

Dynamisches und risikobasiertes Fahrwerksverbund-Testverfahren

Von der Fakultät Konstruktions-,Produktions-
und Fahrzeugtechnik der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Ingenieurwissenschaften (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Dominique Xavier Kiefner

aus Bad Cannstatt

Hauptberichter: Prof. Dr.-Ing. H.C. Reuss
Mitberichter: Prof. Dr.-rer. nat W. Hardt

Tag der mündlichen Prüfung: 09.01.2014

Institut für Verbrennungsmotoren und Kraftfahrwesen
der Universität Stuttgart

2014

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit am Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart. Dem Hauptberichter Herr Prof Dr. Ing. Hans Christian Reuss danke ich für seine langjährige Unterstützung und dem Mitberichter Herr Prof. Dr.-rer. nat W. Hardt danke ich für die freundliche Übernahme des Mitberichts.

Bei meinen Kollegen und Kolleginnen möchte ich mich für den Gedankenaustausch und das angenehme Arbeitsumfeld bedanken. Ich möchte auch meinen Eltern Annita und Paul herzlich für ihre Unterstützung danken und auch meiner gesamten Familie für den Zuspruch und die Nachsicht. Meinen Freunden danke ich für ihre Geduld und meinen Kollegen, die auch an ihre Doktorarbeiten geschrieben haben, für das gegenseitige Motivieren.

Ich möchte meiner geliebten Ehefrau Claudia für ihre Unterstützung und ihren Zuspruch danken. Ihr möchte ich diese Arbeit widmen.

Dominique Xavier Kiefner Korb, Januar 2014

Inhaltsverzeichnis

Vorwort	3
Zusammenfassung	8
Abstract	10
1 Einleitung	11
2 Technologischer Überblick	14
2.1 Mechatronische Systeme im Automobil	14
2.1.1 Steuergeräte	15
2.1.2 Entwicklung von Steuergeräten	17
2.1.3 Verteilte Funktionen	19
2.2 Testen von mechatronischen Systemen	19
2.3 Risiko	25
2.3.1 ISO 26262	27
2.3.1.1 Automotive-Sicherheits-Integritätslevel (ASIL)	29
2.3.1.2 Werkzeugvertrauenseinstufung (TCL)	34
2.3.2 Fehlermöglichkeits- und Einflussanalyse (FMEA)	35
2.3.3 Fehlerbaumanalyse (FBA)	36
2.4 Endlicher Automat	37
2.5 Expertensysteme	40
3 Anforderungen aus der Praxis an das Prüfverfahren	43
3.1 Aktives Fahrwerk	45
3.2 Bremse	46
3.3 Lenkung	46
3.4 Vernetzte Sicherheitsfunktionen	47
3.4.1 Testen von vernetzten Sicherheitsfunktionen	47
3.4.2 Verbunds-HiL	48
3.4.3 Prototyp	49

3.4.4	Fahrsimulator	50
3.4.5	Wiederverwendung von Testfällen	51
3.5	Forschung am FKFS	52
3.5.1	ALPAS	52
3.5.2	TESAM	53
4	Die risikobasierten Testmethoden	55
4.1	Risiko	55
4.2	Methoden zur syntaxbasierten Testfallerzeugung	59
4.3	Methode zur Erstellung eines Beobachters	67
4.4	Umsetzung	71
4.4.1	Umsetzung des Beobachters	72
4.4.2	Umsetzung der syntaxbasierten Testfallerstellung	75
4.4.3	Umsetzung Fahrerprobung	77
4.4.4	Umsetzung am HiL	78
4.4.5	Umsetzung am Fahrsimulator	79
4.4.6	Risikoanalyse	80
5	Ergebnis und Ausblick der Arbeit	98
	Abkürzungen und Formelzeichen	110

Tabellenverzeichnis

2.1	ASIL	33
2.2	Werkzeugvertrauenslevel nach [iso1]	35
4.1	Testfallgenerierung nach Risikoeinstufung	65
4.2	Abbildung UML nach CLIPS	74
4.3	Fehlertabelle	83
4.4	Lenkunterstützungsausfall	84
4.5	Unkontrollierte Lenkbewegung	86
4.6	ESP-Ausfall während einer Regelung	87
4.7	ESP-Ausfall vor einer Regelung	90
4.8	Fehlerhafte ESP-Regelung	93
4.9	Fehlerhaftes Wanken	94
4.10	Ausfall Wankunterstützung	95
4.11	Nach ASIL geordnete Fehler	96
5.1	Laufzeitanalyse	100

Abbildungsverzeichnis

2.1	Produktlebenszyklus	17
2.2	V-Modelle	18
2.3	V-Modell nach [SZ05]	23
2.4	Modellbasiertes Testen nach [Sch07]	24
2.5	Risikozahl	25
2.6	Schaubild der Bände zur Ordnung der Norm ISO 26262 . . .	27
2.7	Fehlerbaum	37
2.8	Petri-Netz	38
2.9	Hierarchie nach [Har87]	38
2.10	Orthogonalität nach [Har87]	39
3.1	Kammchen Kreis nach [Ise06]	43
3.2	HiL-Modell	49
3.3	Beispiel eines Messfahrzeugs	50
3.4	Fahrsimulatormodell	51
3.5	Stuttgarter Fahrsimulator	52
3.6	TESAM Arbeitsschritte	54
4.1	Beispiel mit eingefärbtem Ergebnismodell	69
4.2	Beispiel Fahrzeug	72
4.3	Beobachter Software Architektur	73
4.4	Erweiterte Arbeitsschritte	77
4.5	Beispiel Fahrzeug	78
4.6	Beispiel Verbund HiL	79
4.7	Beispiel Fahrsimulator	80
4.8	Zweimaliger Fahrbahnwechsel	97

Zusammenfassung

Die verstärkte Automatisierung bei der Absicherung der Fahrerassistenzfunktionen durch Tests führt dazu, dass das richtige Testen immer wichtiger wird. Nur wenn der Test Fehler findet, ist dieser erfolgreich. Die Qualitätsaussage eines nicht erfolgreichen Tests ist aufgrund des Testraums sehr gering. Erst das Verwenden einer Teststrategie, die eine große Anzahl an Testfällen umfasst, führt zu einer belastbaren Qualitätsaussage. Ein wichtiger Aspekt einer guten Teststrategie ist eine Priorisierung der Fehler, insbesondere im Automobil, wo die Bandbreite der Auswirkungen einer fehlerhaften Funktion von einem störenden Innengeräusch bis hin zu einer fehlerhaften Regelung, die zu schweren Personenschäden führt, reicht. Deswegen sollte der Test zuerst in den Bereichen durchgeführt werden, die ein hohes Risiko bzgl. Personenschäden haben. Solche Bereiche werden durch eine Risiko- und Gefährdungsanalyse des Autos gefunden. In dieser Arbeit wird untersucht, ob sich die automatisierte Absicherung mit Ergebnissen einer Risiko- und Gefährdungsanalyse verbessern lässt.

Der Fahrer wird im heutigen Automobil bei gefährlichen Fahrsituationen durch eine Vielzahl von Fahrerassistenzfunktionen unterstützt. Natürlich ist ein Fehler in diesen Funktionen sehr gefährlich, deswegen müssen diese Funktionen intensiv getestet werden. Die Funktionen verteilen sich auf die Lenkung, die Bremse und das aktive Fahrwerk. Die endgültige Absicherung der verteilten Funktionen kann erst nach der Integration der drei Systeme im Fahrwerksverbund erfolgen. Zur Absicherung stehen unterschiedliche Testplattformen zur Verfügung. Die Arbeit geht auf die folgenden drei Testplattformen ein: den Fahrzeugprototyp, den Fahrsimulator und den Verbunds-HiL.

Der modellbasierte Test wird zur automatisierten Absicherung der Funktionen verwendet. Die verschiedenen Verfahren des modellbasierten Tests werden daraufhin untersucht, ob sie erweiterbar sind bzgl. des Einsatzes von Kontextwissen in Form der Ergebnisse der Risiko- und Gefährdungsanalyse. Für das ausgewählte syntaxbasierte Testverfahren wird eine Strategie entwickelt, um das Kontextwissen zu nutzen. Das Testmodell wird

durch UML-Zustandsautomaten grafisch modelliert, damit diese Informationen zur Reportdarstellung wiederverwendet werden können. Es stellt sich heraus, dass zwei Modelle übersichtlicher sind und die Testfallerstellung vereinfachen. Das eine Testmodell erzeugt die Testfälle, während das andere Testmodell die Testfälle bewertet und zur Reportdarstellung verwendet wird. Diese Aufteilung ermöglicht auch eine Anwendbarkeit des zweiten Modells auf alle drei Testplattformen.

Abstract

The increased automation of safety functions with tests demands the right kind of testing. A successful test finds errors. The quality statement of an unsuccessful test is very low. Using a test strategy with a large number of tests has a high statement of quality. The effects of errors have different levels of impact. It varies from a bad sound to injuries due to accidents caused by a wrong control. Because of this, testing high risks is important. To reveal parts with high risks, it is necessary to run a risk and hazard analysis. The main focus of this thesis lies in the improvement of the automation processes for tests by using risk based information.

The driver assistance control supports the driver in hazardous situations. An error in the driver assistance control can be a high risk. The driver assistance control therefore has a high test level. Some parts of the driver assistance control units are split on the steering, the brake and the active chassis. The final test should only take place after these parts have been integrated. The final test can be held at three test beds: the prototype, the driving simulator and the system-HiL.

The model based test can be used for the safety functions. The different model based testing methods were reviewed to find the best model based test to be improved with risk information. The syntax based test was to be the best one. The test model is based on the UML state chart. The graphic information is used for creating the result report. As a result, the usage of two models is better. One to create the test sequence and the other one to analyse the results and to report them. Especially the second model can be integrated in all three testing platforms, which makes a split in two models the best choice.

1 Einleitung

Durch die Zunahme von Funktionen im Automobil, die auch in das Fahrverhalten des Automobils eingreifen, steigt der Bedarf an Testverfahren für diese Funktionen [Ise06]. Dieser Bedarf spiegelt sich auch in der neuen Norm ISO¹ 26262 wieder [isoc]. Es gibt einen Kostendruck bei der Entwicklung neuer Automobile durch den globalen Wettbewerb im Automobilmarkt, so dass bessere Testverfahren bei gleichzeitiger Kostensenkung einen Wettbewerbsvorteil darstellen. Die beiden Ziele lassen sich durch eine verstärkte Automatisierung der Testverfahren erreichen [RBGW10].

Die verstärkte Automatisierung sollte nicht blind für die Auswirkung des Fehlverhaltens der Funktion bei einem Fehler sein, da ein vollständiger Test in endlicher Zeit nicht möglich ist. Auch ein Testverfahren mit 90% Abdeckung kann nicht im Entwicklungszeitrahmen für alle Funktionen durchgeführt werden. Die Grundidee ist, dass ein Testobjekt mit einem hohen Risiko eine hohe Testdichte hat und ein Testobjekt mit einem niedrigen Risiko eine niedrige Testdichte hat [PVDBJV04], [SMP08], [Ber09]. Risiko wird allgemein als ein Produkt aus „Eintrittswahrscheinlichkeit \times Auswirkung“ definiert, doch kann das Risiko wie folgt unterschiedlich betrachtet werden:

- Geschäftsrisiko, welche unternehmerischen Risiken es gibt und welche Auswirkungen diese haben.
- Prozessrisiko, welche Fehler während eines Ablaufs des Prozesses auftreten und welche Auswirkungen diese auf den Prozess haben.
- Produktrisiko, ob das Produkt ausfällt und welche Auswirkungen dieser Ausfall hat, kann in zwei Unterklassen aufgeteilt werden:
 - Funktionsrisiko, ob die Funktion durch einen Fehler beeinträchtigt wird und ob der Fahrer dies wahrnimmt.
 - Sicherheitsrisiko, ob die Funktion durch einen Fehler beeinträchtigt wird und ob es dabei zu Personenschäden kommen kann.

¹Internationale Organisation für Normung

Das Sicherheitsrisiko des Automobils ist das Thema der Norm ISO 26262. Die Umsetzung einer neuen Norm im Herstellungsprozess erfordert einen Mehraufwand. Die Umsetzung eines modellbasierten Testverfahrens hat zudem einen erhöhten Startaufwand [RBGW10]. Die beiden Mehraufwände sollten zu einem qualitativ besseren Produkt und dadurch in der Servicezeit zu geringeren Kosten führen.

Die Absicht dieser Arbeit ist es, einen modellbasierten Ansatz und einen risikobasierten Ansatz zu einem neuen Testverfahren zu kombinieren. Das neue Testverfahren wurde an den verteilten Funktionen des Fahrwerksverbands erprobt. Dabei wurde der Systemtest als Testzeitpunkt gewählt. Für den Systemtest standen folgende drei Testplattformen zur Verfügung:

- Der Fahrzeugprototyp, in dem die Funktionen mit einem realen Fahrzeug im Fahrversuch erprobt wurden.
- Der Fahrsimulator, mit dem die Interaktion zwischen Fahrer und der Funktion in einer simulierten Umwelt getestet wurde.
- Der Hardware-in-the-Loop Verbund (V-HiL), in dem die Interaktionen der Funktionen mit einer simulierten Umwelt und einem simulierten Fahrer getestet wurden.

UML-Zustandsautomaten wurden als Modellgrundlage für den modellbasierten Ansatz verwendet. Die UML-Zustandsautomaten dienen sowohl als Grundlage für die Erzeugung von Testfällen als auch zur Analyse der Testdaten. Verschiedene Testfallerzeugungsstrategien wurden daraufhin untersucht, ob sie sich mit Risikoinformationen verbessern lassen. Die strukturbasierten Verfahren sind die am besten geeigneten Verfahren zum Kombinieren mit Risikoinformationen. Beim Erzeugen von Tests kann in Abhängigkeit des Risikos des jeweiligen Strukturelements ein dem Risiko angepasstes Verfahren gewählt werden.

Die grafischen Elemente des Verhaltensmodells werden zum Strukturieren der Testdaten verwendet. Dadurch können die Testergebnisse sehr kompakt dargestellt werden, was zu einer besseren Rückmeldung führt und verhindert, dass Fehler übersehen werden. Eine farbige und grafische Darstellung kann wichtige Ergebnisse hervorheben. Das Einfärben in unterschiedlichen Risikostufen soll automatisch erfolgen. Dazu wurden anhand der Risikoinformation Regeln entwickelt und beim Erzeugen der grafischen Elemente

angewendet. Der Einsatz der grafischen Elemente zur Testfalldarstellung war für alle drei Plattformen möglich und fand daher seine Anwendung auf allen drei Testplattformen.

2 Technologischer Überblick

Das heutige Automobil wird gerne als ein mechatronisches System bezeichnet [Ise06]. Deswegen wird der Begriff mechatronisches System definiert und die Beziehung zum heutigen Automobil aufgezeigt. Mit dieser Grundlage des mechatronischen Systems werden verschiedene Möglichkeiten aufgezeigt, wie das mechatronische System zu testen ist und dabei werden auch die Grundlagen des Testens eingeführt.

Danach werden eine Definition für das Risiko vorgenommen und verschiedene Modelle vorgestellt, die in Beziehung zum Risiko stehen. Dann wird der Begriff Metrik eingeführt und verschiedene Kategorien von Metriken und Eigenschaften von Metriken dargestellt. Mit diesen Schemata werden dann verschiedene Risikometriken bewertet. Danach werden endliche Automaten und deren Fähigkeit zu semiformalen Beschreibungen von reaktiven Systemen eingeführt, die notwendig sind, wenn modellbasiert getestet wird. Als Ausführungsplattform für endliche Automaten werden Expertensysteme vorgestellt und erklärt.

2.1 Mechatronische Systeme im Automobil

Ein mechatronisches System ist ein System, das aus drei Teilen besteht, die relevant sind, damit das System voll funktionsfähig ist. Der erste Teil ist der mechanische Anteil des Systems wie zum Beispiel das Getriebe. Der zweite Teil ist der elektronische Anteil wie die Stromversorgung des Getriebes und der dritte Teil ist der softwaretechnische Anteil wie die Regelung für das Getriebe.

Ein heutiges Auto ist in seiner Gesamtheit ein mechatronisches Kraftfahrzeug [Ise06], wird aber während seiner Entwicklung in kleinere Teilsysteme aufgeteilt. Diese Teilsysteme stellen in der Regel die Steuergeräte des Autos dar. Bei der Entwicklung der mechatronischen Anteile eines Fahrzeugs gibt es unterschiedliche Sichtweisen, um so die Entwicklung der Komponenten durch Vereinfachung zu erleichtern. Eine sinnvolle und praxisnahe Auftei-

lung wird in [SZ05] vorgenommen, in dem unter anderem zwischen einer Funktionsebene und einer Steuergeräteebene unterschieden wird. Dadurch können zuerst die notwendigen Funktionen bestimmt und definiert werden und erst in einem zweiten Schritt wird die Aufteilung auf die verschiedenen Steuergeräte vorgenommen.

2.1.1 Steuergeräte

Ein Steuergerät ist in der Regel ein Mikrocontroller, der mit Sensoren, Aktoren und mit mindestens einer Kommunikationsschnittstelle ausgestattet ist. Je nach Einsatzort im Fahrzeug muss das Steuergerät besonderen Umwelteinflüssen wie EMV¹, Hitze, Kälte, mechanischen Beanspruchungen oder Wasser dauerhaft standhalten [SZ05]. Die Steuergeräte werden zu Netzwerken zusammengefasst, so dass die meisten verteilten Funktionen nur auf ein bis zwei Netzwerke verteilt sind. Eine gängige Netzwerktopologie nach [SZ05] ist:

Antrieb In diesem Netzwerk sind alle Steuergeräte des Antriebsstranges für zum Beispiel Motor, Getriebe und Energiemanagement zusammengefasst.

Fahrwerk In diesem Netzwerk sind alle Steuergeräte der Fahrwerkskontrolle zusammengefasst für zum Beispiel Federung, Wankstabilisierung oder die Bremse.

Komfort In diesem Netzwerk sind alle Steuergeräte aus dem Fahrerinnenraum zusammengefasst, wie zum Beispiel der Bordcomputer, die Türsteuergeräte oder die Mensch-Maschine-Schnittstelle.

In den heutigen Oberklasse-Fahrzeugen kommen noch weitere Netzwerke zum Einsatz. Im aktuellen Touareg [20110] kommen die vier folgende Protokolle zum Einsatz: LIN², CAN³, MOST⁴ und Flexray⁵. Zu den oben beschriebenen Netzwerken kommen noch folgende Netzwerke hinzu:

- Infotainment auf Basis von MOST

¹Elektromagnetische Verträglichkeit

²Local Interconnect Network

³Controller Area Network

⁴Media Oriented Systems Transport

⁵Feldbussystem

- Das Netzwerk für die Anzeigesteuergeräte und die Bedienungsteuergeräte der Klimaanlage auf Basis von CAN
- Sensornetzwerk auf Basis von CAN
- Sensornetzwerk auf Basis von Flexray
- Bordnetz auf Basis von LIN

Zudem haben noch einige Steuergeräte ihre eigenen privaten Netzwerke auf Basis von LIN, um mit Sensoren und Aktoren zu interagieren.

2.1.2 Entwicklung von Steuergeräten

Nach [SZ05] umfasst der Entwicklungszyklus eines Fahrzeugs drei Jahre und der Produktionszeitraum umfasst sieben Jahre. Desweiteren wird eine Servicezeit von insgesamt 10-15 Jahren genannt. Zusätzlich gibt es für neue Funktionen eine Vorentwicklungsphase, in der diese neuen Funktionen entwickelt oder verschiedene Lösungen miteinander verglichen werden. Die so neuentwickelten Funktionen können zum Teil in bestehende Steuergeräte integriert werden, die Übernahmeteile aus früheren Produktlinien sind. Dadurch erhöht sich der Lebenszyklus von Hardwarebestandteilen wie auch Softwarefunktionen. Ansonsten wird für die Funktionen ein neues Steuergerät entwickelt. Verschiedene Hersteller führen noch eine Modellpflege durch, die in der Regel zur Hälfte des Produktlebenszyklus stattfindet. Bei der Modellpflege werden neue Funktionen wie beispielsweise die Start-Stop-Automatik eingeführt. Das führt zu folgendem zeitlichen Verlauf, der in Abbildung 2.1 dargestellt ist. Die ursprüngliche Grafik stammt von [SZ05] und wurde erweitert. Der vorherrschende Entwicklungsprozess

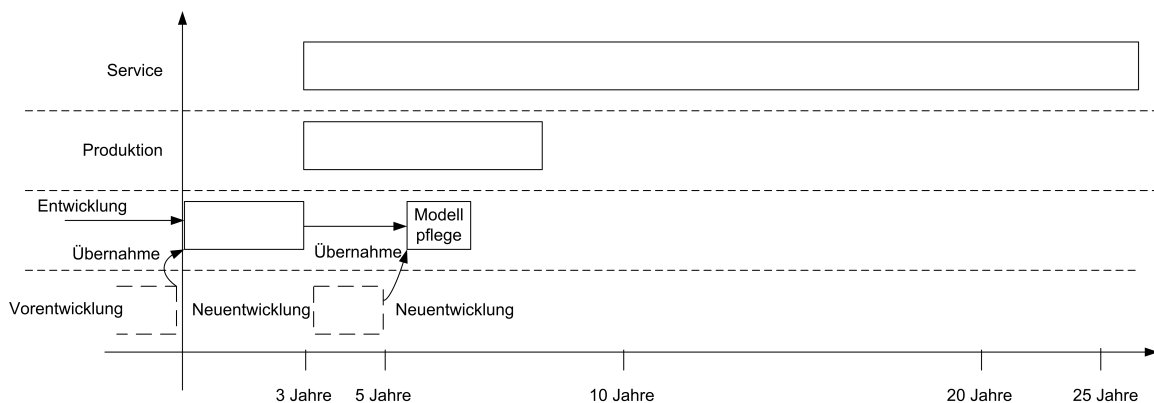


Bild 2.1 Produktlebenszyklus

ist ein iteratives V-Modell (siehe Abbildung 2.2). In den drei Jahren Entwicklungszeit wird das V-Modell mehrmals durchlaufen. Bei den früheren Iterationen sind die Steuergeräte bzgl. ihres Entwicklungsgrads sehr heterogen. Der heterogene Entwicklungsgrad führt in frühen Iterationen zu Reibungsverlust, dadurch geht Testzeit verloren und eine deutlich höhere Testvorbereitungszeit ist erforderlich, um in frühen Phasen verteilte Funktionen zu testen. Die Alternative, darauf zu warten, bis sich der Entwicklungsgrad der Steuergeräte angeglichen hat, kann nur bei nicht feststehendem

Zeiträumen durchgeführt werden, was in der Automobilentwicklung nicht üblich ist.

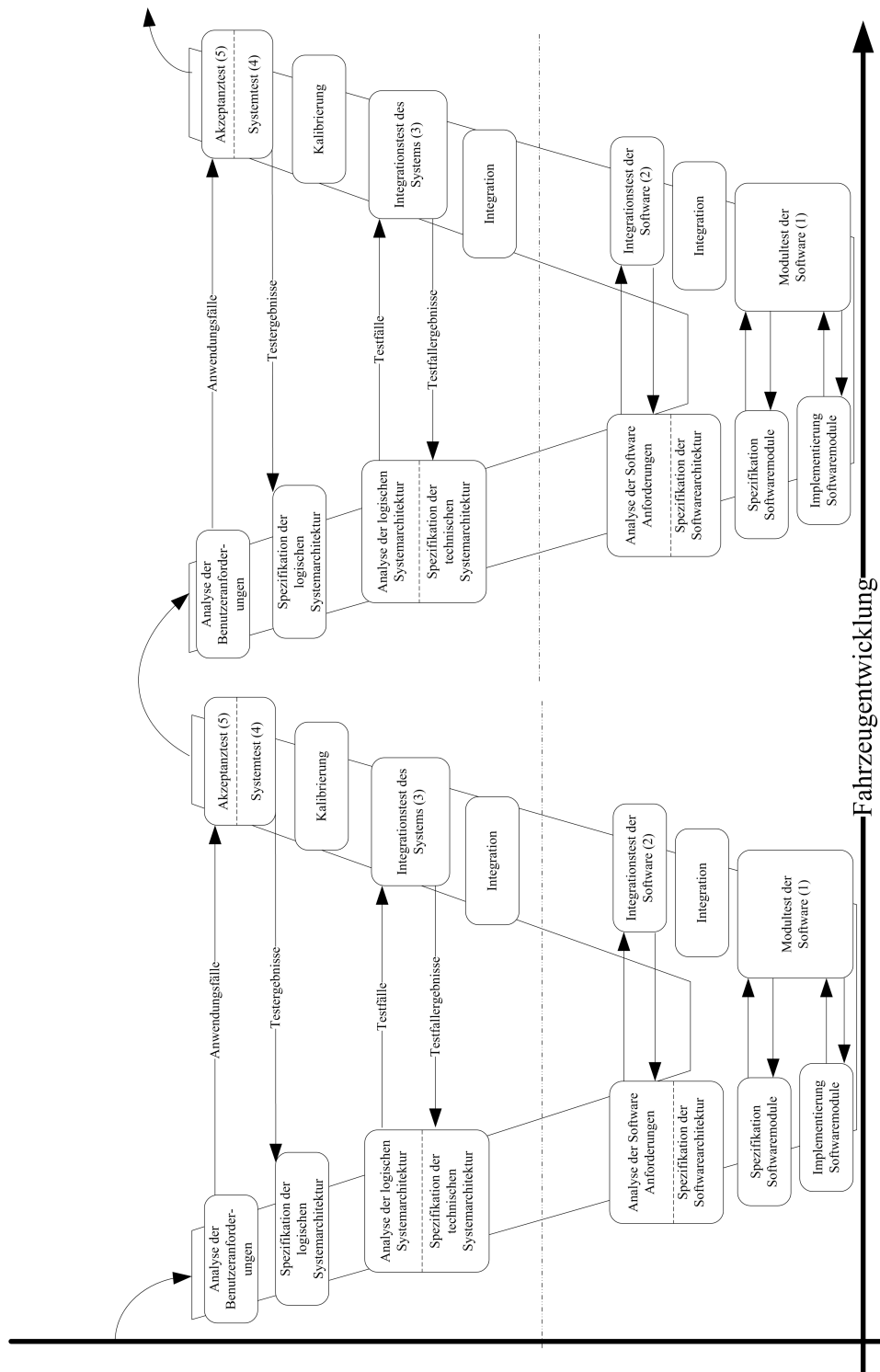


Bild 2.2 V-Modelle

2.1.3 Verteilte Funktionen

Der Fokus dieser Dissertation liegt wie bereits erwähnt auf einem Testverfahren für verteilte Funktionen im Fahrwerk. Im Moment zeichnet sich ein Trend von einer wachsenden Anzahl neuer Funktionen im Auto ab. Die Zahl der Steuergeräte wächst im Verhältnis zu den Funktionen langsamer [Ise06], was zu einer deutlich steigenden Anzahl von verteilten Funktionen führt. Der Trend zu verteilten Funktionen wird durch die neue Systemarchitektur von Autosar und die höheren Übertragungsraten der Kommunikationssysteme unterstützt. In anderen Bereichen der Informatik wurden verteilte Funktionen und verteilte Systeme deutlich früher eingesetzt zum Beispiel in den Bereichen IT Business oder beim Wissensmanagement. Die Erfahrung aus diesen Bereichen sollte genutzt werden, um die negativen Erfahrungen bei der Entwicklung von Steuergeräten nicht zu wiederholen.

2.2 Testen von mechatronischen Systemen

Die besondere Anforderung für das Testen von mechatronischen Systemen liegt darin, dass die jeweiligen Teile zwar vorab testbar sind, aber erst nach der Integration der verschiedenen Komponenten zu einem System kann der anschließende Systemtest alle Fehler finden. So erfordert das Testen von verteilten Funktionen eine sehr gute Testplanung in Bezug auf die Koordination der unterschiedlichen Entwicklungszyklen, so dass die Testphasen bzgl. der folgenden Faktoren zu optimieren sind:

- Aufwand zum Aufspüren der Fehler
- die Entdeckungswahrscheinlichkeit der Fehler
- frühester möglicher Testzeitpunkt

Nach [Lig09] unterteilt man die Testmethoden in statische und dynamische Testmethoden. Statische Methoden, die formale Beweistechniken sind, haben zwar den Vorteil der Vollständigkeit, können aber bei nicht trivialen Systemen nicht immer ein Ergebnis in endlicher Zeit liefern. Zudem besteht hier das Problem, dass viele Beweistechniken einen vollständigen Zugriff auf den Quellcode und auf die Schaltpläne erfordern. Das ist in dem aktuellen arbeitsteiligen Entwicklungsprozess nicht durchsetzungsfähig, weil gerade

im Fahrwerksbereich eine Technologieführerschaft je nach Funktion entweder beim Automobilhersteller oder beim Automobilzulieferer liegt und die erforderlichen Daten zu den Geschäftsgeheimnissen gehören. In der Zukunft könnte sich die erste Aussage durch eine deutliche technische Verbesserung der formalen Beweistechniken wie Model Checking oder Translation Validation relativieren, so zum Beispiel hat das Forschungsprojekt Verisoft XT diesbezüglich vielversprechende Ergebnisse erzielt [spr08].

