

Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen

Dipl.-Wirtsch.-Ing. Fabian Schuldt, Dipl.-Ing. Falko Saust,
Dr.-Ing. Bernd Lichte, Prof. Dr.-Ing. Markus Maurer

Institut für Regelungstechnik

Hans-Sommer-Straße 66, 38106 Braunschweig
{schuldt,saust,lichte,maurer}@ifr.ing.tu-bs.de

Dr.-Ing. Stephan Scholz

Volkswagen AG

stephan.scholz1@volkswagen.de

Kurzfassung

Im Fokus dieses Beitrags steht die Entwicklung eines Testkonzepts, welches dazu geeignet ist, mit Hilfe von systematisch-erzeugten virtuellen Test-szenarien verschiedene Fahrerassistenzsysteme effizient zu testen und validieren. Um die Anwendbarkeit und zugleich den Nutzen des Testkonzepts beurteilen und sukzessive optimieren zu können, wird das Testverfahren praxisnah angewendet. Hierzu werden exemplarisch aus den beiden Forschungsprojekten UR:BAN-KA und Stadtpilot verschiedene Teilfunktionen herangezogen. Das Konzept testet die Assistenzsysteme durch eine Vielzahl von verschiedenen Szenarien, die sich hinsichtlich einzelner Faktoren voneinander unterscheiden. Hierzu gehören unter anderem verschiedenste Straßengeometrien und -topologien, Verkehrssituationen sowie Umwelteinflüsse. Die einzelnen Parameter der Szenarien werden durch kombinatorische Verfahren variiert und auf diese Weise redundanzarme effiziente Testfälle erzeugt. Das Konzept umfasst ebenfalls eine simulationsgestützte automatisierte Testdurchführung und -auswertung. Realtests können durch das Testkonzept nicht vollständig ersetzt werden, jedoch ist es möglich, die Anzahl der notwendigen Realtests erheblich zu reduzieren. Somit können schon zahlreiche Tests in einer frühen Phase der Entwicklung durchgeführt und die Qualität der Systeme verbessert werden.

Einleitung

Das Testen und die Absicherung von software-intensiven Systemen stellt einen wichtigen Schritt im Entwicklungsprozess von Systemen dar. Nur durch eine ausreichende Anzahl von Tests kann gewährleistet werden, dass die Systeme robust, wartbar und zuverlässig sind und somit eine hohe Qualität aufweisen. Die Bedeutung von Tests wird besonders deutlich bei sicherheitsrelevanten Systemen, bei denen kein Fehlverhalten erlaubt ist. Hier muss eine hohe Testtiefe durch ausreichende Tests generiert werden. Aufgrund der steigenden Komplexität der Systeme wird es zunehmend schwieriger, die Systeme hinreichend zu testen und die erforderliche hohe Testtiefe zur Absicherung zu generieren. Dies ist ein Grund dafür, dass der Kostenanteil für die Absicherung oftmals über 50% des Gesamtkostenbudgets einnimmt. Bei sicherheitsrelevanten Systemen kann der Anteil der Kosten durchaus noch höher sein. [Ammann und Jeff, 2008]

Fahrerassistenzsysteme sind ein Beispiel für Systeme, die zur Absicherung ausreichend getestet werden müssen, aber aufgrund der Komplexität immer schwieriger zu testen sind. Assistenzsysteme haben die Aufgabe den Fahrer während der Fahrt zu entlasten und Unfälle zu verhindern. Um diese Funktionen zu erfüllen, greifen die Systeme im Rahmen der Assistenz immer stärker in die Längs- und Querverführung des Fahrzeugs ein. Fehlerbehaftete Eingriffe in die Längs- und Querverführung können zu gravierenden Folgen führen. Aus diesem Grund muss sichergestellt werden, dass ein sicherer und ausgereifter Softwarestand im öffentlichen Straßenverkehr eingesetzt wird, der zu jedem Zeitpunkt ein robustes Verhalten zeigt und Fehleingriffe verhindert. Dies kann wiederum nur gewährleistet werden, wenn eine hohe Testtiefe durch eine ausreichende Anzahl von Testfällen und einem geeigneten Testkonzept generiert wurde.

Aktuell werden Fahrerassistenzsysteme auf verschiedene Arten getestet. Es gibt Ansätze, die Systeme in stehenden Simulatoren zu testen [Wisselmann et al., 2004]. Eine Erweiterung stellen Simulationen mit Bewegungssystemen dar. In diesem Fall können im Simulator die Bewegungen in der Simulation in einem gewissen Rahmen nachgeahmt werden. Ein weiterer Ansatz stellt der Vehicle-Hardware-in-the-Loop (VEHIL) dar, der in [Verburg et al., 2002] im Detail vorgestellt wird. Das Vehicle-in-the-loop (VIL) ist eine weitere Mög-

lichkeit, Fahrerassistenzsysteme zu testen [Bock, 2008].

Assistenzsysteme, die sich in Serie befinden, sind überwiegend für den Bereich außerhalb der Stadt ausgelegt. Um die Zahl und Schwere der Unfälle auch in Städten nachhaltig zu senken, sollen Fahrerassistenzsysteme den Fahrer in Zukunft nicht nur während der Fahrt auf Autobahnen oder Schnellstraßen sondern auch in der Stadt unterstützen. Hierbei ergeben sich besondere Anforderungen durch den urbanen Raum. Der Verkehr ist dichter und die anderen Verkehrsteilnehmer können nicht sicher wahrgenommen werden. Neben ein- und ausscherenden Fahrzeugen müssen abbiegende Verkehrsteilnehmer, sowie das Verhalten des Gegenverkehrs erfasst werden. Kreuzungssituationen stellen durch Längs- und Querverkehr, sowie Abbiegevorgänge komplexe und unübersichtliche Situationen dar.

