

## Testfall-Erkennung in Erprobungsfahrten

Julian Fuchs (1), Christopher Kober (2)

(1) FZI Forschungszentrum Informatik  
Haid-und-Neu-Str. 10–14, 76131 Karlsruhe

(2) Mercedes-Benz AG, E/E Testing Prüfstand  
Werk 059, HPC L334, 71063 Sindelfingen  
fuchs@fzi.de  
christopher.kober@mercedes-benz.com

**Abstract:** Zur Absicherung assistierender und automatisierender Fahrzeugfunktionen werden bei der Mercedes-Benz AG Erprobungsfahrten und Fahrzeugtests basierend auf Testfallkatalogen eingesetzt. Bei Erprobungsfahrten, seien sie real oder im Hardware-in-the-Loop-(HiL)Umfeld, werden große Mengen an Fahrzeugdaten aufgezeichnet. Diese Erprobungsfahrten beinhalten zu einem großen Teil die gleichen Manöver und Fahrscenarien, die in Prüfgeländeversuchen oder Systemtests am HiL-Prüfstand durchgeführt und analysiert werden. In diesem Beitrag soll eine Methodik vorgestellt werden mit deren Hilfe Testfälle aus Erprobungsfahrten extrahiert und ausgewertet werden können. Dadurch kann eine höhere Testabdeckung erreicht und Testzeit eingespart werden. Die Methodik umfasst die Auswahl repräsentativer Erprobungsfahrten, die Extraktion relevanter Testfälle aus den aufgezeichneten Daten, die Abstraktion der Testfälle aus den Testfallkatalogen sowie die Entwicklung und Anwendung von Analyse- und Auswertungsalgorithmen. Die Evaluation erfolgt anhand verschiedener Kriterien wie Testabdeckung, Testzeitersparnis und Überprüfung der Genauigkeit der extrahierten Testfälle im Vergleich zu den ursprünglichen Testfallkatalogen prototypisch am Beispiel diverser Fahrfunktion der Mercedes-Benz-AG.

## 1 Einleitung

Die fortschreitende Komplexität moderner Fahrzeugsysteme und der stetige Anstieg der Anforderungen an Sicherheit und Zuverlässigkeit erfordern eine immer umfassendere Validierung und Verifikation der eingesetzten Technologien. Um sicherzustellen, dass alle relevanten Situationen abgedeckt werden, ist es entscheidend, eine hohe Testabdeckung zu gewährleisten. Die bisherigen Ansätze im Testprozess sind jedoch oft zeit- und ressourcenintensiv, insbesondere wenn Tests auf dem Prüfgelände durchgeführt werden müssen [1] [2] [3]. Zudem besteht die Herausforderung, dass nicht alle realen Fahrbedingungen in den definierten Testfällen erfasst werden.

Hier setzt die entwickelte Methodik an, die darauf abzielt, aufgezeichnete Realdaten aus Erprobungsfahrten systematisch auf das Vorkommen von Testfallinhalten zu untersuchen. Durch diese Methodik wird es möglich, bereits in realen Fahrsituationen durchgeführte Tests zu erkennen und so die Testabdeckung zu erhöhen. Dadurch können nicht nur Testaufwände reduziert werden, sondern es wird auch eine effizientere Nutzung der vorhandenen Ressourcen ermöglicht.

## 2 Herausforderungen und Zielsetzung

Die Implementierung dieser Methodik steht jedoch vor mehreren Herausforderungen:

