Entwicklung eines Verfahrens zur Modellierung von Abhängigkeiten zwischen Modulen für die modulare Absicherung automatisierter Fahrfunktionen

Masterthesis Nr. 835/21

Bearbeiterin: Svenja Katharina Kohls | 2253088
Betreuer: Björn Klamann, M. Sc.

TECHNISCHE UNIVERSITÄT DARMSTADT

FZD FAHRZEUGTECHNIK TU DARMSTADT

Svenja Katharina Kohls
Matrikelnummer: 2253088
Studiengang: Mechanical and Process Engineering (M.Sc.)

Masterthesis Nr. 835/21
Thema: Entwicklung eines Verfahrens zur Modellierung von Abhängigkeiten zwischen Modulen für die modulare Absicherung automatisierter Fahrfunktionen

Eingereicht: 27. April 2022

Veröffentlicht unter der Creative Commons Lizenz: CC BY 4.0 International

Technische Universität Darmstadt
Fachgebiet Fahrzeugtechnik
Prof. Dr. rer. nat. Hermann Winner
Otto-Berndt-Straße 2
64287 Darmstadt

Erklärung

Erklärung zur Abschlussarbeit gemäß § 22 Abs. 7 und § 23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Svenja Katharina Kohls, die vorliegende Master-Thesis gemäß § 22 Abs. 7 APB TU Darmstadt ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Falle eines Plagiats (§38 Abs.2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß § 23 Abs. 7 APD TU Darmstadt überein.

English translation for information purposes only:

Thesis Statement pursuant to § 22 paragraph 7 and § 23 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I, Svenja Katharina Kohls, have written the submitted thesis independently pursuant to § 22 paragraph 7 of APB TU Darmstadt. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources, which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form.

I am aware, that in case of an attempt at deception based on plagiarism (§38 Abs. 2 APB), the thesis would be graded with 5,0 and counted as one failed examination attempt. The thesis may only be repeated once.

In the submitted thesis, the written copies and the electronic version for archiving are pursuant to § 23 paragraph 7 of APB TU Darmstadt identical in content.

Matrikelnummer: 2253088

Datum / Date: 27.04.2022

Unterschrift / Signature: S.Kohls

Masterthesis Nr. 835/21 im Studiengang Mechanical and Process Engineering (30 CP)

von Svenja Kohls

Voraussichtlicher Beginn: 04.10.2021
Bearbeitungsdauer: 6 Monate

Thema: **Entwicklung eines Verfahrens zur Modellierung von Abhängigkeiten zwischen Modulen für die modulare Absicherung automatisierter Fahrfunktionen**

Topic: ***Development of a method for modelling dependencies between modules for the modular safety approval of automated driving functions***

Am Fachgebiet Fahrzeugtechnik der TU Darmstadt (FZD) wird für das vom Bundesministerium für Bildung und Forschung (BMBF) geförderte Projekt UNICAR*agil* in Zusammenarbeit mit mehreren deutschen Universitäten an modularen und diensteorientierten automatisierten Fahrzeugen geforscht. Der Fokus liegt dabei auf der Vermeidung technologischer Altlasten und darauf, diese durch neue, disruptive Konzepte zu ersetzen. Die Absicherung erfolgt im Projekt nicht mit Schwerpunkt auf dem Gesamtsystem, sondern auf der individuellen Erprobung der einzelnen Module.

Die individuelle Erprobung einzelner Module erfordert die Berücksichtigung der Abhängigkeiten zu anderen Modulen.
Ziel dieser Arbeit ist es, ein Verfahren zu entwickeln, das diese Abhängigkeiten ermittelt und in modularen Testfällen berücksichtigt. Die Umsetzung der Testfälle erfolgt dabei in der vorhandenen Simulationsumgebung.

Als Ergebnisse der Masterthesis sind nachzuweisen:

1. Die grundlegende funktionale Architektur im Projekt UNICAR*agil* und die bestehenden Simulationsmodelle sind nachvollzogen und in der Arbeit dargestellt.
2. Der Lösungsraum möglicher Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung ist analysiert und dargestellt.
3. Ein Verfahren zur Ermittlung und Darstellung von Abhängigkeiten ist ausgewählt, ggf. für die gewünschte Anwendung angepasst und beschrieben.
4. Ein Verfahren zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten ist beschrieben.

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachgebiet Fahrzeugtechnik

FZD

Prof. Dr. rer. nat. Hermann Winner

Otto-Berndt-Straße 2
64287 Darmstadt

Bearbeiter:
Björn Klamann M. Sc.
Tel. +49 6151 16 – 24235
Fax +49 6151 16 – 24205
bjoern.klamann@tu-darmstadt.de
www.fahrzeugtechnik-darmstadt.de

Datum
30.09.21

CP	Dauer Zwischenkolloq (Vortrag / gesamt)	Dauer Abschlusskolloq (Vortrag / gesamt)
< 18	5-7 / 15 min	15 / 45 min
> 18	10 / 20 min	20 / 60 min

5. Für zuvor ausgewählte Module in UNICAR*agil* sind Abhängigkeiten und zugehörige Testfälle mit Hilfe der beschriebenen Verfahren ermittelt.
6. Die Testfälle sind in der bestehenden Simulationsumgebung umgesetzt und die Ergebnisse ausgewertet.
7. Die Methodik und Ergebnisse des Vorgehens sowie vorhandene Unzulänglichkeiten sind dokumentiert und diskutiert.

Schwerpunkte der Bewertung:

- Methodik des Vorgehens
- Vollständigkeit
- Nachvollziehbarkeit und Belastbarkeit der Argumentation
- Qualität folgender abzuliefernder Ergebnisse:
    - Schriftliche Ausarbeitung und Dokumentation
    - Erstellter Programmcode und zugehörige Dokumentation
- Abschlusskolloquium

Die Abgabe sämtlicher Daten und des Quellcodes wird vorausgesetzt. Die Arbeit bleibt Eigentum des Fachgebiets. Auf das Merkblatt des Fachgebiets wird hingewiesen.

Prof. Dr. rer. nat. Hermann Winner

Björn Klamann M. Sc.
(Betreuer)

Zusammenfassung

Das automatisierte Fahren ist ein Forschungszweig, der zunehmend mehr Bedeutung und Aufmerksamkeit erfährt und als Innovationstreiber in der Automobilindustrie wirkt. Eine wesentliche Herausforderung stellt dabei die Gewährleistung der funktionalen Sicherheit des automatisierten Fahrens dar. Diese Herausforderung ist Teil des vom Bundesministerium für Bildung für Forschung geförderten Projekts UNICAR*agil*. Die Projektziele sind die Entwicklung und Realisierung von vier hochautomatisierten modularen Fahrzeugen und die Entwicklung von Methoden zur Erbringung eines validen Sicherheitsnachweises. Konventionelle Ansätze zur Absicherung wie reale Testfahrten sind technisch und ökonomisch nicht realisierbar, um den erforderlichen Abdeckungsgrad an Testszenarien und Teststrecke zu erreichen. In UNICAR*agil* werden modulare und dienstorientierte automatisierte Fahrzeuge entwickelt, deren Architektur eine individuelle Erprobung und somit eine Abkehr von der Absicherung auf Gesamtsystemebene ermöglichen. Die individuelle Absicherung einzelner Module erfordert die Berücksichtigung der Abhängigkeiten zu anderen Modulen und zur Umgebung. Letzteres stellt sich bisher als schwierig dar.