Bei den dynamischen Testmethoden wird nur ein Stichprobentest durchgeführt. Wenn aber die Stichproben von ausreichender Güte sind, kann aus dem Stichprobentestergebnis eine belastbare Qualitätsaussage über den Prüfkandidaten getroffen werden. Als Beispiel für eine sinnvolle und belastbare Stichprobenauswahl kann die MC/DC ⁶ genommen werden. Für 2^n Ausprägungen werden nur $n + 1$ Ausprägungen gewählt, die aber 90% der Fehler finden [Chi01]. Allerdings ist dies eine Qualitätsaussage, die sich nur auf Testqualitätsmaßnahmen begründet; sie ist nicht vollständig, da in den verbleibenden 10 % der Fehler noch Fehler mit kritischen Auswirkungen beinhaltet sein können. Deswegen sollten die Eigenschaften der eingesetzten dynamischen Testmethoden bei der Qualitätsaussage berücksichtigt werden. Die dynamischen Testmethoden können in folgende drei Klassen aufgeteilt werden [Lig09]:

Black Box Die Klasse umfasst alle Testmethoden die eingesetzt werden, wenn das System ohne Zugriff auf die Quelldaten geprüft werden soll. Die Testmethoden leiten aus den Eingabewerten und Ausgabewerten die Testdaten ab.

White Box Die Klasse umfasst alle Testmethoden die eingesetzt werden, wenn neben dem System auch die Quelldaten zur Verfügung stehen. Die Testmethoden verwenden zusätzlich zu den Eingabewerten und den Ausgabewerten noch Informationen aus den Quelldaten wie den Kontrollfluss oder den Datenfluss, um die Testdaten zu erzeugen.

Grey Box Die Klasse umfasst alle Testmethoden die eingesetzt werden, wenn neben dem System auch Teilinformationen über die Quelldaten wie Struktur oder Schnittstelle zur Verfügung stehen. Die Testmethoden verwenden zusätzlich zu den Eingabewerten und Ausgabe-

⁶MC/DC steht für Modified Condition Decision Coverage

werten noch Informationen aus den Strukturdaten und Schnittstellenbeschreibungen, um die Testdaten zu erzeugen.

Die dynamischen Testverfahren können in der Regel automatisiert werden. Hierbei werden vier Disziplinen unterschieden, wobei jede Disziplin jeweils unterschiedliche Abstufungen aufweist [RBGW10]:

- Testfallerstellung
 - Die Testfälle können manuell programmiert werden.
 - Das Grundgerüst der Testfälle wird automatisiert erstellt und dann manuell angepasst.
 - Die Testfälle werden automatisch erstellt.
- Testfallausführung
 - Die Testfälle können für die Testplattform manuell konfiguriert und manuell ausgeführt werden.
 - Die Testfälle können für die Testplattform automatisch konfiguriert und automatisch ausgeführt werden.
- Testfallanalyse
 - Die Testfallergebnisse werden automatisiert gesammelt.
 - Die Testfallergebnisse werden noch für eine Analyse vorbereitet und gefiltert.
 - Die Testfallergebnisse werden automatisiert verarbeitet und bewertet.
- Informationsmanagement

Beim Informationsmanagement geht es um die Aufgabe, die richtigen Informationen zum richtigen Zeitpunkt dem berechtigten Personenkreis zur Verfügung zu stellen. Es ist ein Unterstützungsprozess, der aber zum Erfolg der Teststrategie beiträgt.

Jede Umsetzung eines dynamischen Testverfahrens kann unterschiedliche Grade der Automatisierung besitzen. Eine hohe Automatisierung kann zu einer Entlastung des Testers führen. Gleichzeitig kommt es aber zu einer Steigerung der benötigten Qualifikation des Testers. Wenn die automatische Niveauregelung durch einen Fahrzeugprototyp getestet wird, gibt es

folgende Anforderungen an den Tester: Erfahrungen in der Testplanung, Erfahrungen in der Vorbereitung des Testmusters für den Fahrzeugprototyp und Funktionskenntnis der Niveauregelung. Bei der Testdurchführung muss der Tester vor Ort sein und diesen selbst durchführen.

Wenn die automatische Niveauregelung automatisiert an einem HiL getestet wird, kommt noch folgende spezielle Anforderung hinzu: Testfallerstellungswissen am HiL. Der Tester bereitet die Testfälle vor, die dann automatisiert am HiL durchgeführt werden. Bei der Testdurchführung am HiL muss der Tester nicht vor Ort sein. Die Auswertung des Tests erfolgt in der Regel durch den Tester persönlich.

Der zweite Aspekt zur Bestimmung der Anforderungen an die Qualifikation der Personen zur Testdurchführung ist das Testobjekt. Die Funktionen des Testobjekts können wie folgt unterschieden werden:

- Einfache Funktionen erfordern nur ein einfaches Testverfahren in der Art, dass nur eine einfache Aktion ausgelöst wird und es kommt zu einer bestimmten Reaktion der Funktion. Dieses einfache Testverfahren ohne geschlossenen Regelkreis wird als „Open Loop“ bezeichnet.
- Komplexe Funktionen erfordern einen geschlossenen Regelkreis mit entsprechender Simulation und dies wird „Closed Loop“ bezeichnet.

Nach den Testmethoden und dem Automatisierungsgrad ist für das Testen von mechatronischen Systemen die Testplattform wichtig. Die Testplattform hängt zum einen von der Teststufe und zum anderen von der Komplexität der zu testenden Funktion ab. Aus dem Bild der Teststufen des V-Modells 2.3 wurden folgende Testplattformen abgeleitet:

1. Modultest: Der Modultest kann durch die Testmethode „Function in the Loop“ oder durch einen „Unit Test“ durchgeführt werden.
2. Softwaretest: Der Softwaretest kann durch die Testmethode „Software in the Loop“ (SiL) [MMR] oder durch einen „Back to Back“ [Lig09] Test durchgeführt werden.
3. Integrationstest: Nach der Integration der Hardware mit der Software zu einem System kann der Integrationstest bei einem System mit einfachen Funktionen durch eine „Open Loop Failsafe“ Testmethode oder bei komplexen Funktionen durch eine „Closed Loop HiL“ Testmethode durchgeführt werden.

4. Systemtest: Am Verbunds-HiL können Diagnose- und Sicherheitstests durchgeführt werden, um das Verhalten aller Steuergeräte zu berücksichtigen.
5. Akzeptanztest:
 - Am Fahrsimulator kann die Zusammenarbeit zwischen Fahrer und Steuergeräten unter Laborbedingungen getestet werden.
 - Am Fahrzeug können Langzeiterprobung und Leistungserprobung durchgeführt werden. Regelgüte kann auch nur für dynamische Systeme im Fahrzeug getestet werden.

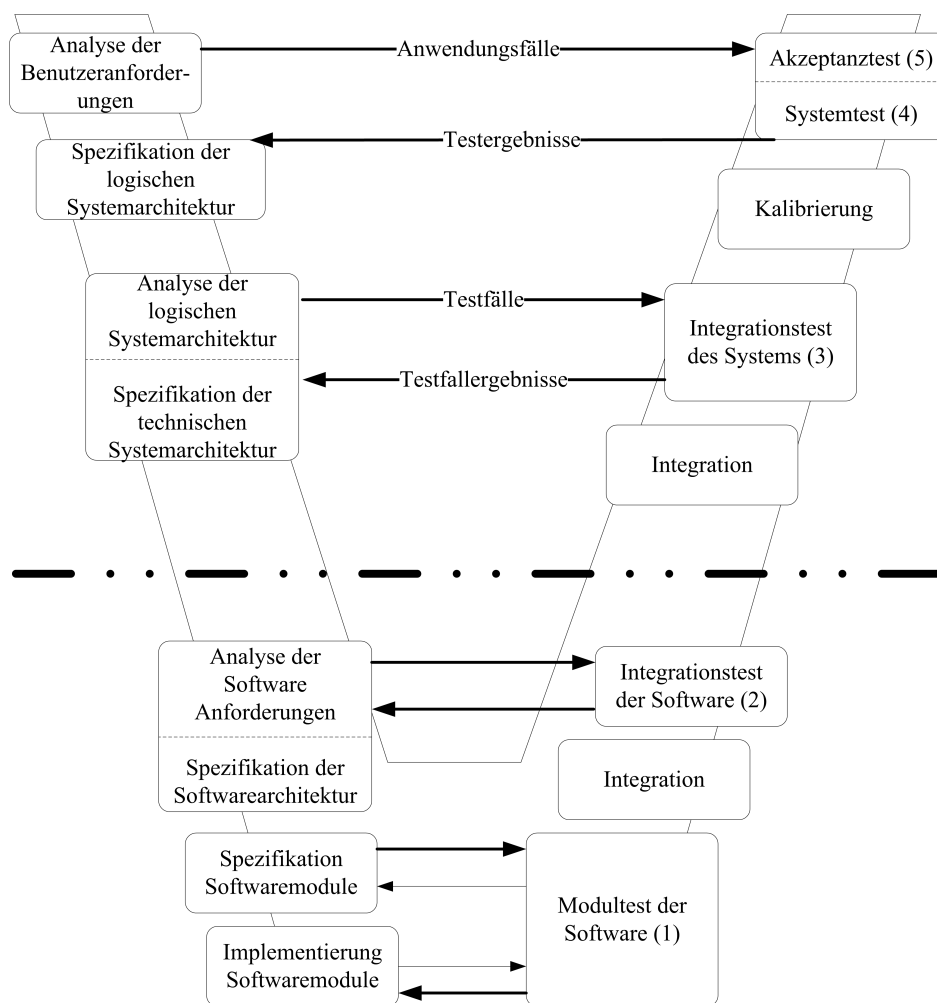


Bild 2.3 V-Modell nach [SZ05]

Ein modellbasiertes Testverfahren (MBT) [RBGW10] hat in allen Disziplinen der Testautomatisierung einen hohen Automatisierungsgrad. Das MBT-Verfahren erfordert, dass eine Spezifikation des Systems zur Verfügung steht, aus der ein Modell des Systems und ein Testmodell abgeleitet

werden können. Somit gibt es zwei Modelle. Die Beziehungen zwischen dem Modell des Systems und dem Modell des Testsystems beschreibt das Bild 2.4.

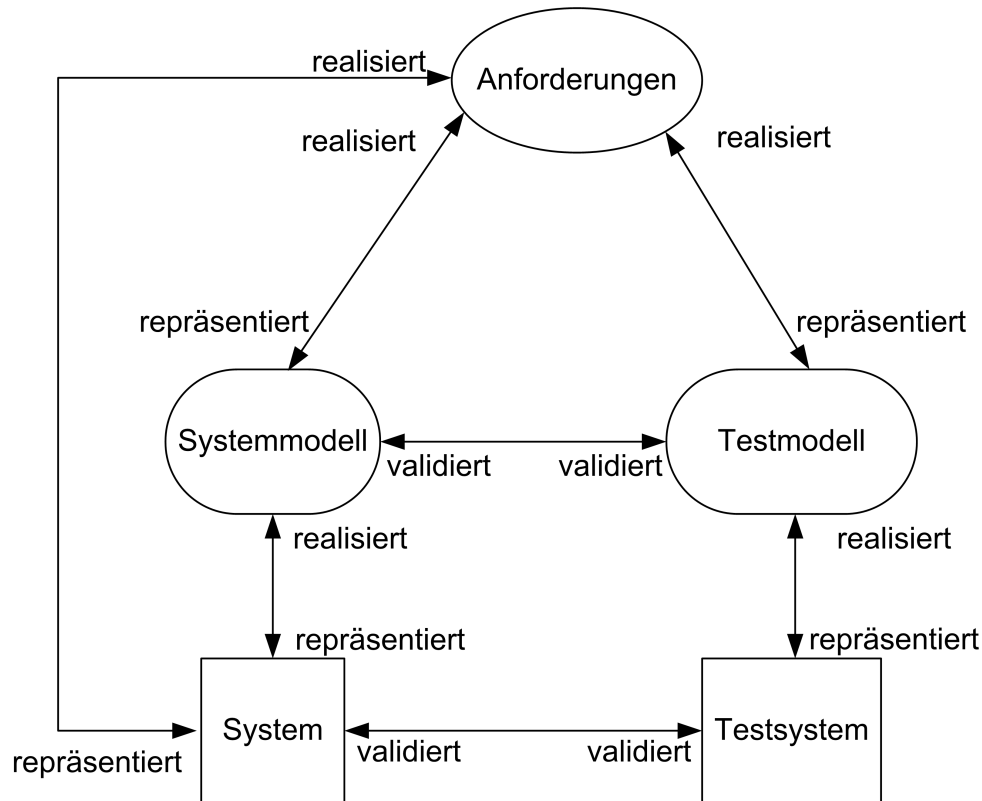


Bild 2.4 Modellbasiertes Testen nach [Sch07]

Kommerzielle Werkzeuge für das modellbasierte Testen stehen bereits zur Verfügung. Allerdings setzen sie nach der IX-Studie [GNRS09] nicht alle Fähigkeiten zur syntaxbasierten Testfallerzeugung um, die in der wissenschaftlichen Forschung bereits untersucht wurden.

Ein besonderer Typ des modellbasierten Testens ist das passive Testen. Beim passiven Test wird aus der Spezifikation ein Beobachter erstellt. Der Beobachter überwacht das Testobjekt und bewertet das Verhalten des Testobjekts. Die Methode ist sehr gut geeignet für verteilte Funktionen. Diese müssen beim Test gleichzeitig überwacht werden. Der Beobachter ist dazu in der Lage.

Damit der Beobachter funktioniert, muss durch das Kommunikationsmedium eine einheitliche zeitliche Sicht ermöglicht werden [Hes06], vor allem wenn sich die verteilte Funktion über mehr als ein Netzwerk verteilt. Wenn keine einheitliche Sicht möglich ist, muss entweder durch eine Hard-

Die Metriken können in drei Kategorien sortiert werden [LL07]. Somit lassen sich alle Risikomodelle nach folgenden Eigenschaften klassifizieren: Typ, Struktur und Eigenschaften der verwendeten Metriken. Zuerst werden der Typ und die Struktur untersucht und in einem zweiten Schritt werden die Metriken nach ihren Eigenschaften untersucht und eingestuft. Hierbei werden die Kategorien [LL07] verwendet. Es gibt drei Arten von Metriken:

objektive Metrik Die grundlegenden Größen der Metrik werden entweder gemessen oder werden gezählt.

subjektive Metrik Die Beurteilung erfolgt durch einen Gutachter, der dann die Bewertung entweder verbal oder auf einer vorgegebenen Skala festhält. Dabei gibt es Vorschriften, die bei verschiedenen Gutachtern eine grundlegende Vergleichbarkeit sicherstellen. Die Vorschriften können zum Beispiel in Form einer Checkliste für Softwarequalität vorliegen, in der Anhaltspunkte und deren Bedeutung stehen wie:

- sind die Variablennamen verständlich
- der Code ist nach Styleguide formatiert
- GoTo Befehle werden verwendet

Pseudometrik Werte werden durch Berechnungen und Schätzungen ermittelt. Diese Werte können dann zu Voraussagen auf zukünftige Entwicklungen verwendet werden. Diese Art der Metrik sollte eingesetzt werden, wenn die Werte nicht direkt ermittelt werden können, aber zu einem frühen Zeitpunkt für eine Entscheidung notwendig sind.

In der Regel setzt sich die Kennzahl bei quantitativen Risikomodelle aus verschiedenen Arten von Metriken zusammen. Zudem sollte eine gute Risikometrik nach [LL07] folgende Attribute besitzen:

differenziert Eine ausreichende Breite der Skala, um unterschiedlichen Risiken unterschiedliche Werte zuzuweisen.

vergleichbar Die Bewertungen müssen sich miteinander vergleichen lassen, wobei bei Risikometriken in der Regel eine Ordinalskala verwendet wird, die diese Attribute erfüllt.

reproduzierbar Dasselbe System mit den gleichen Eigenschaften sollte auch bei mehrmaligen Bewertungen das gleiche Ergebnis liefern.

relevant Die Metrik muss einen Nutzen haben. Eine gute Risikometrik sollte Ergebnisse erzielen, die zur Steuerung des Entwicklungsprozesses eingesetzt werden.

rentabel Der Aufwand der Erhebung sollte nicht mehr Aufwand erfordern, als nach ihrer Relevanz gerechtfertigt ist. Deswegen sollte die Risikoinformation an allen relevanten Stellen eingesetzt werden, um ihrem hohen Aufwand einen hohen Ertrag entgegenzustellen.

plausibel Wenn die Metrik eine bestimmte Interpretation der Bewertung impliziert, dann muss diese Interpretation plausibel sein. Bei Risikometriken erfolgt so ein Abgleich zwischen Bewertung und Beobachtung in der Regel nach einem Entwicklungszyklus.

Qualitative Risikomodelle versuchen unterschiedliche Fehlerquellen oder Einflussfaktoren zu trennen um eindeutige Fehlerpfade zu bestimmen. Diese Modelle versuchen auch alle relevanten Einflussfaktoren zu bestimmen, die das System negativ beeinflussen.

2.3.1 ISO 26262

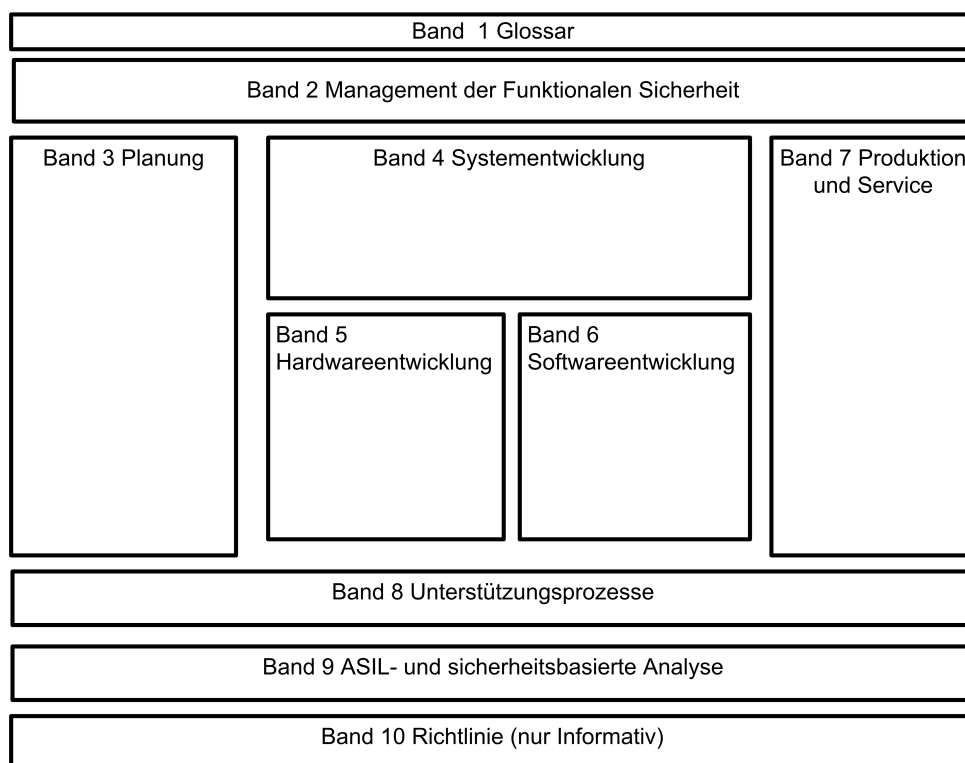


Bild 2.6 Schaubild der Bände zur Ordnung der Norm ISO 26262

Die ISO 26262 ist eine Norm für die funktionale Sicherheit von mechatronischen Systemen im Bereich der Automobilentwicklung. Die Norm ist eine Adaption der Norm DIN-EN-61508 [DIN] und wurde für die besonderen Anforderungen des Bereichs Automobilentwicklung angepasst. Seit dem 14.11.2011 liegt sie als Final Draft vor und ist aus Sicht der ISO auf den Entwicklungsprozess anwendbar. Die Norm umfasst 10 Bände und ist nach der Abbildung 2.6 gegliedert.

Der erste Band [isoa] umfasst den Glossar, in dem die Begriffe definiert werden, auch die allgemein verwendeten Begriffe wie Risiko oder Test. Der zweite Band [isoc] beschreibt die Richtlinien, um das ordnungsgemäße Management von sicherheitsrelevanten Systemen sicherzustellen. Dabei wird auf das Gesamtsicherheitsmanagement, auf das Sicherheitsmanagement der Systementwicklung und auf das Sicherheitsmanagement während der Produktion und des Service eingegangen. Der dritte Band [isod] beschreibt die Anforderungen, die bei der Planung eines sicherheitsrelevanten Systems aus Sicht der Norm notwendig sind. Dazu wird definiert, was die Norm unter einem „Item“⁷ versteht und wie dem „Item“ der dazugehörige Sicherheitsprozess zugeordnet wird. Es werden verschiedene Risiko- und Auswirkungenanalysen empfohlen sowie Handlungsempfehlungen, wie sie durchgeführt werden sollen, aufgelistet. Desweiteren werden die Anforderungen an das funktionale Sicherheitskonzept aufgeführt und erklärt.

Der vierte Band [isoe] umfasst die Anforderungen und Richtlinien für die Systementwicklung des Elements und die Erweiterung des V-Modells um die Aspekte der funktionalen Sicherheit. Als Entwicklungsprozess wird grundsätzlich das V-Modell angenommen. Die rechte Seite des V-Modells wird um die folgenden Punkte erweitert: die Maßnahmen beim Start des Projekts, die Spezifikation der technischen Anforderungen, die Spezifikation der funktionalen Anforderungen und der Systementwurf. Die linke Seite des V-Modells wird um die folgenden Punkte erweitert: die Systemintegration, Überprüfung, ob die Sicherheitsanforderungen ausreichend sind, die Verifikation der Sicherheitsanalyse und die Freigabe des Produktes.

Der fünfte Band [isof] beschreibt die Anforderungen an die Hardwarebestandteile des Systems, sowie die Sicherheitsanforderungen, die bei der Pla-

⁷Ein System oder mehrere Systeme, die eine Funktion auf Fahrzeugebene realisieren, auf der die Norm angewendet wird.

nung und dem Entwurf der Hardware zu berücksichtigen sind. Es werden verschiedene Sicherheitsmetriken für die Hardwarearchitektur vorgestellt. Verschiedene Arten von zufälligen Hardwarefehlern und deren Auswirkung werden dargestellt. Desweiteren listet der Band verschiedene Verfahren für den Test der Hardware auf und bewertet diese.

Der sechste Band [isog] beschreibt die Anforderungen an die Softwarebestandteile und an den Softwareentwurf. Es gibt Vorgaben, wie die Softwareanforderungen gesammelt und dokumentiert werden. Außerdem gibt es Anforderungen an die Softwarearchitektur, an die Softwareimplementierung und an den Softwaretest. Dazu gibt es Vorgaben, wie der Nachweis der Sicherheitsanforderungen erfolgen sollte. Der siebte Band [isoh] beschreibt die Anforderung für die Produktion und Servicephase des Systems. Der achte Band [isoi] beschreibt die Anforderungen für die Unterstützungsprozesse der Systementwicklung wie der Dokumentation, Anforderungsmanagement oder Änderungsmanagement. Im neunten Band [isoj] wird der Automotive Sicherheits-Integritätslevel beschrieben.

Der zehnte Band [isob] ist noch nicht freigegeben und hat nur informativen Charakter. Der zehnte Band erläutert die Norm näher und gibt Beispiele. Auch wenn dieser Band nur zur Information gedacht ist, so ist er wichtig für die Anwendung der Norm. Die Norm soll als technischer Standard in der europäischen Automobilindustrie gelten. Eine zentrale Vorschrift der Norm ist die Risiko/Gefährdungsanalyse zum Bestimmen der Sicherheitseinstufung. Die Norm hat zwei Modelle, die in Beziehung zu den Ergebnissen der Risikoanalyse stehen. Das erste Modell ist das Sicherheitsintegritätsstufenmodell und das zweite Modell ist das Werkzeugvertrauenseinstufungsmodell.

2.3.1.1 Automotive-Sicherheits-Integritätslevel (ASIL)

Das Risikomodell von ASIL ist ein quantitatives Modell [isoj]. Es gibt drei Ebenen der Anwendung, nämlich System, Software und Hardware. Die Systemebene stellt das Fahrzeug als Gesamtsystem dar und liegt in der alleinigen Verantwortung des OEM [LPP07]. Die Softwareebene kann in verteilte und unabhängig Funktionen unterteilt werden. Die verteilten Funktionen erfordern das größte Maß an Abstimmung, um eine Risikobe-

wertung vorzunehmen. Bei unabhängigen Funktionen kann das Entwicklungsteam die Risikoanalyse vornehmen. Bei der Hardwareebene kann das Entwicklungsteam die Risikobewertung vornehmen. Die Kennzahl hat folgende Abstufung: QM, ASIL-A, ASIL-B, ASIL-C und ASIL-D. Sie wird aus drei Faktoren gebildet: Exposure (E), Severity (S) und Controlability (C). Jeder Faktor wird aus einer Metrik gebildet wie folgt gebildet:

1. Exposure (E)

E0: Unvorstellbar

E1: Sehr niedrige Wahrscheinlichkeit (bei Überfahren eines Bahnübergangs kommt es zu einem Motorausfall)

E2: Niedrige Wahrscheinlichkeit (mehrmals im Jahr z.B. Anhängerfahrt)

E3: Mittlere Wahrscheinlichkeit (nicht täglich aber häufig wie eine nasse Straße)

E4: Hohe Wahrscheinlichkeit (bei jeder Fahrt)

Wenn diese Metrik mit den oben genannten Kriterien analysiert wird, kommt man zu folgenden Ergebnissen:

Die Abstufung umfasst fünf Stufen. Die Einstufung ist eine subjektive Einstufung durch den Gutachter, da das Verhalten des Fahrers und welche Witterungsverhältnisse bei der Fahrt vorliegen werden nur Schätzwerte sind. Die Metrik differenziert unterschiedlich wahrscheinliche Ereignisse. Durch die Verwendung von Fahrsituationen ist die Metrik im hohen Maß plausibel, da auch weite Personenkreise die Einstufung nachvollziehen können. Die Metrik ist durch die Verwendung einer Ordinalskala grundlegend vergleichbar. Die Metrik ist nur bedingt reproduzierbar, da es sich um eine subjektive Metrik handelt. Die Bestimmung der Eintrittswahrscheinlichkeit ist für eine Risikoanalyse sehr relevant, um die wichtigen Funktionen zu bestimmen, die regelmäßig ausgeführt werden. Durch die Beschreibung und einfache Einstufung hält sich der Aufwand in Grenzen.

2. Severity (S)

S0: Keine Verletzungen von Fahrinsassen (Auffahren auf einen Zaun oder Begrenzungspfahl mit $v < 15km/h$)

- S1:** Leichte und mittlere Verletzungen von Fahrinsassen (Frontalzusammenstoß von zwei PKW mit $v < 20\text{km/h}$)
- S2:** Schwere Verletzungen von Fahrinsassen - Überleben wahrscheinlich (Frontalzusammenstoß von zwei PKW mit $v > 20\text{km/h}$)
- S3:** Lebensgefährliche Verletzungen von Fahrinsassen - Überleben unwahrscheinlich (Seitenaufprall eines anderen Fahrzeugs mit $v > 35\text{km/h}$)

Die Abstufung umfasst vier Stufen. Die Einstufung erfolgt aufgrund von objektiven Verletzungseinstufungen, welche aber bei den ersten Einstufungen noch Schätzwerte sind, da in der Regel noch keine Crashtest Ergebnisse vorliegen. Die Metrik differenziert zwischen den verschiedenen Verletzungsarten. Durch die schriftliche Beschreibung der angenommenen Fahrsituation ist die Metrik im hohen Maß plausibel. Die Metrik ist durch die Verwendung einer Ordinalskala grundlegend vergleichbar. Die Metrik ist reproduzierbar, da die Verletzungsarten gut definiert sind. Die Bestimmung der Auswirkung ist für eine Risikoanalyse sehr relevant, um die Fahrsituation zu bestimmen, die die größte Gefährdung darstellt. Durch die Beschreibung und einfache Einstufung hält sich der Aufwand in Grenzen.