Der urbane Raum stellt somit einen komplexen Aktionsraum für die Assistenzsysteme dar. Es gibt eine Vielzahl von Einflussfaktoren, die in einem bestimmten Wertebereich situationsabhängig stark variieren können. Aus den Kombinationen der Einflussfaktoren und den jeweiligen Wertebereichen können theoretisch unendlich-viele Testfälle generiert werden. Da jedoch in der Praxis nur eine begrenzte Anzahl von Tests durchgeführt werden kann und jeder Test mit finanziellem und zeitlichem Aufwand verbunden ist, muss darauf geachtet werden, möglichst wenige aber dennoch repräsentative, fehlersensitive, redundanzarme und ökonomische Testfälle zu identifizieren [Liggesmeyer, 2009].

Das in diesem Beitrag vorgestellte Testkonzept arbeitet in dem Sinne effizient, als dass es durch kombinatorische Verfahren die Anzahl der notwendigen Tests nur so lange iterativ erhöht, bis eine ausreichende Testtiefe erreicht wird. Das Verfahren zur systematischen Testgenerierung für Fahrerassistenzsysteme soll sowohl im Rahmen des nationalen Förderprojekts UR:BAN-KA als auch im Projekt Stadtpilot zur Anwendung kommen. Beide Projekte werden im Folgenden kurz vorgestellt.

UR:BAN

Im Forschungsprojekt UR:BAN (Urbane Raum: Benutzergerechte Assistenzsysteme und Netzmanagement) werden im Rahmen der Projektsäule Kognitive Assistenz (KA) Fahrerassistenzsysteme für den urbanen Raum entwickelt. Fahrerassistenzsysteme, die sich aktuell in Serie befinden, unterstützen den

Fahrer hauptsächlich auf Autobahnen oder Schnellstraßen. Die Herausforderungen, die durch den urbanen Raum gestellt werden, sind aktuell noch nicht abgedeckt. Daher soll im Forschungsprojekt UR:BAN-KA die Fahrerassistenz auf den urbanen Raum erweitert werden. Der Fahrer soll in komplexen Situationen, wie beispielsweise Kreuzungen, Engstellen oder beim Spurwechsel unterstützt werden. Es sollen Gefahrensituationen erkannt und drohende Kollisionen durch eine assistierte Längs- und Querführung verhindert werden.[Scholl, 2012]

Stadtpilot

Das Forschungsprojekt Stadtpilot der TU-Braunschweig hat das Ziel eine vollständig autonome Fahrt auf dem Braunschweiger Stadtring zu absolvieren. Das Team setzt sich aus Mitarbeitern und Studenten aus dem Institut für Flugführung und dem Institut für Regelungstechnik zusammen. Bei der autonomen Fahrt sind Situationen wie das Einfädeln in den fließenden Verkehr, Abbiegevorgänge an Kreuzungen und Spurwechselmanöver zu bewältigen.[Saust et al., 2011]

Zum besseren Verständnis wird das erarbeitete Testkonzept am Beispiel eines konkreten Fahrerassistenzsystems erläutert. Hierfür wurde der sogenannte Engstellenassistent ausgewählt, der im Rahmen von UR:BAN-KA entwickelt werden soll. Prinzipiell lässt sich die Vorgehensweise auf alle Fahrerassistenzsysteme übertragen, die in die Längs- und Querführung eingreifen. Die Abbildung 1 zeigt einige urbane Beispielszenarien, die in den Projekte UR:BAN-KA und Stadtpilot, bewältigt werden sollen.

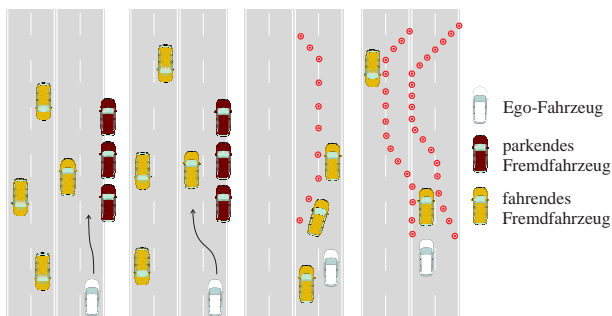


Abbildung 1: Beispielszenarien der Forschungsprojekte UR:BAN-KA und Stadtpilot

Das effiziente Testkonzept

Im vorherigen Kapitel wurde eine Problemstellung erläutert, die sich für das Testen von Fahrerassistenzsystemen im urbanen Raum ergibt. Im Folgenden sollen die Kernbestandteile des effizienten Testkonzepts vorgestellt werden. Das Konzept generiert eine ausreichende Testtiefe für Fahrerassistenzsysteme im urbanen Raum und kann somit einen möglichen Lösungsansatz für das skizzierte Problem darstellen. Das Konzept lässt sich durch die vier Vorgehensschritte: Analyse des Fahrerassistenzsystems auf Einflussfaktoren, generische Testfallgenerierung, Testdurchführung und Testauswertung unterteilen. Die Vorgehensweise innerhalb des Konzepts ist in Abbildung 2 dargestellt.