Im Rahmen der vorliegenden Masterthesis wird im ersten Schritt ein Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung entwickelt. Das Ziel des Verfahrens ist, den Test- und Absicherungsaufwand für die Freigabe automatisierter Fahrfunktionen und -systeme zu reduzieren. Im Anschluss daran wird dieses Verfahren auf die Systemarchitektur von UNICAR*agil* angewendet. Im zweiten Schritt dieser Arbeit wird ein Verfahren zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten entwickelt. Bei der anschließenden Anwendung dieses Verfahrens werden die zuvor identifizierten Abhängigkeiten verwendet. Mit der abschließenden Umsetzung und Durchführung der Testfälle in der bestehenden Simulationsumgebung werden die ermittelten Abhängigkeiten aufgezeigt und weitere unbekannte Abhängigkeiten aufgedeckt.

Nach der Einarbeitung in das Fahrzeugkonzept und in die Systemarchitektur von UNICAR*agil* wird eine Literaturrecherche zum Stand der Technik beider zu entwickelnden Verfahren durchgeführt. Basierend auf den Rechercheergebnissen und der Anwendung in UNICAR*agil* werden Anforderungen an die beiden Verfahren definiert. Der Lösungsraum möglicher Verfahren aus der Literatur wird anhand der definierten Anforderungen bewertet. Die daraufhin ausgewählten Verfahren werden an die gewünschte Anwendung angepasst. Das entwickelte Verfahren zur Ermittlung und Darstellung von Abhängigkeiten basiert auf einem Modellierungswerkzeug, welches für kritische Infrastrukturen entwickelt wurde. In einem iterativen Prozess mit Erkenntnissen aus der Literatur und Expertengesprächen über u.a. Auslöser von Zustands- und Verhaltensänderungen wird die Systemarchitektur von UNICAR*agil* modelliert und das Abhängigkeitsnetz aufgebaut. Mit diesem Verfahren werden mit Fokus auf ausgewählte Module die vorliegenden Abhängigkeiten in der Systemarchitektur von UNICAR*agil* systematisch ermittelt. Das entwickelte Verfahren zur Definition von Testfällen beruht auf einem szenariobasierten Ansatz. Mittels einer experten- und anforderungsbasierten Szenariogenerierung werden Szenarien für einige ermittelte Abhängigkeiten erzeugt. Im Anschluss werden anhand der Testziele und Bewertungskriterien die in den durchgeführten Testfällen untersuchten Abhängigkeiten bestätigt und weitere unbekannte Abhängigkeiten aufgedeckt.

Inhaltsverzeichnis

Zusammenfassung I
Inhaltsverzeichnis II
Abkürzungsverzeichnis IV
Abbildungsverzeichnis V
Tabellenverzeichnis VI
1 Einleitung 1
1.1 Motivation 2
1.2 Konkretisierung der Aufgabenstellung 3
1.3 Aufbau der Arbeit und Methodik des Vorgehens 4
2 Grundlagen 5
2.1 ISO 26262 Road vehicles – Functional safety 5
2.2 UNICAR*agil* 6
2.2.1 Gesamtkonzept 6
2.2.2 Architektursichten 7
2.2.3 Dienste und Module 11
2.3 Testmethoden in der Entwicklung im Automobilbereich 19
2.3.1 Terminologie 19
2.3.2 Allgemeiner Testprozess 20
2.3.3 Testverfahren für automatisierte Fahrfunktionen 20
3 Entwicklung des ersten Verfahrens: Ermittlung und Darstellung von Abhängigkeiten 23
3.1 Anforderungsanalyse – Erstes Verfahren 23
3.2 Stand der Forschung – Erstes Verfahren 26
3.2.1 Bayesian Network 27
3.2.2 Functional Resonance Analysis Method 27
3.2.3 Infrastructure-Resilience-oriented Modelling Language 28
3.3 Bewertung des Lösungsraums – Erstes Verfahren 29
3.3.1 Verifizierung der Verfahren 29
3.3.2 Auswahl der Methode 30
3.4 Methodik – Erstes Verfahren 31
3.4.1 Anpassung an die Anwendung 31
3.4.2 Verfahren zur Ermittlung und Darstellung von Abhängigkeiten 32
4 Anwendung des ersten Verfahrens: Ermittlung und Darstellung von Abhängigkeiten im Rahmen des Projekts UNICAR*agil* 35
4.1 Systemmodellierung und Aufbau des Abhängigkeitsnetzes 35
4.1.1 Identifizierung der beteiligten Funktionen 36
4.1.2 Auflistung der erforderlichen Domänen und Ressourcen 36
4.1.3 Ermittlung der Datenflüsse und Kontrollbeziehungen und Teilsystemdefinition 38
4.1.4 Identifizierung neuer und von den Teilsystemen abhängigen Funktionen 39

4.2 Expertenbefragung ... 40
4.2.1 Vorgehensweise bei den Gesprächen ... 40
4.2.2 Erkenntnisse aus den Gesprächen ... 40
4.3 Ergebnisse – Erstes Verfahren ... 42
4.3.1 Abhängigkeitskategorien ... 42
4.3.2 Abhängigkeitsnetz ... 43
4.3.3 Abhängigkeitssteckbrief ... 46
4.3.4 Ermittelte Abhängigkeiten ... 47
4.3.5 Bewertung der Ergebnisse – Erstes Verfahren ... 49
5 Entwicklung des zweiten Verfahrens: Definition von Testfällen ... 52
5.1 Anforderungsanalyse – Zweites Verfahren ... 52
5.2 Stand der Forschung – Zweites Verfahren ... 54
5.2.1 Szenariogenerierung ... 55
5.3 Bewertung des Lösungsraums und Methodik – Zweites Verfahren ... 56
5.3.1 Verifizierung des Testprozesses ... 56
5.3.2 Verfahren zur Definition von Testfällen ... 57
6 Anwendung des zweiten Verfahrens: Definition von Testfällen im Rahmen des Projekts UNICAR*agil* ... 59
6.1 Definition der Testszenarien ... 59
6.1.1 Testfallgruppe I – Codebasierte Abhängigkeiten ... 59
6.1.2 Testfallgruppe II – Systemunabhängige Abhängigkeiten ... 61
6.1.3 Testfallgruppe III – Systemdesignbasierte Abhängigkeiten ... 62
6.1.4 Testfallgruppe IV – Inter-Datenbasierte Abhängigkeiten ... 64
6.1.5 Testfallgruppe V – Datentransferbasierte Abhängigkeiten ... 65
6.2 Ergebnisse – Zweites Verfahren ... 66
6.2.1 Bewertung der Testfälle ... 66
6.2.2 Bewertung der Ergebnisse – Zweites Verfahren ... 68
7 Diskussion der Ergebnisse ... 70
7.1 Einflüsse und Diskussion – Erstes Verfahren ... 70
7.2 Einflüsse und Diskussion – Zweites Verfahren ... 71
8 Fazit und Ausblick ... 72
Anhang ... 75
A.1 Abhängigkeitsnetz ... 75
A.2 Ermittelte Abhängigkeiten – Fahrdynamikzustandsschätzung ... 81
A.3 Ermittelte Abhängigkeiten – Verhaltens- und Trajektorienplanung ... 84
A.4 Ermittelte Abhängigkeiten – Trajektorienvorverarbeitung ... 86
A.5 Ermittelte Abhängigkeiten – Plattformsensorik ... 87
A.6 Ermittelte Abhängigkeiten – Fahrschlauchabsicherung ... 88
A.7 Ermittelte Abhängigkeiten – Dynamikmodul ... 91
Literaturverzeichnis ... 92