3. Controlability (C)

- C0:** Im Allgemeinen beherrschbar
- C1:** Einfach beherrschbar (Fahrer kann in aller Regel Personenschäden durch Bremsen vermeiden, wenn die Lenksäule beim Anfahren blockiert)
- C2:** Normalerweise beherrschbar (Fahrer kann bei Ausfall des ABS eine Notbremsung durchführen)
- C3:** Schwierig beherrschbar (Fahrer kann in der Kurve bei Ausfall der ESP-Regelung das Fahrzeug sicher abfangen)

Die Abstufung umfasst vier Stufen. Die Einstufung ist eine subjektive Einstufung durch den Gutachter. Da das tatsächliche Verhalten der Fahrer in der jeweiligen Fahrsituation nicht vorliegen kann, werden nur Schätzwerte verwendet. Die Metrik differenziert die unterschiedlichen Grade der Beherrschbarkeit einer Fahrsituation ausreichend. Durch die schriftliche Be-

schreibung der angenommenen Fahrsituation ist die Metrik im hohen Maß plausibel. Die Metrik ist durch die Verwendung einer Ordinalskala grundlegend vergleichbar. Die Metrik ist nur bedingt reproduzierbar, wenn keine Standards von der Automobilindustrie zusätzlich eingeführt werden.

Die Bestimmung der Beherrschbarkeit wird bei anderen Risikoverfahren nicht als relevant angesehen, aber gerade im Zuge der verstärkten Automatisierung des Automobils nimmt die Bedeutung der Fragestellung zu, ob die Fahrsituation beim Ausfall der Fahrerassistenzfunktionen noch vom Fahrer beherrschbar ist.

Durch die Einführung von modernen Sicherheitsfunktionen können Fahrsituationen beherrschbarer werden als ohne die Sicherheitsfunktion. So wurde durch den Einsatz des ESP-Systems [Rei10] die Kontrollierbarkeit bei Kurvenfahrten deutlich erhöht. Trotz der Komplexität von Sicherheitsfunktionen kann eine allgemeine erhebliche Minimierung des Risikos durch eine bessere Kontrollierbarkeit herbei geführt werden und somit die funktionale Sicherheit verbessern. Durch die Beschreibung und einfache Einstufung hält sich der Aufwand in Grenzen. Die Einstufung erfolgt nach weichen Faktoren, die nur ein Gutachter, der auch ein Systemexperte ist, korrekt beurteilen kann.

Tabelle 2.1 ASIL

		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	ASIL-A
	E4	QM	ASIL-A	ASIL-B
S2	E1	QM	QM	QM
	E2	QM	QM	ASIL-A
	E3	QM	ASIL-A	ASIL-B
	E4	ASIL-A	ASIL-B	ASIL-C
S3	E1	QM	QM	ASIL-A
	E2	QM	ASIL-A	ASIL-B
	E3	ASIL-A	ASIL-B	ASIL-C
	E4	ASIL-B	ASIL-C	ASIL-D

Aus den drei Faktoren wird der ASIL Level bestimmt, wie in der Tabelle 2.1 dargestellt. Die Abstufung der ASIL Einstufung hat fünf Stufen. Die Einstufung erfolgt durch den Risikograph wie in der Tabelle 2.1 dargestellt. Die Metrik differenziert die unterschiedlichen Grade der Risikoauswirkung ausreichend. Durch die angenommenen Fahrsituationen ist die Metrik im hohen Maß plausibel. Die Metrik ist durch die Verwendung einer Ordinalskala grundlegend vergleichbar. Die Metrik ist reproduzierbar, wenn die Teilmetriken bei der Analyse zum gleichen Ergebnis kommen. Durch die einfache Beschreibung und einfache Einstufung hält sich der Aufwand in Grenzen. Die Einstufung erfolgt nach Faktoren, die nur ein Gutachter, der auch ein Systemexperte ist, korrekt beurteilen kann. Gleichzeitig sollte zur Gewährleistung der Unabhängigkeit der Gutachter kein Mitglied der Fachabteilung sein.

2.3.1.2 Werkzeugvertrauenseinstufung (TCL)

Die Norm ISO 26262 schreibt eine Einstufung der Testwerkzeuge vor und somit auch indirekt die Testplattform in diesem Fall ein HiL. Ein HiL ist ein System aus Hardware- und Softwareteilsystemen, die erst als Komplettsystem als Testwerkzeug das Testobjekt testen können. Dadurch kann der HiL als ein Werkzeug angesehen werden. Deswegen kann für den HiL eine Gesamteinstufung vorgenommen werden. Das Werkzeugvertrauenslevel (TCL) hat drei Stufen 1-3. Die Kennzahl besteht aus zwei Faktoren, Tool Impact (TI) und Tool Error Detection (TD).

1. Tool Impact (TI)

TI 0: Das Werkzeug kann keine Sicherheitsanforderung verletzen.

TI 1: Wenn nicht TI 0 dann TI 1.

2. Tool Error Detection (TD)

TD 1: Das Vertrauen ist hoch, das Werkzeug selber verhindert, dass eine Sicherheitsanforderung verletzt wird, oder es ist leicht zu erkennen, wenn es eine Sicherheitsanforderung verletzt.

TD 2: Das Vertrauen ist mittel, das Werkzeug kann in der Regel eine Sicherheitsanforderungsverletzung verhindern oder es ist zu erkennen, wenn es eine Sicherheitsanforderung verletzt.

TD 3: Das Vertrauen ist gering, das Werkzeug kann teilweise verhindern, dass eine Sicherheitsanforderung verletzt wird oder es ist schwer zu erkennen, wenn es eine Sicherheitsanforderung verletzt.

Die Einstufung TCL-Kennzahl erfolgt durch einen Gutachter und ist ein Schätzwert, der entweder auf Betriebserfahrung, einer Verifikation oder einer Validation basiert. Die Verfahren können nur eine Abwesenheit von Fehlern aber keine Fehlerfreiheit sicherstellen. Die Metrik differenziert unterschiedliche Vertrauensebenen. Die Metrik ist plausibel, da erprobte Werkzeuge in der Regel zuverlässig sind. Die Metrik ist durch die Verwendung einer Ordinalskala grundlegend vergleichbar. Die Metrik ist reproduzierbar, da die Gutachter bei der gleichen Verfahrensweise zum gleichen Ergebnis kommen. Die Bestimmung des Vertrauenslevels ist für die Norm sehr relevant. Gerade bei neuen Werkzeugen ist der Aufwand erheblich.

Tabelle 2.2 Werkzeugvertrauenslevel nach [isoi]

TI	TD	TCL
0	1	1
	2	
	3	
1	1	2
	2	
	3	3

Durch die einfache Bildung der Kennzahl können alle oben genannten Aussagen auch auf die Kennzahl übernommen werden.

2.3.2 Fehlermöglichkeits- und Einflussanalyse (FMEA)

Die FMEA wurde ursprünglich von der NASA [20003] entwickelt. Ford hat die FMEA nach Qualitätsproblemen eingeführt und von Ford aus hat sich die FMEA in der gesamten Automobilindustrie verbreitet. Die Kriterien der FMEA sind Entdeckungswahrscheinlichkeit (E), Auftrittswahrscheinlichkeit (A) und Bedeutung (B). Jede der Kriterien hat eine Skala von 1-10. Die Risikoprioritätszahl (RPZ) ist ein Produkt der drei Faktoren und umfasst einen Wertebereich von 1-1000 [20003].

Die FMEA gibt es in verschiedenen Ausprägungen, die unterschiedliche Metriken zur Bildung der drei Kriterien nutzen. Die bekanntesten FMEAs sind die System-, Prozess- und Produkt-FMEA [20003]. In diesem Abschnitt wird die FMEA für mechatronische Systeme nach der Definition der VDA [20003] untersucht. Für diese FMEA wurden einige wichtige Definitionen getroffen. Die erste Definition ist, dass die Hardware wie auch die Software des mechatronischen Systems gesamtheitlich betrachtet werden. Es wurden zwei zusätzliche Fehlerklassen eingeführt: der Scheinfehler und der Ausfall im Anforderungsfall.

Ein Scheinfehler tritt auf, wenn das System einen Fehler erkennt, obwohl kein Fehler vorliegt. Ein Ausfall im Anforderungsfall liegt dann vor, wenn

der Erstfehler nicht erkannt wird und die Fehlerreaktion nicht wie vorgesehen reagiert. Der FMEA-Strukturbaum wird in zwei Teilbäume aufgeteilt. Der erste Strukturbaum gilt für den Normalbetrieb und der zweite Strukturbaum gilt für Funktionen im Fehlerbetrieb. Im zweiten Strukturbaum werden die Diagnosefunktionen modelliert. Der Fehler wird nach den folgenden sieben Eigenschaften bewertet:

- Bedeutung der beschriebenen Fehlerfolge
- Auftrittswahrscheinlichkeit der Fehlerursache im Entwicklungsprozess
- Auftrittswahrscheinlichkeit der Fehlerursache im Kundenbetrieb
- Entdeckungswahrscheinlichkeit im Kundenbetrieb
- Ausfallwahrscheinlichkeit der Fehlerreaktion
- Entdeckungswahrscheinlichkeit beim Service
- Wirksamkeit der Maßnahme zur Reduktion einer der oberen Eigenschaften

Somit können mehrere RPZ in Abhängigkeit des Kundenbetriebs oder Entwicklungsprozesses dargestellt werden, um die jeweilige Maßnahme zu der Reduktion der Auftrittswahrscheinlichkeit oder zu der Reduktion der Entdeckungswahrscheinlichkeit im Kundenbetrieb oder im Entwicklungsprozess zu bewerten. Die Metrik differenziert unterschiedliche Eigenschaften eines Fehlers. Durch die Analyse der Eigenschaften eines Fehlers ist die Metrik im hohen Maß plausibel. Die Metrik ist durch die Verwendung einer Ordinalskala grundlegend vergleichbar. Die Metrik ist nur bedingt reproduzierbar, da es sich um eine Schätzung von Seiten der Gutachter handelt. Die Bestimmung der Schwere eines Fehlers ist für eine Risikoanalyse sehr relevant. Für eine vollständige FMEA ist der Aufwand sehr hoch.

2.3.3 Fehlerbaumanalyse (FBA)

Die FBA [Dut] leitet aus einem unerwünschten Ereignis alle Ursachen mit Hilfe der kombinatorischen Dekomposition ab. Somit können Fehlerpfade gefunden werden. Solche Fehlerpfade können dann gut in der Diagnose verwendet werden, um geeignete Diagnoseverfahren zu bestimmen. Als

Beispiel wird ein kleiner Fehlerbaum (siehe Bild 2.7) gewählt, der drei Fehlerquellen hat. Die Fehlerquellen sind durch ein „logisches Oder“ verknüpft. In einem zweiten Schritt wird der Fehlerbaum vereinfacht. In einem

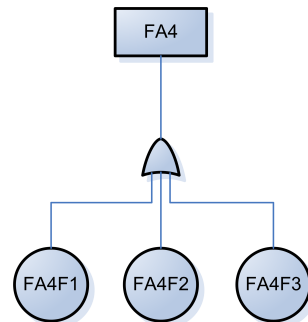


Bild 2.7 Fehlerbaum

dritten Schritt kann aus dem Fehlerbaum die Nichtverfügbarkeit und Häufigkeitsdichte des Ausfalls berechnet werden. Die Nichtverfügbarkeit einer Betrachtungseinheit ist die Wahrscheinlichkeit, sie zum Zeitpunkt t ausgefallen anzutreffen. Die Verfügbarkeit $V(t)$ ist das Komplement von $U(t)$. Das führt zur Formel:

$$V(t) = 1 - U(t)$$

Die Ausfallhäufigkeit $H(t)$ einer Betrachtungseinheit ist der Erwartungswert im Zeitintervall $(0, t)$. Die Abstufung umfasst so viele Stufen, wie weit man das System dekomponiert. Die Metrik differenziert unterschiedliche Fehlerpfade. Durch die kombinatorische Dekomposition eines Fehlers ist die Metrik im hohen Maß plausibel. Die Metrik ist durch die Verwendung der Kombinatorik zur Erstellung des Fehlerbaums und durch die Wahrscheinlichkeitsmathematik im hohen Maß vergleichbar. Die Metrik ist durch die Vereinfachung in der zweiten Stufe des Verfahrens im hohen Maß reproduzierbar. Die Bestimmung aller Fehlerpfade und der Eintrittswahrscheinlichkeit ist für eine Risikoanalyse sehr relevant. Der Aufwand für das gesamte System ist auch mit Werkzeugen zur Durchführung einer FBA sehr hoch.

2.4 Endlicher Automat

Für reaktive Systeme bietet sich die Modellierung als endlicher Automat an, aber komplexe Funktionen führen zu sehr großen und unübersichtlichen Automaten. Deswegen wurden verschiedene Methoden zur besseren

Darstellung entwickelt. Zwei bekannte Darstellungen sind zum einen die Petri-Netze [CS01] zum anderen die Statecharts nach Harel [Har87]. Petri-Netze wie im Bild 2.8 dargestellt sind sehr abstrakte Automaten, in denen es zwei verschiedene Typen von Zuständen gibt, um das Erzeugen und den Konsum von Elementen darzustellen.

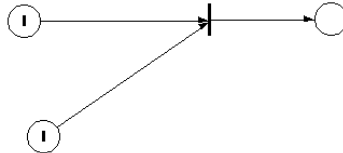


Bild 2.8 Petri-Netz

Für das Modellieren eines reaktiven Systems ist diese Darstellung schlechter geeignet als die Verwendung eines rein endlichen Automats, weil sich zwar das Verhalten bzgl. des Zugriffs auf Ressourcen gut modellieren lässt, aber Petri Netze, die das exakte Verhalten von reaktiven Systemen beschreiben, größer sind als reine endliche Automaten.

Harel hat seine Methode so definiert: Statechart = Zustandsdiagramm + Hierarchie + Orthogonalität, was eine übersichtlichere und kompaktere Darstellung ermöglicht. Die Hierarchie wie im Bild 2.9 ist formal gesehen eine „exklusive ODER“-Beziehung von Zuständen. Es werden alle Zustände in einen Superzustand zusammengefasst. Dadurch kann die Gesamtdarstellung der Zustandsautomaten reduziert werden, in dem der Superzustand die ihm zugeordneten Zustände ersetzt. Der innere Ablauf des Superzustands bleibt erhalten. Dieser Ablauf kann wiederhergestellt werden oder separat angezeigt werden, wenn auf die inneren Zustände umgeschaltet werden soll.

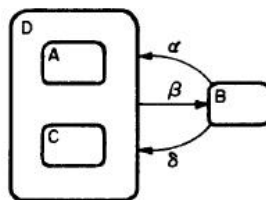


Bild 2.9 Hierarchie nach [Har87]

Orthogonalität ist formal gesehen eine „UND“-Beziehung von Zuständen. Die „UND“-Beziehung wird durch mindestens zwei Superzustände darge-

stellt. Jede dieser Superzustände umfasst mehrere Zustände. Die Zustände innerhalb eines Superzustands beschreiben einen Teilautomat. Durch das verwenden von mehreren orthogonalen Superzuständen kann ein verteiltes System kompakt beschrieben werden. Jeder Superzustand beschreibt ein unabhängiges Teilsystem. Das Gesamtsystem kann durch ein Produkt aller inneren Zustände miteinander wieder hergestellt werden.

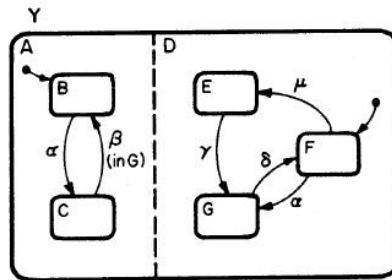


Bild 2.10 Orthogonalität nach [Har87]

Die Statecharts wurden weiterentwickelt zu den Stateflows für Matlab/Simulink sowie zu den UML Zustandsautomaten [UML]. UML erweitert das Statechart um weitere Mengen von Zustandstypen, um das Modellieren zu vereinfachen. Außerdem erweitert es den Zustandsübergang um eine Reaktion bzgl. des eingetretenen Ereignisses. Das führt zum folgenden Konstrukt: „Ereignis[Wächter]/Aktion“, wobei die Sprachsyntax frei wählbar ist. Der Zustandsautomat ist wie folgt definiert:

- $\Sigma \in e$: Ist die Menge der akzeptierten Ereignisse. Jedes Ereignis kann 0..n Attribute besitzen.
- $S \in s$: Ist die Menge der Zustände.
- $S \subset S_0$: Ist die Menge der Startzustände und ist eine nicht leere Teilmenge der Zustände.
- $S \subset P$: Ist die Menge von Pseudozuständen, die eine Teilmenge der Zustände darstellen.
- $S \subset T$: Ist die Menge der Zustandsübergänge.
- $S \subset F$: Ist die Menge der Endzustände und ist eine Teilmenge der Zustände.
- $W \in w$: Menge von kombinatorischen Ausdrücken, die auf die Attribute angewendet werden.

- $A \in a$: Ist die Menge der Aktionen.
- ω : Die Menge der Relationen für die gilt: $s_n : e \times w \times a \rightarrow s_n \vee s_n : e \times w \times a \in T \rightarrow s_m$

Die Relationen der Menge ω können auch als Regeln für das Verhalten des Systems in dem jeweiligen Zustand betrachtet werden. Die Regeln beschreiben primär das erwartete Verhalten; wenn das System eine Eigendiagnose besitzt, wie es bei Softwarefunktionen von SG normal ist, wird dadurch auch das unerwartete Verhalten des Systems beschrieben.

2.5 Expertensysteme

Expertensysteme stellen eine etablierte Systemarchitektur [CGJ89] dar, die in der Lage ist, Expertenwissen in eine automatisierbare Form zu bringen und dieses Expertenwissen anzuwenden. In vielen Teilbereichen der Informatik werden sie erfolgreich eingesetzt. Grundlegend funktioniert ein Expertensystem so, dass zuerst Daten hinzugefügt und anschließend die Regeln auf die Daten angewandt werden, um so an Informationen zu kommen.

In der Form der Workflow Systeme [Bau04] haben sie im Fachbereich der Wirtschaftsinformatik einen sinnvollen Anwendungsfall und werden als Ausführungsplattformen von Business-Rule-Management-Systems eingesetzt. Im Bereich des Wissensmanagements werden auch Expertensysteme [BKI06] angewendet. Expertensysteme werden im Automotive Bereich als Diagnosesysteme eingesetzt. Hierbei gibt es zwei Arten. Zum einen als Dialogsystem in der Werkstatt, um bei der Problemanalyse zu unterstützen, zum anderen als Diagnosesystem im SG zur Fehleranalyse [PC00]. Gerade im zweiten Fall zeigt sich eine Schwäche der Expertensysteme. Sie sind nicht in der Lage, auf nicht vorgesehene Ereignisse, für die es keine Repräsentation im System gibt, zu reagieren.

Die Expertensysteme werden durch eine Regelbasis und eine Wissensbasis programmiert. Die Regelbasis besteht aus einer Liste von Regeln der Struktur „WENN X gilt DANN führe Y aus“. Die Wissensbasis stellt eine Datenstruktur dar. Die Struktur kann verschiedene Paradigmen umsetzen wie beispielsweise eine Tabellenrelation einer Datenbank, Structs-ähnliche

Notationen eines prozeduralen Programms oder eine Objekthierarchie. Die Wissensbasis hat zum einen eine initial gefüllte Datenstruktur und zum anderen wird während der Laufzeit des Expertensystems die Wissensbasis erweitert. Die Wissensbasis kann durch externe Daten erweitert werden. Allerdings kann die Wissensbasis auch durch das Ausführen von Regeln intern verändert werden. Zum Ausführen der Regel benutzen Expertensysteme ein Interferenzsystem. Das Kernelement des Interferenzsystems ist ein Mustererkennungsalgorithmus, der in der Lage ist zu erkennen, welche der Regeln der Regelbasis durch die Wissensbasis erfüllt sind.

Ein weitverbreiteter Mustererkennungsalgorithmus ist der Rete Algorithmus [For82]. Als erster Schritt wird ein Netz aus Knoten erzeugt. Die Knoten werden aus den Regeln abgeleitet. Eine Regel wird in zwei Teile aufgeteilt: der erste Teil umfasst „WENN X“ während der zweite Teil „DANN Y“ umfasst. X stellt einen booleschen Ausdruck dar. Jeder Knoten stellt eine erfüllte Bedingung von X dar. Wenn neue Daten hinzugefügt werden, können neue Knoten erzeugt werden, wenn sie neue Bedingungen von X erfüllen. Eine Regel wird dann ausgeführt, wenn der X-Teil einer Regel durch die Wissensbasis erfüllt ist. Es kann durchaus der Fall vorkommen, dass durch das Einfügen eines Knotens mehrere Regeln aktiviert werden. Um diesen Konflikt aufzulösen können verschiedene Lösungsstrategien angewendet werden. Sie sind aber nicht Bestandteil des Algorithmus. Jede aktivierte Regel wird in eine Ausführungsliste geschrieben. Dann wird die Liste abgearbeitet, indem der Y-Teil der Regel ausgeführt wird. Wenn der Fall auftritt, dass die Y-Regel die Wissensbasis verändert, werden zuerst die Veränderungen im Netzwerk erzeugt und gegebenenfalls neue Regeln auf der Ausführungsliste hinzugefügt. Erst dann wird die nächste aktive Regel der Ausführungsliste ausgeführt. Ein Durchlauf endet erst, wenn die Ausführungsliste leer ist.

Durch die Rete-Implementierung ist eine Regel im Expertensystem atomar. Es gibt viele verschiedene Expertensysteme. Eines der ältesten und ausgereiftesten Expertensysteme ist CLIPS. CLIPS [Ril] hat eine Rete-Implementierung und befolgt folgende Lösungsstrategien:

Tiefe Wenn die Regeln den gleichen Rang haben, werden die Regeln zuerst ausgeführt, deren Knoten zuletzt eingefügt wurden. Werden zwei

Regeln vom gleichen Knoten aktiviert, ist die Ausführung nicht bestimmt, sondern die auszuführende Regel wird willkürlich bestimmt.

Breite Wenn die Regeln den gleichen Rang haben, werden die Regeln zuerst ausgeführt, deren Knoten zuerst eingefügt wurden. Werden zwei Regeln vom gleichen Knoten aktiviert, ist die Ausführung nicht bestimmt, sondern die auszuführende Regel wird willkürlich bestimmt.

Einfach Die Regeln werden bzgl. der Anzahl der Argumente bewertet. Es wird die Regel zuerst ausgeführt, die die wenigsten Argumente besitzt.

Komplex Die Regeln werden bzgl. der Anzahl der Argumente bewertet. Es wird die Regel zuerst ausgeführt, die die meisten Argumente besitzt.

LEX Jeder Knoten erhält einen Zeitstempel und alle Knoten einer Regel werden zeitlich geordnet. Jede neue aktivierte Regel wird bzgl. ihres höchsten zeitlichen Rangs ihrer Knoten eingeordnet. Die Regel mit dem höchsten Zeitstempel wird ausgeführt.

MEA Jeder Knoten erhält einen Zeitstempel und alle Knoten einer Regel werden zeitlich geordnet. Jede neue aktivierte Regel wird bzgl. der Zeitstempel ihres ersten Knotens eingeordnet. Nur dann, wenn zwei Regeln den gleichen ersten Knoten besitzen, wird LEX zur weiteren Konfliktauflösung angewendet.

Zufall Die Regel wird zufällig aus der Liste der aktivierten Regeln ausgeführt.

CLIPS ist eine Public Domain Software: Dadurch ist es möglich, das System an die Anforderungen anzupassen. Desweiteren steht für eine Vielzahl von Betriebssystemen und Compilern der Quellcode zur Verfügung. CLIPS ist dafür ausgelegt, eingebettet verwendet zu werden und hat eine umfangreiche Selbstdiagnose. Deswegen war es die erste Wahl. Das Expertensystem wurde auch deswegen gewählt, weil sich die Daten und die Regeln unabhängig beschreiben lassen: dadurch kann ein Zustandsautomat leicht in eine Regelbasis eines Expertensystems umgewandelt werden, während die Fakten sich aus den Datenquellen automatisch erzeugen lassen. Dadurch war mit wenig Aufwand ein Beobachter auf Basis von UML zu realisieren. Die Frage der Performance und der zeitlichen Auflösung wurden während der Umsetzung untersucht.

3 Anforderungen aus der Praxis an das Prüfverfahren

Das mechatronische Fahrwerk [Ise06] ist aufgrund der historischen Entwicklung ein verteiltes System, in dem sich die Lenkung, die Bremse und das aktive Fahrwerk unterschiedlich schnell entwickelt haben. In einem ersten Schritt stellen die einzelnen Systeme den anderen Systemen ihre Sensorinformationen zur Verfügung. Dann werden Regelinformationen übertragen, was dann zu verteilten Funktionen führt. Die Entwicklung geht zum einen in die Richtung der Elektrifizierung und zum anderen in Richtung zusätzlicher Softwarefunktionen. Es gibt bei den meisten Automobilherstellern eine Arbeitsteilung, so dass das Bremssystem, das Lenksystem und das Fahrwerkssystem von verschiedenen Modulherstellern oder Systemherstellern entwickelt werden.

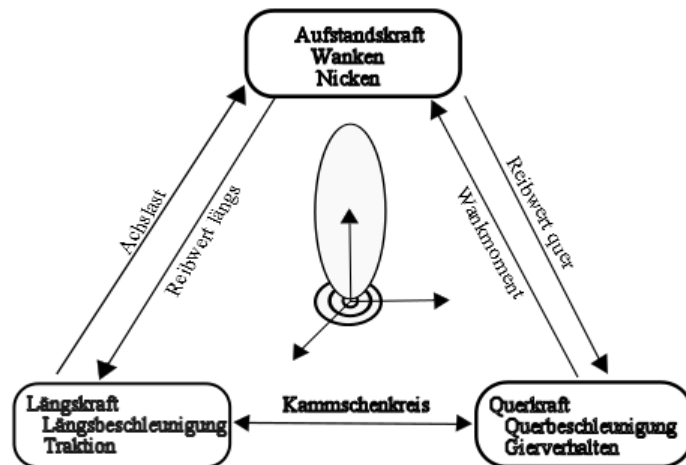


Bild 3.1 Kammschen Kreis nach [Ise06]

Da aber alle drei Systeme durch den Kammschen Kreis im Grenzbereich verbunden sind, wie im Bild 3.1 dargestellt ist, müssen alle Systeme bei der Regelung im Grenzbereich aufeinander abgestimmt, entwickelt und getestet werden. Die Kernaussagen des Kammschen Kreises sind, dass nur bei der Geradeausfahrt die maximale positive wie negative Beschleunigung

möglich ist und dass die Längsbeschleunigung und Querschleunigung durch einen Kraftschluss wie im Bild 3.1 dargestellt beschränkt sind. Die verschiedenen Funktionen müssen als ein Gesamtsystem für die Fahrdynamik zusammenarbeiten. Daher muss eine Koordination der Funktionen der Systeme erfolgen. Es gibt nach [Ise06] drei Arten der Koordination der Systemfunktionen:

- Die friedliche Koexistenz der Funktionen. Die Systeme sind lose gekoppelt und tauschen über den Bus Sensorinformationen aus. Jedes System stellt sicher, dass es andere Systeme und deren Funktionen nicht negativ beeinflusst. Der Vorteil ist, dass nach der Abstimmung der Systeme, diese unabhängig von einander entwickelt werden können. Erst bei der Integration muss die friedliche Koexistenz ausführlich überprüft werden. Eine direkte Kooperation der Funktionen ist nicht möglich und für verteilte Funktionen schlecht geeignet.
- Die kooperative Koexistenz der Funktionen. Die Systeme sind stark gekoppelt. Neben den Sensorinformationen werden auch Abstimmungsinformationen übertragen. Die Regelung kann besser abgestimmt werden. Hierbei ist auch die Überprüfung bei der Integration die aufwändigste aller Koordinationsformen. Ein Beispiel aus der Praxis für eine kooperative Koexistenz ist der VW Touareg aus dem Jahr 2010 [20110].
- Die Zentralregelung der Funktionen. Die Systeme sind so zusammengefasst, dass es einen Zentralregler gibt, der alle relevanten Informationen sammelt und regelt. Aus Regelungssicht und aus Abstimmungssicht ist der Zentralregler die beste Lösung. Die Lösung hat aber auch Nachteile. Der erste Nachteil ist, dass ein Zentralregler einen Flaschenhals der Regelstrecke der Fahrdynamik für das Gesamtsystem darstellt und eine erhöhte Sicherheitsanforderung hat. Der zweite Nachteil ist, dass das Ausführen der zentralen Regelung eine leistungsfähige Steuergeräteplattform notwendig macht. Der Zentralregler wird als ein viertes System umgesetzt, das die Partnersteuergeräte regelt. Somit müssen wie bei der kooperativen Koexistenz neben Sensorinformationen auch Zustandsinformationen ausgetauscht werden. Als Beispiel aus der Praxis für einen Zentralregler dient der aktuelle 7er BMW [20008].

