



Tests für automatisierte und autonome Fahrsysteme

Wiederverwendung aufgezeichneter Fahrten ist nicht zu rechtfertigen

Alexander Pretschner¹ · Florian Hauer¹ · Tabea Schmidt¹

Angenommen: 19. April 2021 / Online publiziert: 1. Juni 2021
© Der/die Autor(en) 2021

Zusammenfassung

Der Test automatisierter und autonomer Fahrsysteme ist aus verschiedenen Gründen sehr schwierig, u. a. wegen der Abwesenheit klarer Spezifikationen der zu testenden Systeme, wegen Unklarheit über die Auswahlkriterien für „gute“ Testfälle und wegen des Fehlens von Testendekriterien. Ein populärer Ansatz schlägt deswegen vor, aufgezeichnete Fahrten unabhängig vom zu testenden System als Testfälle zu verwenden. Wir zeigen, dass dies im Wesentlichen einem rein zufälligen Test des Systems gleichkommt, dass dieselbe aufgezeichnete Testfahrt dementsprechend einmal einen „guten“ und einmal einen uninteressanten Testfall darstellen kann und dass es notwendig ist, für jedes neue System neue Testfälle abzuleiten. Die simulationsbasierte heuristische Suche nach Testfällen auf Basis sogenannter Szenariotypen, die typische Verkehrssituationen beschreiben, stellt eine Lösung des Problems dar.

Abstract

Testing automated and autonomous driving systems is challenging for various reasons, e.g., because of the absence of clear and precise specifications of the system under testing, because of the unclear test selection criteria for “good” test cases, and because of missing test exit criteria. As a consequence, a popular approach suggests reusing recorded test drives as test cases regardless of the system being tested. We show that this is generally equivalent to purely random testing of the system, that a recorded test drive may be sometimes “good” and sometimes irrelevant, and that it is necessary to derive new test cases for each new system. A solution to this issue is a simulation-based heuristic search of test cases for so-called scenario types that capture recurring traffic situations.

In Industrie, großen Verbundprojekten und Zertifizierungsstellen wird derzeit die direkte Verwendung aufgezeichneter Fahrten für den Test automatisierter und autonomer Fahrsysteme diskutiert. Wir zeigen, dass dieser Ansatz höchstens als in seiner Effektivität unklares Komplement zu systemspezifisch erzeugten Testfällen dienen kann, wenn zer-

tifizierungsrelevante Begründungen für Güte und Vollständigkeit von Testsuites als notwendig erachtet werden.

Automatisierte Tests von Fahrerassistenzsystemen und autonomen Fahrzeugen spielen eine große Rolle bei der Bewertung der Sicherheit solcher Systeme. Ein zunehmend häufiger aufgegriffener Vorschlag ist, aus einer großen Menge aufgezeichneter Fahrten mit speziellen Auswahlkriterien oder zufällig auszuwählen und diese Untermenge aufgezeichneter Fahrten als Testfälle zu verwenden [1–5]. Wenn man den Anspruch erhebt, „gute“ Testfälle zu verwenden und somit über den – in Teilen gerechtfertigten – Anspruch zufällig ausgewählter Tests hinausgeht, kann dieser Ansatz allerdings nicht schlüssig begründet werden. Stattdessen müssen im Allgemeinen systemspezifisch neue Testfälle generiert werden, was durch die intelligente Instanziierung sogenannter Szenariotypen mit heuristischer Suche möglich ist.