Abkürzungsverzeichnis

ASOA	Automotive Service-Oriented Software Architecture
BMBF	Bundesministerium für Bildung und Forschung
BN	Bayesian Network
E/E	Elektrik/Elektronik
ECU	Electronic Control Unit
FRAM	Functional Resonance Analysis Method
FSA	Fahrschlauchabsicherung
FTR	Fahrdynamik- und Trajektorienregelung
FZD	Fahrzeugtechnik Darmstadt
FZS	Fahrdynamikzustandsschätzung
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
IRML	Infrastructure Resilience-oriented Modelling Language
Lidar	Light detection and ranging
PerCollECT	Perception sensor Collaborative Effect and Cause Tree
Radar	Radio detection and ranging
TVV	Trajektorienvorverarbeitung
UML	Unified Modeling Language
VTP	Verhaltens- und Trajektorienplaner

Abbildungsverzeichnis

Abbildung 2-1: Gesamtkonzept von UNICAR*agil* 6
Abbildung 2-2: Schematische Darstellung der E/E-Architektur 8
Abbildung 2-3: Konzept der *ASOA* bei unterschiedlichen Betriebsmodi nach Woopen et al. 10
Abbildung 2-4: *A-Modell* der funktionalen Architektur in UNICAR*agil* 11
Abbildung 2-5: Systemarchitektur in UNICAR*agil* nach Gottschalg et al. 12
Abbildung 2-6: Struktur der Sensordatenfusion der FZS in UNICAR*agil* nach Buchholz et al. 15
Abbildung 2-7: Systemstruktur des VTP nach Kamran et al. 15
Abbildung 2-8: UNICAR*agil*-Fahrzeug mit einem örtlichen Solltrajektorienverlauf (grün - gestrichelt) und einem örtlichen Notbahnverlauf zu einem risikominimalen Ort (rot - durchgezogen) nach Ackermann et al. 16
Abbildung 2-9: Seitliches Einparkmanöver (blau - durchgezogen) vom auto*SHUTTLE* 18
Abbildung 3-1: Struktur einer *FRAM*-Funktion nach Carvalho et al. 28
Abbildung 3-2: *IRML*-Modell am Beispiel eines Gasförderungssystems nach Hu et al. 29
Abbildung 3-3: Anpassungen an die Modellierungssprache *UML* 32
Abbildung 3-4: Methodik zur Ermittlung und Darstellung von Abhängigkeiten 34
Abbildung 4-1: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - Legende 35
Abbildung 4-2: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 1. Zwischenschritt 36
Abbildung 4-3: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 2. Zwischenschritt (Ausschnitt FTR) 37
Abbildung 4-4: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 3. Zwischenschritt (Ausschnitt FTR) 38
Abbildung 4-5: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 4. Zwischenschritt (Ausschnitt FTR) 39
Abbildung 4-6: Vollständigkeit der definierten Abhängigkeitskategorien in UNICAR*agil* 43
Abbildung 4-7: Abhängigkeitsnetz (Ausschnitt FTR) 44
Abbildung 4-8: Legende für Abhängigkeitsnetz 44
Abbildung 4-9: Entwickelter Entwurf des Abhängigkeitssteckbriefs 46
Abbildung 5-1: Methodik zur Definition von Testfällen 58
Abbildung 6-1: Szenario 1 – „Seitliches Parkmanöver“ 60
Abbildung 6-2: Szenario 3 – „Fahrt über Bodenwellen“ 62
Abbildung 6-3: Szenario 5 – „Scharfe Kurve“ 63
Abbildung 6-4: Szenario 7 – „180° Wendung“ 65
Abbildung 6-5: Szenario 8 – „Tunnelfahrt“ 66
Abbildung A-1: Abhängigkeitsnetz – Ausschnitt Fahrdynamikzustandsschätzung 75
Abbildung A-2: Abhängigkeitsnetz – Ausschnitt Verhaltens- und Trajektorienplaner 76
Abbildung A-3: Abhängigkeitsnetz – Ausschnitt Trajektorienvorverarbeitung 77
Abbildung A-4: Abhängigkeitsnetz – Ausschnitt Plattformsensorik 78
Abbildung A-5: Abhängigkeitsnetz – Ausschnitt Fahrschlauchabsicherung 79
Abbildung A-6: Abhängigkeitsnetz – Ausschnitt Dynamikmodul 80

Tabellenverzeichnis

Tabelle 3-1: Anforderungen an das Verfahren zur Ermittlung und Darstellung von Abhängigkeiten 26
Tabelle 4-1: Definition der Abhängigkeitskategorien und Beispiele für UNICAR*agil* 42
Tabelle 4-2: Ermittelte Abhängigkeiten – Fahrdynamik- und Trajektorienregelung 48
Tabelle 5-1: Anforderungen an das Verfahren zur Definition von Testfällen 54
Tabelle A-1: Ermittelte Abhängigkeiten – Fahrdynamikzustandsschätzung 81
Tabelle A-2: Ermittelte Abhängigkeiten – Verhaltens- und Trajektorienplanung 84
Tabelle A-3: Ermittelte Abhängigkeiten – Trajektorienvorverarbeitung 86
Tabelle A-4: Ermittelte Abhängigkeiten – Plattformsensorik 87
Tabelle A-5: Ermittelte Abhängigkeiten – Fahrschlauchabsicherung 88
Tabelle A-6: Ermittelte Abhängigkeiten – Dynamikmodul 91

1 Einleitung

Das automatisierte Fahren ist neben der Elektrifizierung des Antriebs und dem vernetzten Fahren ein Trend in der Automobilbranche. Die Entwicklung und die Einführung verlaufen dabei nicht abrupt, sondern evolutionär aufgrund der sukzessiven Integration von Fahrerassistenzfunktionen und -systemen, deren Automatisierungsgrad immer weiter zunimmt. Diese Assistenzsysteme werden nach dem Standard SAE J3016[1] in die Automatisierungsstufen null bis fünf klassifiziert. Die 5. Automatisierungsstufe ist die höchste und beschreibt die *Vollautomatisierung* und somit die vollständige Ausführung der dynamischen Fahraufgabe unter allen Fahrbahn- und Umgebungsbedingungen durch das automatisierte Fahrsystem.[2,3,4]

Aktueller Stand der Technik ist die Einführung von Assistenzsystemen der Automatisierungsstufe 3 – *Bedingte Automatisierung* – und der Automatisierungsstufe 4 – *Hochautomatisierung.*[1] Bei diesen Stufen ist der Fahrer im Vergleich zu den darunterliegenden Stufen während des automatisierten Fahrens nicht zu einer Überwachung des Fahrzustands verpflichtet. Ab der Automatisierungsstufe 4 ersetzen die Systeme den Fahrer als Rückfallebene gänzlich, um das Fahrzeug in einen risikominimalen Zustand zu überführen. Das automatisierte Fahren ermöglicht eine Erhöhung der Auslastung der Verkehrskapazität und einen Gewinn an Komfort. Dies ist darin begründet, dass der Fahrer die Option hat, die Fahrzeit im Fahrzeug für andere Tätigkeiten als die Fahraufgabe zu nutzen, da die Möglichkeit, diese zeitweise oder vollständig nicht wahrzunehmen, besteht.[3,5]

Die schrittweise voranschreitende Entwicklung bei der Automatisierung von Fahrzeugen wirft neben technischen Fragen u.a. auch rechtliche, ethische und soziologische Fragen auf. Eine wesentliche Anforderung dabei ist die Gewährleistung der funktionalen Sicherheit der Assistenzsysteme und schlussendlich des automatisierten Fahrens, um ein fehlerfreies Verhalten zu garantieren. Gemäß der ISO-Norm 26262[6] konzentriert sich die funktionale Sicherheit auf die Vermeidung oder Verringerung von Systemausfällen aufgrund von Fehlern in den elektronischen Hardware- und Softwarekomponenten, die das Fahrzeug betreiben. Für die Sicherstellung der funktionalen Sicherheit beim automatisierten Fahren ist folglich die Erbringung eines validen Sicherheitsnachweises erforderlich. Dazu werden Anpassungen der herkömmlichen Entwicklungsprozesse und effiziente Test- und Absicherungsverfahren benötigt.[3,7]

Im Detail bedeutet dies, dass die Entwicklung und Einführung von Fahrerassistenzsystemen mit den Automatisierungsstufen 3, 4 und 5 den Nachweis erfordern, dass der Austausch des menschlichen Fahrers durch das automatisierte Fahrsystem die Sicherheit der Insassen sowie des gesamten

[1] SAE International: SAE J3016: Taxonomy and Definitions (2021).