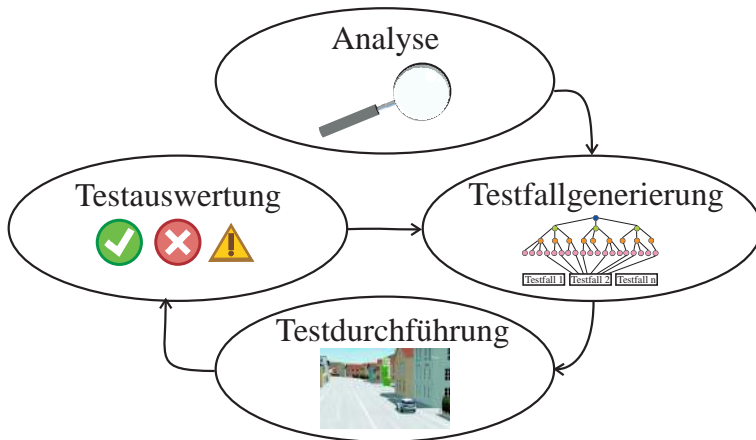


Abbildung 2: Vorgehensweise beim effizienten Testkonzept

1. Analyse des Fahrerassistenzsystems auf Einflussfaktoren

Im ersten Schritt müssen die Faktoren identifiziert werden, die einen Einfluss auf das Assistenzsystem haben. Dabei können vorhandene Spezifikationen oder Szenarienkataloge genutzt werden. Innerhalb der Spezifikationen werden die einzelnen Funktionen definiert, die das Fahrerassistenzsystem im Rahmen der Assistenz erfüllen soll.

2. Generische Testfallgenerierung

Im zweiten Schritt werden anhand der Einflussfaktoren Testszenarien generisch erzeugt. Hierbei werden Methoden des Black-Box-Testings sowie der Kombinatorik angewendet, um redundanzarme und effiziente

Testscenarien zu generieren. Diese werden auf Basis von vier Ebenen aufgebaut.

3. **Testdurchführung**

Die Testdurchführung findet in der Simulation statt. Dabei wird der Umfang der Simulation auf verschiedenen Stufen variiert, um die Assistenzsysteme mit unterschiedlichsten Daten zu stimulieren.

4. **Testauswertung**

Die Testauswertung sollte aufgrund der Vielzahl von erstellten Szenarien automatisiert durchgeführt werden. Es werden verschiedene Kriterien herangezogen, um die Ergebnisse der Tests zu vergleichen. Dabei wird zwischen K.O.-Kriterien sowie weichen Kriterien unterschieden. Zur Bewertung der Kriterien werden verschiedene Metriken eingeführt. Nach der Auswertung können neue Testfälle zur Erhöhung der Testtiefe generiert werden und wiederum mit Testgenerierung begonnen werden.

Die einzelnen Schritte der generischen Testfallgenerierung, Testdurchführung sowie Testauswertung werden in den folgenden Kapiteln detaillierter erläutert.

Generische Erzeugung der Testfälle

In diesem Kapitel werden die einzelnen Schritte der generischen Testfallgenerierung anhand des Fahrerassistenzsystems Engstellenassistent beispielhaft erläutert. Die Szenarien werden auf Basis eines 4-Ebenenmodells aufgebaut. Ein Szenario kann flexibel durch eine Auswahl der vier Ebenen zusammengestellt werden. Statische Szenarien können auch auf Basis der ersten beiden Ebenen definiert werden. Auf jeder Ebene der Szenariengenerierung wird das Prinzip der systematischen Testfallgenerierung angewendet, auf das später im Detail eingegangen wird. Die Abbildung 3 stellt den Ablauf der Generierung der Szenarien auf Basis des 4-Ebenenmodells dar.

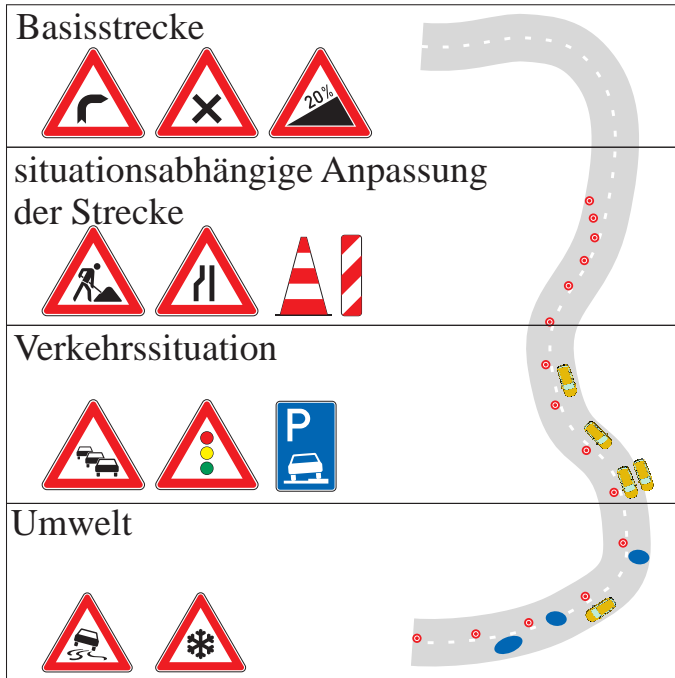


Abbildung 3: Szenariengenerierung auf Basis des 4-Ebenenmodells

Szenariengenerierung auf Basis des 4-Ebenenmodells

1. Straßengeometrie und Topologie

Auf der ersten Ebene der Szenariengenerierung wird die Straßengeometrie sowie Topologie der Basisstrecke definiert. Der Verlauf der Straße orientiert sich bei der Generierung an den geltenden Richtlinien für die Anlage von Stadtstraßen und Zufahrtsstraßen, die beispielsweise minimal zulässige Kurvenradien sowie die maximal erlaubten Quer- und Längsneigungen vorgeben. Zusätzlich werden verschiedene Regelquerschnitte definiert. Diese geben die Anzahl der Fahrstreifen sowie deren Breite und Markierungen vor [Baier, 2007; Durth, 1995; Rohloff, 2008]. Diese und weitere Faktoren können innerhalb des Testaufbaus variiert und kombiniert werden. Die Abbildung 4 zeigt eine mögliche Basisstrecke in der Simulation.