✉ Alexander Pretschner
alexander.pretschner@tum.de

Florian Hauer
florian.hauer@tum.de

Tabea Schmidt
tabea.schmidt@tum.de

¹ Lehrstuhl für Software und Systems Engineering, Technische Universität München, Boltzmannstraße 3, 85748 Garching b. München, Deutschland

Unser Argument besteht aus 3 Schritten. **Erstens** wiederholen wir die Doppelfunktion des Tests zur Überprüfung von Funktionalität und zum Finden von Fehlern und zeigen, dass ein sinnvoller Begriff „guter“ Testfälle in einer fehlerbasierten Lesart gründet – letztlich weil nichtfehlerbasierte Ansätze keine Diskriminierung zwischen Testfällen zulassen können und dementsprechend kein Testfall „besser“ als ein anderer ist. **Zweitens** erklären wir die Idee des Aufzeichnens von Fahrten und des Wiederabspielens solcher Fahrten als Testfälle – und argumentieren, warum dies problematisch ist: Eine solche aufgezeichnete Fahrt kann etwa einen Spurwechselassistenten provozieren, die Spur zu wechseln – oder eben nicht. Wenn der Spurwechsel nicht erfolgt – ist der entsprechende Test sinnvoll? Und wenn er erfolgt, ist die entsprechende Situation in irgendeiner Weise gegenüber anderen Spurwechseln ausgezeichnet und insofern „besser“ als eine andere Situation bzw. ein anderer Testfall? Unsere Antwort ist „nein“: Grob gesagt, ist ein guter Test für einen VW Käfer nicht automatisch ein guter Test für einen Ferrari. Das bedeutet, dass aufgezeichnete Fahrten/Testfälle nicht „blind“ wiederverwendet werden sollten. **Drittens** zeigen wir, wie das Problem mit testlingspezifischer Testfallgenerierung aus Szenariotypen gelöst werden kann.

Erstens: Gute Testfälle sind fehlerbasiert!

Testen hat einerseits den Zweck, die Übereinstimmung eines Systems mit seiner Spezifikation zu zeigen. Es ist bekannt, dass Testen ein endlicher und teurer Prozess ist und dass dieser Nachweis deswegen zwingend immer nur für wenige ausgewählte Abläufe des Systems möglich ist. Andererseits ist der komplementäre Zweck des Testens, Fehler zu finden. „Gute“ Testfälle sind dann solche, die potenzielle Fehler mit guter Kosteneffektivität finden: *Potenziell* müssen in dieser Definition die Fehler sein, weil bei Annahme *tatsächlicher* Fehler sonst keine guten Testfälle für ein hypothetisches fehlerfreies System existieren könnten; und die *Kosteneffektivität* bezieht sich auf die Schwere und Auftretenswahrscheinlichkeit des Fehlers, aber auch auf den Aufwand der Testerstellung und -durchführung [6]. Diese beiden dualen Sichtweisen umfassen auch das risikobasierte und das anforderungsbasierte Testen: Risikobasiertes Testen ist getrieben durch Auftretenswahrscheinlichkeit und Schwere potenzieller Fehler. Beim anforderungsbasierten Testen wird zwar nicht die Schwere eines potenziellen Fehlers explizit berücksichtigt, aber implizit die als häufiger ausgeführte angenommene Funktionalität eines Systems (nämlich die, für die Anforderungen explizit formuliert wurden, siehe dazu unten).

Die Idee *potenzieller Fehler* findet sich entsprechend in den meisten Testauswahlkriterien wie Grenzwerttesten oder

kombinatorischem Testen; ebenfalls zu finden ist die Idee in statischen Analysen oder beim Fuzzing von Situationen, in denen nach Arrayüberläufen, Null Pointer Exceptions oder Divisionen durch Null gesucht wird; und sie zeigt sich auch in der Idee der Fehlerinjektion zur Bemessung der Qualität von Testfällen. In der Praxis am verbreitetsten ist wohl die Idee des Grenzwerttestens. Hier versucht man, Eingabe- oder interne Variablen mit empirisch besonders interessanten, weil in der Vergangenheit häufig fehlerhaften, Werten zu belegen. Wenn die entsprechenden Typen der Variablen geordnet sind, sind dies häufig die Enden der entsprechenden Intervalle: MININT oder MAXINT; NEGINFTY oder POSINFTY; leere Liste bzw. NULL; usw. Bei Divisionen versucht man einerseits, eine Null im Nenner zu erzwingen, und andererseits, sofern das nicht möglich ist, Programmabläufe mit Werten nahe dieser Null im Nenner zu provozieren und das Verhalten des Systems zu beobachten.