[2] Shuttleworth, J.: SAE J3016 (2019).

[3] Pfeffer, R.: Diss., Funktionale Absicherung hochautomatisierter Fahrfunktionen (2020), S. 2.

[4] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 12.

[5] UNICARagil: UNICARagil news - 4. Ausgabe (2021), S. 5.

[6] International Organization for Standardization: ISO 26262 (2018), S. 6.

[7] Winner, H.: Quo vadis, FAS? (2015), S. 1173.

Straßenverkehrs nicht verringert. Für einen solchen Sicherheitsnachweis werden konventionell reale Fahrversuche während der Fahrzeugentwicklung durchgeführt. Im Fall des automatisierten Fahrens ist diese herkömmliche Methode jedoch technisch und ökonomisch nicht realisierbar.[8a,9a] Die zu berücksichtigende Anzahl an Fahrsituationen steigt exponentiell mit zunehmendem Automatisierungsgrad. Dieser Zusammenhang endet in der sogenannten *Freigabefalle.*[10a] Nach Wachenfeld und Winner[9b] sind für die Absicherung eines Autobahnautomaten Testfahrten mit einem menschlichen Fahrer auf einer Referenzstrecke von 6-7 Mrd. km notwendig.[8b,9c]

Des Weiteren erhöht die Verwendung von technisch immer höher entwickelten Assistenzsystemen die Komplexität dieser und erschwert die Absicherung der Assistenzsysteme zusätzlich. Darüber hinaus sind die vorherrschenden Entwicklungsmethoden in der Automobilindustrie für die erwarteten kurzen Innovationszyklen nicht zweckmäßig. Folglich ist das Ziel, die für den Sicherheitsnachweis relevanten Testfälle zu identifizieren, um die Anzahl der Testszenarien zu verringern und alternative Testmethoden zu finden, die reale Testfahrten vollständig oder teilweise ersetzen.[8c,10b]

Virtuelle Fahrversuche in einer Simulationsumgebung sind eine Alternative zu realen Testfahrten für die Freigabe in der Fahrzeugentwicklung. Diese haben Vorteile hinsichtlich Aufwandsreduktion, Reproduzierbarkeit und Flexibilität. Ferner sind weitere Ansätze zu einer möglichen Lösung der Problematik notwendig. Ein Ansatz ist die Entwicklung neuer, disruptiver Konzepte der Systemarchitektur für automatisierte Fahrzeuge. Ein Beispiel für einen neuen Ansatz ist die Modularisierung der Systemarchitektur. Eine modulare Fahrzeugarchitektur ermöglicht eine individuelle Absicherung der Module und somit einen Verzicht auf eine Absicherung auf Gesamtsystemebene.[8c,9d,11]

1.1 Motivation

Zur modularen Absicherung von automatisierten Fahrfunktionen und -systemen wird am Fachgebiet Fahrzeugtechnik der Technischen Universität Darmstadt (FZD) in Zusammenarbeit mit Industriepartnern und anderen deutschen Universitäten geforscht. Für eine modulare Absicherung ist eine klare Trennung der verschiedenen Funktionen mit zuvor definierten Schnittstellen bei den vorliegenden Hardware- und Softwarekomponenten notwendig. Durch eine anschließende Identifizierung der Abhängigkeiten zwischen den Modulen und zur Umgebung werden individuelle Modultests ermöglicht, welche den enormen Aufwand heutiger herkömmlicher System- und Integrationstests umgehen. Ein weiterer Grund für den reduzierten Aufwand beim Sicherheitsnachweis und somit bei der Freigabe von automatisierten Fahrfunktionen und -systemen ist, dass die Module in mehreren Fahrzeugmodellen verwendbar sind, ohne dass die Durchführung einer erneuten Gesamtabsicherung aufgrund der neu zusammengestellten Module notwendig wird. Außerdem verringert sich der Aufwand für die Absicherung bei Änderungen oder einem Update eines Moduls, denn für das betroffene Modul wird

[8] Pfeffer, R.: Diss., Funktionale Absicherung hochautomatisierter Fahrfunktionen (2020), a: S. 2; b: S. 2 f; c: S. 3.

[9] Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015), a: S. 440; b: S. 458; c: S. 457 f; d: S. 458 ff.

[10] Winner, H.: Quo vadis, FAS? (2015), a: S. 1177; b: S. 1178.

[11] Amersbach, C.; Winner, H.: Test Coverage for Automated Driving (2019), S. 3 f.

lediglich eine erneute Absicherung notwendig, für die anderen Module jedoch nicht. Somit wird der Testaufwand bei Änderungen oder Updates reduziert.[12,13a]

Hierfür ist das Fachgebiet FZD an dem Forschungsprojekt UNICAR*agil*[14] beteiligt. Im Rahmen des vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekts forscht das Fachgebiet FZD in Zusammenarbeit mit mehreren deutschen Universitäten an automatisierten Fahrzeugen mit einer modularen und dienstorientierten Architektur aus Hardware- und Softwarekomponenten. Die Forschungsziele von UNICAR*agil* sind zum einen die Entwicklung und Realisierung von vier hochautomatisierten modularen Fahrzeugen und zum anderen das Erforschen und Entwickeln von weiteren Methoden für den Sicherheitsnachweis.[15,16]

1.2 Konkretisierung der Aufgabenstellung

Im Projekt UNICAR*agil* wird das Ziel verfolgt, den Validierungs- und Verifikationsaufwand beim Sicherheitsnachweis für die Freigabe automatisierter Fahrfunktionen und -systeme zu reduzieren, indem die einzelnen Module, welche sich aus einer Gruppierung von verschiedenen Hardware- und Softwarekomponenten zusammensetzen, individuell abgesichert werden. Bislang ist die Forschung trotz einer Modularisierung an einer separaten Absicherung der einzelnen Module gescheitert, da die Komponenten weiterhin starke und meist unbekannte Abhängigkeiten untereinander aufweisen und diese größtenteils nur anhand von Gesamtsystemtests aufgedeckt werden. Demzufolge wird in dem Projekt UNICAR*agil* konsequent von Beginn an ein modulares System auf allen Architekturebenen aufgebaut, bei dem die Schnittstellen der Module eindeutig definiert sind.[13b]

Ziel dieser Masterthesis ist die Entwicklung eines Verfahrens, das die Abhängigkeiten zwischen den Modulen ermittelt und in modularen Testfällen berücksichtigt. Die Umsetzung der Testfälle erfolgt dabei in der bestehenden Simulationsumgebung. Die Ermittlung und Darstellung von Abhängigkeiten zwischen den Modulen stellen die notwendige Basis für die individuelle Erprobung und Absicherung der Module dar. Die anschließende Definition von Testfällen auf Grundlage der ermittelten Abhängigkeiten und die Durchführung dieser Testfälle dienen in erster Linie dazu, Abhängigkeiten aufzuzeigen, die mit dem Verfahren zuvor identifiziert wurden, obwohl sie vorher nicht offensichtlich waren. Bei der Durchführung der Testfälle werden zudem weitere mögliche unbekannte Abhängigkeiten aufgedeckt. Mit dem in dieser Masterthesis entwickelten Verfahren wird angestrebt, eine weitgehende Erfüllung bei der Identifizierung aller Abhängigkeiten für die erforderliche Berücksichtigung bei der individuellen Absicherung einzelner Module zu erreichen.