Abbildung 4: Beispiel einer Basisstrecke

2. Situationsabhängige Anpassung der Strecke

Auf der zweiten Ebene werden zu der Basisstrecke situationsabhängige Anpassungen hinzugefügt. Um geeignete Szenarien für beispielsweise den Engstellenassistenten zu generieren, wird die Strecke um Engstellen bzw. Baustellen erweitert. Diese werden nach den geltenden Richtlinien für die Absicherung von Arbeitsstellen an Straßen (RSA) angelegt [Bundesministerium für Verkehr, 2009]. Die Abbildung 5 zeigt eine normgerechte Engstelle nach den geltenden Richtlinien.

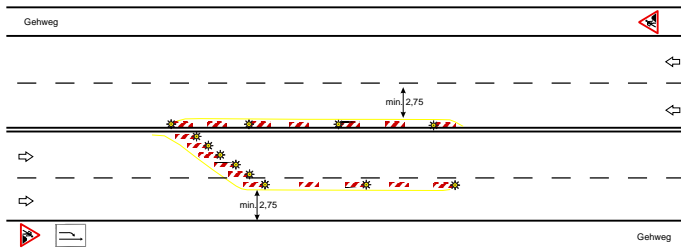


Abbildung 5: Normgerechte Engstelle nach RSA [Bundesministerium für Verkehr, 2009]

Der linke Fahrstreifen wird durch eine Querabspernung verjüngt, deren Verschwenkungsmaß in einem vorgegebenen Rahmen variiert werden kann. Das Verschwenkungsmaß gibt das Verhältnis aus Querversatz und Länge der Querabspernung an. Je nach Querversatz ergeben sich verschiedene Breiten für den rechten Fahrstreifen. Zur Absicherung müssen zusätzlich gelbe Markierungen auf die Straße aufgebracht und statische Leitobjekte zum Szenario hinzugefügt werden. Die Leitobjekte können Leitbaken, Leitkegel oder Leitwände sein, deren Abstand in Längs- und Querrichtung variiert werden kann. Die Basisstrecke liegt unter der neu angepassten Strecke. Die ursprünglichen Fahrstreifenmarkierungen können als widersprüchliche Informationen für die Fahrerassistenzsysteme verwendet werden. Auf dieser Ebene können somit eine Vielzahl von Parametern variiert werden, um die unterschiedlichsten Engstellenszenarien zu generieren. Die Abbildung 6 zeigt eine mögliche Ergänzung der Basisstrecke um eine Engstelle in der Simulation.



Abbildung 6: Erweiterung der Basisstrecke um eine Engstelle

3. Verkehrssituation

Nachdem auf den ersten beiden Ebenen die Basisstrecken sowie deren Erweiterungen und statische Objekte definiert wurden, werden die Sze-

narien auf der dritten Ebene um dynamische Objekte erweitert. Es ist möglich den Verkehrsfluss und -stärke gezielt zu variieren. Der Verkehr wird soweit beeinflusst, dass die Anzahl der übrigen Verkehrsteilnehmer und deren Differenzgeschwindigkeiten zum Ego-Fahrzeug variiert werden können. Zusätzlich werden die zulässigen Geschwindigkeiten sowie die Abstände der Fahrzeuge untereinander variiert. Es werden gezielt Verkehrskonstellationen, wie Folge-Fahrten, Parallelfahrten, Fahrstreifenwechsel und Ein- bzw. Ausschervorgänge erzeugt, die durch zahlreiche Faktoren parametrisiert werden können. Somit können beliebig viele dynamische Situationen erstellt werden. Das Verhalten der anderen Verkehrsteilnehmer ist durch gegebene Parameter individuell einstellbar. In der Abbildung 7 ist eine mögliche Ergänzung der Engstelle um dynamische Verkehrsteilnehmer dargestellt.



Abbildung 7: Erweiterung der Engstelle um dynamischen Objekte

4. Umwelt

Auf der vierten Ebene können regionale Besonderheiten in das Szenario eingebaut werden. Dabei können beispielsweise andere Linienfarben oder spezielle Fahrstreifenbreiten realisiert werden. Die Anpassung an andere Regionen als Deutschland ist allerdings nur möglich, wenn ähnliche Richtlinien vorhanden sind wie die RASt. Zusätzlich

kann auf der vierten Ebene die Tageszeit variiert werden. Somit können Szenarien mit verschiedenen Lichtverhältnissen generiert werden und so getestet und analysiert werden, wie Assistenzsysteme bei Nacht oder Dämmerung performen. Neben der Tageszeit kann auch das Wetter variiert werden und damit das Verhalten der Systeme bei Regen oder Nebel analysiert werden, was besonders bei videobasierten Assistenzsystemen von Bedeutung ist. Wie ein Szenario in der Simulation bei Regen aussehen kann, zeigt die Abbildung 8.