Je mehr Informationen der Regelung über den Bus ausgetauscht werden, desto besser können die Systeme überwacht werden. Bei der Zentralregelung und bei der kooperativen Koexistenz werden die meisten Informationen übertragen und können so am besten überwacht werden. In den letzten Jahren sind die Speichermedien so günstig geworden, dass es möglich ist, alle Daten der Fahrzeugprototypen und an den HiLs während der Entwicklung eines Automobils zu archivieren. Die Daten stellen einen Informationsschatz dar, der einer tiefer gehenden Analyse zu unterziehen ist.

3.1 Aktives Fahrwerk

Es gibt verschiedene Umsetzungen und Zielsetzungen eines aktiven Fahrwerks. Grundsätzlich ist es möglich, durch ein aktives Fahrwerk eine bessere Reaktion des Fahrwerks bzgl. der aktuellen Fahrsituation zu erreichen als durch ein passives Fahrwerk. Das kann entweder eine bessere Rückkopplung oder eine stärkere Dämpfung sein [Ise06]. Ein aktives Fahrwerk kann in drei Teilsysteme aufgeteilt werden:

- Die „aktive Dämpfung“ ermöglicht eine an Geschwindigkeit und Bodenbeschaffenheit angepasste Dämpfung.
- Die „aktive Wankstabilisierung“ ermöglicht es, eine Gegenkraft aufzubringen, um der Neigung bei Kurvenfahrten entgegen zu wirken, um so eine stabilere und angenehmere Kurvenfahrt zu ermöglichen.
- Die „aktive Niveuregelung“ ermöglicht eine der Geschwindigkeit angepasste Bodenfreiheit.

Je nach Fahrzeugtyp besteht das Gesamtsystem aus allen oder aus einer Teilmenge der oben genannten Teilsysteme. Aus Vernetzungssicht können durch eine Kooperation mit der Bremse und Lenkung Sensoren eingespart werden, da alle Systeme grundlegende Sensorinformationen brauchen und diese über den Bus von einem Sensor an alle Systeme verteilen.

3.2 Bremse

Die Bremse war eine der ersten Plattformen für mechatronische Funktionen wie ABS und ESP [Ise06]. Das aktuelle Bremssystem der Oberklassenfahrzeuge besitzt noch weitere Funktionen wie beispielsweise die Stabilisierung des Fahrzeugs nach einem Unfall. Die Sicherheitsfunktionen erfordern hochwertige und zeitlich hoch aufgelöste Sensorinformationen, wie die Längsbeschleunigung, Querbeschleunigung und die Gierrate, um rechtzeitig einzugreifen. Diese Informationen werden anderen Systemen zur Verfügung gestellt. Aus Vernetzungssicht arbeitet die Bremse mit weiteren Systemen im Fahrzeug wie dem Antrieb und dem Getriebe zusammen. Aus Sicherheitssicht ist die Bremse ein System, an das die höchsten Anforderungen bzgl. der Diagnose und Verfügbarkeit gestellt werden. In der Regel hat die Bremse eine aufwändige Degenerationstrategie für einen Fehlerbetrieb des Steuergeräts, um selbst bei einem Teilausfall zumindest eine grundlegende Bremsfunktion bereitzustellen. Für die Sensoren [Bau03] wird eine mehrstufige Strategie verwendet:

1. Die Sensoren werden während des ganzen Fahrbetriebes auf Leitungsbruch und nicht plausible Werte überwacht.
2. Bei wichtigen Sensoren wird, wenn es möglich ist, beim Einschalten und während des Fahrbetriebes aktiv getestet.
3. Es gibt ein internes Modell, das die Sensoren im stationären Fahrbetrieb analytisch überprüft.

3.3 Lenkung

Seit der Einführung der Überlagerungslenkung kann die Lenkung das Fahrzeug unabhängig vom Fahrer beeinflussen [Ise06]. Dadurch ist es möglich, je nach Geschwindigkeit unterschiedliche Unterstützungskräfte zur Lenkbewegung des Fahrers aufzubringen. Dadurch kann der Fahrer beispielsweise beim Einparken stark unterstützt werden, während bei hohen Geschwindigkeiten die Lenkung eine hohe Gierwinkeldämpfung erreicht [Ise06]. Es gibt bei den aktiven Lenkungen verschiedene Systeme, zum Beispiel: die rein elektrische Lenkung oder die elektro-hydraulische Lenkung [Bau03]. Aus

funktionaler Sicherheitssicht hat die elektro-hydraulische Lenkung noch eine Reservefunktion, wenn es zu einem Spannungseinbruch kommt, da der Drucktank als Puffer dient, während die elektrische Lenkung eine deutliche Gewichtseinsparung darstellt und weniger Energie verbraucht [Bau03]. Die Lenkung arbeitet mit der Bremse beim „ μ Split-Anfahren“ oder bei der „Spurführungsunterstützung“ zusammen [Ise06]. Die Lenkung darf keine ungewollten und keine plötzlichen Lenkbewegungen ausführen.

3.4 Vernetzte Sicherheitsfunktionen

Die aktive Sicherheit wurde durch die Vernetzung der drei Systeme deutlich erhöht. Durch die Vernetzung und die Eingriffsmöglichkeiten wird das Fahrzeug im Grenzbereich besser beherrschbar bei gleichzeitiger Dämpfung von Fahrerfehlern. Zudem können durch die Vernetzung auch neue Komfortfunktionen und Kraftstoffsparfunktionen umgesetzt werden. Dabei muss sichergestellt sein, dass diese Funktionen keinen negativen Einfluss auf die Fahrsicherheit haben. Die aktiven Systeme besitzen für verschiedene komplexe Fahrsituationen zusätzliche Sicherheitsfunktionen, wie zum Beispiel „Überholen bei hohen Geschwindigkeiten mit kritischem Übersteuern“.

3.4.1 Testen von vernetzten Sicherheitsfunktionen

Der Akzeptanztest und Systemtest sind Bestandteil der letzten Stufe im V-Modell. Somit ist es die letzte Möglichkeit, Fehler zu finden, bevor der nächste Entwicklungszyklus oder die Kundenfreigabe erfolgt. In der beschränkten Zeitspanne beim Durchführen der Tests sollte der Testfokus auf sicherheitsrelevante Systemeigenschaften sowie auf Systemeigenschaften, die in den unteren Stufen nur eingeschränkt oder gar nicht testbar sind, liegen. Die sicherheitsrelevanten Systemeigenschaften leiten sich aus den Ergebnissen der Sicherheitsanalyse ab. Die Einschränkungen leiten sich aus der Analyse der unteren Teststufen des V-Modells ab. Diese sind in der Regel vernetzte Funktionen, da sich die beteiligten Steuergeräte nur schlecht simulieren lassen. Bei nicht verteilten Funktionen für frühe Stufen des V-Modells wird das Testen von sicherheitsrelevanten Funktionen von [LBK] untersucht, während [Wie] Funktionen ermittelt, die mit einem

geringen Aufwand in frühen Phasen des V-Modells testbar sind. So bleiben verteilte und sicherheitsrelevante Funktionen übrig, die am besten nach der Systemintegration zu testen sind.

3.4.2 Verbunds-HiL

In den seltensten Fällen umfasst ein Verbunds-HiL alle Steuergeräte eines Fahrzeugs. Es werden mehrere Verbunds-HiLs aufgebaut, die eine Teilmenge der SG testen. Dieser Verbund von SG wird unter verschiedenen Kriterien zusammengefasst wie zum Beispiel nach den Kommunikationstopologien oder nach der Unternehmenstruktur. Der Testfokus liegt auf verteilten Funktionen und deren Reaktion auf Störgrößen [Ise06]. In absehbarer Zeit ist es nicht möglich, eine Aussage bzgl. der Regelgüte mit einem Verbunds-HiL zu treffen. Dies hat mehrere Gründe:

- Einige Sensorsignale sind nicht unter der Echtzeitanforderung aus der Simulation mit der üblichen Rechnerausstattung heraus zu erzeugen wie zum Beispiel die Radarreflektionen eines ACC.
- Der hohe Platzbedarf der Prüfstandsaktorik für Systeme wie das Fahrwerk oder die Lenkung um die Kraft zu erzeugen und aufzunehmen, die in extremen Fahrsituationen auftreten.
- Die notwendigen Modellvereinfachungen für einen ökonomischen Einsatz von Rechnerkapazität haben eine negative Auswirkung auf Güteaussagen.

Deswegen liegt der Testfokus am Verbunds-HiL auf dem Testen der Fehlererkennung und dem Fehlerbetrieb der Steuergeräte im Verbund bei dynamischer wie statischer Simulation. Schon dieser Testraum ist ausreichend groß, um die vorhandene Testzeit vollständig auszufüllen. Die Auswirkungen der so gefundenen Fehler sind sicherheitskritisch und rechtfertigen einen hohen Testaufwand. Bei der dynamischen Simulation am Verbunds-HiL besteht das HiL-Modell aus einem Fahrermodell, einem Umgebungsmodell und einem Teilfahrzeugsystem für die nicht real vorhandenen Fahrzeugsysteme, die nicht direkt aus der Simulation aus dem Umweltmodell ableitbar sind.

Das Bild 3.2 zeigt ein typisches HiL-Modell. Die drei realen Steuergeräte sind hervorgehoben. Das Fahrermodell steuert alle Steuergeräte, diese sind

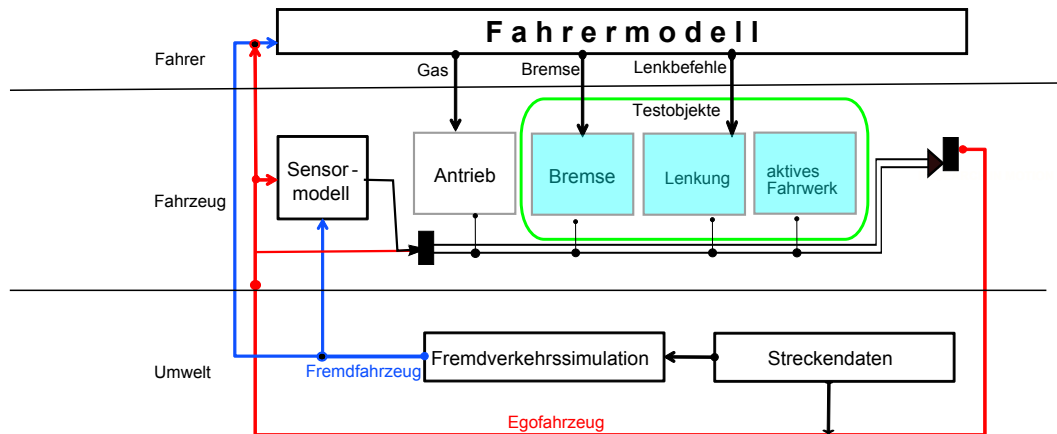


Bild 3.2 HiL-Modell

durch das Netzwerk verbunden. Der Motor wird simuliert wie auch die Umwelt und die Sensoren. Damit der Verbunds-HiL wirtschaftlich betrieben werden kann, sollte eine hohe Automatisierung umgesetzt sein. Das szenariobasierte Testen am V-HiL kann dazu beitragen. Auch bei verteilten Funktionen der Fahrwerkssysteme ist das szenariobasierte Testen möglich und sinnvoll. Hierbei sollten dann die Szenarien aus der Sicherheitsanalyse überprüft werden. Die Testmächtigkeit eines Verbunds-HiL hängt von der Güte der Simulation ab und von der Möglichkeit zur Manipulation der SG, um Fehler in das System einzubringen.

3.4.3 Prototyp

Ein Prototyp ist ein Fahrzeug, das als ein Steuergeräteträger dient. Für jeden Systemtest wird das Fahrzeug mit unterschiedlichen Steuergeräten bestückt. Je nach Entwicklungsstand und Testzweck werden alte Steuergeräte mit neuen Steuergeräten gemischt [SZ05]. Bei einem Prototyp wird die Parametrierung der Steuergeräte überprüft, ein Dauerlaufstest durchgeführt oder eine gezielte Fahrerprobung durchgeführt [Sax08]. Durch den Kostendruck wird bei der Entwicklung neuer Fahrzeuge die Anzahl an Prototypen bei einer steigenden Anzahl von verteilten Funktionen gleich bleiben oder sich sogar reduzieren. Als Maßnahme zur Beherrschung des Entwicklungsprozesses wird mehr Messtechnik in die Prototypen eingebaut und eine Datenhaltungsinfrastruktur aufgebaut, so dass die Messungen nicht

nur während der Fahrt analysiert werden, sondern auch zu einem späteren Zeitpunkt. Hierbei werden große Datenmengen erzeugt, die nur werkzeuggestützt auswertbar sind.



Bild 3.3 Beispiel eines Messfahrzeugs

3.4.4 Fahrsimulator

Bei einem Fahrsimulator wird die Mensch-Maschine-Interaktion geprüft. Die Realitätsnähe des Fahrsimulators wird zum einen durch die Güte der Bilderzeugung und zum anderen durch die Anzahl der Bewegungsachsen bestimmt. Durch ein leistungsfähiges Simulationsmodell und eine schnelle Ansteuerung sind komplexe Fahrmanöver möglich. Der Fahrsimulator ermöglicht es, normale Fahrer extremen Fahrsituationen auszusetzen, ohne sie zu gefährden. Dabei können dann sicherheitsrelevante Systeme im Normalbetrieb wie im Fehlerbetrieb getestet werden. Damit kann eine Einstufung der Kontrollierbarkeit einer Funktion in der Gefährdungs- und Sicherheitsanalyse validiert werden. Am FKFS [BPL10] wurde gerade ein leistungsfähiger Fahrsimulator gebaut, der als geeignete Testplattform dienen kann. Die Fragestellung der Kontrollierbarkeit von fehlerhaften Funktionen können am Fahrsimulator geklärt werden. Die Fragestellung beschäftigt sich mit der Einstufung eines Fehlers mit C2 oder C3. Für die Einstufung in C3 wird von Seitens der Norm ISO 26262 kein Nachweis gefordert [isod]. Für die Einstufung in C2 wird von Seitens der Norm ISO 26262 ein Nachweis gefordert [isod]. Der Nachweis muss der Art erfolgen, dass mindestens

3.5 Forschung am FKFS

Seit über 15 Jahren arbeitet das FKFS im Bereich Test und Diagnose von Steuergeräten. In diesem Bereich arbeitet das FKFS mit Automobilherstellern und Zulieferern zusammen. Das FKFS stellt Fahr simulatoren im Auftrag für Kunden her und entwickelt Hardware und Software für die Echtzeit-Fahrzeugsimulation. Das FKFS und die Universität Stuttgart errichteten gemeinsam den europaweit größten bewegten Fahr simulator an einer Forschungseinrichtung. Die Inbetriebnahme des Fahr simulators erfolgte im Juni 2012. Beide Bereiche haben zu einem großen Erfahrungsschatz am FKFS geführt.



Bild 3.5 Stuttgarter Fahr simulator

3.5.1 ALPAS

ALPAS [BRZ08] ist im Rahmen verschiedener Forschungsprojekte am FKFS entwickelt worden und kann je nach Erfordernis an das konkrete Projekt angepasst werden. ALPAS dient als zentrale Datenverwaltung und ermöglicht so Abbildungen zwischen verschiedenen Modellen und ausführbarem Code.

ALPAS kann die verschiedenen Datenquellen wie Busbeschreibungen, Diagnoseservicebeschreibungen und Signallisten der Testhardware einlesen und zu einer einheitlichen Datenbasis zusammenführen. Aus den Datenquellen werden neben den Signalen auch Attribute wie Zeitverhalten, Hierarchie und Codierung ausgelesen. Die Datenbasis kann dann entweder in eine Datenverwaltung wie eine XML Datei oder Datenbank überführt werden oder, wenn ein UML-Modell eingelesen wird, in ausführbaren Code umgewandelt werden. Ein angepasstes ALPAS wird auch zur Unterstützung der risikobasierten Testfallerstellung genutzt.

3.5.2 TESAM

TESAM war ein Forschungsprojekt [BRZ08], das für ein „Open-Loop“ Testsystem für Steuergeräte aus dem Komfortbereich den Einsatz einer automatisierten und syntaxbasierten Testfallerstellung entwickelt und untersucht hat. Dieser Ansatz stellt die Grundlage dieser Dissertation dar und wird weiterentwickelt. Die zu testende Funktion der Steuergeräte wurde durch ein UML-Zustandsdiagramm beschrieben. Aus diesem Modell wurden mit verschiedenen Strategien Testfälle erzeugt, die dann automatisiert auf einem „Open-Loop“ Testsystem ausgeführt werden. Die Methode führt folgende Arbeitsschritte aus, um die Testfälle zu erstellen:

- Im ersten Schritt wird die Funktion durch einen Zustandsautomaten modelliert. Dabei wird sowohl die Stimulation der Funktion wie auch die zu erwartende Reaktion der Funktion durch den Zustandsautomaten modelliert. Ein Zustandsautomat besteht aus Zuständen und Zustandsübergängen. Ein Zustandsübergang besteht aus dem Triple „Ereignis“, „Wächter“ und „Aktion“. Für die Signale und die Aktionen werden symbolische Namen verwendet.
- Im zweiten Schritt wird der Zustandsautomat aus dem UML-Editor exportiert.
- Im dritten Schritt wird der Automat eingelesen und dabei werden die Hierarchie und Nebenläufigkeit des Automaten aufgelöst.
- Im vierten Schritt wird der Automat je nach Strategie unterschiedlich durchlaufen. So werden unterschiedliche Pfade gefunden und gespeichert.

- Im fünften Schritt werden die Zustandsübergänge der gefundenen Pfade in Stimuli und Ausgaben umgewandelt.
- Im sechsten Schritt werden die symbolischen Namen aufgelöst. Die Platzhalter werden durch die konkreten Namen und Aktionen des Zieltestsystems ersetzt. Dabei wird das zeitliche Verhalten des Testsystems berücksichtigt. Eine Symboldatenbank vom Typ ALPAS stellt die Information über das Testsystem zur Verfügung. So wird ein Testskript erzeugt, das dann die so gefundenen Testsequenzen automatisch durchführt.

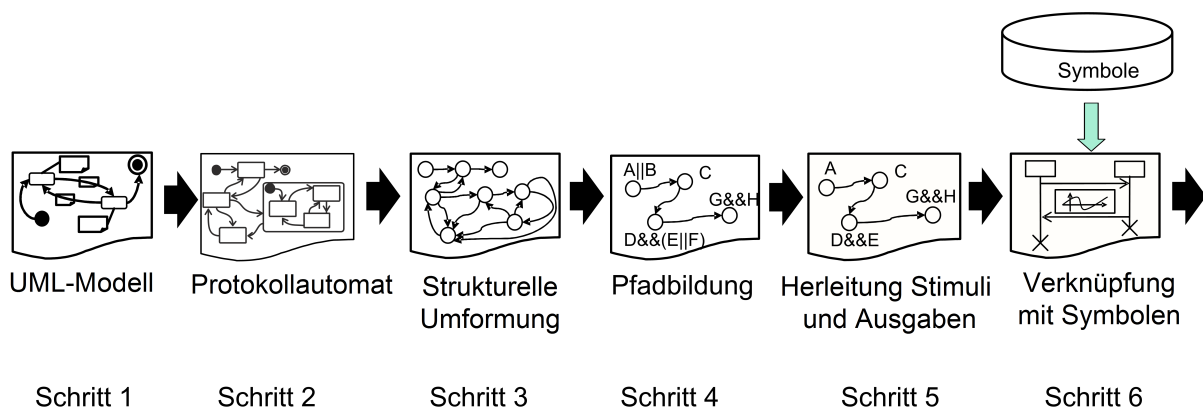


Bild 3.6 TESAM Arbeitsschritte

Die zugrundeliegende Methode von TESAM ist für ein „Open-Loop“ Testsystem mit der üblichen Aktion-Reaktion-Beziehung entwickelt worden. Nur die erstellten Aktionen der Methode stimulieren das Testsystem und die Reaktion oder Nicht-Reaktion kann einfach definiert werden. Diese Eigenschaft ist nicht allgemein gültig für „Closed-Loop“ Systeme. Durch die Rückkopplungen in der Regelschleife der komplexen Funktion ist das Systemverhalten nicht immer einfach vorherzusagen, deshalb ist es aufwändiger, eine gültige Reaktion zu definieren.

4 Die risikobasierten Testmethoden

Folgende Anforderungen werden an die risikobasierten Testmethoden gestellt: Es sollen verteilte Funktionen bzgl. der Systemsicht geprüft werden. Die Testziele sollen aus der Gefährdungsanalyse abgeleitet werden. Dazu stehen auf Systemebene der Verbund-HiL, der Fahrsimulator und der Prototyp zur Verfügung. Um die notwendigen Testumfänge leisten zu können, ist ein hoher Automatisierungsgrad notwendig. Das führt zum modellbasierten Testen (MBT). Damit der Aufwand des MBT nicht zu groß wird, soll eine Wiederverwendungsstrategie die Testdurchführung unterstützen. Aufgrund der Unterschiedlichkeit von V-HiL, Fahrzeug und Fahrsimulator als Testplattform sollte die Wiederverwendung im Bereich der Testfallanalyse stattfinden.

Eine zentrale Fragestellung in dieser Arbeit ist, ob die Testfallerstellung und Testfallanalyse durch den Einsatz von Kontextwissen in Form des Risikos zu verbessern sind. Deswegen wird zuerst eine Gefährdungsanalyse für das Fahrwerk anhand einer Fahrsituation durchgeführt. Dabei werden Risikoinformationen gesammelt. Dann werden die verschiedenen Methoden zur zustandsbasierten Testfallerstellung und der Fähigkeit der Fehlerfindung betrachtet und bewertet, ob sie mit einem risikobasierten Vorgehen zu optimieren sind. Bei der Testdatenanalyse wird der Beobachter um Risikoinformationen erweitert. An einigen Analyseverfahren werden die Verbesserungen durch den Einsatz von Risikoinformationen aufgezeigt. Die bestehenden Werkzeuge ALPAS und TESAM werden für die Verarbeitung der Risikoinformationen erweitert. Anschließend wird der Testablauf für die jeweilige Testplattform V-HiL, Fahrzeug und Fahrsimulator dargestellt.

4.1 Risiko

Die Norm ISO 26262 ist die Hauptquelle der Risikoinformationen, da sie für jedes SG und jede Funktion eine Gefährdungsanalyse vorschreibt. Somit stehen Risikoinformationen zur Verfügung, um die Teststrategie zu verbes-

sern. Die verwendbaren Informationen sind die Fahrsituationen, die Fehler und die Fehlerbewertung.

Die Gefährdungsanalyse verwendet Fahrsituationen zur Bestimmung des ASIL-Levels der Fehler, diese Fahrsituationen können auch als Testszenario verwendet werden. Diese Informationen werden dann in die Elemente eines Modells des MBT abgebildet. So ein Modell hat grundsätzlich Zustände und Zustandsübergänge, diese werden dann mit Risikoinformationen angereichert. Das Verfahren umfasst folgende Schritte:

- Definition des Systems oder des Elements der Gefährdungsanalyse.
- Durchführung der Gefährdungsanalyse für das definierte System.
- Extraktion der Risikoinformationen aus der Gefährdungsanalyse.
- Die verwendete Fahrsituation wird als Testszenario verwendet und dabei die Fahrsituation in Abschnitte aufgeteilt, welche auf Zustände des Modells abgebildet werden.
- Die Fehler und Fehlerquellen werden als Grundlage für die Wächter der Zustandsübergänge verwendet und übernehmen dabei die Kombinatorik und die Risikoeinstufung, wobei Eigenschaften der unterschiedlichen Fehlerart berücksichtigt werden.

Gerade der letzte Schritt hängt davon ab, wie die Gefährdungsanalyse durchgeführt wird. Bei Fehlern, die eine Einstufung von ASIL-B-D haben, fordert die Norm 26262 noch die Anwendung von zusätzlichen Analysen zur Absicherung. Die im vorherigen Kapitel vorgestellte FBA wird unter anderen von der Norm als geeignet vorgeschlagen [LPP07]. Die FBA untersucht, welche Ursachen ein Fehler hat. Der Fehler ist die Wurzel der Baumstruktur und die Ursachen sind die Knoten der Baumstruktur. Diese sind mit logischen Operatoren verknüpft. Aus so einem Baum wird eine logische Formel abgeleitet. Aus dieser Formel kann für den MBT die Kombinatorik der Regeln der Ereignisse bestimmt werden. Damit kann aus der Kombinatorik folgendes bestimmt werden: bei unterschiedlichen Risikoeinstufungen, die durch ein logisches „UND“ oder durch ein logisches „Äquivalenz“ verknüpft sind, wird die höhere Risikoeinstufung übernommen. Bei Risikoeinstufungen, die durch ein logisches „ODER“ oder durch ein logisches „exklusives ODER“ verknüpft sind, werden die Variablen der Formel

nach ihrem Risiko sortiert und in unterschiedliche Formeln aufgeteilt. So wird eine homogene Risikoeinstufung erreicht.

Am V-HiL kann in den dynamischen Situationen der Einfluss von Fehlern auf die Steuergeräte und die Fehlerreaktion der Steuergeräte untersucht werden. Bei der Fahrzeugerprobung wird die Fahrsituation zur Beurteilung des fahrdynamischen Verhaltens im Grenzbereich durch Testfahrer eingesetzt. Dazu treten ähnliche Fahrsituationen auch während der Dauerlauferprobung auf, bei der das System bzgl. eines unerwarteten Verhaltens überwacht wird. Am Fahr Simulator werden Fehler eingebracht, um die Reaktion von normalen Fahrern zu überprüfen, insbesondere inwieweit solche Situationen noch beherrschbar sind, denn wenn sie nicht beherrschbar sind, kann das zu Unfällen mit Personenschaden führen. Die Gefährdungsanalyse wird auf die folgenden drei Ebenen angewendet: System, Hardware und Software, dadurch können die Fehler in die vier folgenden Klassen eingeteilt werden:

System Aus technischer Sicht ist jedes SG im Fahrzeug mit seinen Sensoren und seiner Aktorik ein System. Aus funktionaler Sicht ist eine verteilte Funktion ein System. Die Funktion wird technisch durch die Aufteilung auf die SG und Erstellung der Kommunikationsmatrix¹ umgesetzt. Systemfehler umfassen alle Fehler. Aber es gibt Fehler, die erst auf der Systemebene auftreten. Diese Systemfehler können nur durch den abschließenden Test der parametrisierten Steuergeräte entdeckt werden.

Hardware Hardwarefehler können in der Produktion, durch Umwelteinflüsse, durch Verschleiß oder durch Alterung der Bauteile entstehen. Bei der Absicherung sind die sicherheitsrelevanten Hardwarefehler und deren Sicherungsmaßnahmen wie die Diagnose oder die Replikation im Fokus. Schlafende und Mehrfachfehler müssen ab der ASIL Stufe B abgesichert sein.