---

[12] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), S. 22.

[13] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 50 f; b: S. 36 f.

[14] RWTH Aachen: UNICARagil (2018).

[15] Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), S. 3.

[16] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 5 f.

1.3 Aufbau der Arbeit und Methodik des Vorgehens

Das in der Aufgabenstellung geforderte Verfahren wird in dieser Arbeit in zwei zu entwickelnde Verfahren aufgeteilt. Zuerst wird ein Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung automatisierter Fahrfunktionen entwickelt. Anschließend wird das zweite Verfahren zur Definition von Testfällen auf Grundlage der ermittelten Abhängigkeiten erarbeitet.

Zum Verständnis erfolgt zu Beginn in Kapitel 2 eine Einarbeitung in die ISO-Norm 26262. Neben dieser Grundlage werden das Gesamtkonzept und die Architekturen im Projekt UNICAR*agil* analysiert. Zusätzlich folgt eine Recherche zum Stand der Technik von Testmethoden in der Entwicklung im Automobilbereich.

Nach dieser Einarbeitung folgt in Kapitel 3 die Entwicklung des ersten Verfahrens, der Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung. Zu Beginn erfolgt die Erstellung einer Anforderungsliste für dieses Verfahren. Danach wird eine Literaturrecherche zu möglichen Verfahren durchgeführt. Die Verifizierung und Evaluierung der recherchierten Verfahren finden durch einen Vergleich mit der zuvor erstellten Anforderungsliste statt, sodass ein Verfahrenskonzept ausgewählt wird. Aufgrund von unterschiedlichen Anwendungsgebieten im Vergleich zu den in der Literatur verwendeten und den Anforderungen für das Projekt UNICAR*agil* sind Anpassungen an das ausgewählte Verfahren gegebenenfalls notwendig. Aus dem ausgewählten Verfahren und möglichen Anpassungen wird das zu entwickelnde Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung automatisierter Fahrfunktionen abgeleitet.

In Kapitel 4 wird das zuvor entwickelte Verfahren angewendet. Somit werden die Abhängigkeiten in der Architektur vom Projekt UNICAR*agil* ermittelt sowie dargestellt.

Die Entwicklung des zweiten Verfahrens, welches die Definition von Testfällen umfasst, ist in Kapitel 5 beschrieben. Für die Entwicklung dieses Verfahrens werden ebenfalls Anforderungen identifiziert und in Form einer Anforderungsliste dargestellt. Anschließend wird eine Recherche zu entsprechenden Verfahren in der Literatur durchgeführt. Die identifizierten Anforderungen dienen der Verifizierung und Evaluierung der Rechercheergebnisse, um anschließend ein Verfahren auszuwählen. Das ausgewählte Verfahren mit den gewünschten Anpassungen für die Anwendung ermöglicht die Ableitung eines Verfahrens zur Definition von Testfällen auf Grundlage der ermittelten Abhängigkeiten.

Nach der Entwicklung des zweiten Verfahrens werden in Kapitel 6 Testfälle auf Grundlage der zuvor ermittelten Abhängigkeiten definiert. Anschließend werden die Testfälle in der bestehenden Simulationsumgebung umgesetzt und durchgeführt.

Eine abschließende Diskussion der Ergebnisse in Kapitel 7 hinterfragt die beiden Verfahren und die jeweiligen Einflussfaktoren. Das Fazit und der Ausblick in Kapitel 8 schließen das Vorgehen dieser Arbeit ab und bieten Anknüpfungspunkte für nachfolgende Untersuchungen.

2 Grundlagen

In diesem Kapitel werden die benötigten Grundlagen für die Arbeit erklärt. Zunächst wird in Abschnitt 2.1 die ISO-Norm 26262 vorgestellt, welche die normative Grundlage im Bereich der funktionalen Sicherheit und Absicherung automatisierter Fahrfunktionen und -systeme bildet. Der in dieser Masterthesis liegende Fokus, das Projekt UNICAR*agil,* wird in Abschnitt 2.2 vorgestellt. Dazu werden das Gesamtkonzept, die verschiedenen Architektursichten sowie die Dienste und Module erläutert. Abschnitt 2.3 behandelt Testmethoden in der Entwicklung im Automobilbereich.

2.1 ISO 26262 Road vehicles – Functional safety

Vor der Freigabe zum Verkauf oder Gebrauch eines technischen Produkts besteht die Notwendigkeit, einen Nachweis zu führen, dass das Produkt ausreichend sicher ist. Das Teilgebiet der sicheren und korrekten Funktion eines technischen Produkts in dieser allgemeinen Sicherheitsbetrachtung wird als funktionale Sicherheit bezeichnet. Die Bewertung, ob ein Produkt sicher ist, wird anhand der Referenzgröße, dem tolerierbaren Grenzrisiko, welches durch den aktuellen Stand der Technik definiert ist, durchgeführt. Größtenteils wird der verbindliche Stand der Technik in Normen definiert, welche die zu erfüllenden Anforderungen bezüglich der Entwicklungsmethodik, der Dokumentation und der Produkteigenschaften enthalten.[17]

Die internationale Normenserie IEC 61508[18] umfasst die Anforderungen an die funktionale Sicherheit bei der Entwicklung von elektrischen, elektronischen und programmierbaren elektronischen Systemen, die eine Sicherheitsfunktion ausführen. Die ISO-Norm 26262 Road vehicles – Functional safety[19a] ist eine Anpassung der Normenserie IEC 61508 an den Automobilbereich. Mit der ISO-Norm 26262 wird die funktionale Sicherheit von sicherheitskritischen Komponenten und Systemen in Straßenfahrzeugen, die aus elektrischen, elektronischen und Softwarekomponenten bestehen, geregelt. Durch die sukzessive Integration von Fahrerassistenzfunktionen und -systemen und deren zunehmende technologische Komplexität steigt das Risiko von zufälligen und systematischen Hardwareausfällen. Dies verstärkt die Nachfrage nach funktionaler Sicherheit und die Notwendigkeit, den Nachweis zu erbringen, dass die Ziele der funktionalen Sicherheit erfüllt werden. Die ISO-Norm 26262 enthält Definitionen, Leitlinien und Entwicklungs- und Kontrollmethoden zur Minderung dieser Risiken und Gewährleistung der Integration von Sicherheitsanforderungen schon zu Beginn des Entwicklungsprozesses.[17,19b]

Die ISO-Norm 26262 beschreibt eine Vorgehensweise, welche sich am *V-Modell* der Produktentwicklung orientiert. Das *V-Modell* wurde ursprünglich als Entwicklungsstandard für IT-Systeme des Bundes entwickelt.[20a] Das Ziel dieses Vorgehensmodells ist die Minderung eines Fehlschlags sowie die Steigerung der Produktqualität während des Entwicklungsprozesses.[20b] Das Vorgehen in der ISO-

[17] Wilhelm, U. et al.: Funktionale Sicherheit und ISO 26262 (2015), S. 86.

[18] International Electrotechnical Commission: IEC 61508 (2010).

[19] International Organization for Standardization: ISO 26262 (2018), a: S. 6; b: S. 6 ff.

[20] Höhn, R.; Höppner, S.: Das V-Modell XT (2008), a: S. 2; b: S. 6.