Aus Sicht des praktischen Testers ist bei diesem Ansatz sehr hilfreich, dass bei dieser fehlerbasierten Art des Testens per definitionem eine starke Einschränkung der möglichen (Eingabe-)Werte erfolgt, nämlich auf die genannten wenigen Extremwerte. Beim anforderungsbasierten wie auch dem strukturbasierten Testen fehlt ein solcher Filter: alle Abläufe eines Typs sind zunächst einmal äquivalent. Diese Äquivalenzklassen von Abläufen entstehen häufig durch die Zerlegung des Eingaberaums in aus Anwendungssicht sinnvolle Klassen, etwa implementierte Szenarien oder User Stories. Manchmal entstehen sie auch, wenn Codeabdeckungskriterien für die Auswahl von Tests herangezogen werden. Häufig gelingt es dann, die Werte in den entsprechenden Äquivalenzklassen zu ordnen und dann wiederum Testfälle an den „Rändern“ dieser Klassen auszuwählen. Man geht auf diese Weise vor, weil die Erfahrung zeigt, dass Probleme üblicherweise an den Rändern von Intervallen stattfinden – und wenn man sich für diese Art der Testauswahl entscheidet, dann basieren die zugrundeliegenden Tests wiederum auf einer Fehlerhypothese.

Wenn eine solche Fehlerhypothese nicht begründet werden kann, sind tatsächlich alle Elemente einer Klasse oder sogar des gesamten Eingabedatenraums bzgl. der a-priori-Fehlerwahrscheinlichkeit äquivalent und Testfälle können zufällig aus ihrer Klasse oder dem gesamten Eingabedatenraum ausgewählt werden. Wie „gut“ diese Testfälle dann sind, ist allerdings unklar – im Gegensatz zu den auf Fehlerhypothesen basierenden Testverfahren, die Eingabedaten mit a-priori erhöhten Wahrscheinlichkeiten potenzielle Fehler zu finden auswählen, beispielsweise Eingabedaten am Rand von Intervallen. Diese Argumentation gilt auch für den Test von Fahrerassistenzsystemen: Wenn wir begründen wollen, warum ein Testfall „besser“ als ein anderer ist, bedarf es einer Annahme über potenziell auftretende Fehler.

Zweitens: Aufgezeichnete Fahrten, also Testfälle, sollten nicht automatisch für den Test autonomer Fahrzeuge wiederverwendet werden!

Aufgezeichnete Fahrten aus Perspektive des Testlings (des Ego-Fahrzeugs) sind Beschreibungen seiner Umwelt. Für einen Spurwechselassistenten sind das die für die Spurwechselentscheidung relevanten Fahrzeuge in der Umgebung. Ein typisches Ziel ist es, zu testen, ob der Spurwechselassistent bei einem Spurwechsel ausreichend Abstand zu diesen Fahrzeugen der Umgebung hält. Hierfür wird der tatsächlich gehaltene Abstand mit einem vorgegebenen Sicherheitsabstand verglichen.

Nehmen wir im Folgenden an, dass während der Ausführung eines Testfalls festgestellt wird, dass der Spurwechselassistent beim Spurwechsel den Sicherheitsabstand – z. B. zu einem Vorderfahrzeug auf der Zielspur – unterschreitet (siehe Abb. 1) und dass das zugrundeliegende Testziel die Minimierung des Abstands zum Vorderfahrzeug beim Spurwechsel ist. Intuitiv ist dies ein nützlicher Testfall, da er ein Fehlverhalten des Systems aufzeigt. Nehmen wir weiterhin an, das zugrundeliegende Problem im System wird dadurch behoben, dass der Planer des Spurwechselassistenten „defensiver“ eingestellt wird, d. h. nur bei großen Abständen zum Vorderfahrzeug die Spur wechselt. Wenn nun die neue Systemversion des Spurwechselassistenten getestet werden soll, kann man auf die Idee kommen, denselben Testfall, der zuvor Fehlverhalten aufgezeigt hat, wiederzuverwenden – und genau das passiert, wenn einmal aufgezeichnete Fahrten unabhängig vom System zum Testen verwendet werden.