- **Abstraktion der Testfälle:** Testfälle aus den bestehenden Testfallkatalogen müssen so abstrahiert werden, dass eine automatisierte Erkennung ihres Inhalts in den aufgezeichneten Realdaten möglich ist. Dies erfordert eine sorgfältige Analyse und Modifikation der Testfallbeschreibungen, um die essenziellen Merkmale herauszufiltern und sie auf ein Niveau zu bringen, das für die automatisierte Verarbeitung geeignet ist.
- **Integration in bestehende Datenpipelines:** Die Methodik soll in die bestehende Datenpipeline integriert werden, was technologische und organisatorische Anpassungen mit sich bringt. Die nahtlose Einbindung in bestehende Prozesse und Systeme ist entscheidend, um den maximalen Nutzen zu erzielen und Reibungsverluste zu vermeiden.
- **Datenmengen und Visualisierung:** Die Menge an Daten, die bei Erprobungsfahrten generiert wird, stellt eine weitere Herausforderung dar. Um die relevanten Informationen effizient zu verarbeiten und darzustellen, müssen Konzepte und Mockups für eine geeignete Visualisierung entwickelt werden. Diese sollen es den Ingenieuren und Entwicklern ermöglichen, die wesentlichen Informationen schnell zu erfassen und fundierte Entscheidungen zu treffen.

Das übergeordnete Ziel ist die kontinuierliche, systemunabhängige Erkennung und Bewertung von Testfällen, unabhängig von der jeweiligen Testinstanz und der zu bewertenden Funktionalität.

### 3 Konzept zur Abstraktion von Testfällen zur Anwendung auf realen Daten

Normalerweise bestehen Testfälle aus mehreren Testschritten. Diese Testfälle werden sequentiell nacheinander ausgeführt. Testfahrer erhalten ihren klaren Testauftrag und führen diesen aus. Es wird dabei meist nur eine Funktionalität gleichzeitig bewertet. Zusätzlich werden Erprobungsträger in Abend- und Wochenenderproben genutzt, um Daten aufzuzeichnen und Auffälligkeiten zu finden. Hierbei fallen große Datenmengen an, deren manuelle Auswertung mit einem großen zeitlichen und ressourcentechnischen Aufwand verbunden ist [4].

#### 3.1 Ermittlung von Aktionen und Bewertungen auf Basis der Testspezifikation

Als Eingangsgröße in das entwickelte Konzept dienen Testspezifikationen und aufgenommene Zeitreihen, der Fahrzeugbusse (oder auch von Referenz Sensorik). Dabei werden zunächst die bestehenden Testspezifikationen genutzt und die einzelnen Testschritte unterteilt. Für jeden Testschritt werden Aktionen und Bewertungen definiert. Die Aktionen werden auf Basis der Stimulationen im Testschritt (bspw. Fahrer schnallt sich ab) festgesetzt. Die Bewertungen beschreiben den idealen Soll-Zustand der Systemfunktionalität. Jede Aktion hat einen Start- und einen Endzeitpunkt, genauso wie eine Bedingung (Aktivierungsbedingung), die erfüllt sein muss, damit die Aktion aktiv sein kann (bspw. Fahrer ist abgeschnallt). Jede Bewertung hat genauso wie eine Aktion einen Start- und Endzeitpunkt und eine Bedingung, die erfüllt sein muss, damit die Bewertung aktiv ist, hat aber zusätzlich noch ein Bewertungskriterium, das ein Testergebnis (pass/ failed) hat [5] [6].

Die Testfälle werden in einer Abstraktionsmatrix dargestellt, die zur Erstellung einer analytischen Testfalldatenbank verwendet werden kann. Die Abstraktionsmatrix soll dabei an die Testfälle und deren Testschritte gekoppelt werden können, um eine Parametrisierung der Analyseskripte zu gewährleisten. Auch Metainformationen, wie Testfall ID, werden in der Abstraktionsmatrix gespeichert.

Sobald die Aktionen und Bewertungen definiert wurden, kann in die zyklische Abarbeitung der Messdaten gestartet werden. Dieser Prozess besteht aus vier Schritten (siehe Abbildung 1).