Norm 26262 basiert auf einer zu Beginn des Produktlebenszyklus durchgeführten Gefahrenanalyse und einer darauffolgenden Quantifizierung der Risiken. Auf Basis der bestimmten Risiken werden Sicherheitsziele festgelegt, welche die Minderungsstrategien und Anforderungen den funktionalen Komponenten für den gesamten Produktlebenszyklus zuordnen.[21,22] Das Projekt UNICAR*agil* verfolgt einen neuen Ansatz beim Sicherheitskonzept, welcher über die Ansätze in der ISO-Norm 26262 hinausgeht.

2.2 UNICAR*agil*

Die Aufgabenstellung dieser Masterthesis stammt aus dem Projekt UNICAR*agil*. Für die Bearbeitung und besonders den ersten Teil, die Ermittlung und Darstellung der Abhängigkeiten, ist ein tiefes Systemverständnis erforderlich. Infolgedessen werden in diesem Abschnitt im Detail das Gesamtkonzept, die verwendeten Architektursichten und die Dienste und Module vorgestellt.

2.2.1 Gesamtkonzept

Im Folgenden wird das Gesamtkonzept vom Projekt UNICAR*agil* anhand Abbildung 2-1 zusammengefasst erläutert. Auf die nachfolgend erwähnten Komponenten wird in den Abschnitten 2.2.2 und 2.2.3 näher eingegangen.

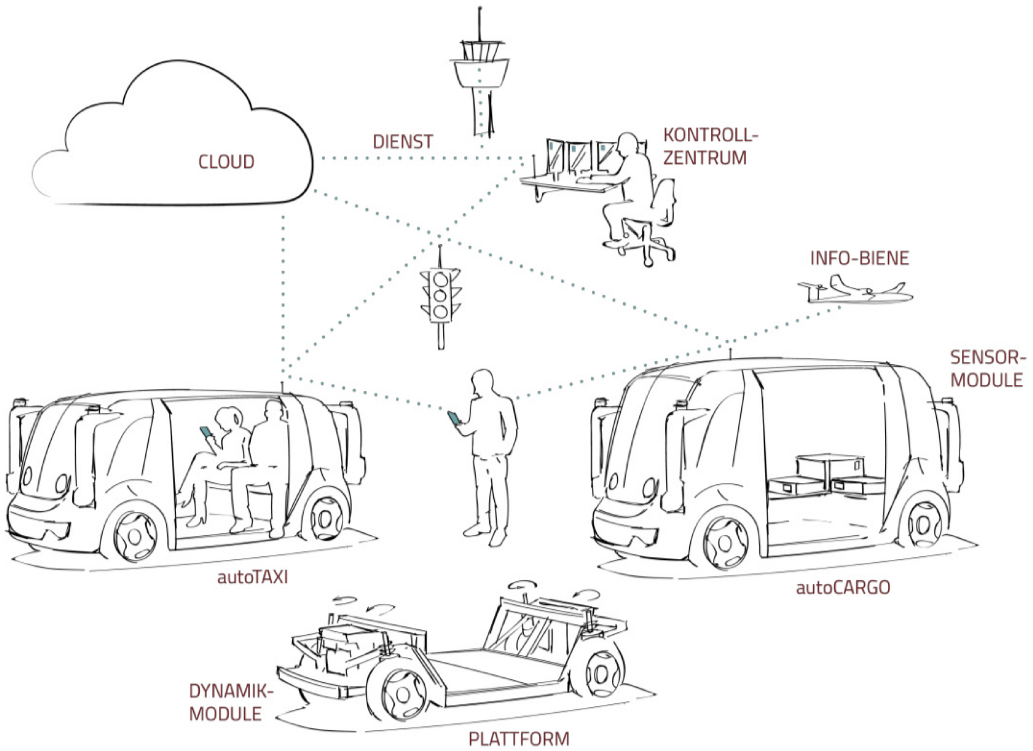

Abbildung 2-1: Gesamtkonzept von UNICAR*agil*[23]

[21] Wilhelm, U. et al.: Funktionale Sicherheit und ISO 26262 (2015), S. 86.

[22] Salay, R. et al.: ISO 26262 and Automotive Software (2018), S. 1 f.

[23] UNICARagil: UNICARagil news - 4. Ausgabe (2021), S. 25.

Das Zentrum für die Umsetzung der vier UNICAR*agil*-Fahrzeuge bilden die skalierbare Fahrplattform und die insgesamt vier Aufbaumodule. Die Aufbaumodule unterscheiden sich je nach Fahrzeugtyp. Die vier Typen der UNICAR*agil*-Fahrzeuge sind das auto*SHUTTLE*, das auto*CARGO*, das auto*TAXI* und das auto*ELF*. Vier Dynamikmodule erweitern die Plattform. Die Dynamikmodule stellen den Antrieb, die Bremse, die Lenkung und die Federung und Dämpfung bereit. Die für die Umweltwahrnehmung notwendigen Sensoren und Verarbeitungseinheiten werden in Form von disruptiven Sensormodulen an den vier Ecken der Aufbaumodule integriert.[24,25a]

Im Projekt UNICAR*agil* befindet sich im Betrieb der Fahrzeuge kein Fahrer. Da in manchen Situationen ein menschlicher Entscheider benötigt wird, ist eine Leitwarte, das Kontrollzentrum, ein weiteres Element des Gesamtkonzepts, das mit den fahrerlosen Fahrzeugen vernetzt ist und die Fahrzeugsteuerung übernehmen kann.[24,25b]

Neben der Vernetzung mit der Leitwarte sind die Fahrzeuge auch untereinander mit einer Cloud vernetzt, um auf Informationen von verschiedenen Fahrzeugen, aber auch von Infrastrukturelementen oder unbemannten Luftfahrzeugen, zuzugreifen. Ein solches unbemanntes Luftfahrzeug wurde im Rahmen des Projekts UNICAR*agil* entwickelt. Die *Info-Biene* ist eine automatisierte, mit Sensoren ausgestattete Drohne.[25c] Diese Drohne dient zur luftgestützten Verkehrsüberwachung und sammelt somit beispielsweise Informationen nach einem Unfall über blockierte Fahrstreifen oder über freie Parkplätze. Somit wird das externe Umfeldmodell erweitert und gibt Aufschluss über den Zustand des Straßennetzes und der Umgebung, was relevant für die Routen- und Verhaltensplanung automatisierter Fahrzeuge ist.[25d]

2.2.2 Architektursichten

Die Verwendung von verschiedenen Architektursichten in der Softwareindustrie ist bereits ein Bestandteil bei der systematischen Modellierung von komplexen Systemen. Im Automobilbereich herrscht bislang kein Konsens über einen Standardarchitekturrahmen.[26]

Die Notwendigkeit eines ganzheitlichen Ansatzes verstärkt sich aufgrund der zunehmenden Komplexität und der folglich erschwerten Gewährleistung der Sicherheit automatisierter Fahrfunktionen und -systeme. Mehrere Sichtweisen auf die Systemarchitektur dienen als möglicher Ansatz für ein Sicherheitskonzept von automatisierten Fahrzeugen.[27a]

Aufgrund keiner einheitlichen Übereinstimmung sind in der Literatur diverse Ansätze für verschiedene Architektursichten in einem Architekturrahmen vorhanden. Bagschik et al.[27b] erweitern das System mit der funktionalen Sicht, der Softwaresicht und der Hardwaresicht, um die *Capability-Sicht*. Die Sicherheitsanforderungen aus der Verhaltenssicherheit werden dieser Sicht zugeordnet.[27c] Bei

[24] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 5 f.

[25] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 16 f; b: S. 17; c: S. 18; d: S. 30 f.

