG '09. Sixth International. [S. l.]: IEEE Xplore, 2009: 769.

[24] WAYKAR Y. A study of importance of uml diagrams: with special reference to very large-sized projects [C/OL]// International Conference on Reinventing Thinking Beyond Boundaries to Excel. 2013. https://www. researchgate. net/publication/322991896 A Study of Importance of UML diagrams With Special Reference to Very Large-sized Projects.

[25] ANKE J, BENTE S. Uml in der hochschullehre: Eine¨ kritische reflexion[EB/OL]. 2019 [2021-05-25]. http://www.researchgate.net.

[26] DANIEL F, EDUARD E, WASIF A, *et al*. From natural language requirements to passive test cases using guarded assertions[C/OL]// 2018 IEEE International Conference on Software Quality, Reliability and Security (QRS). [S. l.]: IEEE, 2018. DOI:10.1109/QRS.2018.00060.