Abbildung 8: Erweiterung der Situation um Wettereinflüsse

Systematische Testfallgenerierung

Auf den einzelnen Ebenen der Szenariengenerierung soll das Prinzip der systematischen Testfallgenerierung angewendet werden. Dieses besteht aus den Schritten Äquivalenzklassenbildung, Grenzwertbetrachtung und kombinatorische Testfallgenerierung. Die Testverfahren sind ausgewählte Verfahren aus dem Pool von Prüfmethoden der Black-Box-Tests. Bei Black-Box-Tests sind die inneren Strukturen (Source Code) des Testobjekts unbekannt. Die Tests werden lediglich anhand der Eingangs- und Ausgangswerte bewertet. Das mit den vorgegebenen Eingangswerten generierte Ergebnis wird zur Bewertung mit den erwarteten Ausgangswerten verglichen und auf Korrektheit überprüft. Die Generierung der Testfälle basiert dabei auf den Anforderungs- und

Schnittstellenbeschreibungen [Hoffmann, 2008]. Die Abbildung 9 zeigt die einzelnen Schritte der systematischen Testfallgenerierung.

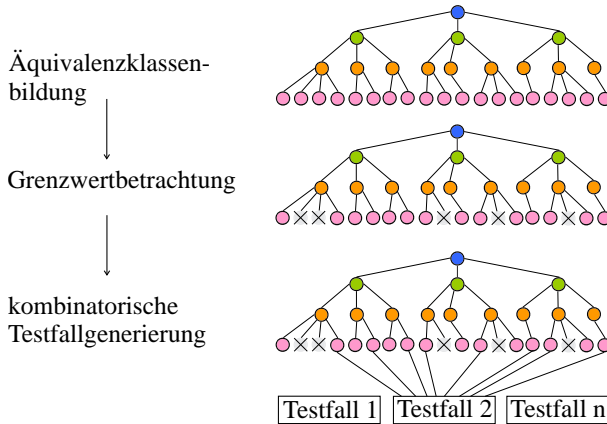


Abbildung 9: Ablauf der systematischen Testfallerzeugung

Äquivalenzklassenbildung

Die Äquivalenzklassenbildung verfolgt das Ziel eine hohe Testabdeckung bei einer geringen Anzahl von Testfällen zu generieren. Dazu wird der Wertebereich aller Einflussgrößen in disjunkte Partitionen (Äquivalenzklassen) aufgeteilt. Zur Testfallgenerierung wird aus jeder Klasse eine beliebige Anzahl von Parameterwerten ausgewählt. Die Partitionierung kann partiell oder vollständig durchgeführt werden. Während bei der partiellen Partitionierung nur gültige Werte verwendet werden, können bei der vollständigen Partitionierung auch Partitionen mit ungültigen Werten gebildet werden. Die Partitionierung der Äquivalenzklassenbildung soll im effizienten Testkonzept zur Vorbereitung der Szenariengenerierung verwendet werden. Die Wertebereiche der zahlreichen Einflussfaktoren, die sich auf die Fahrerassistenzsysteme auswirken können, werden auf diese Weise sinnvoll partitioniert. Es soll eine vollständige Partitionierung durchgeführt werden, da durch die Verwendung von ungültigen Werten die Robustheit des Testobjekts getestet werden kann.

Grenzwertbetrachtung

Die Grenzwertbetrachtung stellt eine Erweiterung der Äquivalenzklassenbildung dar. Es wird die gleiche Partitionierungsstrategie zur Bildung der Äquivalenzklassen verwendet. Jedoch wird eine andere Strategie zur Bildung der

Testfälle verwendet. Im Fall der Äquivalenzklassen werden zufällige Werte zu Testfallgenerierung ausgewählt. Bei der Grenzwertbetrachtung werden die Randwert-Tupel sowie alle Tupel, für die ein einzelner Wert außerhalb der Äquivalenzklasse liegt, für die Testfallgenerierung verwendet. Diese Strategie stellt eine nützliche Erweiterung für die Erstellung der Testfälle dar. Jedoch wird auch deutlich, dass je nach Dimension der Äquivalenzklasse eine große Anzahl von Testfällen erzeugt wird. Die Zahl der Testfälle kann nach folgender Formel berechnet werden:

$$k = 2^n \cdot (n + 1) \text{ für } n: \text{ Dimension der Äquivalenzklasse}$$

Die Grenzwertbetrachtung soll ebenfalls im effizienten Testkonzept verwendet werden. Es sollen auf diese Weise relevante Parameterbereiche für die Einflussfaktoren bestimmt werden und so eine effiziente Vorauswahl der Wertebereiche getroffen werden.

Die Grenzwertbetrachtung kann die Werte der Einflussparameter schon geeignet eingrenzen. Jedoch ergeben sich durch die Kombinationen der übrigen Parameter noch eine große Menge von möglichen Szenarien. Alle Szenarien können aufgrund von zeitlichen Restriktionen nicht getestet werden. Als Konsequenz werden die Szenarien im effizienten Testkonzept durch kombinatorische Überlegungen erstellt. Eine Gegenüberstellung von vorhandenen Strategien wird im folgenden Abschnitt vorgestellt.

Kombinatorische Testverfahren

Kombinatorische Testverfahren verfolgen das Ziel die Parameter von Tests so zu variieren, dass keine redundanten Tests entstehen. Es haben sich verschiedene Verfahren etabliert, die unterschiedliche Abdeckungskriterien erfüllen. Die Abdeckungskriterien können zwischen „each-used“, „pair-wise“, „t-wise“ und „N-wise“ unterschieden werden. Bei einer „each-used“ Abdeckung müssen alle Werte der einzelnen Parameter lediglich nur einmal durch einen Testfall verwendet werden. Diese Abdeckung kann schon mit einer relativ geringen Anzahl von Testfällen erreicht werden. Jedoch kann das Testergebnis nicht als repräsentativ angesehen werden, da eine Vielzahl von Fehlern erst bei bestimmten Kombinationen auftreten und diese nicht mit Sicherheit getestet werden. Ein Abdeckungskriterium, welches repräsentativer ist, ist „pair-wise“. In diesem Fall werden die Testfälle so gewählt, dass alle Werte der Einflussparameter einem Paarvergleich unterzogen werden. Dies bedeutet, dass jeder Wert eines Parameters mit den übrigen Werten eines anderen

Parameters getestet wird. Das Abdeckungskriterium kann erweitert werden, wenn eine t-wise Abdeckung gewählt wird. In diesem Fall werden die Werte einem t-Vergleich unterzogen. Bei „N-wise“ handelt es sich um die absolut vollständige Abdeckung. Um diese Abdeckung zu erreichen, müssen alle möglichen Kombination der Werte der einzelnen Parameter überprüft werden.