[26] Dajsuren, Y. et al.: Correspondence Rules for Automotive Architecture (2014), S. 1.

[27] Bagschik, G. et al.: Architecture Framework for Automated Vehicles (2018), a: S. 1; b: S. 2; c: S. 7.

Dajsuren et al.[28] weist der Architekturrahmen neben der funktionalen Sicht, der Softwaresicht und der Hardwaresicht, fünf weitere Architektursichten auf. Dazu gehören die *Use-Case-Sicht* sowie die *Feature-Sicht,* welche Variationen der funktionalen Sicht sind und somit für die Entwicklung der funktionalen Architektur verwendbar sind. Die dritte eingeführte Sicht ist die *Allocation-Sicht*. Bei der Kombination der Software- und Hardwaresicht wird diese bereits berücksichtigt. Die letzten beiden Sichten, die *Topology-Sicht* und die *Timing-Sicht*, sind spezifischere Sichten für die Software- und Hardwaresicht.[28]

Die im Projekt-UNICAR*agil* verwendeten Architektursichten zur Verifizierung und Validierung des automatisierten Fahrens sind die geometrische Architektur, die Elektrik/Elektronik-Architektur (E/E-Architektur), die Softwarearchitektur, die funktionale Architektur und die Verhaltenssicherheit.[29a] Die E/E-Architektur, die Softwarearchitektur und die funktionale Architektur werden im Folgenden näher beschrieben, da diese entscheidend für die Automation sind.

2.2.2.1 E/E-Architektur

Für die Umsetzung der Ziele im Projekt UNICAR*agil* ist eine neue E/E-Architektur notwendig. Die in Fahrzeugen derzeit etablierten E/E-Architekturen basieren auf einer großen Anzahl von elektronischen Steuergeräten, dezentraler Datenspeicherung und begrenzter Datenkommunikation. Dies hat einen enormen Aufwand bei Änderungen an solchen Systemen für die Verifikation und Validierung zur Folge. Zudem wird eine Erweiterung und Gewährleistung der Ausfallsicherheit des Systems erschwert. Die neue E/E-Architektur orientiert sich bei der Terminologie am menschlichen Nervensystem und setzt sich aus den Ebenen *Großhirn*, *Stammhirn* und *Rückenmark* zusammen.[29b] Der vereinfachte Aufbau der E/E-Architektur ist in Abbildung 2-2 dargestellt.

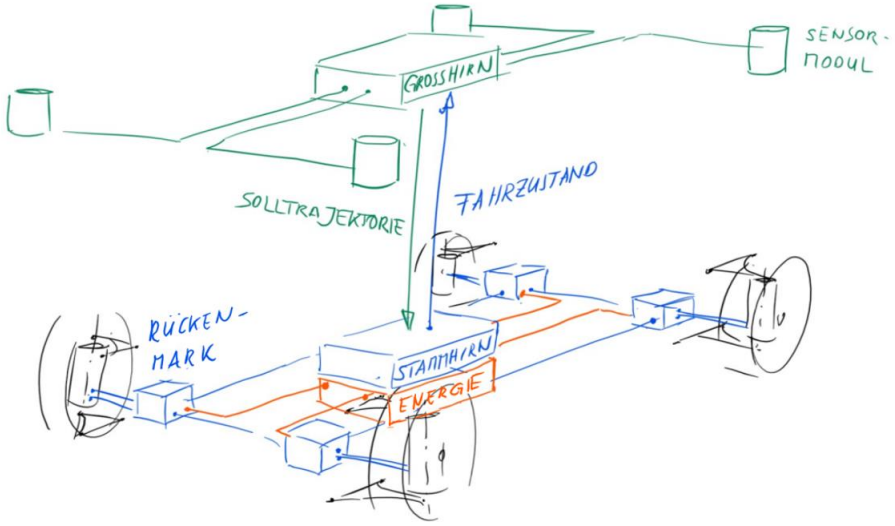

Abbildung 2-2: Schematische Darstellung der E/E-Architektur[30]

[28] Dajsuren, Y. et al.: Correspondence Rules for Automotive Architecture (2014), S. 3.

[29] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), a: S. 7; b: S. 4.

[30] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 6.

Auf der obersten Ebene, dem *Großhirn*, werden die Umgebungsdaten durch die vier Sensormodule erfasst. Neben der Umfeldwahrnehmung erfolgt im *Großhirn* die Fusion der Umfeldinformationen zu einem Umfeldmodell. Die anschließende Verhaltens- und Trajektorienplanung berechnet auf Basis des Umfeldmodells und Informationen aus einer digitalen Karte eine Solltrajektorie. Die berechneten Solltrajektorien werden an das darunterliegende *Stammhirn* weitergeleitet. Das *Stammhirn* übernimmt die Aufgaben der Selbstwahrnehmung und der Fahrdynamik- und Trajektorienregelung des Fahrzeugs und ist dazu mit einer Fahrdynamikzustandsschätzung, welche den Ist-Zustand des Fahrzeugs erfasst, verbunden. Zusätzlich wird im *Stammhirn* die Rückfallebene, das *Sichere Anhalten*, ausgeführt, welche das automatisierte Fahrzeug in einen risikominimalen Zustand überführt. Das *Stammhirn* ist in der Lage, mit dem Fahrzustand und dem vom *Großhirn* durch die Solltrajektorie beschriebenen Sollzustand Stellgrößen für die nächste Ebene, das *Rückenmark*, bereitzustellen. Mit diesen Stellgrößen steuert das *Rückenmark* die Aktorik der vier Dynamikmodule an. Das *Rückenmark* besitzt eine eigene Energieversorgung, die bei Defekten oder Ausfällen des *Stammhirns* die Ausfallsicherheit des Gesamtsystems absichert. Das *Großhirn*, das *Stammhirn* und das *Rückenmark* werden auch als Hardwareebenen bezeichnet und stellen somit die Hardwarearchitektur dar.[31,32a,33,34a]

2.2.2.2 Softwarearchitektur

Eine Softwarearchitektur beschreibt, wie der Aufbau von Computerprogrammen auf den Steuergeräten eines technischen Systems aussieht und wie die Kommunikation zwischen den Computerprogrammen abläuft. Bei den in heutigen Fahrzeugen existierenden Softwarearchitekturen sind bereits im Entwicklungsprozess die Abhängigkeiten zwischen den Softwarekomponenten festgeschrieben und somit sind die Updatefähigkeit und Wiederverwendbarkeit dieser nicht gegeben. Dies erschwert die Realisierung der immer kürzeren Entwicklungs- und Lebenszyklen im Automobilbereich und ist problematisch bei der Entwicklung automatisierter Fahrzeuge.[32b,34b]

In der im Projekt UNICAR*agil* verwendeten dienstbasierten Softwarearchitektur, auch als *Automotive Service-Oriented Software Architecture* (*ASOA*) bezeichnet, sind die Softwarekomponenten als lose gekoppelte Dienste implementiert. Die *ASOA* ermöglicht eine dynamische Rekonfiguration des Systems und bietet Möglichkeiten für die Integration neuer Softwarekomponenten. Zudem ist die *ASOA* für den Datentransport zwischen den Diensten verantwortlich. Mit *ASOA* sind die Module fähig, die erforderlichen Funktionen als Dienste anzubieten und abzufragen.[34c]

Die modularen Dienste werden in Abhängigkeit des Betriebsmodus durch die zentrale Komponente in der Architektur, den Orchestrator, dynamisch verschaltet, sodass eine optimale Realisierung des gewünschten Fahrverhaltens vorliegt. Dazu regelt der Orchestrator den Verbindungsaufbau und -abbau sowie die Steuerung über den Zustand der Dienste. Die Integration und Kopplung von Diensten ist beispielhaft für zwei Betriebsmodi, dem automatisierten Betrieb und Leitwartenbetrieb, in

[31] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 6.