Bei Verwendung desselben Testfalls wird sich in unserem Beispiel zeigen, dass die neue Systemversion bei Ausführung dieses Testfalls überhaupt keinen Spurwechsel mehr durchführt (siehe Abb. 2), da ja der Assistent „defensiver“ eingestellt wurde und daher in dieser Situation auf den Spurwechsel verzichtet. Der zuvor nützliche Testfall erfüllt für die neue Systemversion unser Testziel, den Abstand zum Vorderfahrzeug beim Spurwechsel zu minimieren, nicht mehr, weil ja gar kein Spurwechsel stattfindet (und selbst wenn ein Spurwechsel doch stattfände, könnten wir nicht wissen, ob das zweite Kriterium der Minimalität

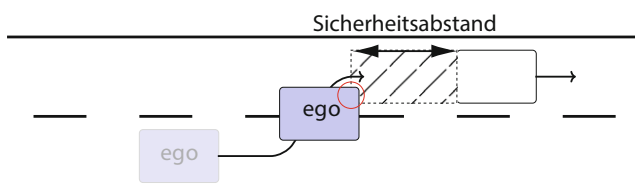


Abb. 1 Schematische Darstellung eines Spurwechsels des Testlings (des Ego-Fahrzeugs) und des Sicherheitsabstandes zum Vordermann auf der Zielspur. Der Testling verletzt den Sicherheitsabstand zum Vordermann

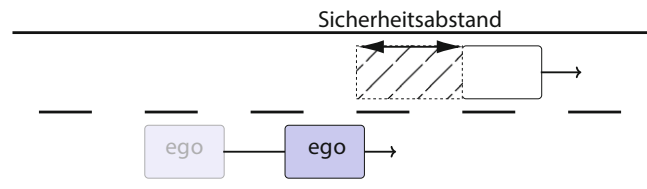


Abb. 2 Schematische Darstellung eines Spurwechsels des Testlings (des Ego-Fahrzeugs) und des Sicherheitsabstandes zum Vordermann auf der Zielspur. Nachdem das Problem im System behoben wurde, führt der Testling keinen Spurwechsel mehr durch

des Abstands erfüllt ist und ob der Testfall in dem Sinn „gut“ wäre, dass er eine *minimale* Distanz zum Vorderfahrzeug provoziert). Weil der Test also für das neue System sein Testziel nicht mehr erfüllt, können wir nicht davon ausgehen, dass er gut ist! Dieses Beispiel zeigt, dass die Güte von Testfällen systemspezifisch ist: Sie können „gut“ für eine Version des Systems und „schlecht“ für eine andere Systemversion sein! Wiederverwendete Testfälle sind in diesem Sinn vor erneuter Ausführung i. A. nicht unterscheidbar von zufällig ausgewählten Testfällen. Sie sollten daher generell nicht „blind“ ausgeführt werden [7], wenn man die Ansicht teilt, dass zufällige Tests die systematischen Tests der uns interessierenden sicherheitskritischen Systemen vielleicht komplementieren, aber nicht ersetzen sollten. Dabei spielt es keine Rolle, woher ein wiederverwendeter Testfall stammt, ob dieser also per Hand erstellt, per automatischer Testfallerzeugung generiert oder eben aus aufgezeichneten Fahrdaten extrahiert wurde. Das Wiederabspielen von Fahrten als Testfälle ist nicht besser als zufällig zu testen und kann – soweit es das Ziel ist, nachgewiesen besser als zufälliges Testen zu sein – nicht schlüssig begründet werden.