Im Bereich des kombinatorischen Testens gibt es zahlreiche Ansätze, die im Folgenden kurz vorgestellt und gruppiert werden. Zusätzlich wird die von dem jeweiligen Verfahren erfüllte Testabdeckung identifiziert. Die Verfahren können je nach Art des zu testenden Systems unterschieden werden. Es gibt Verfahren für deterministische und nicht deterministische Systeme. Ein System ist deterministisch, wenn es bei gleichen Eingabewerten immer die gleichen Ergebnisse liefert. Ist diese Bedingung nicht erfüllt, handelt es sich um ein nicht deterministisches System.

Die Verfahren für nicht deterministische Systeme können wiederum unterteilt werden in Heuristiken, Verfahren des Prinzips „Artificial Life (AL)“ und Zufallstests. Zwei Verfahren aus dem Bereich der Heuristiken stellen die Verfahren „Automatic Efficient Test Generator (AETG)“ und „Simulated Annealing (SA)“ dar. Sowohl AETG als auch SA generieren ein Blockset von Testfällen, die eine „pair-wise“ Abdeckung erfüllen [Cohen et al., 1994, 1997]. Bei der Anwendung von Verfahren aus dem Zweig des AL können auf der einen Seite genetische Algorithmen als auch „Ant Colony Algorithm (ACA)“ [Shiba et al., 2004] verwendet werden. Beiden Algorithmen erfüllen das Kriterium der „pair-wise“ Abdeckung. Bei Zufallstest kann aufgrund der Vorgehensweise keine Aussage über ein Abdeckungskriterium getroffen werden.

Die deterministischen Verfahren können ebenfalls feiner gruppieren werden. Hier können Verfahren unterschieden werden, die die Testfälle instantan und iterativ erstellen. Beim instantanen Aufbau werden alle Testfälle vor der Testdurchführung erstellt. Für die Erstellung können „orthogonal Arrays (OA)“ oder „Covering Arrays (CA)“ verwendet werden [Cohen et al., 2003]. Bei „orthogonal Arrays (OA)“ handelt es sich um zweidimensionale Arrays, wo in jeder Zeile und Spalte eine Zahl nur einmal auftritt. Ein bekannter Vertreter sind die Arrays aus dem Spiel „Sudoku“. Beiden Verfahren erfüllen die „pair-wise“ Abdeckung.

Beim iterativen Aufbau werden die Testfälle während der Testdurchführung erzeugt. Innerhalb dieser Gruppe gibt es zahlreiche Verfahren von denen einigen vorgestellt werden. Ein Basisverfahren ist die „each-choice“-Teststrategie (EC). Die Testfälle werden so gewählt, dass jeder Wert eines Parameters in mindestens einem Testfall vertreten ist. Damit ist das Kriterium der „each-used“ Abdeckung erfüllt. Eine weitere Strategie ist „base-choice“ (BC). Bei dieser Strategie wird von jedem Parameter der „wichtigste“ Wert für einen Testfall verwendet. Die übrigen Testfälle werden so aufgebaut bis eine „each-used“ Abdeckung erreicht ist.

Ein weiteres iteratives Verfahren, welche einen Testkatalog mit „pair-wise“ Abdeckung erzeugt, ist „Combinatorial Auction Test Suite (CATS)“. Es wird eine Startkombination ausgewählt und alle möglichen „pair-wise“ Kombinationen gebildet, die in einer Menge Q abgebildet werden. Im Folgenden wird die Kombination aus der Menge Q für einen Test ausgewählt, die die meisten „pair-wise“ Kombinationen abdeckt. Alle abgedeckten Kombinationen werden aus der Menge Q entfernt. Dies wird solange wiederholt bis die Menge Q keine Kombinationen mehr enthält.[Sherwood, 1994]

Die einzige Strategie, die eine N-wise Abdeckung generiert, ist die Wahl aller möglichen Testkombinationen. Diese Strategie ist bei komplexen Systemen in den meisten Fällen nicht durchführbar, da die Anzahl der Testfälle zu umfangreich ist. Jedoch kann der Test in Einzelfällen als Benchmark zu anderen Strategien verwendet werden.

Fahrerassistenzsysteme stellen aufgrund der vielen Einflussfaktoren und deren weiten Wertebereich komplexe Systeme dar. Jedes System sollte aus Gründen der Sicherheit und Robustheit durch eine Strategie getestet werden, welches eine „N-wise“-Testabdeckung generiert. Jedoch ist dies bei komplexen Systemen nicht mit vertretbarem Aufwand möglich. Aus diesem Grund sollte eine Teststrategie gewählt werden, die eine „pair-wise“ oder „t-wise“ Abdeckung erreicht. In einer Vielzahl von Fällen kann der Fehler auf eine Kombination von zwei Parametern zurückgeführt werden [Liggesmeyer, 2009]. Strategien mit einer „each-used“-Abdeckung sind aufgrund der Komplexität von Fahrerassistenzsystemen ungeeignet.