Software Softwarefehler sind systematisch, d.h. sie sind immer vorhanden, sobald die Software in das System geladen und parametrisiert ist. Auch wenn die Softwarefehler systematisch sind, sind nicht al-

¹Die Kommunikationsmatrix, beschreibt welche Netzwerke welche Steuergeräte als Sender und Empfänger enthält, zusätzlich werden die Botschaften definiert und dem Sender und dem Empfänger zugeordnet.

le Softwarepfade auch erreichbar. Das hat die folgenden Ursachen: Funktionen sind ausgeschaltet, Fehlerfunktionen sind nur durch den Hardwarefehler erreichbar oder entstehen durch sporadische Umwelteinflüsse. Ein Hardwarefehler, der ein Programm direkt negativ beeinflusst, ist beispielsweise eine beschädigte Speicherzelle eines Registers, die dadurch die enthaltene Variable verfälscht. Ein sporadischer Umwelteinfluss kann beispielsweise ein elektrischer Störimpuls sein, der in einer Botschaft zu einem Bitdreher führt. Es gibt statistische Untersuchungen bzgl. der Anzahl von Fehlern pro Codezeile oder Maschinencodenumfang, um abzuschätzen, wie viele Fehler zu erwarten sind. Bei Black-Box-Systemen kann die Anzahl der Fehler durch Schätzverfahren bestimmt werden wie zum Beispiel über „Functions Points“² [Lon] oder indem stochastische Methoden angewendet werden, wie sie hier beschrieben werden [Lig09].

Mensch Durch Fehlverhalten des Fahrers, Servicepersonals oder Dritten können Fehler in das System eingebracht werden. Einige so eingebrachte Fehler können durch die Sicherheitsfunktionen des Systems verhindert oder abgemildert werden. Ein kompletter Schutz ist nicht möglich. Fehlerhafte Wartung kann durch verteilte Systeme besser erkannt werden. Bei Fahrfehlern kann es keine absolute Absicherung geben. Gerade bei einer sportlichen Abstimmung des Fahrzeugs, die ein erfahrener Fahrer richtig nutzen kann, kann ein unerfahrener Fahrer überfordert sein.

Die verteilten Funktionen können nur auf Systemebene mit realen SG abschließend getestet werden. Die Hardwarefehler können nur eingeschränkt auf der Systemebene getestet werden. Primär werden die Sensorik der Steuergeräte und die Spannungsversorgung der Steuergeräte getestet. Durch das aktive Aufschalten von Hardwarefehlern kann die Softwarediagnose der Steuergeräte insbesondere die Zusammenarbeit der Steuergeräte überprüfen. Der Dauerlaufstest oder die Fahrerprobung kann durch extreme Umwelteinflüsse zusätzlich genutzt werden, um Hardwaredefekte aufzuspüren. Tritt dabei ein Hardwarefehler in einem sicherheitskritischen System auf, muss auch die Risikoanalyse überprüft werden.

²Functions Points ist eine Einheit zum Bewerten des Aufwands der Implementierung eines Softwareentwurfs [GH01].

Die Softwarefehler, die nicht im Zusammenhang mit Umwelteinflüssen oder Hardware stehen, können nur durch eine ausreichende Systematik entdeckt werden. Für Softwarefehler lassen sich aus der Systemspezifikation Testvektoren ableiten. Aufgrund dessen, dass ein vollständiger Test der Software wegen der Kombinatorik nicht möglich ist, da es für N Werte genau 2^N Kombinationen gibt, werden Testvektoren ausgewählt, die eine gute Abdeckung bzgl. der möglichen Softwarefehler sicherstellen.

4.2 Methoden zur syntaxbasierten Testfallerzeugung

Es gibt unterschiedliche Methoden zur Testfallerzeugung. Die Methoden haben unterschiedliche Fähigkeiten im Auffinden von Fehlern. Die syntaxbasierte Testfallerstellung setzt eine semiformale Systemspezifikation voraus. Zur Spezifikation wird der UML-Zustandsautomat verwendet, da dieser aufgrund der UML- Beschreibungsmittel und der weiten Werkzeugverbreitung [Bro09] besonders geeignet ist. Grundsätzlich können folgende Fehlertypen nach [Bin05] unterschieden werden:

- Ein Wächterfehler: entweder die Aktion soll ausgeführt werden, wird aber nicht ausgeführt oder die Aktion soll nicht ausgeführt werden, wird aber ausgeführt.
- Ein Zustandsübergang fehlt.
- Ein Zustandsübergang ist überzählig.
- Eine Aktion fehlt oder eine falsche wird ausgeführt.
- Eine Ereigniskette erreicht fälschlicherweise einen Akzeptanzzustand.
- Ein fehlender Zustand.
- Ein unerwartetes Ereignis.
- Ein zusätzlicher Zustand; um die Fähigkeiten zu beurteilen werden zwei Variablen definiert:
 - i stellt die Anzahl der Zustände der Implementierung dar
 - z stellt die Anzahl der Zustände des Zustandsautomaten dar

Dabei stellt ein zusätzlicher Zustand für die meisten Verfahren ein Problem dar, wenn die Anzahl der Zustände im Implementierungsmodell die Anzahl der Zustände im Referenzmodell deutlich übersteigt, denn eine Signatur³ für einen Zustand ist abhängig von der maximalen Anzahl der Zustände des zu testenden Systems. Die verschiedenen Methoden können in zwei Klassen eingeteilt werden [Bin05]. Die erste Klasse umfasst primär die strukturbasierten Methoden. Der grundlegende Ansatz der strukturbasierten Methode ist eine Traversierung der Automaten entweder durch eine Tiefensuche [CS01] oder eine Breitensuche [CS01] und das Erfüllen eines strukturbasierten Abdeckungskriteriums des Automaten. Die strukturbasierten Abdeckungskriterien sind:

- Zustände: jeder Zustand wird mindestens einmal besucht.
- Zustandsübergänge:
 - jeder Zustandsübergang wird mindestens einmal besucht.
 - jeder Zustandsübergang wird durch einen Pfad aus dem Startzustand besucht.
- Ereignisse:
 - jedes Ereignis wird mindestens einmal ausgelöst.
 - jedes gültige Ereignis wird mindestens einmal in jedem Zustand ausgelöst.
 - jedes Ereignis wird mindestens einmal in jedem Zustand ausgelöst.
- Aktionen:
 - jede Aktion wird mindestens einmal durchgeführt.
 - jede gültige Aktion wird mindestens einmal pro Zustand und Zustandsübergang durchgeführt.
- Wächter:
 - jeder Wächter wird nach wahr und falsch ausgewertet.
 - jeder Wächter wird durch „maskiert MC/DC“ [Chi01] belegt.
 - jede Belegung jedes Wächters wird vollständig abgedeckt.

³Eine Ereigniskette mit einer zugehörigen Aktionsfolge, die den Zustand eindeutig identifiziert.

Als Beispiel werden folgende Methoden ausgewählt und erläutert: „alle Zustandsübergänge“ Methode und „N+“-Methode. [Bin05] „Alle Zustandsübergänge“ ist eine Tiefensuche durch den Zustandsautomaten, die aber alle Zustandsübergänge berücksichtigt. Es werden Pfade erzeugt, so dass der Automat abgerollt ist, dadurch werden die Zustände mehrmals verwendet. Der Vorteil ist, dass die Pfade in Ebenen besser darstellbar sind. Die Methode umfasst folgende Schritte:

- Markiere alle Zustandsübergänge als nicht besucht.
- Finde einen Spannbaum⁴ und markiere alle benutzten Zustandsübergänge des Spannbaums als besucht.
- Durchlaufe den Spannbaum erneut und wähle einen Zustand aus, dann wähle wenn möglich einen neuen Zustandsübergang aus, wenn der Zustandsübergang einen Wächter besitzt, wähle eine wahre Belegung aus und markiere den Zustandsübergang als besucht.
- Das Ende ist erreicht, wenn alle Zustandsübergänge als besucht markiert sind.

Die „N+“ Methode beruht auf einer Tiefensuche durch den Zustandsautomaten. Die Zustandsübergänge, die einen Wächterausdruck besitzen, werden durch neue Zustandsübergänge ersetzt, die jeweils nur einen Minterm⁵ besitzen, die den Wächterausdruck erfüllt. Dadurch erhöht sich die Anzahl an Pfaden im Vergleich zu vorherigen Verfahren, da nur eine Ausprägung und nicht alle Mintermausprägungen gewählt werden. Die Methode umfasst folgende Schritte:

- Markiere alle Zustandsübergänge als nicht besucht.
- Finde einen Spannbaum und markiere alle benutzten Zustandsübergänge des Spannbaums als besucht.
- Durchlaufe den Spannbaum erneut und wähle einen Zustand aus, wähle wenn möglich einen neuen Zustandsübergang, wenn der Zustandsübergang einen Wächter besitzt, ersetze den gewählten Zustandsübergang durch so viele Zustandsübergänge wie es Minterme

⁴Ein Spannbaum ist ein Zustandsautomat, der alle Zustände enthält und die minimale Menge an Zustandsübergängen, um alle Zustände zu erreichen [CS01].

⁵Die Variablen sind durch ein logisches „Und“ verknüpft. Aus diesem Minterm kann eine disjunktive Normalform gebildet werden [CS01].

gibt, die zu einer wahren Belegung des Wächters führen, wähle einen der neuen Zustandsübergänge aus und markiere den gewählten Zustandsübergang als besucht.

- Das Ende ist erreicht, wenn alle Zustandsübergänge als besucht markiert sind.

Die zweite Methode ist eine Erweiterung der ersten, so dass für jeden Zustand und jeden Zustandsübergang entschieden werden kann, welches Verfahren für den jeweiligen Zustand oder den jeweiligen Zustandsübergang durchgeführt wird. Somit könnten beide Verfahren kombiniert werden, wodurch in einem Durchlauf anhand eines weiteren Kriteriums entschieden werden kann, ob nur eine Belegung oder die „N+“-Belegung verwendet werden soll. Zusätzlich kann die „N+“-Methode bei der Belegung erweitert werden.

Die zweite Klasse umfasst primär die Methoden, die eine einzigartige Signatur erzeugen, um so eine Abweichung zwischen dem Modell und der Implementierung zu finden. Dazu gehören folgende Methoden: Die „Word Sequence (W)“-Methode nach [Cho78] oder die „Unique Input Output Sequence (UIO)⁶“ nach [VCI90]. Alle Signaturmethoden haben einfache Zustandsübergänge, die UML-Zustandsübergänge müssen umgewandelt werden, in dem die Ereignisse, Attribute und Wächter in Symbole umgewandelt werden. Nach [SS97] führt dies zu einem deutlich größeren Zustandsautomaten. Die „W“-Methode hat vier Anforderungen an den Zustandsautomaten:

- Der Zustandsautomat ist vollständig spezifiziert, minimal, zusammenhängend und deterministisch.
- Der Zustandsautomat hat einen festen Startzustand.
- Der Zustandsautomat hat eine Resetfunktion, um in den Startzustand zurückzukehren, dabei wird das leere Element ausgegeben.
- Der Zustandsautomat und das zu testende System haben das gleiche Eingabealphabet.

Die „W“-Methode umfasst folgende Schritte, um einen Zustandsautomaten zu testen:

⁶Eine Signatur für einen Zustand, die mit dem Verfahren nach [VCI90] erzeugt wird.

- Schätze die maximale Anzahl(i) der Zustände des zu testenden Systems, ob das zu testende System noch zusätzliche Fehlerzustände besitzt.
- Erzeuge eine Menge von charakteristischen Eingabewörtern W für den Zustandsautomaten, indem für jedes Zustandspaar eine Eingabesequenz erzeugt wird, die die Zustände unterscheidet und in einer minimalen Menge zusammenfasst.
- Erzeuge eine deckende Menge P von Zustandsübergängen, die folgende Eigenschaften erfüllt: alle Zustände sind vom Startzustand mit einer Menge an Zustandsübergängen erreicht und alle Zustandsübergänge sind mindestens einmal ausgeführt worden.
- Die Menge Z hängt von der Beziehung zwischen i und z ab:
 - $i \leq z : Z = W$
 - $i \geq z : Z = X^{[i-z]} \times W$
- $P \times Z$ liefert die Testfälle.

Die „UIO“-Methode beruht auf dem Erzeugen von „UIO“-Sequenzen. Im Gegensatz zur „W“-Methode gibt es nicht für jeden Zustand eine „UIO“-Sequenz [SD88]. Die Länge der Sequenz ist bzgl. der Anzahl der Zustände begrenzt. Die Methode zum Auffinden funktioniert grundsätzlich so, dass nach Eingabe und Ausgabe von Paaren die Zustände in Mengen sortiert werden. Wenn die Mengen nur einen Zustand enthalten, ist schon eine „UIO“-Sequenz gefunden. Dann wird auf die Menge, die mehr als einen Zustand umfasst, die Eingabe angewendet und untersucht, ob sich die in der Menge befindlichen Zustände in ihrer Ausgabe unterscheiden. Wenn das der Fall ist, wird die Menge erneut geteilt. Dieser Schritt wird so oft wiederholt, bis die Menge nur ein Element besitzt. Für Zustände, die keine „UIO“-Sequenz haben, werden „W“-Sequenzen bestimmt.

- Finde für jeden Zustand eine „UIO“-Sequenz.
- Erzeuge einen minimalen Spannbaum P .
- Erzeuge einen Pfad für jeden Zustandsübergang, so dass gilt: entnehme aus dem Spannbaum P den Pfad vom Startzustand des Spannbauams zum Ausgangszustand des Zustandsübergangs, dann führe den

Zustandsübergang aus, um anschließend die „UIO“-Sequenz des Zielzustands auszuführen.

Die beiden Methoden unterscheiden sich primär darin, wie die Signatur eines Zustands gefunden wird. Dabei stellt die „UIO“ eine Verbesserung bzgl. der Laufzeit dar, so dass eine weitere Kombination keine Verbesserung bringt. Die „UIO“-Signaturen werden schneller gefunden und sind in der Regel kürzer als die „W“-Signaturen [Mat08]. Hier kann keine Verbesserung durch das Kontextwissen bzgl. des Risikos erreicht werden

Die oben eingeführten Fehler können sowohl die signaturbasierten Methoden als auch die „N+“-Strategie entdecken. Der Aufwand bzgl. der Laufzeit ist jedoch sehr hoch (O^3)⁷ [Bin05]. Die Eingabemenge ist der Zustandsautomat. Der Zustandsautomat unterteilt sich in die Zustände, die Zustandsübergänge und die Wächter. Grundsätzlich sind die strukturbasierten Methoden optimaler bzgl. der Laufzeit, sie haben aber Schwächen in den letzten beiden oben genannten Fehlertypen [Bin05] [Mat08]. Was aber wesentlich für die strukturbasierten Methoden spricht, ist die bessere Kombinationsfähigkeit der Methoden, um so für einen Teilautomaten eine bessere Methode einzusetzen, während die Signaturmethoden unterschiedliche Lösungen der gleichen Fragestellung sind.

Wie oben ausgeführt, lassen sich die strukturbasierten Testverfahren gut kombinieren und in Abhängigkeit eines weiteren Kriteriums gezielt anwenden. Als zusätzliches Kriterium wurde das Risiko gewählt. Da das Risiko unterschiedliche Darstellungen hat, wurde die Darstellung in Form der ASIL Einstufung gewählt. Diese kann dann auf Komponenten, Fehler und Fehlerquellen angewendet werden. Dadurch kann ein Zustandsautomat mit Risikoinformationen angereichert werden, um so kontextabhängig das Verfahren anzuwenden.

Die Norm ISO 26262 schlägt als Testmethoden für ASIL-C und ASIL-D Elemente sowohl MC/DC als auch den zustandsbasierten Test vor. Dadurch gibt es eine Vielzahl von Ansatzpunkten, um die strukturbasierten Methoden zu kombinieren. Die Kombination von Traversierungsmethoden und der Abdeckung der Wächter mit der impliziert vollständigen Abdeckung der Ereignisse und Aktionen ist von besonderem Interesse, da die Kombination der Empfehlungen der Norm bzgl. MC/DC entspricht. Die

⁷O-Notation ist die Abschätzung der Laufzeit im schlimmsten Fall bzgl. der Eingabemenge.

Anzahl der Fehlerquellen, die trotz konstruktiver Maßnahmen in der Einstufung ASIL-D verbleiben, ist kleiner als die Anzahl der Fehlerquellen, die in die ASIL-C und ASIL-B eingestuft sind. Dadurch bietet es sich an, eine bessere Testmethode als MC/DC für die ASIL-D Fehlerquellen zu wählen. Der Mehrfachbedingungsüberdeckungstest (MC) deckt durch den Test aller 2^n Ausprägungen alle Entscheidungen und alle Konditionen zu 100% ab.

Bei den Traversierungsmethoden gibt es die „N+“-Methode, in der sich die Auflösung der Wächtersaudrücke im Vergleich zu MC/DC in zwei wesentlichen Punkten unterscheidet: Der erste Punkt ist die Bildung der Teilformel, die auf Mintermen basiert, die eine echte Teilmenge der Ausprägung vom MC/DC sind. Der zweite Punkt ist, dass auch negative Auswertungen bei MC/DC erzeugt werden. Somit erzeugt die „N+“-Methode Testvektoren, die eine Teilmenge von „N+“ erzeugten Testvektoren mit MC/DC kombiniert. So wurde dann für ASIL-C „N+“ mit MC/DC kombiniert gewählt und für ASIL-B alleinig „N+“ gewählt. Für ASIL A wurden alle Zustandsübergänge gewählt, die schwächer als „N+“ sind, aber die gleichen Traversierungsmethoden wie „N+“ verwenden. Die QM sollte einmal ausgeführt werden. Die Methodenkombination wird in der folgenden Tabelle 4.1 zusammengefasst.

Tabelle 4.1 Testfallgenerierung nach Risikoeinstufung

ASIL	Kombinatorik der Wächter	Pfadbildung
QM	-	einmalig Ausführen
A	-	Alle Zustandsübergänge
B	-	N+
C	MC/DC	N+
D	MC	N+

Durch die so angepasste syntaxbasierte Testfallerzeugung wird eine bessere Abdeckung in einem Durchlauf erreicht. In der Regel ist die Testzeit beim Vorgehen nach dem V-Modell beschränkt, wodurch alle Phasen für die drei Jahre Entwicklungszeit beim Start festgelegt sind. Zwar sind auch Pufferzeiten eingeplant, die aber auch begrenzt sind und Aufgrund des

Testaufwands nicht beliebig erweiterbar sind. Deswegen gibt es für jeden Durchlauf des V-Modells für den Verbundtest einen festen Zeitrahmen. Bei der Testplanung wird dann für jedes Testobjekt, jede Fahrsituation und jede Plattform ein Zeitrahmen festgelegt. Durch risikobasierte und syntaxbasierte Testfallerzeugung kann eine bessere Ausnutzung erreicht werden und dabei eine hohe Testabdeckung von Hochrisikobereichen gewährleistet werden. Neben der Testfallerzeugung gibt es zwei Aspekte, die zu einer Überschreitung des Zeitrahmens führen können. Beide sind auf Wechselwirkungen mit dem „Closed-Loop“ zurückzuführen.

Der erste Aspekt sind die Seiteneffekte, die durch die Wechselwirkungen der verschiedenen Modelle zur Simulation, der verschiedenen Steuergeräte und dem geschlossenen Regelkreis entstehen. Diese Seiteneffekte können Fehler aufdecken. Ein Beispiel für so einen Seiteneffekt: Der Test einer Fehlererkennung in der Bremse wird durch die Fehlererkennung in der Lenkung verhindert, da die Lenkung zuerst den Fehler feststellt und durch ihre Fehlermeldung einen Notlauf bei der Bremse einleitet. Solche Seiteneffekte lassen sich entweder durch zusätzliche Zustandsübergänge und Zustände entdecken, die aber im Gegenzug die Testfallerstellung erweitern, was zu mehr Testfällen führt. Sie lassen sich aber auch durch eine Überwachung durch ein zweites Modell entdecken, das nicht in die syntaxbasierte Testfallerstellung einfließt. Dieser zweite Ansatz ist durch das Einhalten des Zeitrahmens besser geeignet und die so ausgelagerte Reaktionsanalyse und Bewertung in das zweite Modell führen zu einem vereinfachten Testmodell und zu weniger Testfällen.

Der zweite Aspekt ist die große Menge von Testfällen, die bei der syntaxbasierten Testfallerzeugung erzeugt werden, während die Analyse der Testfälle gerade auf Systemebene aufwändig ist. Dadurch können Fehler übersehen oder zu spät für den Entwicklungszyklus entdeckt werden. Deswegen muss sichergestellt werden, dass kein Fehler übersehen wird. Auch hierbei kann durch ein zweites Beobachtungsmodell eine automatisierte erste Analyse durchgeführt werden.

Die beiden Aspekte können durch eine Aufteilung in zwei Modelle besser kontrolliert und gelöst werden, indem das erste Modell die Testfälle erzeugt sich aber nicht um die Reaktionsanalyse kümmert. Das zweite Modell überwacht das System als ein Beobachter. Das zweite Modell fließt

nicht in die Testfallerzeugung ein und hat eine breite Überwachung, da alle Wächter eines aktiven Zustands das System überwachen. So ein Beobachter kann auch zur Voranalyse und Reporterstellung verwendet werden. Die Trennung führt zu einer besseren Selbstdiagnose, was wiederum zu einer besseren TCL Einstufung führt.

4.3 Methode zur Erstellung eines Beobachters

Der Beobachter sollte in einer ähnlichen Form modelliert werden wie das Ausführungsmodell. Dazu bieten sich UML Zustandsautomaten an. Damit die verteilten Sicherheitsfunktionen gut analysiert und beurteilt werden können, wird die Modellierung um vier Eigenschaften erweitert:

- Das Risiko wird als Parameterattribut für Zustände und Zustandsübergänge eingefügt.
- Die Verwendung von Referenzen wird eingefügt, um eine Abbildung auf die Anforderungen zu ermöglichen. Durch die Referenzen kann überprüft werden, welche Anforderungen durch den Beobachter überwacht werden. Wenn dann alle Referenzen aus allen Beobachtern zusammengeführt werden, kann überprüft werden, ob auch alle Anforderungen abgedeckt sind.
- Die Ereignisse sollten noch einen Zeitstempel als Attribut besitzen, um so eine bessere Verknüpfung mit einer Messaufzeichnung und dem Testreport zu erreichen.
- Die Menge der Aktionen werden um grafische Symbole erweitert, die sich auf das Ursprungsdiagramm beziehen, wobei die besuchten Zustände und Zustandsübergänge entsprechend den folgenden Definitionen eingefärbt werden:
 - unbesucht **Grau**.⁸
 - besucht **Blau**,⁹ wenn nicht eins der folgenden Kriterien erfüllt ist:

⁸RGB R:192 G:192 B:192

⁹RGB R:0 G:0 B:255

- * erwartetes Verhalten **Grün**.¹⁰
- * Fehler der Stufe QM: **Gelb**.¹¹
- * Fehler der Stufe ASIL A: **gelbes Rot**.¹²
- * Fehler der Stufe ASIL-B: **Rot**.¹³
- * Fehler der Stufe ASIL-C: **dunkles Rot**.¹⁴
- * Fehler der Stufe ASIL-D: **sehr dunkles Rot**.¹⁵

Durch das Einfärben und dadurch, dass jedes Element eines UML-Diagramms eine eindeutige Identität besitzt, ist es möglich, das Ergebnis in zwei verschiedenen Darstellungen zurückzuspiegeln. Die erste Darstellung sind die Browser, die in der Lage sind, „HTML“¹⁶ und interaktive Grafiken in „SVG“¹⁷ darzustellen. Durch die in der Grafik eingebauten Hyperlinks¹⁸ kann der gesamte Report strukturiert werden. Die zweite Darstellung umfasst die UML-Programme, die das XMI¹⁹ wieder importieren können und so das Ergebnis darstellen können. Auch hier können Dokumentteile mit dem Diagramm strukturiert werden. Die Unterscheidung zwischen besuchten und erwarteten Verhalten in Form der Zustände und Zustandsübergänge hat folgende Gründe:

- Bei der Interaktion der syntaxbasierten Testfallerstellung mit dem Fahr Simulator oder dem „V-HiL“ kann das erwartete Verhalten als Nachweis dienen, ob die Testsequenz wie vorgesehen durchgeführt wurde.
- Bei der Interaktion mit einem Fahrzeug kann der Beobachter durch das erwartete Verhalten helfen, festzustellen, ob das Fahrmanöver den Vorgaben entspricht oder ob das Fahrmanöver variiert werden muss für eine erfolgreiche Erprobung. Eine Suche nach nicht erfülltem aber

¹⁰RGB R:0 G:255 B:0

¹¹RGB R:255 G:255 B:0

¹²RGB R:255 G:66 B:0

¹³RGB R:255 G:0 B:0

¹⁴RGB R:170 G:0 B:0

¹⁵RGB R:128 G:0 B:0

¹⁶Hyper Text Markup Language

¹⁷Scalable Vector Graphics

¹⁸Zeiger mit URI basierten Adressen

¹⁹XML Metadata Interchange

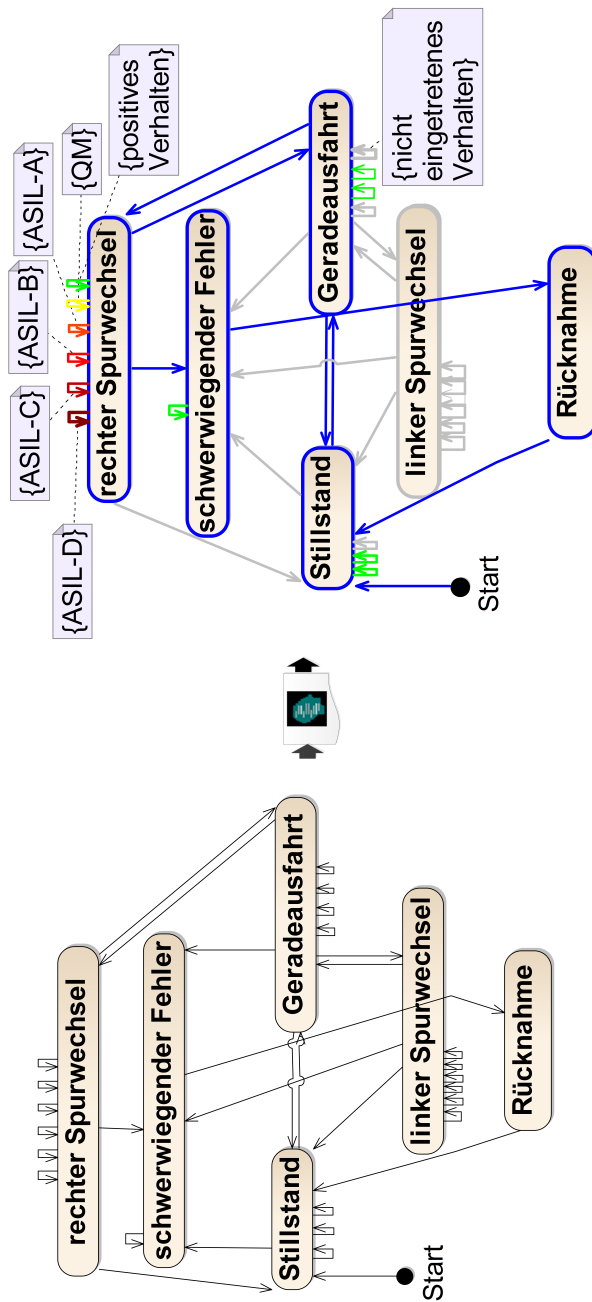


Bild 4.1 Beispiel mit eingefärbtem Ergebnismodell

erwartetem Verhalten kann in Abhängigkeit des Risikos zu weiteren Fahrerproben führen.

- Wenn beim Test keine Fehler gefunden wurden, kann die Abdeckung des erwarteten Verhaltens im Beobachter festgestellt werden. Wenn dabei Lücken oder seltene Ereignisse gefunden werden, können so weitere Testfälle abgeleitet werden.

Um den Beobachter auf Vollständigkeit zu prüfen, werden die Werte in Äquivalenzklassen²⁰ eingeteilt, diese Einteilungen werden aus den erwarteten Verhaltensregeln und aus den Fehlerregeln für jeden Zustand abgeleitet. Dadurch kann eine Lücke bzgl. der Abdeckung der Werte je Zustand aufgezeigt werden.

Wie bereits erwähnt wird Speicher heute immer billiger, wodurch es möglich ist, die Datenmenge, die während der Entwicklung eines Autos wie zum Beispiel beim Dauerlaufstest oder am V-HiL anfällt, aufzuzeichnen und zu archivieren. Die Datenarchive stehen dann verschiedenen Abteilungen zur Verfügung. Das beim Beobachter im vorhergehenden Abschnitt beschriebene Rückspiegeln des Verhaltens in formalen Strukturen kann die Ergebnisse der Datenanalyse übersichtlich darstellen. Diese Art der Klassifikation der Daten erkennt der Anwender wieder. Durch die Kombination der interaktiven Grafiken mit den Zeitstempeln der Ereignisse wird der zeitliche Verlauf durch eine Zeitleiste darstellbar. Das Testergebnis kann durch das Einblenden von Kennzahlen in das Diagramm besser bewertet werden.