Man könnte argumentieren, dass der nun keinen Spurwechsel mehr provozierende Testfall T trotzdem „gut“ ist, weil er ja auch eine Anforderung abtestet, nämlich die, in genau dieser Situation keinen Spurwechsel durchzuführen. Das ist zunächst plausibel. *Es gibt jedoch keinen einsichtigen Grund, diesen Testfall T irgendeinem anderen Testfall vorzuziehen, der ebenfalls keinen Spurwechsel provoziert* – und in diesem Sinne ist T nicht mehr „gut“ bzw. „besser“ als zufälliges Testen! Die Auswahl von T ist nicht fehlerbasiert, nicht risikobasiert und auch nicht anforderungsbasiert, weil es i. A. keine spezifische Anforderung gibt, die genau diese eine Situation beschreiben würde (und als zusätzliche Erschwernis in genau diesem Fall ist es in der Praxis auch noch so, dass keine explizite Spezifikation des Spurwechselassistenten existiert).

Zusammengefasst müssen Testfälle i. A. offenbar systemspezifisch erzeugt werden, wenn eine Änderung des Systems erfolgt ist, die mit der im Testfall adressierten Funktionalität zusammenhängt. Im Bereich des Regressions-tests ist das selbstverständlich: Diese Tests beziehen sich

auf die (hoffentlich) *nicht* modifizierte Funktionalität des Systems. Wenn nun im Test automatisierter Fahrerassistenzsysteme einmal aufgezeichnete Fahrten gleichsam als Referenztestsuite für alle Systeme verwendet werden, dann ist das nicht nur wegen unserer theoretischen Herleitung problematisch, sondern widerspricht auch direkt der Praxis von Regressionstests.

Drittens: Testlingspezifische, gute Testfälle können fehlerbasiert mit heuristischer Suche generiert werden!

Ein vielversprechender und seit Langem in der Literatur bekannter Ansatz zur Generierung von Testfällen [8–11] basiert auf der Idee, abstrakte Szenariotypen in konkrete Szenarien zu instanziierten [12] und dabei eine die „Güte“ des Testfalls codierende Zielfunktion zu optimieren, etwa den Abstand zu umgebenden Fahrzeugen zu minimieren. Szenariotypen (Abb. 3) sind typische Situationen, etwa ein Spurwechsel auf der Autobahn, für die wesentliche Parameter und deren Wertebereiche festgelegt werden, beispielsweise die Anzahl und Breite der Spuren, die Krümmung oder die Anzahl und Position der umgebenden Fahrzeuge (die parametrisierten Szenarien in Abb. 3). In einer XiL-Simulation werden dann iterativ und durch heuristische Suche gezielt unterschiedliche Werte für diese Parameter ausprobiert, um den Wert einer Zielfunktion zu optimieren, also einen Grenzwert zu erreichen (siehe Abb. 3). In unserem Beispiel ist die Zielfunktion, die Distanz zu mindestens einem umgebenden Fahrzeug zu minimieren. Das Resultat, eine Szenarioinstanz in Abb. 3, stellt einen gemäß unserer Definition „guten“ Testfall dar: Wenn eine Szenarioinstanz gefunden wird, in der ein minimaler Sicherheitsabstand zu einem umgebenen Fahrzeug unterschritten wird, wurde ein tatsächlicher Fehler im System gefunden. Falls keine solche Unterschreitung gefunden wird, ist der Testfall im Sinn des Grenzwerttestens trotzdem „gut“, weil er

sich einer potenziellen Unterschreitung annähert und damit einen *potenziellen* Fehler abtestet! Und im Gegensatz zu einem wiederverwendeten Test, dessen Effektivität nicht bemessen werden kann, liegt hier eine aus Zertifizierungssicht wünschenswerte klare Argumentation für die Sinnhaftigkeit und Notwendigkeit dieses generierten Tests vor.