Abbildung 1: Konzept zur Abstraktion der Testfälle

## 3.2 Extraktion von Aktionen und Bewertungen

Anschließend werden die Aktionen und Bewertungsbedingungen auf die Auffälligkeiten angewandt. Eine Aktion ist dann aktiv, wenn die in Kapitel 3.1 ermittelten Bedingungen erfüllt sind. Das heißt konkret, wenn die Signalwerte des aktuell betrachteten Zeitpunkts den ermittelten Werten entsprechen. Der Zeitpunkt, bei dem die gefundene Aktion zum Ersten Mal aktiv ist, wird als Startzeitpunkt protokolliert. Sobald die Bedingung wieder nicht erfüllt ist, wird der Endzeitpunkt geschätzt und eine gefundene Aktion-Instanz abgespeichert.

Genauso passiert dies bei den Bewertungen. Auch hier wird zunächst überprüft, ob die Aktivierungsbedingung erfüllt ist. Sobald diese Bedingung erfüllt ist, wird der Zustand der Signale überwacht und mit dem Bewertungskriterium verglichen. Wird das formulierte Bewertungskriterium innerhalb der Aktivierungszeit verletzt, wird die Bewertung als „failed“ markiert – ansonsten als „passed“ (vgl. Ansatz nach King et al. [7])

Zum Schluss erhält man eine Liste über gefundenen Aktionen und Bewertungen. Die Bewertungen können dabei unabhängig von der späteren Aggregation zu Testfällen zur Bewertung der Systemperformance bzw. -Reifegrad verwendet werden.

## 3.3 Zusammenfassen von Aktionen und Bewertungen zu Testfällen

Im nächsten Schritt werden die Aktionen und Bewertungen automatisch zu Testfällen zusammengebaut und für jeden Testfall ein Ergebnis ermittelt. Hierfür ist die zeitliche Abfolge von Aktivierung und Deaktivierung entscheidend. Die sequentielle Kombination von mehreren Aktionen untereinander beschreiben den Testfall. Mit Hilfe eines Suchalgorithmus werden passende Sequenzen an Aktionen in der Zeit gefunden. Hierzu wird die Liste an in der Messung gefundenen Aktionen für jeden Testfall separat untersucht. Anschließend werden die gültigen Bewertungen für jeden gefundenen Testfall überprüft und ebenso wie die gefundenen Aktionen zum Testfall hinzugefügt. Das Gesamtergebnis des Testfalls bildet sich aus einer logischen Verknüpfung der Teilergebnisse der gefundenen Bewertungen (siehe beispielsweise Abbildung 2: Beispielhafte Abstraktion).

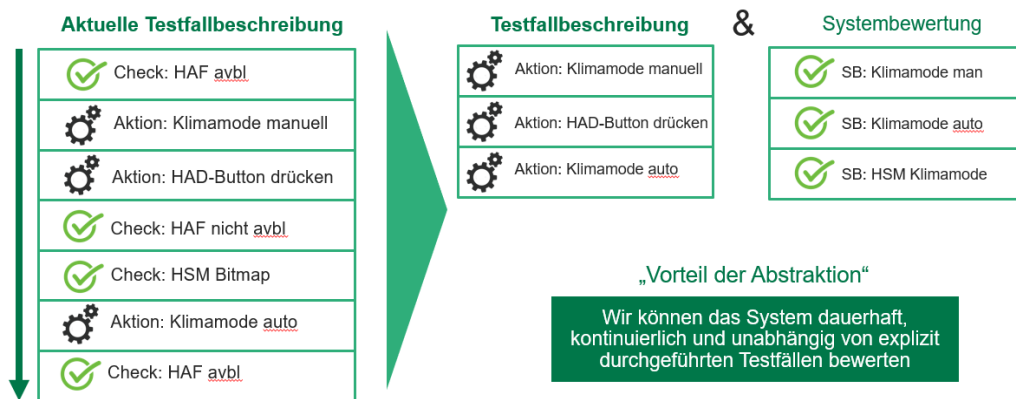


Abbildung 2: Beispielhafte Abstraktion in Testfallbeschreibung (Aktion) und Systembewertung

Ein Testfall kann somit mehrmals in einer Messung gefunden werden und wird zusammen mit den gefundenen Aktionen, Bewertungen und dem aggregierten Ergebnis dann in der in Kapitel 4 eingeführten Datenstruktur abgelegt.