Zu weiteren Datenanalysen in Abhängigkeit der Klassifikation werden zwei Methoden vorgestellt, um den Vorteil der Klassifikation durch Zustandsautomaten aufzuzeigen. Die erste Methode ist eine kontextabhängige Zeitverlaufsdarstellung. Das Problem von Zeitverlaufsdigrammen²¹ ist, dass es manuell aufwändig ist, den passenden Signalverlauf zu finden, um die gewünschte Information zu erhalten. In einigen Fällen kann die Einstellung zur passenden Darstellung im Testvorfeld festgelegt werden und ist gut zu automatisieren oder eine Liste der relevanten Signale kann angegeben werden. Gerade komplexe Darstellungen haben eine hohe Berechnungsanforderung, so dass eine Reduktion wünschenswert ist, die durch die Steuerung der Zeitverlaufsdigramme über Aktionen im Beobachter erreicht wird und das Ergebnis in den Report einbindet. Die Reihenfolge und Ordnung der Zeitverlaufsdigramme erfolgt in Abhängigkeit des Risikos. Die zweite Methode ist die Überlagerung von Messaufzeichnungen. Wenn bereits mehrere Messaufzeichnungen überprüft worden sind und keine Fehler entdeckt wurden, stellen sich zwei Fragen: wo und wie soll weiter getestet werden, um neue Fehler zu finden. Durch das Überlagern der gleichen Diagramme

²⁰Einteilung der Wertemenge anhand von Relationen in Unterklassen.

²¹Ein Zeitverlaufsdigramm ist ein Diagramm mit einer X-Achse und einer Y-Achse, wobei die Y-Achse die Werte eines Signals darstellt, während die X-Achse die Zeit darstellt [UML].

kann erkannt werden, ob das erwartete Verhalten vollständig erreicht ist. Das erwartete Verhalten wird in Form von markierten Zuständen und Zustandsübergängen festgelegt. Desweiteren kann die gesamte Verweildauer in den erwarteten Zuständen und Zustandsübergängen angezeigt werden. Durch diese Informationen können nicht besuchte Zustände mit hohem Risiko entdeckt werden. Alternativ können Bereiche gefunden werden, die ein hohes Risiko besitzen, aber eine geringe Verweildauer aufweisen.

4.4 Umsetzung

Die oben eingeführten Methoden wurden für alle drei Plattformen V-HiL, Fahrzeug und Fahrsimulator umgesetzt. Bei der Umsetzung wurde der Aspekt der „TCL“-Level berücksichtigt. Auf allen Plattformen wird der Beobachter eingesetzt. Deswegen muss bei der Umsetzung des Beobachters sichergestellt sein, dass keine kritischen Fehler übersehen werden und dass auch keine falschen Fehler angezeigt werden. Neben einer guten Softwareumsetzung sollte eine Absicherung des Beobachters sowohl durch einen Einsatz in früheren Stufen im „V-Modell“ als auch durch eine Kopplung mit syntaxbasierter Testfallerzeugung erfolgen.

Damit die Risikoinformationen in beiden Umsetzungen nachvollziehbar sind, müssen sie durch ein Werkzeug verwaltet werden. ALPAS wurde hierzu weiterentwickelt. Die Risikoinformationen werden wie folgt in die ALPAS Parameterstruktur abgebildet. Die ASIL Einstufung der Fehlerquelle wird bestimmt. Die Fehlerquelle wird dann auf Parameter oder Operationen der Parameterstruktur abgebildet.

Durch die Aufteilung auf zwei Modelle erfolgt auch eine Veränderung der ursprünglichen Ereignis-Wächter-Aktion Struktur. Bei der Testfallerzeugung werden nur die Ereignisse und die Wächter zur Beschreibung der Zustände und der Zustandsübergänge verwendet. Aus diesem reduzierten Modell wird die Eingabemenge zum Testen erzeugt. Die Aktionen werden im Analysemodell in eine Ereignis-Wächter-Aktion abgebildet. Die alten Aktionen werden so zu den Ereignissen. Auf diese Ereignisse werden Wächter angewendet. Wenn die Wächter positiv ausgewertet worden sind, werden Aktionen zur Bewertung des Systemverhaltens oder zur erweiterten Analyse ausgelöst.

Zur Analyse der Testfallerzeugung wird mit Hilfe von CANoe²² und der Möglichkeit der Kommunikationssimulation ein Analysesystem aufgebaut, das ein V-HiL System mit drei simulierten SG nachbaut. So können verschiedene Strategien der Testfallerzeugung untersucht werden. Die drei SG wurden als vereinfachte Matlab-Simulink-²³Modelle umgesetzt und als SG in CANoe compiliert. Dadurch konnten die Testsequenzen in einem „Closed-Loop“-System untersucht werden. Der Beobachter war durch die Verwendung des gleichen Werkzeugs mit integriert. Die Testausführungsprache von CANoe wurde dann als Zielsprache der Testsequenz verwendet. Der Aufbau wurde in CANoe wie in der folgenden Abbildung 4.2 dargestellt umgesetzt.

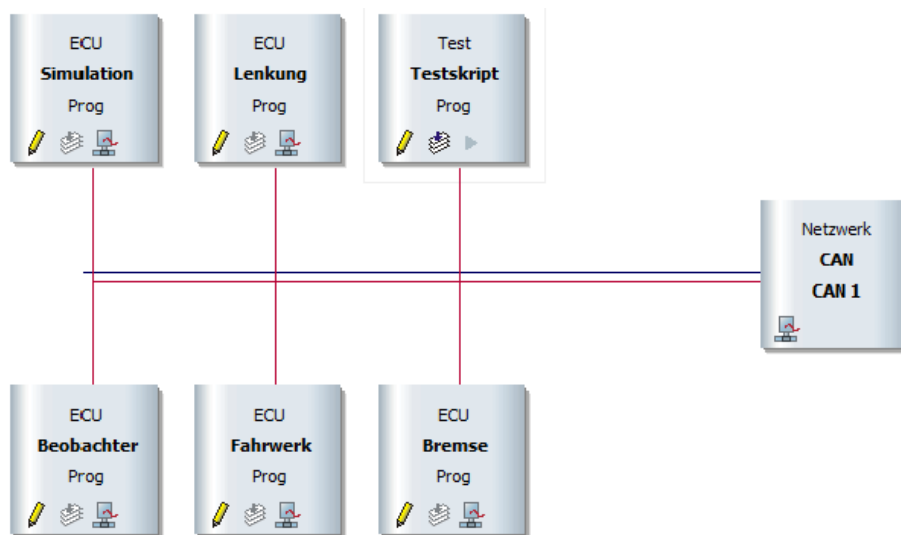


Bild 4.2 Beispiel Fahrzeug

4.4.1 Umsetzung des Beobachters

Die im vorherigen Abschnitt vorgestellten Methoden zum Modellieren von Beobachtern auf Basis der UML sollten mit allen Plattformen zusammenarbeiten. Als Schnittstelle wurde das Programm CANoe gewählt. In das Programm CANoe wurde die Visualisierung und das Reportmanagement integriert. Der Beobachter wird ebenfalls als DLL integriert. ALPAS wurde

²²Ein Softwarewerkzeug der Firma Vector Informatik zur Entwicklung von Steuergeräten und Netzwerken von Steuergeräten.

²³Ein Softwarewerkzeug zur Berechnung und zum Simulieren von mathematischen Systemen.

um eine Importmöglichkeit der Zustandsautomaten und um eine Exportmöglichkeit für die Beobachterkonfigurationsdaten erweitert. Dadurch ist auch eine Verknüpfung mit der syntaxbasierten Testfallerstellung möglich. Der Ablauf in der Werkzeugkette wird in der folgenden Abbildung 4.3 dargestellt.

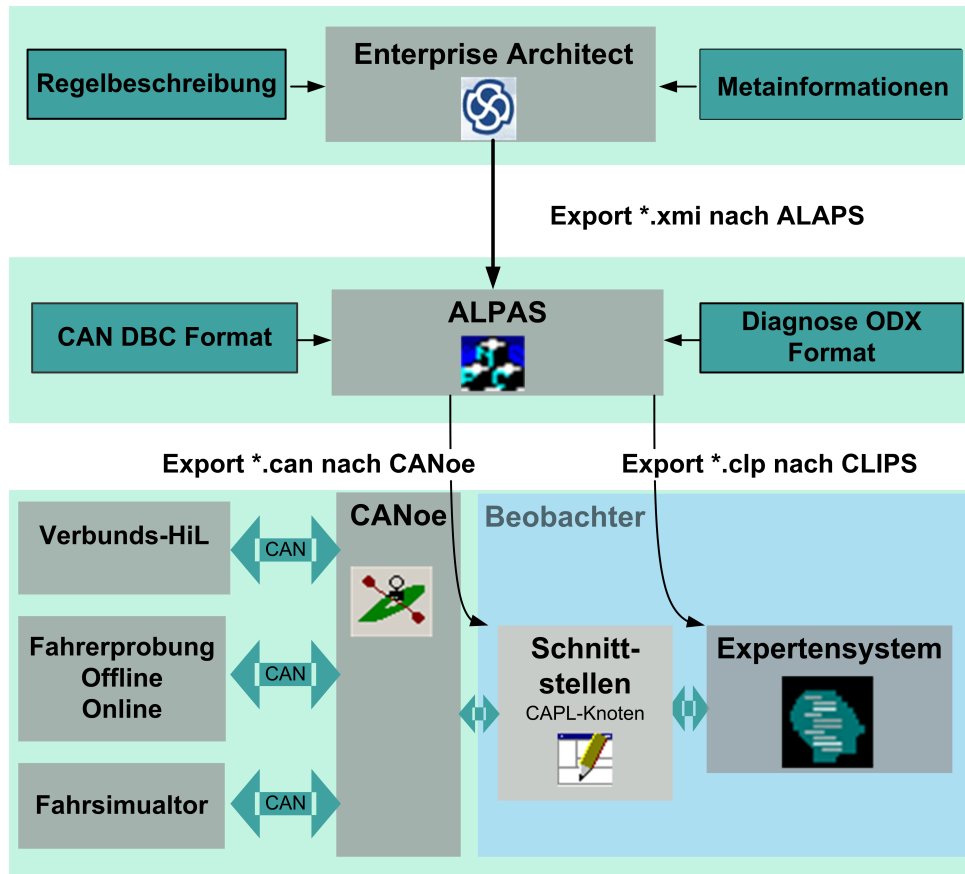


Bild 4.3 Beobachter Software Architektur

Das Expertensystem CLIPS stellt die Ausführungsplattform des Beobachters dar. Das Expertensystem ist dafür ausgelegt, in andere Systeme eingebettet zu werden und wurde in den vorgegebenen DLL Rahmen von Vector integriert. Das Expertensystem ist ein sehr reifes und gepflegtes System mit integrierten Diagnosesystem und hat deshalb ein gutes TCL von 1. Im erweiterten ALPAS wird aus dem Zustandsmodell des Beobachters eine Regelbasis und eine Schnittstelle zum Datenmanagement zwischen CANoe und CLIPS automatisch erzeugt. Die Abbildung der UML-Elemente auf die CLIPS Elemente werden in der Tabelle 4.2 dargestellt.

Tabelle 4.2 Abbildung UML nach CLIPS

UML	CLIPS Umsetzung
UML Ereignis	Das Ereignis wird auf einen CLIPS Fakt abgebildet, wobei jeder Fakt einen Zeitstempel hat
UML Wächter	Der Wächter wird auf ein Konditionselement von CLIPS abgebildet
UML Aktion	Die Aktion wird auf ein Aktionselement von CLIPS abgebildet
UML Startzustand	Der Starttoken in der Tokenliste
UML Endzustand	Die Tokenliste wird geleert
UML Zustand	Es wird durch ein Token dargestellt, wenn der Zustand aktiv ist und der Zustandstoken ein Mitglied der Tokenliste ist
UML innere Zustände eines Zustands	Sie werden auch durch Token umgesetzt, zusätzlich sind sie als innere Regel auch davon abhängig, ob der umschließende Zustand aktiv ist und in der Tokenliste enthalten ist
UML innere Zustände von orthogonalen Regionen	Für jede aktive Region wird ein Token in die Tokenliste hinzugefügt. Wird die Region verlassen, werden alle Token wieder entfernt
UML Vereinigung	In der Tokenliste wird für jeden Zustandsübergang, der hier endet, ein Token aus der Liste entfernt
UML Gabelung	In der Tokenliste wird für jeden Zustandsübergang ein Token in die Liste eingefügt
UML tiefe Historie	Ein Fakt mit einem Listenspeicher
UML einfache Historie	Einfacher Fakt

UML	CLIPS Umsetzung
UML Zustandsübergang: Ereignis-Wächter-Aktion	Sie werden als Regel abgebildet, die die Ereignisse auf die dazu passenden Fakten abbilden, die Wächter werden auf die Konditionselemente abgebildet und die Aktion wird auf die CLIPS Aktion abgebildet. Die Regel ist in Abhängigkeit des Ausgangszustands aktiv und beim Ausführen der Regel wird in der Tokenliste der Startzustandstoken durch das Endzustandstoken ersetzt
UML Entry: Ereignis-Wächter-Aktion	Werden als Regeln abgebildet, die in Abhängigkeit des erstmaligen Betretens des Zustands aktiv ausgeführt werden
UML Ausgang: Ereignis-Wächter-Aktion	Werden als Regeln abgebildet, die in Abhängigkeit des Verlassens des Zustands ausgeführt werden

In der Report- und Ergebnisdarstellung wird noch die jeweilige Plattform unterschieden, die jeweils unterschiedliche Informationen darstellt. Die gefundenen Fehler werden in Abhängigkeit der Risikoeinstufung eingefärbt.

4.4.2 Umsetzung der syntaxbasierten Testfallerstellung

Die Umsetzung der risikobasierten und syntaxbasierten Testfallerstellung erfordert eine Anpassung der beiden Softwarewerkzeuge TESAM und ALPAS. Die allgemeinen Anpassungen werden im folgenden Abschnitt erläutert, während die spezifische Plattformanpassung im folgenden Kapitel erläutert wird. ALPAS sollte das Risiko verwalten und die Beziehung zwischen den Regeln des Beobachters und den TESAM Testfällen ermöglichen. Dazu werden die Parameter um ein Attribut erweitert, das die Risikoeinstufung anzeigt. Die Parameter werden in einer Datenstruktur in ALPAS verwaltet. Die Ereignisse und Wächterbestandteile eines Zustandsautomats verweisen auf Parameter, die in der jeweiligen Plattform manipulierbar

sind. Damit vereinfachte und verkürzte Namen möglich sind, werden symbolische Namen verwendet. Die symbolischen Namen sind beispielsweise der Lenkwinkel oder FA1F1. Die beiden symbolischen Namen stellen folgende reale Parameter des Testsystems dar:

- Der Lenkwinkel stellt das CAN Signal Lenkwinkel der Nachricht Lenk-01 vom Lenkungs-SG des Fahrwerksbusses aus der CAN-DBC²⁴ dar mit einem Wertebereich und einer Auflösung wie auch einem festen Sendezyklus.
- FAF1 stellt die Operation dar, die die Stromversorgung des Lenkungs-SG kurzschließt.

Die Beziehung zwischen den symbolischen Namen und den Plattformparametern werden in ALPAS verwaltet. Im sechsten Schritt der Testfallerzeugung von TESAM, wie im Bild 3.6 dargestellt ist, wird eine Rückkopplung eingebaut, da hier die symbolischen Namen aufgelöst werden. So können die Testschritte in Beziehung zu den Parametern gesetzt werden, die in den Testschritten enthalten sind. Über den Risikoattributen des Parameters kann der Testschritt mit den Anforderungen in Beziehung gesetzt werden. Über die Parameter kann auch die Beziehung zwischen Beobachterregeln und den Testschritten bestimmt werden.

TESAM muss so erweitert werden, dass TESAM automatisch das Risiko eines Wächters bestimmen kann und danach die dem Risiko zugeordnete Methode anwendet. Das Risiko wird über den symbolischen Namen an ALPAS angefragt und ALPAS liefert die ASIL Einstufung zurück. Wenn ein Wächter verschiedene ASIL Einstufungen ermittelt hat, wird mit Hilfe der Kombinatorik eine einheitliche Risikoeinstufung bestimmt. Die Erweiterungen sind gekapselt. Zum einen gibt es zusätzliche syntaxbasierte Testfallerstellungen, die automatisch die passende Methode anhand des Risikos selektieren und zum anderen gibt es zwei Ausgabeprofile für die beiden Plattformen Verbunds-HiL und Fahrsimulator.

Die wesentlichen Schritte der syntaxbasierten Testfallerstellung sind die beiden folgenden Schritte: der vierte Schritt von TESAM und der fünfte Schritt von TESAM wie im Bild 4.4 dargestellt. Im vierten Schritt werden die Pfade erstellt. Im Gegensatz zur ursprünglichen Implementie-

²⁴CANdb-Datenbank-Datei, die eine Vector-Kanaldefinition für CAN-Bus beschreibt.

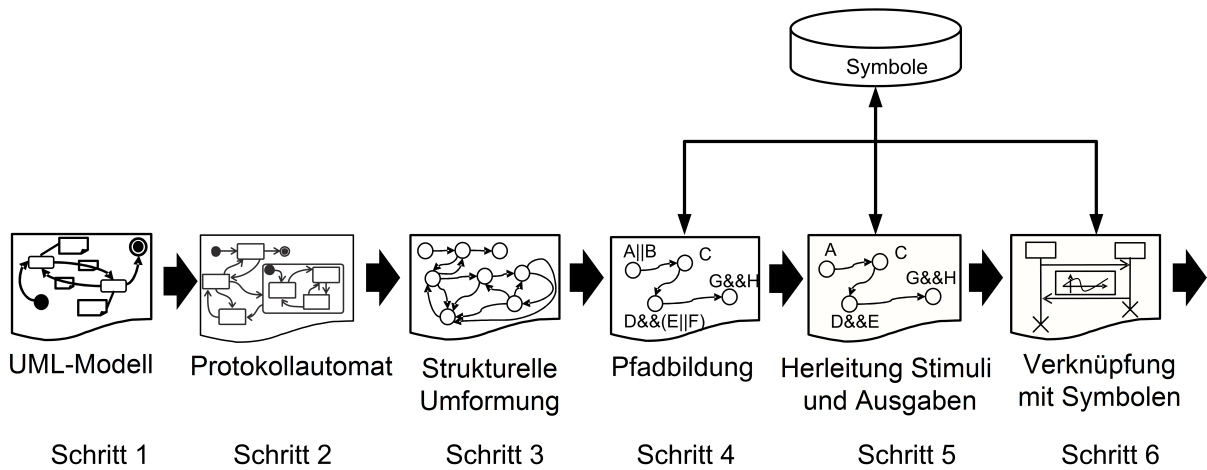


Bild 4.4 Erweiterte Arbeitsschritte

ung werden mehrere Methoden in einer Iteration verwendet. Im fünften Schritt beim Erzeugen der Eingabevektoren und Ausgabevektoren werden verschiedene Methoden angewendet.

4.4.3 Umsetzung Fahrerprobung

Es wird nur ein Beobachter verwendet, der in der Lage ist, mehrere Modelle gleichzeitig zu unterstützen und dadurch ein weites Spektrum an Fahrerprobungen überwachen soll. Ein Einsatz von 24 Stunden an 7 Tagen in der Woche ist das Ziel. Der Beobachter soll mit der bestehenden Werkzeugkette zusammenarbeiten. Neben der aktiven Messung während einer Testfahrt werden auch gesammelte Daten analysiert. Deswegen wird nach einer Messung ein Report und eine schnellere visuelle Rückmeldung innerhalb von CANoe ausgegeben. In ALPAS wurde ein neues Attribut für die Parameter eingeführt, das einen Text darstellt. Damit kann in der Oberfläche ein Text angezeigt werden, wenn keine Fehler gefunden wurden, um dem Testfahrer Hilfestellungen zum besseren Testen zu geben.

Der Ablauf ist wie in der Abbildung 4.5 dargestellt und umfasst zusätzlich nur die Erzeugung des Beobachters für CANoe. Der Beobachter ist in einer CAPL-DLL eingebettet und die Auflösung der symbolischen Namen wird über einen automatisch erzeugten CAPL-Knoten erzeugt. Die Risikoinformationen werden dem Beobachter für die Testfallergebnisanalyse und

Fahrerprobung

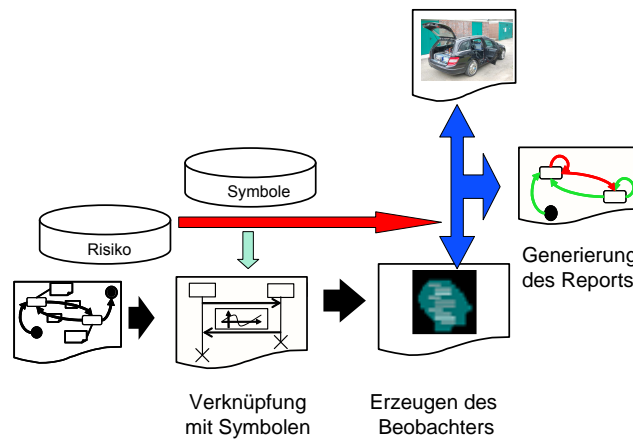


Bild 4.5 Beispiel Fahrzeug

Testfallergbnisdarstellung bereit gestellt, so dass die gefundenen Fehler anhand des Risikos eingefärbt werden.

4.4.4 Umsetzung am HiL

Der Verbund-HiL umfasst alle relevanten Steuergeräte des Fahrwerksverbunds, also die Lenkung, die Bremse und das aktive Fahrwerk und ist in der Lage, das Beispielfahrmanöver mit einem Einspurfahrermodell [Ise06] zu durchfahren. Die CAN Kommunikation eines SG kann gezielt manipuliert werden. In dieser Umsetzung gibt es eine Fehlerschiene zur Manipulation der Leitungen. Folgende Operationen können durchgeführt werden: Kurzschluss, Kabelbruch, Überspannung und Unterspannung. Für die Steuergeräte wurde jeweils ein Datenbankeintrag angelegt und für die Parameter jeweils ein Eintrag in die Parameterstruktur mit jeweils symbolischen und absoluten Namen angelegt. Da sich die Ablaufsteuerung für „Closed-Loop“-Systeme mit einem Testszenario von „Open-Loop“-Systemen unterscheidet, muss die Testfallerzeugung angepasst werden. Das Testszenario vereinfacht die Parametrierung, da die Simulationen vieler Einstellungen schon festgelegt sind. Aber bei jedem Test muss beim Start die Fehlerfreiheit geprüft werden. Beide Vorgänge können in Routinen zusammengefasst werden.

Bei der Manipulation muss der richtige Zeitpunkt abgewartet werden. Jeder Pfad stellt einen Testdurchlauf des Testszenarios dar. Bei einer Fehlfunktion des Testablaufs kann es zu einer Abweichung der erwarteten Testlaufzeit kommen. Diese muss erkannt und abgefangen werden um den Gesamtestdurchlauf nicht zu gefährden. Wie das Bild 4.6 zeigt werden zwei Modelle verwendet: das erste, um die Testablaufsteuerungen zu erstellen, und das zweite zur Überwachung und zur Reporterstellung.

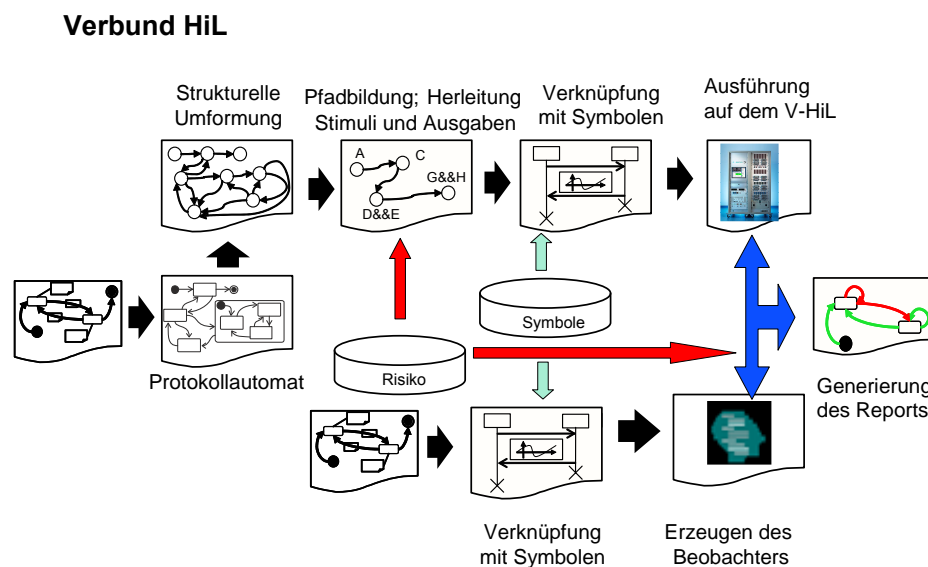


Bild 4.6 Beispiel Verbund HiL

4.4.5 Umsetzung am Fahr Simulator

Der Fahr Simulator stellt eine besondere Testplattform dar. Die Fahrer haben einen großen Freiheitsgrad in der Interaktion mit der Simulation. Zudem sollte ein heterogenes Fahrer Kollektiv zum Einsatz kommen, das ein breites Spektrum an Fahrern widerspiegelt. Hierdurch wird die Testzeit begrenzt. Deswegen kann nur eine sehr begrenzte Anzahl an Testdurchläufen durchgeführt werden. Es steht eine eingeschränkte Manipulation der Steuergeräte im Vergleich zu einem Verbund-HiL zur Verfügung. Aber auch hier kann die CAN Kommunikation eines SG manipuliert und die simulierten Sensordaten verändert werden. Die Testdurchführung dauert

deutlich länger als am V-HiL und deshalb sollten nur Testfälle einer hohen Risikoeinstufung ausgewählt werden. Der Ablauf sieht wie im Bild 4.7 dargestellt aus. Die Testfälle werden unter der Berücksichtigung des Risikos erzeugt und nach den oben genannten Kriterien aussortiert. Der Beobachter wird erzeugt und dem Beobachter werden die Risikoinformationen zur Verfügung gestellt.

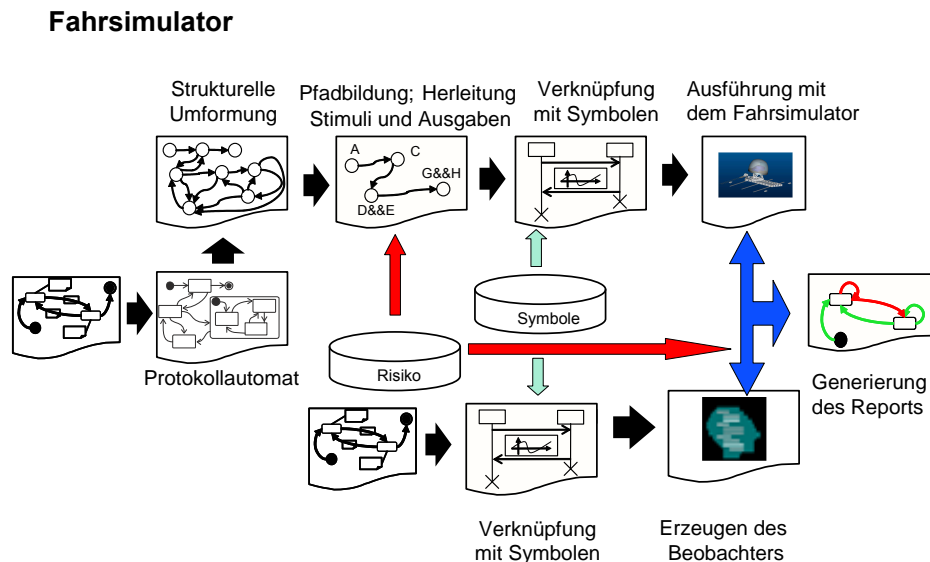


Bild 4.7 Beispiel Fahrsimulator

4.4.6 Risikoanalyse

Das System, das für die praktische Umsetzung gewählt ist, ist der Fahrwerksverbund von Steuergeräten bestehend aus der Bremse, der Lenkung und dem aktiven Fahrwerk. Jedes Steuergerät wird als ein Black-Box-System betrachtet, während die Kommunikationsschnittstelle bekannt ist. Daher sind die verteilten Funktionen ein Grey-Box-System. Somit gibt es keine vollständige bekannte Gefährdungsanalyse für die Hardware und Software der drei Steuergeräte, sondern es werden externe Fehlerquellen und einige allgemeine interne Fehlerquellen betrachtet. Für die praktische Umsetzung erfolgt die Risikoanalyse anhand der bewährten Fahrsituation

des zweimaligen Fahrbahnwechsels nach VDA²⁵ [Bau03]. Diese Fahrsituation erfordert das Zusammenspiel der drei Steuergeräte insbesondere der verteilten Funktionen. Der zweimalige Fahrbahnwechsel kann in sechs Phasen aufgeteilt werden:

- Das Fahrzeug beschleunigt auf die Testgeschwindigkeit.
- Das Fahrzeug fährt mit konstanter Geschwindigkeit 12 Meter geradeaus.
- Das Fahrzeug wechselt die Fahrspur.
- Das Fahrzeug fährt 12 Meter geradeaus.
- Das Fahrzeug wechselt auf die ursprüngliche Fahrspur zurück.
- Das Fahrzeug fährt 12 Meter geradeaus.