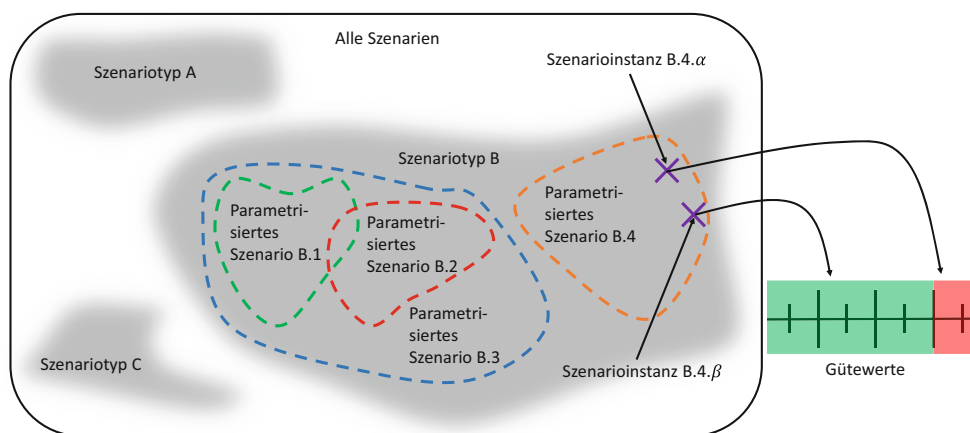
Zusammenfassung und Ausblick

Wir haben argumentiert, dass Testfälle insbesondere im Bereich sicherheitskritischer Systeme begründbar „gut“ sein sollten. Gute Testfälle adressieren potenzielle Fehler und sollten deswegen nicht für beliebige Varianten oder Revisionen eines Systems wiederverwendet werden, wenn sie ihre „Güte“ behalten sollen. Bei Änderungen des Systemverhaltens sind wiederverwendete Testfälle i. A. nicht mehr „gut“ oder es gibt keine Argumentationsgrundlage für ihre „Güte“. Dabei spielt es keine Rolle, ob die Testfälle aus aufgezeichneten Fahrten extrahiert, manuell erstellt oder automatisch generiert wurden. Wir haben am Beispiel autonomer und automatisierter Fahrssysteme argumentiert, aber die Überlegungen gelten allgemein für cyberphysikalische Systeme.

Wir haben per Analogie gesehen, dass auch Regressionstests modifiziert werden müssen, wenn sich die abgetestete Funktionalität ändert. Testfälle müssen daher systemspezifisch neu generiert werden. Dies ist heutzutage mit szenariobasierter Testfallerzeugung mit heuristischer Suche theoretisch und praktisch möglich.

Relevante Fragestellungen beziehen sich nun (1) auf die Vollständigkeit der Kataloge von zu testenden Szenariotypen, (2) auf die Vollständigkeit des relevanten Parametersatzes dieser Szenariotypen, (3) auf die Unmöglichkeit der heuristischen Suche, globale Optima zu identifizieren und (4) auf die praktische Notwendigkeit der Definition von System- und Umgebungsmodellen für die Simulation. Die Punkte (1) und (2) betreffen letztlich die in al-