### 3.4 Visualisieren der Teilergebnisse und manuelle Auswertung

Die Visualisierung der Teilergebnisse kommt nach dem 4. Schritt aus Abbildung 1. Sie ermöglicht es den Ingenieuren, schnell und effizient einen Überblick über die gefundenen Testfälle, Aktionen und Bewertungen zu erhalten. Das Ziel der Visualisierung ist es, die Ergebnisse so aufzubereiten, dass eine intuitive und übersichtliche Auswertung möglich wird.

Das zentrale Element der Visualisierung ist ein Dashboard, das die wesentlichen Informationen zu den durchgeführten Testfahrten und den dabei erkannten Testfällen aggregiert darstellt. Hierbei werden die Ergebnisse in Form von Diagrammen, Tabellen und interaktiven Grafiken präsentiert. Das Dashboard ist modular aufgebaut und erlaubt es, unterschiedliche Ansichten zu wählen, abhängig von der Rolle des Nutzers (z.B. Testingenieur, Entwickler, Management).

Die Ergebnisse können nach verschiedenen Kriterien gefiltert und sortiert werden, z.B. nach Testfahrt, Fahrzeugmodell, Datum der Erprobungsfahrt oder nach spezifischen Testfällen. Eine Zeitachse zeigt die zeitliche Abfolge der erkannten Aktionen und Bewertungen innerhalb einer Testfahrt, wodurch Auffälligkeiten und potenzielle Probleme leicht identifiziert werden können.

## 4 Strukturierung der Daten

Zur strukturierten Ablage der Ergebnisse wurde eine relationale Datenbank aufgebaut, welche in verschiedenen Tabellen die erkannten Aktionen, Bewertungen und Testfälle zu einer Messung speichert. Zusätzlich werden Meta-Informationen zu jeder Messung abgespeichert, um Informationen zur Baureihe, zur verwendeten Software, etc. auch im Nachgang zur Verfügung zu haben. Ziel der Struktur ist eine filter- und sortierbare Speicherung.

Der zentrale Bestandteil des Entitätsdiagramm ist die Liste der Testfahrten (test\_drives), hier werden pro Testfahrt Meta Informationen, wie Länge der Testfahrt, das Datum und die Dauer der Testfahrt, sowie eine Referenz auf ein entsprechendes Fahrzeug gespeichert werden.

Jeder Testfall besteht neben einer eindeutigen ID und einem Namen wiederum aus mehreren Aktionen (actions) und Bewertungen (assessments). Wird nun eine Messung ausgewertet, werden Aktionen (extracted\_actions) und Bewertungen (extracted\_assessments) gefunden, die zu gefundenen bzw. extrahierten Testfällen gespeichert werden. Jeder gefundene Testfall (extracted\_test\_case) ist somit eindeutig mehreren Aktionen und/oder Bewertungen zugeordnet und gehört eindeutig zu einer Messung. Die Datenablage dient als Basis für die Visualisierung und Speicherung der Daten.

## 5 Evaluation

Zwischen den beiden Testinstanzen, die in dieser Methode gemeinsam ausgewertet werden können, besteht der gravierende Unterschied der Referenzsensorik, also einer zusätzlichen Informationsquelle zur Überprüfung der eigentlichen Fahrzeugsensoren. In den Daten aus Fahrten, die am HiL-Prüfstand durchgeführt wurden, und woher die Grundzüge dieser Methode stammen, ist diese Referenzsensorik über die bekannte Simulationsumgebung vorhanden. Dies erhöht die Möglichkeiten einer Aktivierung ungemein, da die Umwelt und der Zustand des Fahrzeuges zu jeder Zeit bekannt sind.