Aus diesen sechs Phasen lassen sich die drei folgenden Fahrsituationen ableiten, die als Grundlage für eine Risikobewertung dienen:

- Das Fahrzeug beschleunigt auf die Testgeschwindigkeit.
- Das Fahrzeug fährt mit konstanter Geschwindigkeit geradeaus.
- Das Fahrzeug wechselt die Fahrspur.

Jede der drei Fahrsituationen hat eine unterschiedliche Gefährdungsklasseneinstufung. So haben Witterungseinflüsse in Kombination mit Ausfall und Teilausfall der Steuergeräte unterschiedliche Einstufungen bzgl. der Kontrollierbarkeit von Seiten des Fahrers zur Folge. Zum Beispiel ist ein Ausfall der Lenkunterstützung beim Spurwechsel kritischer als während der Beschleunigungsphase. Aus Gründen der Übersichtlichkeit werden in diesem Abschnitt nur die Ergebnisse der Risikobewertung einer Phase dargestellt. Es wird die Teilfahrsituation Spurwechsel gewählt, die die kritischste ist und die höchsten Anforderungen an die Steuergeräte stellt. In der Fahrsituation müssen die relevanten Fehler gefunden und betrachtet werden.

Die Tabelle 4.3 bezieht sich auf die Teilfahrsituation Spurwechsel. Der Fahrer lenkt das Fahrzeug, so dass ein Spurwechsel bei gleichbleibend hoher Geschwindigkeit erfolgt. Der Fahrer wird von der Lenkung bei Aufbringen des Lenkmoments unterstützt und das Fahrwerk beeinflusst den Wankwin-

²⁵Verband der Automobilindustrie

kel, so dass ein sicheres und angenehmes Kurvenverhalten sichergestellt ist.

Gerade beim Untersteuern oder Übersteuern kann durch das richtige Zusammenspiel der Steuergeräte das Fahrzeug stabilisiert, kleine Fahrfehler und negative Umweltbedingungen kompensiert werden. Damit der hier umgesetzte Zustandsautomat (siehe Darstellung 4.8) übersichtlicher ist, wird für jeden Fehler ein Label zugeordnet. Die Einstufungen der Fehler sind aus den Vorgaben der Norm [isod] übernommen und aus den Einstufungen von Low [LPP07] abgeleitet. Der Beobachter des Bremse-SG wurde [Bau03] entnommen. Die Lenkung und das aktive Fahrwerk wurde [Ise06] entnommen.

Die Bremse überwacht Fahrverhalten und greift unterstützend ein, wenn eine kritische Untersteuerung oder eine kritische Übersteuerung des Fahrzeugs auftritt. Wenn eine solche Situation eintritt, gibt es sich gegenseitig beeinflussende Kräfte, die auf das Fahrwerk über den Wankwinkel und die Bremse auf die Räder und auf das Fahrzeug einwirken. Dies führt zur folgenden Tabelle 4.3. Zu jedem Fehler gibt es noch eine zweite Tabelle, die die Fehlerquelle der FBA detailliert darstellt und den dazugehörigen Baum darstellt.

Tabelle 4.3 Fehlertabelle

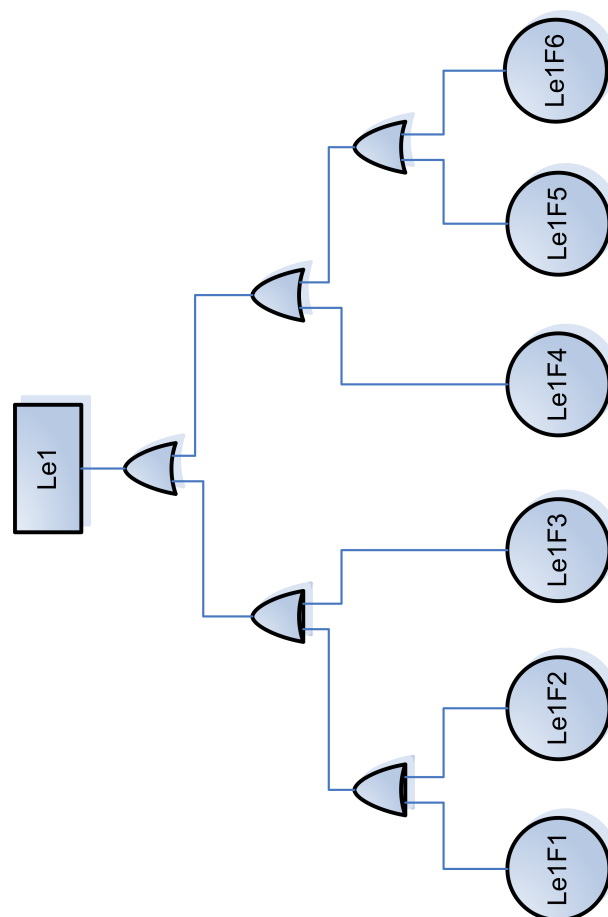
Fehler	Lenkung	Bremse	Fahrwerk	Label	ASIL
Ausfall der Lenkunterstützung	-	ESP Eingriffe, um den Fahrer bei der Lenkung zu unterstützen	-	Le1 siehe Tabelle 4.4	D
unkontrollierte Lenkbewegung	Fehler-speicher-eintrag	-		Le2 siehe Tabelle 4.5	D
ESP Ausfall während eines Eingriffs	Lenkunterstützung	-		Fa1 siehe Tabelle 4.6	D
ESP Ausfall vor einem Eingriff	Lenkunterstützung	-	-	Fa2 siehe Tabelle 4.7	C
fehlerhafter ESP Eingriff	Lenkunterstützung	-	-	Fa3 siehe Tabelle 4.8	D
Kein Aussetzen der Wankunterstützung während eines ESP Eingriffs	Lenkunterstützung	ESP nicht ausreichend	-	Fa4 siehe Tabelle 4.9	C
Ausfall der Wankunterstützung während eines ESP Eingriffs	Lenkunterstützung	ESP ausreichend	-	Fa5 siehe Tabelle 4.10	A

Um das ursprüngliche ASIL Level [isod] zu Kennzeichnen wird dieses in einer Klammer dazugeschrieben. Jeder der Fehler kann mehrere Fehlerquellen besitzen.

Tabelle 4.4 Lenkunterstützungsausfall

Fehler	Fehlerur- sache	Maßnahmen zur Risikoreduktion	Label	ASIL
Le1	Ausfall des E-Motors durch einen Defekt	Fehlermeldung an den Fahrer, Überlastungsschutz und Verwendung eines langlaufenden Motors	Le1F1	(D) C
Le1	Über- spannung im Bordnetz	Fehlermeldung an den Fahrer und Überlastungsschutz	Le1F2	(D) C
Le1	Unter- spannung im Bordnetz	Fehlermeldung an den Fahrer und Rückmeldung über abnehmende Lenkunterstützung	Le1F3	(D) C
Le1	fehlerhafte Software	Fehlermeldung an den Fahrer und Rückmeldung über abnehmende Lenkunterstützung	Le1F3	(D) C
Le1	CRC Fehler in der Botschaft des Lenk- winkel- sensors	Fehlermeldung an den Fahrer und Rückmeldung über abnehmende Lenkunterstützung	Le1F4	(D) B

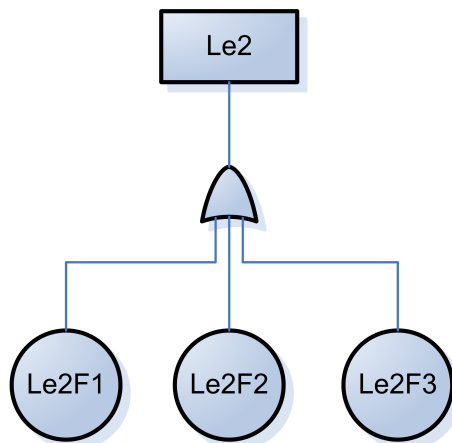
Fehler	Fehlerur- sache	Maßnahmen zur Risikoreduktion	Label	ASIL
Le1	CRC Fehler in der Botschaft der Bremse	Fehlermeldung an den Fahrer und Rückmeldung über abnehmende Lenkunterstützung	Le1F5	(D) B
Le1	fehlerhafte Software	Fehlermeldung an den Fahrer und Rückmeldung über abnehmende Lenkunterstützung	Le1F6	(D) C



Kombinatorik der Fehler nach FBA: $((Le1F1 \text{ XOR } Le1F2 \text{ XOR } Le1F3) \text{ OR } (Le1F4 \text{ OR } Le1F5 \text{ OR } Le1F6))$

Tabelle 4.5 Unkontrollierte Lenkbewegung

Fehler	Fehlerursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Le2	fehlerhafte Software: durch einen Softwarefehler wird ein überhöhtes Lenkmoment eingebracht	-	Le2F1	D
Le2	Lenkwinkelsensorfehler durch Alterung wird ein falscher Lenkwinkel gemessen	Sensorprüfung gegen Lenkmomente	Le2F2	(D) C
Le2	Lenkwinkelbotschaftfehler: durch EMV kommt es zu einem Bitdreher	überwachen der eigenen Botschaft	Le2F3	(D) C



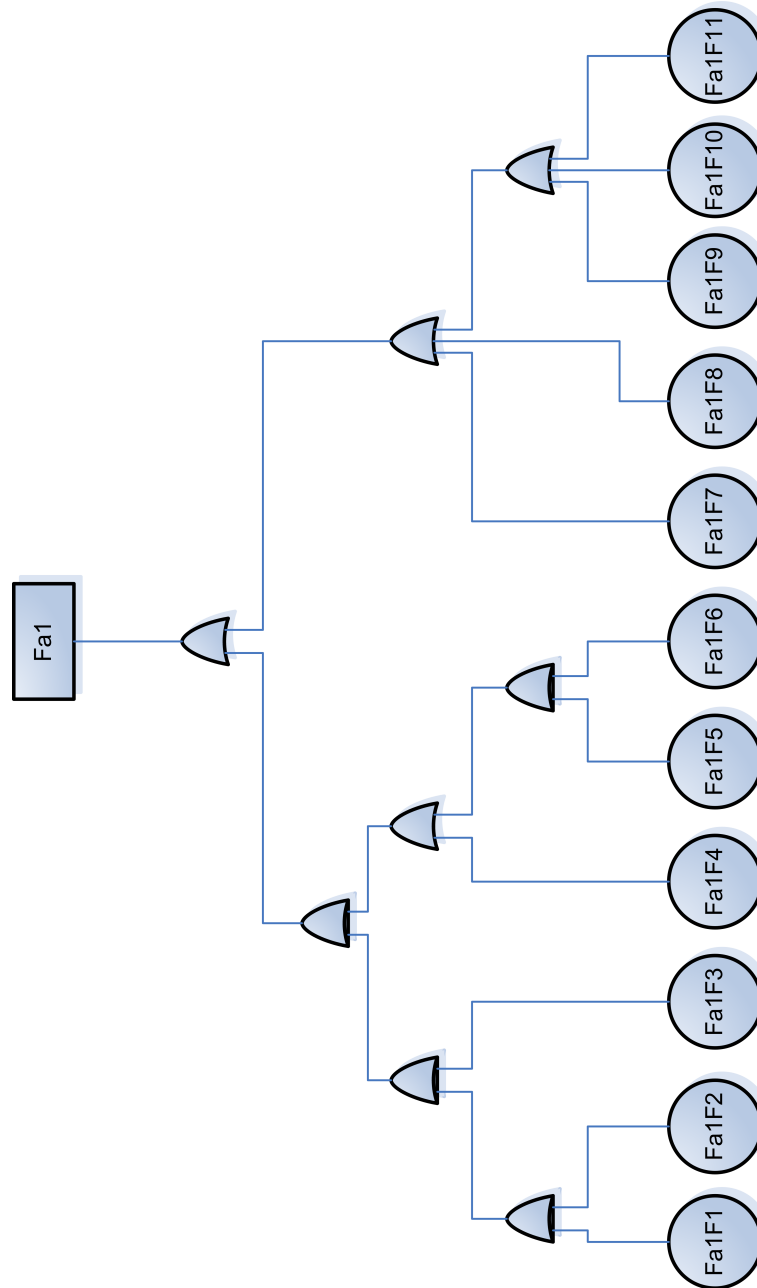
Kombinatorik der Fehler nach FBA: (Le2F1 OR Le2F2 OR Le2F3)

Tabelle 4.6 ESP-Ausfall während einer Regelung

Fehler	Fehlerursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Fa1	Kurzschluss	Hardwareschutz	Fa1F1	(D) C
Fa1	Überspannung	Hardwareschutz, abgestuftes Abschalten von Funktionen und Rückmeldung an den Fahrer	Fa1F2	(D) C
Fa1	Unterspannung	abgestuftes Abschalten von Funktionen und Rückmeldung an den Fahrer	Fa1F3	(D) C
Fa1	Gierrate-sensorfehler	der Beobachter prüft den Sensorwert auf Plausibilität und bei einem Fehler wird mit geschätzten Werten die ESP-Regelung zu Ende geregelt	Fa1F4	(D) C
Fa1	Ausfall eines Raddrehzahl-sensors	der Beobachter, der aus den drei anderen Werten den ausgefallenen Wert berechnet, regelt weiter	Fa1F5	(D) C

Fehler	Fehlerursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Fa1	Ausfall von zwei bis vier Raddrehzahlsensoren	der Beobachter prüft den Sensorwert auf Plausibilität, wenn er einen Fehler feststellt, wird mit geschätzten Werten die ESP-Regelung sicher beendet	Fa1F6	(D) B
Fa1	Ausfall von einem Rad-drucksensor	der Beobachter prüft den Sensorwert auf Plausibilität, wenn er einen Fehler feststellt, wird mit geschätzten Werten zu Ende geregelt	Fa1F7	(D) B
Fa1	Fehler in der Lenkwinkel Botschaft	CRC über die Botschaft und einen Zähler, bei einmaligem Fehler rechnet der Beobachter mit geschätzten Werten weiter, bei mehrmaligen Fehlern wird zu Ende geregelt	Fa1F8	(D) C
Fa1	Fehler im Lenkwinkelsensor	der Beobachter prüft den Sensorwert auf Plausibilität, bei einem Fehler beendet er die Regelung	Fa1F9	(D) C
Fa1	Softwarefehler	der Beobachter prüft und regelt zu Ende	Fa1F10	(D) C

Fehler	Fehlerursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Fa1	Parameterfehler	der Beobachter prüft und regelt zu Ende -	Fa1F11	(D) C



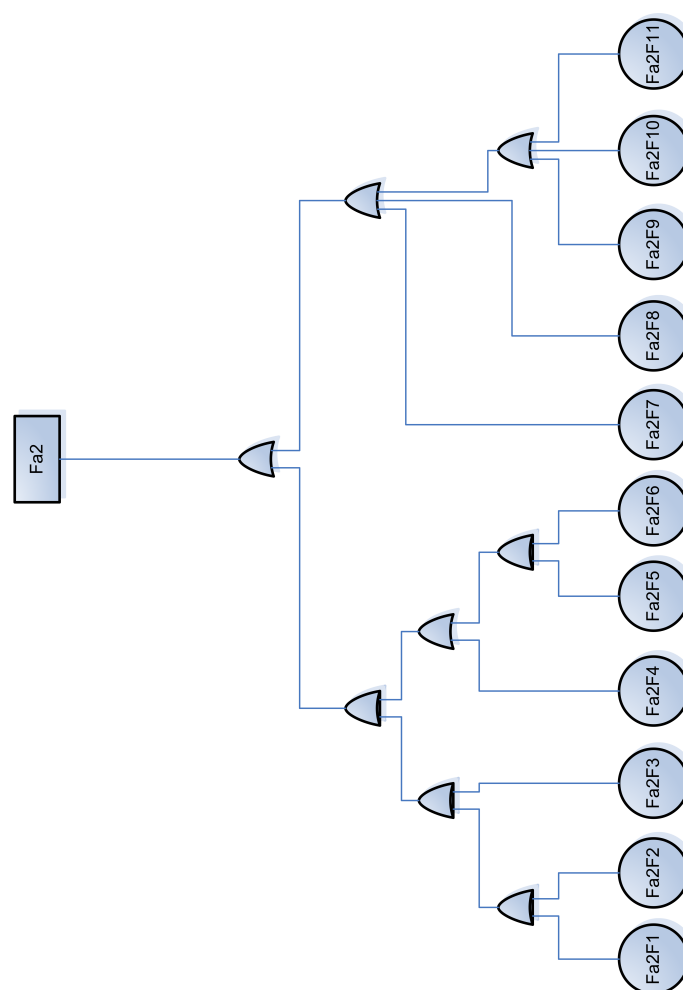
Kombinatorik der Fehler nach FBA: $((Fa1F1 \text{ XOR } Fa1F2 \text{ XOR } Fa1F3 \text{ XOR } (Fa1F4 \text{ OR } (Fa1F5 \text{ XOR } Fa1F6))) \text{ OR } Fa1F7 \text{ OR } Fa1F8 \text{ OR } Fa1F9 \text{ OR } Fa1F10 \text{ OR } Fa1F11)$

Tabelle 4.7 ESP-Ausfall vor einer Regelung

Fehler	Fehler- ursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Fa2	Kurz- schluss	ESP Ausfall und Rückmeldung an den Fahrer	Fa2F1	(C) B
Fa2	Über- spann- ung	ESP Ausfall und Rückmeldung an den Fahrer	Fa2F2	(C) B
Fa2	Unter- spann- ung	ESP Ausfall und Rückmeldung an den Fahrer	Fa2F3	(C) B
Fa2	Gierrate- sensor- fehler	der Beobachter prüft den Sensor und beim Entdecken eines Fehlers kommt es zu einem ESP-Ausfall und Rückmeldung an den Fahrer	Fa2F4	(C) B
Fa2	Ausfall eines Rad- dreh- zahl- sensors	der Beobachter prüft die Sensoren und beim Entdecken eines Fehlers wird ein Mittelwert gebildet und weiter geregelt	Fa2F5	(C) B
Fa2	Ausfall von zwei bis vier Rad- dreh- zahl- sensoren	der Beobachter prüft die Sensoren und beim Entdecken eines Fehlers von mehr als einem Sensor kommt es zu einem ESP Ausfall und einer Rückmeldung an den Fahrer	Fa2F6	(C) A

Fehler	Fehler- ursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Fa2	Ausfall von einem bis vier Rad- druck- sensoren	der Beobachter prüft den Sensor und beim Entdecken eines Fehlers kommt es zu einem ESP-Ausfall und Rückmeldung an den Fahrer	Fa2F7	(C) B
Fa2	Fehler in der Lenk- winkel- Bot- schaft	CRC über die Botschaft und einen Zähler	Fa2F8	(C) B
Fa2	Fehler im Lenk- winkel- sensor	der Beobachter prüft den Sensor und beim Entdecken eines Fehlers kommt es zu einem ESP-Ausfall und Rückmeldung an den Fahrer	Fa2F9	(C) B
Fa2	Soft- ware- fehler	der Beobachter prüft den Sensor und beim Entdecken eines Fehlers kommt es zu einem ESP-Ausfall und Rückmeldung an den Fahrer	Fa2F10	(C) B

Fehler	Fehler-ursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Fa2	Parameterfehler	der Beobachter prüft den Sensor und beim Entdecken eines Fehlers kommt es zu einem ESP-Ausfall und Rückmeldung an den Fahrer	Fa2F11	(C) B



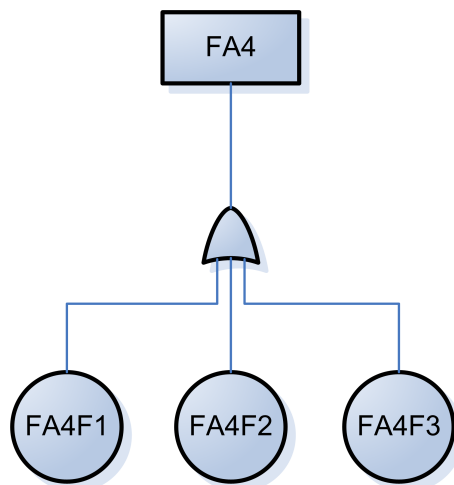
Kombinatorik der Fehler nach FBA: ((F1 XOR F2 XOR Fa2F3 XOR (Fa2F4 OR (Fa2F5 XOR Fa2F6))) OR Fa2F7 OR Fa2F8 OR Fa2F9 OR Fa2F10 OR Fa2F11)

Tabelle 4.8 Fehlerhafte ESP-Regelung

Fehler	Fehlerursache	Maßnahmen zur Risikoreduktion	Label	ASIL
Fa3	Softwarefehler	Codereview und Softwaretests	Fa3F1	D
Fa3	Parameterfehler	Parametertests	Fa3F2	D

Tabelle 4.9 Fehlerhaftes Wanken

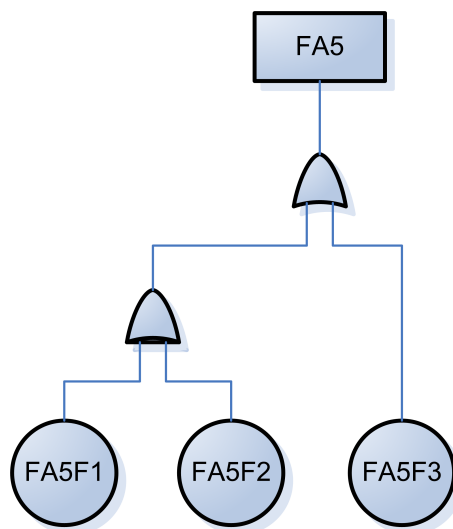
Fehler	Fehlerursache	Maßnahmen zur Risikoreduktion	Label	ASIL
FA4	Softwarefehler: das ESP Signal wird nicht richtig verarbeitet	Test der Reaktion in verschiedenen Extremsituationen	FA4F1	(D) C
FA4	Parameterfehler: es wird eine zu starke Wankkraft in das System eingebracht	Test und Überwachen der Wankkräfte	FA4F2	(D) C
FA4	Sensorfehler: es wird eine zu starke Wankkraft in das System eingebracht	Erkennung des Sensorfehlers durch ein internes Modell	FA4F3	(D) C



Kombinatorik der Fehler nach FBA: FA4F1 OR FA4F2 OR FA4F3

Tabelle 4.10 Ausfall Wankunterstützung

Fehler	Fehlerursache	Maßnahmen zur Risikoreduktion	Label	ASIL
FA5	Leck in der Hydraulik	Drucksensor und Berechnung des Drucks über ein Modell	FA5F1	(A) A
FA5	fehlerhafter Drucksensor	Testroutine bei Einschalten des Drucksensors	FA5F2	(A) A
FA5	Fehler in der ESP-Botschaft	CRC über die Botschaft und einen Zähler	FA5F3	(A) A



Kombinatorik der Fehler nach FBA: $((FA5F1 \text{ XOR } FA5F2) \text{ OR } FA5F3)$

Mit diesen Fehlern kann ein allgemeines Modell mit Risikoinformationen verknüpft werden, welches dann für die folgenden Methoden geeignet ist. Wenn die Formel mit Hilfe der Kombinatorik und bzgl. der Risikoeinstufung geordnet umgeformt ist, gibt es fünf Formeln für jeden Zustand nach

Risiko sortiert, die Elemente der fünf Formeln für den Zustand des „Spurwechsels“ sind in der folgenden Tabelle 4.11 dargestellt.

Tabelle 4.11 Nach ASIL geordnete Fehler

QM	ASIL-A	ASIL-B	ASIL-C	ASIL-D
Fa2F6	Fa2F1, Fa2F2, Fa2F3, Fa2F4, Fa2F5, Fa2F7, Fa2F8, Fa2F9, Fa2F10, Fa2F11, FA5F1, FA5F2, FA5F3	Le1F4, Le1F5, Fa1F6, Fa1F7	Le1F1, Le1F2, Le1F3, Le1F6, Le2F2, Le2F3, Fa1F1, Fa1F2, Fa1F3, Fa1F4, Fa1F5, Fa1F8, Fa1F9, Fa1F10, Fa1F11, FA4F1, FA4F2, FA4F3	Le2F1, FA3F1, FA3F2

Der folgende 4.8 UML Zustandsautomat umfasst alle drei unterschiedlichen Phasen als Zustände. Zusätzlich gibt es den Zustand Fehler für gefundene Fehler, den Startzustand und Endzustand. Aus Gründen der Übersichtlichkeit sind nur die Regeln dargestellt, die den Zustand „Spurwechsel“ betreffen.

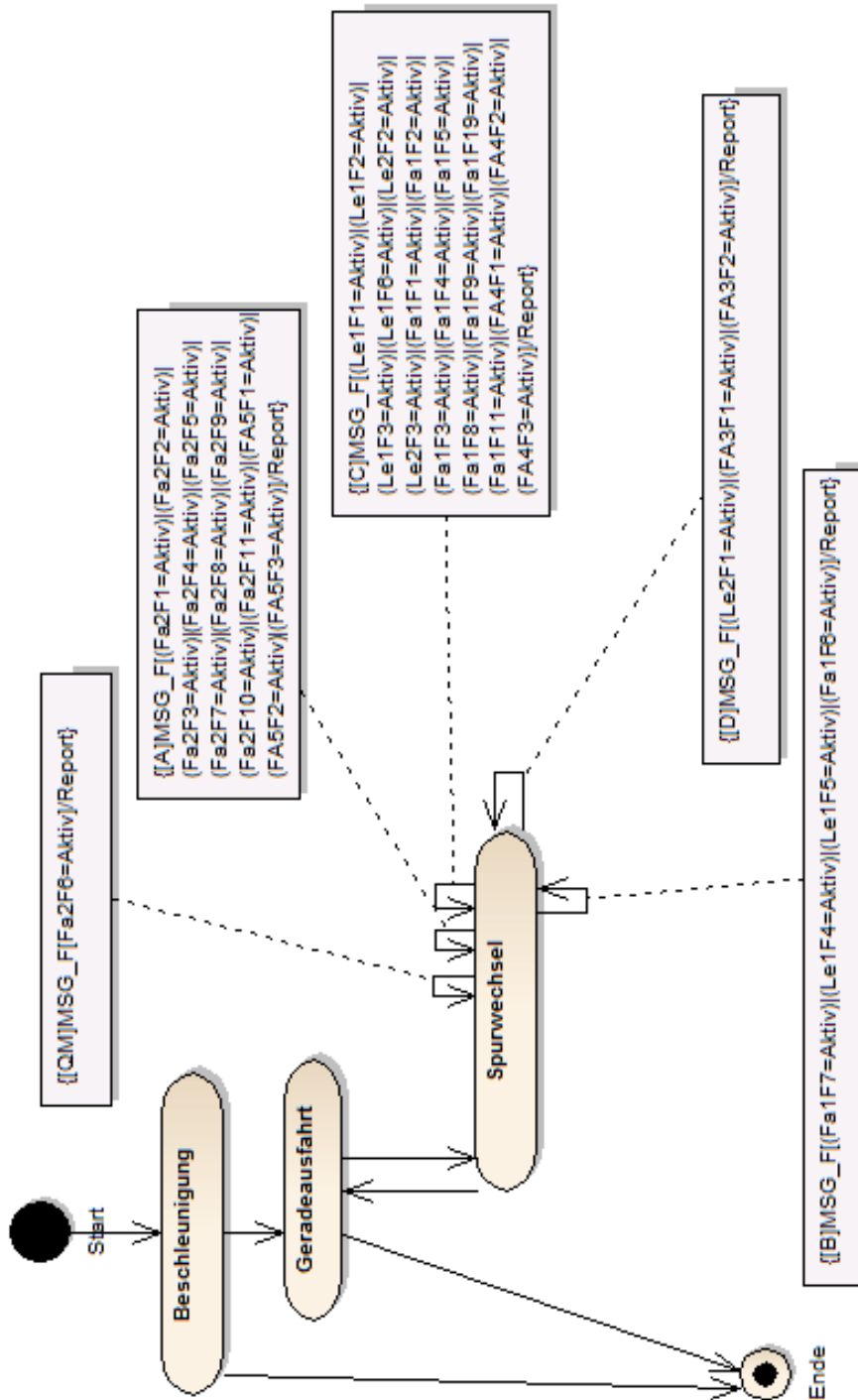


Bild 4.8 Zweimaliger Fahrbahnwechsel

5 Ergebnis und Ausblick der Arbeit

Für die praktische Umsetzung an dem Fahrsimulator wurde eine Testsequenz aus dem Beispiel Testmodell des zweimaligen Fahrbahnwechsels ausgewählt. Diese Testsequenz führte ein Testfahrer aus, die Testfahrt wurde dabei aufgezeichnet. Diese Messung wurde anschließend mit dem Beobachter analysiert. Das Ergebnis der Analyse war, dass die Fahrdynamik noch zu verbessern war, aber der Fahrsimulator grundsätzlich zum Testen der Interaktion zwischen Fahrer und den verteilten Funktionen geeignet ist.