Damit eignen sich die kombinatorischen Verfahren AETG, OA bzw. CA oder CATS für den Schritt der systematischen Testfallgenerierung in dem hier vorgestellten Testkonzept.

Testdurchführung

Im vorherigen Kapitel wurde der Abschnitt der Testfallgenerierung auf Basis des 4-Ebenenmodells sowie die systematische Testfallgenerierung des effizienten Testkonzepts erläutert. Im Folgenden wird weiter auf die Testdurchführung eingegangen, welche größtenteils in der Simulation durchgeführt wird. Simulationen bieten gegenüber Realversuchen den Vorteil, dass die relevanten Situationen reproduzierbar und gefahrlos validiert werden können. Realversuche können durch Simulationen aber nicht vollständig ersetzt werden und müssen somit weiterhin durchgeführt werden. Die Abbildung 10 zeigt die Ebenen der Simulationen, auf denen die Tests durchgeführt werden.

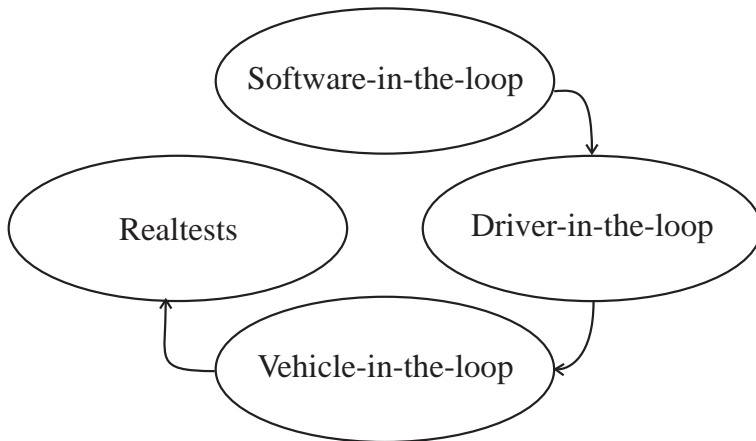


Abbildung 10: Testdurchführung

1. Software-in-the-Loop

Auf der ersten Ebene wird das Assistenzsystem komplett in der Simulation getestet (Software-in-the-loop). Es soll sichergestellt werden, dass das System die vorgegebenen Funktionalitäten erfüllt. Alle benötigten Daten werden für die Tests aus der Simulation generiert. Es werden zum einen idealisierte Sensoren verwendet, aber auch Sensormodelle, die einige reale Effekte abbilden können. Somit kann getestet werden, ob die Funktionalitäten unter variierenden Randbedingungen gewährleistet sind, bzw. wie diese durch die Verfälschung von Daten eingeschränkt werden. Um das Verhalten des Fahrers abzubilden, werden

auf dieser Ebene verfügbare Fahrermodelle verwendet.

2. **Driver-in-the-Loop**

Auf der zweiten Ausbaustufe wird das Fahrermodell durch einen realen Fahrer ersetzt (Driver-in-the-Loop). Dieser wird in einem ortsfesten Simulator mit variierenden Szenarien, die auf Basis des 4-Ebenenmodells erstellt werden, konfrontiert. Damit kann systematisch getestet werden, wie der Fahrer auf die verschiedenen Situationen reagiert und ob das Assistenzsystem in der Lage ist, den Fahrer zu assistieren. Durch den Einsatz des realen Fahrers können Fehler im Assistenzsystem entdeckt werden, die durch die Verwendung des Fahrermodells nicht sichtbar wurden. Zusätzlich wird das komplette System aus Sicht des Kunden getestet.

3. **Vehicle-in-the-Loop**

Auf der dritten Stufe wird das Konzept in einem Vehicle-in-the-loop (VIL) angewendet. Hierbei fährt der Fahrer in einem realen Fahrzeug in einer teil- oder voll-virtuellen Simulation. Der Fahrer erfährt dabei alle Beschleunigungen durch das Fahrzeug und hat somit ein reales Fahrgefühl, was einen großen Vorteil gegenüber dem Driver-in-the-Loop Test aufweist [Bock, 2008]. Durch den Einsatz des VILs sind realitätsnahe Simulationen möglich.

4. **Realtests**

Durch die Tests in der Simulation können keine Realfahrten ersetzt werden. Diese müssen weiterhin durchgeführt werden, um das System unter realen Bedingungen zu testen. Aus diesem Grund sieht das Testkonzept zum Schluss der Testdurchführung weitere Realfahrten ohne den Einsatz der Simulation vor. Jedoch können aufgrund des effizienten Testkonzepts schon zahlreiche Situationen vorab getestet werden, sodass die nötigen Tests für die Realfahrten minimiert werden.

Automatisierte Testauswertung

Ein weiterer wichtiger Bestandteil des effizienten Testkonzepts stellt die Testauswertung dar. Aufgrund der Erzeugung einer großen Anzahl von Testfällen

kann das Testkonzept erst effizient sein, wenn die Testauswertung automatisiert durchgeführt wird und die einzelnen Testfälle miteinander vergleicht.

Für einen aussagekräftigen Vergleich zwischen den Testfällen müssen geeignete Kriterien gefunden werden, die die Ergebnisse der Testfälle vergleichbar machen. Hier kann zwischen harten und weichen Kriterien unterschieden werden, wobei die harten Kriterien auch als K.O.-Kriterien angesehen werden können. Sobald eines der harten Kriterien nicht erfüllt ist, hat das Fahrerassistenzsystem das Szenario bzw. den Test nicht erfolgreich bestanden. Die weichen Kriterien können bei detaillierteren Fragestellungen zur Bewertung herangezogen werden. Sollte nur ein weiches Kriterium bei einem Szenario verletzt sein und alle K.O.-Kriterien erfüllt, hat das Assistenzsystem das Szenario bestanden, jedoch wird die Bewertung für dieses Szenario herabgesetzt. Um die Ergebnisse anhand der Kriterien bewerten zu können, müssen Metriken gefunden werden, die die Ergebnisse messbar machen.