Bei den Daten aus realen Erprobungsfahrten ist keine Referenzsensorik verbaut, das heißt die Implementierung von Aktionen und Bewertung muss sich auf die vorhandenen Bus-Signale stützen. Zusätzliche Informationen, die der Testfahrer bereitstellen könnte, sind schwierig automatisiert zu verarbeiten und mit einiger Ungenauigkeit behaftet, weshalb auch darüber keine Lösung der fehlenden Referenzsensorik erreicht wird.

Aufgrund der, durch die oben beschriebene Problematik, reduzierten Möglichkeiten der Bestimmung der Aktivierung der einzelnen Aktionen kann die Methode bei realen Testfahrten nicht die gleiche Aussagekraft und den Umfang der Prüfungen im Vergleich zu Fahrten in simulierten Umgebungen liefern. Nichtsdestotrotz wurde ein wertvoller erster Schritt getätigt, der mit geringem Aufwand zusätzliche Erkenntnisse aus vorhandenen Fahrdaten liefern kann.

## 6 Ausblick

Es gibt verschiedene Möglichkeiten, wie die Methode in der Zukunft erweitert werden könnte. Zum einen wäre es von Vorteil, die Algorithmen der Methode direkt in das Cloud Portal zu integrieren, in dem die Messdaten von Realfahrten abgelegt werden. Dies würde die Anwendung erleichtern und die Methode einem breiteren Publikum zugänglich machen, als es derzeit bei der „offline“ Auswertung der Fall ist.

Eine Möglichkeit, das Problem der fehlenden Referenzsensorik anzugehen, wäre eine Auswertung von Videoaufzeichnungen der Sensor-Kameras. Hier könnte beispielsweise ein neuronales Netz zur Erkennung verschiedener Situationen eingesetzt werden. Implementierung und Auswertung wären aber sehr aufwändig, darüber hinaus würde man zu einem gewissen Maße, die Fahrfunktion „nachprogrammieren“, die ja ebenfalls über Sensoren das Umfeld klassifiziert und Situationen bewertet.

Zusätzlich könnten GPS-Positionen, die während der Fahrt aufgezeichnet werden, auf interessante oder kritische Straßenpunkte untersucht werden. Dadurch würde sich eine zusätzliche Informationsquelle ergeben, die möglicherweise mehr Aktivierungen oder Prüfungen ermöglicht (Abbildung 3).



Abbildung 3: Aktionen in Bezug zur Kartenposition

## 7 Literaturverzeichnis

- [1] R. Pfeffer, P. Ukas und E. Sax, „Potential of virtual test environments for the development of highly automated driving functions using neural networks,“ *ATZ Fahrerassistenzsysteme*, 2018.
- [2] E. Sax, *Tatort Test „Test is not the Last“*, Stuttgart: Vector Testing Symposium, 2015.
- [3] Waymo, „Waymo Autonomous Vehicle Disengagement Report 2017,“ Waymo, 2017.
- [4] C. Wang und H. Winner, „Overcoming Challenges of Validation Automated Driving and Identification of Critical Scenarios,“ in *IEEE Intelligent Transportation Systems Conference, ITSC*, 2019.
- [5] S. Otten, J. Bach, C. Wohlfahrt, C. King, J. Lier, H. Schmid, S. Schmerler und E. Sax, „Automated Assessment and Evaluation of Digital Test Drives,“ in *Advanced Microsystems for Automotive Applications 2017*, Karlsruhe, Springer Verlag, 2017.
- [6] C. King, J. Bach, S. Otten und E. Sax, „Identifikation von Fahrszenarien während einer virtuellen Testfahrt,“ in *INFORMATIK 2017*, Chemnitz, 2017.
- [7] C. King, L. Ries, C. Kober, C. Wohlfahrt und E. Sax, „Automated Function Assessment in Driving Scenarios,“ in *IEEE International Conference of Software Testing, Verification and Validation*, 2019.