Der Beobachter wurde in zwei Phasen entwickelt. In der ersten Phase wurde die gesamte Werkzeugkette vom Import des UML-Zustandsautomats bis zur CANoe „DLL“ erstellt. Die Umsetzung beinhaltet auch die modellbasierte Reporterstellung. Dadurch konnten die Ergebnisse in den Zustandsautomaten zurückgespiegelt werden. In der zweiten Phase wurde die risiko-basierte Datenanalyse umgesetzt. Die erste Umsetzung des Beobachtermodells wurde in einem breiten Rahmen eingesetzt, um einen Belastungstest des Beobachters durchzuführen. Es wurden folgende Modelle erstellt:

- Es wurde ein Beobachtermodell für das Aufstartverhalten und das Nachlaufverhalten aller Steuergeräte im Fahrwerksverbund in Regeln gefasst und überwacht. In diesem Zeitabschnitt ist die interne Steuergerätediagnose inaktiv.
- Es wurde ein Beobachtungsmodell für das Anfahren und Halten am Berg erstellt. In dieser verteilten Funktion arbeiten mehrere Steuergeräte zusammen.
- Es wurde ein Beobachtungsmodell für das Notlaufverhalten des Fahrwerks erstellt.
- Es wurde ein Beobachtungsmodell zum Überwachen aller Fehlermeldungen des Fahrwerksverbunds eingesetzt.
- Es wurde ein Beobachtungsmodell für das Überwachen des zweimaligen Fahrbahnwechsels eingesetzt.

Die verschiedenen Modelle wurden zunächst erprobt und dann während der Entwicklungsphase der Steuergeräte eingesetzt. Der Beobachter wurde an verschiedenen Testplattformen eingesetzt. Es wurden hierbei einige Fehler bei den SG im Aufstartverhalten entdeckt und ein Nachweis der Belastbarkeit somit erbracht. Die Analyse der Dauerlaufaufzeichnung von über 10 Stunden führte zu einem Messbericht, der größer als 400 Megabyte war. Aber durch das Rückspiegeln in das Testmodell konnte der Tester mit einem Blick die Analyse bewerten.

Die verbesserte Datenanalyse wurde bei verschiedenen Beobachtungsmodellen eingesetzt. Mit der Analyse des Aufstartverhaltens und des Nachlaufverhaltens der Steuergeräte und der verbesserten Datenanalyse durch das Zusammenführen von verschiedenen Messungen zu einem Ergebnismodell konnte der Nachweis geführt werden, dass alle erwarteten Verhalten erreicht wurden. Gerade bei einem großen Testraum und bei der endlichen Testzeit führt dieser Nachweis zu einem erhöhten Vertrauen in die Teststrategie. Die Verknüpfung zwischen dem erwarteten Verhalten des Beobachters und des Wächters des Testmodells wurde zur gegenseitigen Verifikation der beiden Systeme verwendet. Dadurch konnte die Umsetzung mit einem TCL Level von 1 eingestuft werden.

Um das Verfahren der Testfallerzeugung vorab zu prüfen, wurden die aus dem Testmodell erstellten Testsequenzen zuerst im Simulationsaufbau erprobt. Die Ausführung einer Testsequenz im Simulationsaufbau dauerte 17 Sekunden. TESAM erzeugt aus dem Testmodell, wie im Bild 4.8 dargestellt, mit der heuristischen Strategie 132 Testsequenzen, die in der Tabelle 4.1 dargelegt sind. Die Testsequenzen wurden im Simulationsaufbau ausgeführt, was zu einer gesamten Testzeit von 38 Minuten führte. Zum Vergleich erzeugte TESAM weitere Testsequenzen mit verschiedenen Strategien zur Testfallerzeugung, welche dann auch im Simulationsaufbau erprobt wurden. Die umfassendste Strategie, die mit TESAM möglich ist, ist die Strategie der vollen Kombinatorik der Wächter mit einer Tiefensuche zur Pfadbildung. Da diese Strategie zu 2097152 Testsequenzen führt, wurde sie aber nicht im Simulationsaufbau durchgeführt, da sie zu einer Testzeit von ca. 420 Tagen führen würde und im üblichen Testzeitraum von Steuergeräten nicht durchführbar ist. Bei der N+ Strategie erzeugt TESAM 158 Testsequenzen aus dem Testmodell. Bei der Strategie, die N+ mit MC/DC kombiniert erzeugt TESAM 171 Testsequenzen. Dabei zeigt

die Simulation, dass die ASIL-D und ASIL-C Bereiche am besten durch die heuristische Strategie abgedeckt werden. Das führt aber zu einer schwächeren Abdeckung in den Bereichen von ASIL-B, ASIL-A und QM.

Das Testmodell wurde so verändert, dass einige Wächterelemente, die ursprünglich ASIL(D)-C eingestuft waren auf D hochgestuft wurden. Diese Hochstufung erfolgte dann in zwei Schritten: Zuerst mit den beiden Elementen Le1F3, FA4F1 und dann mit den beiden Elementen FA4F2, FA4F3. In beiden Schritten wurden zwei Strategien ausgewählt, um die Auswirkung der Hochstufung zu bewerten. Zum einen die Strategie, die wie folgt arbeitet: N+ mit MC/DC in allen Regeln, die nicht mit ASIL-D bewertet sind, während für die ASIL-D Bereiche mit voller Kombinatorik getestet wurde. Zum anderen wurde die heuristische Strategie gewählt. Beide Strategien haben die gleiche Testabdeckung im ASIL-C Bereich und im ASIL-D Bereich des Testmodells.

Tabelle 5.1 Laufzeitanalyse

maximale Variable in ASIL-D	Testsequenzen in der Vergleichsstrategie	Testsequenzen in der heuristischen Strategie
3	191	132
5	306	226
7	688	608

Es zeigt sich, dass der Abstand zwischen den Testsequenzen beider Strategien pro Schritt bei einer steigenden Anzahl von Variablen im Bereich ASIL-D gleich bleibt. Es ist ein deutlicher Zeitgewinn bei der heuristischen Strategie festzustellen.

Beim Ausführen der Testsequenzen am V-HiL gibt es eine Dominanz der Start- und Stop-Routinen. Diese Routinen sorgen für einen sauberen Systemzustand der realen Steuergeräte. Dadurch beträgt die Ausführungsdauer am V-HiL nicht nur 17 Sekunden wie im Simulationsaufbau, sondern dauert mehr als 3 Minuten, was zu einer deutlichen Steigerung der Testlaufzeit führt. Wenn also jede Testsequenz mehr Zeit braucht führt die heuristische Strategie am V-HiL zu einer merklichen Zeitersparnis ohne

dabei die Testabdeckung im Hochrisikobereich von ASIL-C und ASIL-D zu reduzieren.

Wie sich aus diesen Ergebnissen zeigt, ermöglicht die Weiterentwicklung des bestehenden Ansatzes der syntaxbasierten Testfallerstellung von TESAM hin zu einer risikobasierten Teststrategie schon bei der Erstellung des Testmodells genau zu bestimmen, in welchen Bereichen des Testmodells mehr Testfälle zu erzeugen sein werden. Dadurch müssen nicht die verschiedenen Teststrategien untersucht werden, um die zeitlich beste Testfallmenge zu erreichen. Im Hochrisikobereich kann dennoch die Strategie der vollen Kombinatorik angewendet werden, die sonst insgesamt zu viele Testfälle erzeugt.

Die Verwendung der vollen Kombinatorik im Bereich von ASIL-D führt zu einer deutlich belastbaren Qualitätsaussage, da dadurch ein sehr dichtes Testnetz aufgespannt wird. Der Einsatz der vollen Kombinatorik ist allerdings beschränkt, da sie oberhalb von mehr als 12 Variablen zu einer Testzeit führt, die nicht mehr real testbar ist.

Durch die Verwendung eines Beobachters kann eine Wiederverwendungsstrategie in dem heterogenen Testplattformumfeld angewendet werden und so die Auswertung frühzeitig und breit gefächert erfolgen. Es vereinfacht die visuelle Rückmeldung an dem Tester und ermöglicht eine kompakte Ergebnisdarstellung, um den Testlauf auf einen Blick zu bewerten. Es zeigt sich, dass mit der Umsetzung des Beobachters, durch ein Expertensystem mit dessen Rete Algorithmus auch unvollständige Modelle eingesetzt werden können. So können die Testmodelle mit der Entwicklung der Steuergeräte Schritt halten.

Werkzeuge zur Qualitätssicherung durch Testen müssen aus Sicht der Norm ISO 26262 ein TCL Level von 1 erfüllen, um bei der Steuergeräteentwicklung eingesetzt zu werden. Gerade bei neuen Werkzeugen muss das Werkzeug sowohl qualifiziert werden wie auch über eine geeignete Selbstdiagnose verfügen um Fehler zu entdecken und um den TCL Level von 1 zu erreichen. Durch die Aufteilung in zwei Modelle kann je ein Modell das andere testen. Dadurch wurde das geforderte TCL Level 1 erreicht und die Anwendbarkeit der Methode nachgewiesen, da sie belastbare Qualitätsaussagen liefert.

Literaturverzeichnis

- [20003] *Sicherung der Qualität vor Serieneinsatz: Sicherung der Qualität während der Produktrealisierung; Methoden und Verfahren.* VDA, Qualitätsmanagement-Center (QMC), 2003 (Qualitätsmanagement in der Automobilindustrie). <http://books.google.de/books?id=WSWGPwAACAAJ>
- [20008] WINTERHAGEN, Johannes (Hrsg.): Der neue BMW 7er. 2008. (ATZ Extra). – Forschungsbericht
- [20110] WINTERHAGEN, Johannes (Hrsg.): Der VW Touareg. 2010. (ATZ Extra). – Forschungsbericht
- [Bau03] BAUER, H.: *Kraftfahrtechnisches Taschenbuch.* Vieweg, 2003 <http://books.google.de/books?id=PmMYUqiEVcYC>. – ISBN 9783528238766
- [Bau04] BAUER, Thomas: Kooperation von Projekt- und Workflow-Management-Systemen. In: *Informatik - Forschung und Entwicklung* 19 (2004), 74-86. <http://dx.doi.org/10.1007/s00450-004-0164-6>. – ISSN 0178-3564. – 10.1007/s00450-004-0164-6
- [Ber06] *Tagungsband Dagstuhl-Workshop MBEES : Modellbasierte Entwicklung eingebetteter Systeme II.* 2006 (06022)
- [Ber09] BERTSCHE, B: Grundlagen für eine Zuverlässigkeitsbewertung mechatronischer Systeme. Springer, 2009
- [Bin05] BINDER, R.: *Testing Object-Oriented Systems.* Addison-Wesley, 2005
- [BKI06] BEIERLE, C. ; KERN-ISBERNER, G.: *Methoden wissensbasierter Systeme: Grundlagen - Algorithmen - Anwendungen.* Vieweg, 2006 (Computational Intelligence). <http://books.google.de/books?id=0v1Y37UT0nIC>. – ISBN 9783834800107

- [BN08] BROEKMAN, Bart ; NOTENBOOM, Edwin: *Testing embedded software*. Addison-Wesley, 2008
- [BPL10] BAUMANN, G. ; PIEGSA, A. ; LIEDECKE, C.: *Der neue Fahr Simulator der Universität Stuttgart*. 2010
- [Bro05] BROY, Manfred (Hrsg.): *Model-based testing of reactive systems*. Springer, 2005
- [Bro09] BROST, Michael: *Automatisierte Testfallerzeugung auf Grundlage einer zustandsbasierten Funktionsbeschreibung für Kraftfahrzeugsteuergeräte*. Renningen, Diss., 2009
- [BRZ08] BROST, Michael ; REUSS, Hans-Christian ; ZÖLLER, Rolf: Automatische Testfallgenerierung aus einer formalen Funktionsbeschreibung. (2008)
- [CGJ89] CARRICO, M.A. ; GIRARD, J.E. ; JONES, J.P.: *Building knowledge systems: developing and managing rule-based applications*. Intertext Publications, 1989 (Artificial intelligence series). <http://books.google.de/books?id=AnwZAQAAIAAJ>. – ISBN 9780070234376
- [Chi01] CHILENSKI, John J.: An investigation of three forms of the modified condition decision coverage (mcdc) criterion / Office of Aviation Research. 2001. – Forschungsbericht
- [Cho78] CHOW, Tasun: Testing software design modeled by finite state machines. In: *Transactions on Software Engineering* SE-4 (1978), S. 178–186
- [CS01] CLAUS, V. ; SCHWILL, A.: *Duden Informatik: ein Fachlexikon für Studium und Praxis*. Dudenverl., 2001 <http://books.google.de/books?id=4I3TPwAACAAJ>. – ISBN 9783411052332
- [DHM12] DEICKE, Markus ; HARDT, Wolfram ; MARTINUS, Marcus: Simulation hardwarespezifischer Komponenten von ECU-Software in der virtuellen Absicherung. In: *ATZ Elektronik* 3 (2012), June
- [DHS⁺] DORN, Rüdiger ; HETZEL, Günter ; SCHUMANN, Matthias ; ZÖLLER, Rolf ; MANICKE, Oliver: Testhaus EE: Testpro-

zesse und Testmethoden zur Absicherung verteilter Funktionen innerhalb der Elektrik/Elektronik-Entwicklung der Porsche AG. In: *Integration The Vlsi Journal*

- [Dij70] DIJKSTRA, E.W.: *Notes on Structured Programming*. Technological University, Department of Mathematics, 1970 (TH report). <http://books.google.de/books?id=e6gMNQAACAAJ>
- [DIN] DIN: *DIN EN 61508*
- [DNSVT07] DIAS NETO, Arilo C. ; SUBRAMANYAN, Rajesh ; VIEIRA, Marlon ; TRAVASSOS, Guilherme H.: A survey on model-based testing approaches: a systematic review. In: *Proceedings of the 1st ACM international workshop on Empirical assessment of software engineering languages and technologies: held in conjunction with the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE) 2007*. New York, NY, USA : ACM, 2007 (WEASEL Tech '07). – ISBN 978-1-59593-880-0, 31-36
- [Dut] DUTT, Jörg: *DIN IEC 61025*
- [For] FORCHERT, Thomas M.: Anwendung der technischen Risikoanalyse für die Planung von Tests, Prüfungen und Wartungsmaßnahmen. In: *System*
- [For82] FORGY, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. In: *Artificial intelligence* 19 (1982), Nr. 1, S. 17-37
- [FPS] FERRARA, F ; PLÖGER, M ; SCHÜTTE, H: Automatisierter HIL-Test im Entwicklungsprozess vernetzter, automotiver Elektroniksysteme. In: *Development*
- [Fra90] FRANKS, Paul M.: Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-based Redundancy A Survey and Some New Results. In: *Automatica* 26 (1990), S. 459-474
- [GH01] GARMUS, D. ; HERRON, D.E.: *Function point analysis: measurement practices for successful software pro-*

- jects*. Addison-Wesley, 2001 (Addison-Wesley information technology series). <http://books.google.de/books?id=4J1QAAAAMAAJ>. – ISBN 9780201699449
- [GKS06] GOFUKU, A. ; KOIDE, S. ; SHIMADA, N.: Fault Tree Analysis and Failure Mode Effects Analysis Based on Multi-level Flow Modeling and Causality Estimation. (2006), S. 497–500. <http://dx.doi.org/10.1109/SICE.2006.315478>. – DOI 10.1109/SICE.2006.315478
- [GNRS09] GÖTZ, H. ; NICKOLAUS, M. ; ROSSNER, T. ; SALOMON, K.: *Modellbasiertes Testen: Modellierung und Generierung von Tests ; Grundlagen, Kriterien für Werkzeugeinsatz, Werkzeuge in der Übersicht*. Heise Zeitschriften Verl., 2009
- [Gud03] GUDDAT, Ulrich: *Automatisierte Tests von Telematiksystemen im Automobil*, Eberhard-Karls-Universität Tübingen, Dissertation, 2003
- [HAH04] HECHT, Herbert ; AN, Xuegao ; HECHT, Myron: Computer Aided Software FMEA for Unified Modeling Language Based Software. In: *Development* (2004), S. 243–248
- [Har87] HAREL, David: STATECHARTS: A visual formalism of complex Systems. In: *Science of Computer Programming* 8 (1987), S. 231–274
- [Hes06] HESSEL, Anders: *Model-Based Test Case Generation for Real-Time Systems*, Uppsala, Diss., 2006
- [Ise06] ISERMANN, Rolf: Das mechatronische Kraftfahrzeug. Version: 2006. http://dx.doi.org/10.1007/978-3-8348-9049-8_1. In: ISERMANN, Rolf (Hrsg.): *Fahrdynamik-Regelung*. Vieweg, 2006. – ISBN 978-3-8348-9049-8, 1-26
- [isoa] ISO: *1. Vocabulary ; Road vehicles : Functional safety - ISO*
- [isob] ISO: *10. Guideline ; Road vehicles : Functional safety - ISO*

- [isoc] ISO: 2. *Management of functional safety ; Road vehicles : Functional safety - ISO*
- [isod] ISO: 3. *Concept phase ; Road vehicles : Functional safety - ISO*
- [isoe] ISO: 4. *Product development: system level ; Road vehicles : Functional safety - ISO*
- [isof] ISO: 5. *Product development: hardware level ; Road vehicles : Functional safety - ISO*
- [isog] ISO: 6. *Product development: software level ; Road vehicles : Functional safety - ISO*
- [isoh] ISO: 7. *Production and operation ; Road vehicles : Functional safety - ISO*
- [isoi] ISO: 8. *Supporting processes ; Road vehicles : Functional safety - ISO*
- [isoj] ISO: 9. *ASIL-oriented and safety-oriented analyses ; Road vehicles : Functional safety - ISO*
- [Kap05] KAPICI, Senol: *Ein stochastisches Risikomodell für komplexe Projekte*, Otto-von-Guericke-Universität Magdeburg, Dissertation, 2005
- [KB09] KENETT, Ron S. ; BAI, Xiaoying: Risk-Based Adaptive Group Testing of Web Services. In: *Cybernetics* (2009)
- [KR10a] KIEFNER, Dominique X. ; REUSS, Hans-Christian: *Automatische Testfallgenerierung unter Einbeziehung der Risikopriorität.* 2. Baden-Württemberg Testing Day, 30.09.2010, September 2010
- [KR10b] KIEFNER, Dominique X. ; REUSS, Hans-Christian: DriFT, Dynamisches risikobasiertes Fahrwerksverbund Testverfahren. In: *Auto Test 2010* (2010)
- [LBK] LAMBERG, Klaus ; BEINE, Michael ; KÖHL, Susanne: Funktions-, Software- und Steuergerätestest sicherheitsrelevanter Funktionen gemäß ISO 26262.

- [Lig09] LIGGESMEYER, Peter: *Software-Qualität*. Heidelberg: Spektrum Akad. Verl., 2009
- [LL07] LUDEWIG, Jochen ; LICHTER, Horst: *Software-Engineering*. Heidelberg dpunkt-Verlag, 2007
- [Lon] LONGSTREET, David: *Test Case & Defects*. <http://www.softwaremetrics.com/Articles/defects.htm>
- [LPP07] LOEW, Peter ; PAPST, Roland ; PETRY, Erwin: *Funktionale Sicherheit in der Praxis*. dpunkt, 2007
- [Mat08] MATHUR, A.P.: *Foundations of Software Testing*. Pearson Education, 2008 <http://books.google.de/books?id=TBqmNQAACAAJ>. – ISBN 9788131716601
- [MDF13] MARTINUS, Marcus ; DEICKE, Markus ; FOLIE, Michael: Virtueller Fahrversuch - Hardwareunabhängige Integration von Seriensoftware. In: *ATZ Elektronik* 5 (2013), October, S. 344–349
- [MMR] MAYER, Christian ; MÜLLER, Mark ; RISSLING, Peter: Mit MiL, SiL und HiL den Antrieb unter Kontrolle - eine Simulationsplattform für alle Fälle aus einer Hand. , Nr. MiL
- [MRW⁺77] MCCALL, J.A. ; RICHARDS, P.K. ; WALTERS, G.F. ; CENTER, Rome Air D. ; DIVISION, United States. Air Force. Systems Command. Electronic S.: *Factors in software quality*. Rome Air Development Center, Air Force Systems Command, 1977
- [PC00] PATTON, Paul M. F. Ron J J. Ron J ; CLARK, Robert N.: *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000
- [PVDBJVV04] PINKSTER, I. ; VAN DE BURGT, B. ; JANSSEN, D. ; VAN VEENENDAAL, E.: *Successful Test Management: An Integral Approach*. Springer, 2004 <http://books.google.de/books?id=BkgMqRJG2YAC>. – ISBN 9783540228226
- [RBGW10] ROSSNER, T. ; BRANDES, C. ; GÖTZ, H. ; WINTER, M.: *Basiswissen modellbasierter Test*. Dpunkt.Verlag

- GmbH, 2010 <http://books.google.de/books?id=WfG1QQAACAAJ>. – ISBN 9783898645898
- [RD] ROCH, Matthias ; DEUTSCHMANN, Rocco: Testautomatisierung und HiL für Diagnosetests.
- [Rei10] REIF, Konrad: Elektronisches Stabilitäts-Programm ESP. Version: 2010. http://dx.doi.org/10.1007/978-3-8348-9717-6_5. In: REIF, Konrad (Hrsg.): *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*. Vieweg Teubner, 2010. – ISBN 978-3-8348-9717-6, 58-73
- [Ril] RILEY, Gary: CLIPS User's Guide. – Forschungsbericht
- [Sax08] SAX, E.: *Automatisiertes Testen Eingebetteter Systeme in der Automobilindustrie*. Hanser, 2008. – ISBN 9783446416352
- [Sch07] SCHIEFERDECKER, I.: Modellbasiertes Testen. (2007)
- [SD88] SABNANI, Krishan ; DAHBURA, Anton: A protocol test generation procedure. In: *Comput. Netw. ISDN Syst.* 15 (1988), September, Nr. 4, 285–297. [http://dx.doi.org/10.1016/0169-7552\(88\)90064-5](http://dx.doi.org/10.1016/0169-7552(88)90064-5). – DOI 10.1016/0169-7552(88)90064-5. – ISSN 0169-7552
- [SMP08] STALLBAUM, Heiko ; METZGER, Andreas ; POHL, Klaus: An Automated Technique for Risk-based Test Case Generation and Prioritization. In: *Systems Engineering* (2008), S. 67–70
- [spr08] Die 2000er Jahre bis 2006. In: REUSE, Bernd (Hrsg.) ; VOLLMAR, Roland (Hrsg.): *Informatikforschung in Deutschland*. Springer Berlin Heidelberg, 2008. – ISBN 978-3-540-76550-9
- [SS97] SHEHADY, R.K. ; SIEWIOREK, D.P.: A method to automate user interface testing using variable finite state machines. In: *Fault-Tolerant Computing, 1997. FTCS-27. Digest of Papers., Twenty-Seventh Annual International Symposium on IEEE*, 1997, S. 80–88

- [SZ05] SCHÄUFFELE, J. ; ZURAWKA, T.: *Automotive Software Engineering*. Vieweg, 2005 <http://books.google.de/books?id=41d2QgAACAAJ>. – ISBN 9780768014907
- [TS03] TANENBAUM, A.S. ; STEEN, M.: *Verteilte Systeme*. Pearson Studium, 2003 <http://books.google.de/books?id=qXGn0gAACAAJ>. – ISBN 9783827370570
- [UML] OMG: *Unified Modeling Language: Superstructure UML 2*
- [VCI90] VUONG, S.T. ; CHAN, W.W.L. ; ITO, M.R.: The UIOv-method for protocol test sequence generation. In: *IFIP 2th International Workshop on Protocol Test Systems*, 1990, S. 203–225
- [Weh08] WEHLING, Tobias: *Konzept und Implementierung eines Verfahrens zum regelbasierten Testen von E/E Systemen an einem Hardware in the Loop (HiL) Prüfstand*, Universität Stuttgart, Diplomarbeit, 2008
- [Wie] WIESE, Matthias: *Kosten-Nutzenoptimierung von Tests modellbasiert entwickelter Fahrzeugfunktionen*.

Abkürzungen und Formelzeichen

Abkürzungen

Abkürzung	Definition
ABS	Antiblockiersystem
ACC	Adaptive Cruise Control
ALPAS	Allgemeine Parametrierschnittstelle
ASIL	Automotive Sicherheits Integritätslevel
DIN	Deutsches Institut für Normung
DLL	Dynamic Link Library
ECU	Electronic Control Unit
EMV	Elektromagnetische Verträglichkeit
ESP	Elektrisches Stabilitäts Paket
FBA	Fehlerbaumanalyse
FMEA	Fehler-Möglichkeiten- und Einflussanalyse
FKFS	Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren
FiL	Function in the Loop
HiL	Hardware in the Loop
HTML	Hyper Text Markup Language
IEC	Internationale Elektrotechnische Kommission
ISO	Internationale Organisation für Normung
IT	Informationstechnologie
i	Stellt die Anzahl der Zustände der Implementierung
UML	Unified Modelling Language
MBT	Modellbasiertes Testverfahren

MiL	Model in the Loop
MC/DC	Modified Condition/Decision Coverage
MC	Multiple Condition
NASA	National Aeronautics and Space Administration
OEM	Original Equipment Manufacturer
QM	Qualitätsmanagement
RRBT	Risk Requirement Based Testing
SG	Steuergerät
SiL	Software in the Loop
TCL	Tool Confidence Level
TESMA	Testsequenz-Automaten
SVG	Scalable Vector Graphics
V-HiL	Verbund-Hardware in the Loop
V-Modell	Vorgehensmodell zur Entwicklung
VDA	Verband der Automobilindustrie
XMI	XML Metadata Interchange
XML	Extensible Markup Language
z	Stellt die Anzahl der Zustände des Zustandsautomats

Lebenslauf

Dominique Xavier Kiefner

Im Brühl 9

71404 Korb

Telefon: 07151/1679258

E-Mail: dominique.xavier-
.kiefner@googlemail.com

Geburtsdatum:

13.05.1977

Geburtsort:

Bad-Cannstatt

Staatsangehörigkeit:

deutsch

Berufliche Tätigkeit:

12.2006-12.2011

Wissenschaftlicher Mitarbeiter
beim FKFS

03.2004-08.2004

Praktikum bei Agilent Technolo-
gies

04.2002-10.2002

Tutor des Fachpraktikums für
Java RMI

11.2000-01.2002

Werkstudent bei der Firma Mar-
kenwerke

Akademische Ausbildung:

09.2006

Abschluss des **Studiums** (*Note:
Gut*)

03.2001

Abschluss des Grundstudiums
(Vordiplom)

10.1998	Beginn des Diplomstudiengangs Softwaretechnik an der Universität Stuttgart
Schulbildung:	
1997	Erwerb der Allgemeinen Hochschulreife
1994-1997	Technisches Gymnasium der Gewerblichen Schule Waiblingen
1994	Erwerb der Mittleren Reife
1988-1994	Reinhold-Nägele Realschule in Weinstadt
1987-1988	Hauptschule, Christian-Morgenstern-Schule in Waiblingen
1983-1987	Grundschule, Christian-Morgenstern-Schule in Waiblingen