Ein denkbare K.O.-Kriterium des Engstellenassistenten ist der Abstand des Ego-Fahrzeugs zu anderen Verkehrsteilnehmern bzw. -objekten. Anhand dieses Kriteriums kann bewertet werden, wie sicherheitsrelevant die Situation ist. Sobald der Abstand zu anderen Objekte unter einen definierten Wert sinkt, sollte das Assistenzsystem den Fahrer durch die Engstelle assistieren und eine mögliche Kollision mit anderen Verkehrsteilnehmern oder statischen Objekten verhindern. Sollte der Abstand unter einen minimalen Wert sinken oder eine Kollision nicht verhindern werden können, hat das Assistenzsystem das Szenario nicht bestanden. Im Gegensatz zum K.O.-Kriterium Abstand könnte ein weiches Kriterium die Querbeschleunigung des Fahrzeugs sein.

Somit können durch die automatisierte Testauswertung die Ergebnisse der einzelnen Szenarien bewertet und verglichen werden.

Zusammenfassung

Das vorgestellte Testkonzept dient zum effizienten Test von Fahrerassistenzsystemen und automatisiert fahrenden Fahrzeugen, die im urbanen Raum agieren. Um eine ausreichende Testtiefe für einen sicheren Einsatz der Systeme im öffentlichen Straßenverkehr zu generieren, müssen zahlreiche Tests durchgeführt werden. Durch die Anwendung des Testkonzepts wird die Anzahl der benötigten Testfälle für eine ausreichende Testtiefe durch Verwendung von kombinatorischen Verfahren minimiert. Das Testkonzept sieht vor,

dass ein Großteil der Tests in der Simulation durchgeführt werden. Die Simulation stellt zwar keinen Ersatz für reale Tests dar, jedoch können die Systeme aufgrund der Reproduzierbarkeit der Szenarien in der Simulation vorab getestet werden. Somit wird der Reifegrad des Systems erhöht und die Zahl der nötigen Realtests minimiert. Durch das vorgestellte Konzept kann somit der Testprozess effizienter gestaltet werden und die Anzahl der nötigen Tests für eine ausreichende Testtiefe minimiert werden.

Literaturverzeichnis

- Ammann, P. und Jeff, O. (2008). *Introduction to Software Testing*. Cambridge University Press, New York.
- Baier, R. (2007). *Richtlinien für die Anlage von Stadtstraßen (RAS06)*. Forschungsgesellschaft für Strassen- und Verkehrswesen, Köln.
- Bock, T. (2008). *Vehicle in the loop : Test- und Simulationsumgebung für Fahrerassistenzsysteme*. PhD thesis, Cuvillier Verl., Göttingen.
- Bundesministerium für Verkehr (2009). *Richtlinien für die Sicherung von Arbeitsstellen an Straßen (RSA)*. Kirschbaum, Bonn.
- Cohen, D., Dalal, S., Kajla, A., und Patton, G. (1994). The Automatic Efficient Test Generator (AETG) system. In *Proceedings of 1994 IEEE International Symposium on Software Reliability Engineering*, Seiten 303 – 309.
- Cohen, D. M., Society, I. C., Dalal, S. R., Fredman, M. L., und Patton, G. C. (1997). The AETG System: An Approach to Testing Based on Combinatorial Design. In *Transactions on Software Engineering*, Ausgabe 23, Seiten 437 – 444.
- Cohen, M. B., Gibbons, P. B., Mugridge, W. B., und Colbourn, C. J. (2003). Constructing test suites for interaction testing. In *Proceedings of 25th International Conference on Software Engineering*, Seiten 38 – 48.
- Durth, W. (1995). *Linienführung (RAS-L)*. Forschungsgesellschaft für Strassen- und Verkehrswesen, Köln.
- Hoffmann, D. W. (2008). *Software-Qualität*. Springer Verlag.

- Liggemeyer, P. (2009). *Software-Qualität: Testen, Analysieren und Verifizieren von Software*. Spektrum Akademischer Verlag.
- Rohloff, M. (2008). *Richtlinien für die Anlage von Autobahnen (RAA)*. Forschungsgesellschaft für Strassen- und Verkehrswesen, Köln.
- Saust, F., Wille, J. M., Lichte, B., und Maurer, M. (2011). Autonomous Vehicle Guidance on Braunschweig's inner ring road within the Stadtpilot Project. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, Seiten 169 – 174.
- Scholl, W. (2012). Urban Homepage.
- Sherwood, G. (1994). Effective testing of factor combinations. In *3rd International Conference on Software Testing, Analysis, and Review (STAR94)*, Seiten 151 – 166.
- Shiba, T., Tsuchiya, T., und Kikuno, T. (2004). Using artificial life techniques to generate test cases for combinatorial testing. In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, Ausgabe 1, Seiten 72 – 77.
- Verburg, D. J., van der Knaap, A. C. M., und Ploeg, J. (2002). VEHIL: developing and testing intelligent vehicles. In *Intelligent Vehicle Symposium*, Ausgabe 2, Seiten 537 – 544.
- Wisselmann, D., Gresser, K., und Spannheimer, H. (2004). ConnectedDrive - ein methodischer Ansatz für die Entwicklung zukünftiger Fahrerassistenzsysteme. In *Aktive Sicherheit durch Fahrerassistenz*, München.