Abb. 3 Verschiedene Arten von Szenarien: Abstrakte Szenariotypen können zu parametrisierten Szenarien formalisiert werden. Einzelne Parameterkombinationen eines Szenariotypen stellen Szenarioinstanzen dieses Szenariotypen dar und repräsentieren die ausführbaren Testfälle. Deren Güte kann durch eine Zielfunktion bestimmt werden. Besonders interessant sind Testfälle im Grenzbereich, welche die „guten“ Testfälle beschreiben



len Bereichen des Testens relevante Frage nach fundierten Testendekriterien. Szenariotypen werden üblicherweise von Experten definiert; offensichtliche Kriterien für deren Vollständigkeit existieren nicht. Wir glauben, dass hier der fundamentale Nutzen aufgezeichneter Fahrten zu finden ist. Einerseits kann man berechnen, wie viele aufgezeichnete Fahrten notwendig wären, um die Vollständigkeit einer Liste von Szenariotypen nachzuweisen [13]; andererseits kann man durch automatisiertes Clustering von existierenden Fahrten Szenariotypen automatisiert ableiten [14] und die auf diese Weise erstellten Kataloge mit den manuell erstellten Katalogen vergleichen. Dies kann Hinweise auf Unvollständigkeiten im Datenmaterial oder in den manuell erstellten Katalogen von Szenariotypen geben; es kann ebenfalls bei der Identifikation relevanter Parameter helfen, etwa ob die Feuchtigkeit der Straße eine Rolle für den Test des Systems spielt. Insgesamt sehen wir sehr gute Anhaltspunkte, dass die szenariobasierte Testfallgenerierung eine fundierte und zertifizierungsrelevante Grundlage für Argumentationen über Güte und Vollständigkeit erfolgter Tests sicherheitskritischer Systeme bietet.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

Literatur

1. Bach J, Holzäpfel M, Otten S, Sax E (2017) Reactive-Replay approach for verification and validation of closed-loop control systems in early development. SAE Technical Paper
2. Lages U, Spencer M, Katz R (2013) Automatic scenario generation based on laserscanner reference data and advanced offline processing. IEEE Intelligent Vehicles Symposium Workshops, S 146–148
3. Minnerup P, Kessler T, Knoll A (2015) Collecting simulation scenarios by analyzing physical test drives. IEEE Intelligent Transportation Systems Conference (ITSC), S 2915–2920
4. Wachenfeld W, Junietz P, Wenzel R, Winner H (2016) The worst-time-to-collision metric for situation identification. IEEE Intelligent Vehicles Symposium
5. Wagner S, Groh K, Kühbeck K, Dörfel M, Knoll A (2018) Using Time-To-React based on Naturalistic Traffic Object Behavior for Scenario-Based Risk Assessment of Automated Driving. IEEE Intelligent Vehicles Symposium
6. Pretschner A (2015) Defect-based testing. Dependable Software Engineering
7. Hauer F, Pretschner A, Holzmüller B (2020) Re-using concrete test scenarios generally is a bad idea. IEEE Intelligent Vehicle Symposium (IV)
8. Abdessalem RB, Nejati S, Briand L, Stifter T (2016) Testing advanced driver assistance systems using multi-objective search and neural networks. 31st IEEE/ACM International Conference on Automated Software Engineering (ASE), S 63–74
9. Althoff M, Lutz S (2018) Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. IEEE Intelligent Vehicles Symposium (IV), S 1326–1333
10. Bühler O, Wegener J (2003) Evolutionary functional testing of an automated parking system. Proceedings of the International Conference on Computer, Communication and Control Technologies (CCCT) and the 9th. International Conference on Information Systems Analysis and Synthesis (ISAS)
11. –. „Fitness Functions for Testing Automated and Autonomous Driving Systems.“ International Conference on Computer Safety, Reliability, and Security, 2019: 69–84
12. Menzel T, Bagschik G, Maurer M (2018) Scenarios for development, test and validation of automated vehicles. In: IEEE Intelligent Vehicles Symposium 2018
13. Hauer F, Schmidt T, Holzmüller B, Pretschner A (2019) Did we test all scenarios for automated and autonomous driving systems? IEEE Intelligent Transportation Systems Conference (ITSC), S 2950–2955
14. Hauer F, Gerostathopoulos I, Schmidt T, Pretschner A (2020) Clustering traffic scenarios using mental models as little as possible. IEEE Intelligent Vehicles Symposium 2020