[32] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 61; b: S. 74.

[33] Ackermann, S.; Winner, H.: Systemarchitektur beim Sicheren Anhalten (2020), S. 2 f.

[34] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), a: S. 4 f; b: S. 13; c. S. 14.

Abbildung 2-3 schematisch dargestellt. An diesem Beispiel wird deutlich, dass die Dienste im Fahrzeug im automatisierten Betrieb anders zusammengesetzt sind als im Leitwartenbetrieb, wobei dies durch den Orchestrator und nicht durch die Dienste gesteuert wird. Die Dienste sind aufgrund einer gemeinsamen Ethernet-Verbindung zwischen den Steuergeräten in der Lage, über diese miteinander zu interagieren.[35a] Neben diesen beiden Betriebsmodi im Projekt UNICAR*agil* existieren noch der Modus *Sicheres Anhalten* und der manuelle Betrieb.[36a]

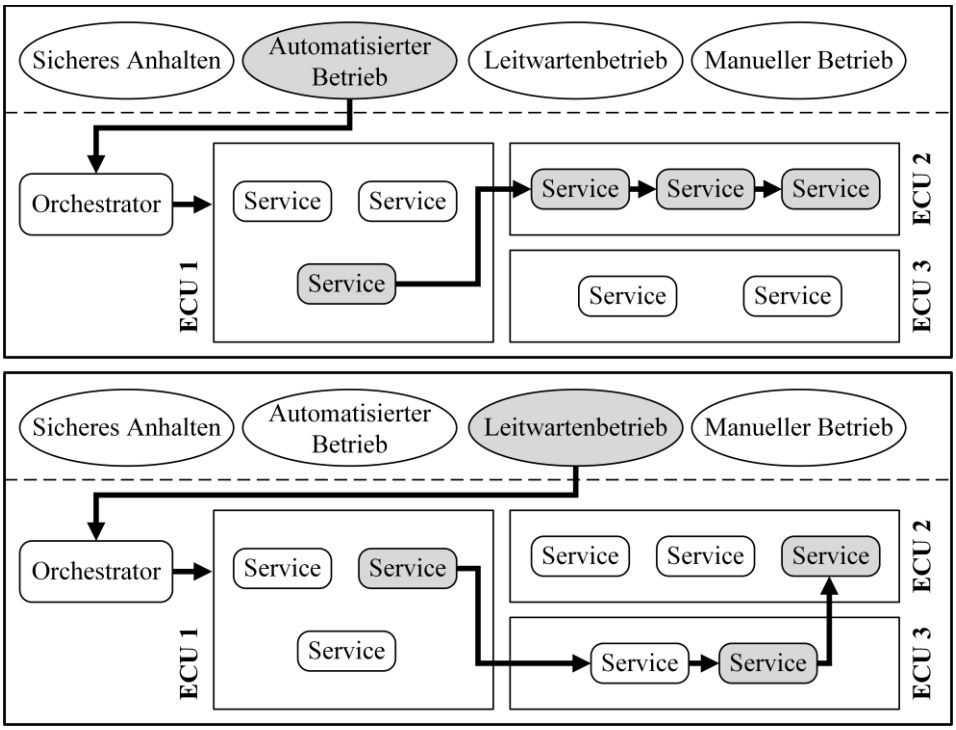

Abbildung 2-3: Konzept der *ASOA* bei unterschiedlichen Betriebsmodi nach Woopen et al.[37]

Aufgrund der dynamischen und adaptiven Systemintegration während der Laufzeit und der konsequenten Modularisierung der Dienste ergibt sich die Möglichkeit, Dienste nachzuladen, zu aktualisieren und neue Funktionen unter Wiederverwendung bestehender Dienste umzusetzen.[36b]

Auf dem *Großhirn*, dem *Stammhirn* und dem *Rückenmark* wird die *ASOA* verwendet. Durch die Einbettung in die drei Hardwareebenen werden eine ebenen- und hardwareübergreifende Kommunikation und die flexible Integration einzelner Komponenten ermöglicht.[35b,38]

2.2.2.3 Funktionale Architektur

Die funktionale Architektur in Fahrzeugen beschreibt, welche Funktionen konkret durch die Hardware und Software für die Erfüllung der Fahraufgabe abgebildet werden. Die funktionale Architektur im Projekt UNICAR*agil* ist in Abbildung 2-4 anhand eines *A-Modells* veranschaulicht und zeigt die

[35] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), a: S. 14; b: S. 6.

[36] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 38; b: S. 72 ff.

[37] Eigene Darstellung nach Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), S. 15.

[38] Ackermann, S.; Winner, H.: Systemarchitektur beim Sicheren Anhalten (2020), S. 3.

Verarbeitungsschritte und das Zusammenwirken der notwendigen Funktionalitäten für die Ausführung der Fahraufgabe.[39,40a]

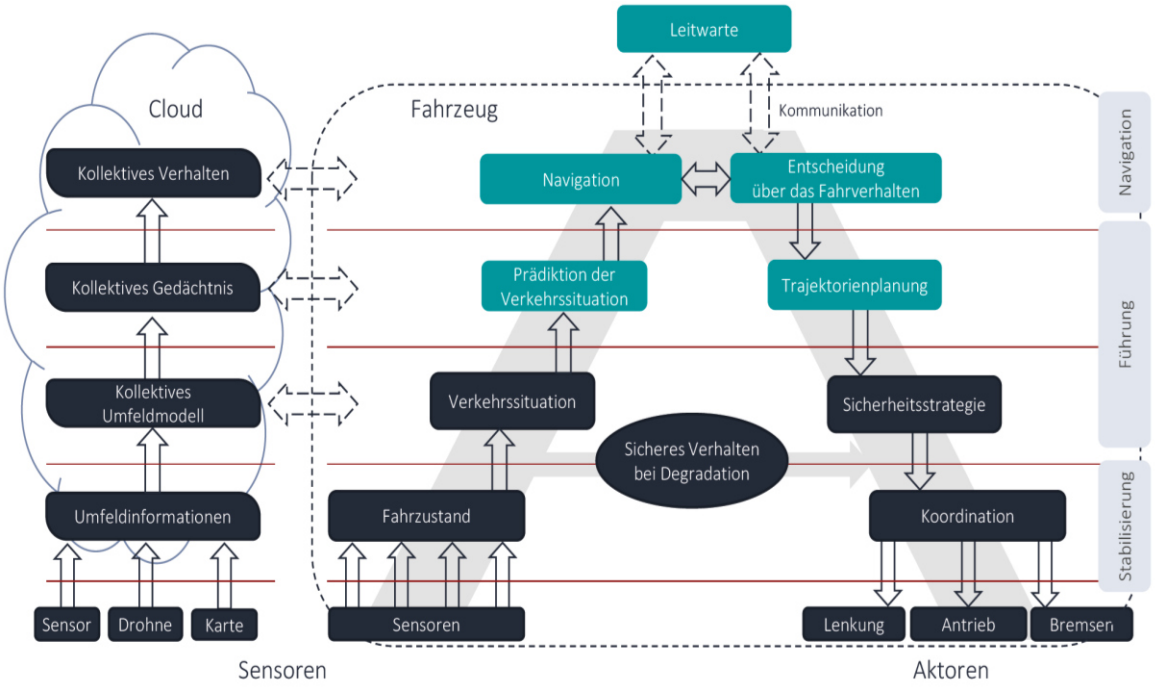

Abbildung 2-4: *A-Modell* der funktionalen Architektur in UNICAR*agil*[40b]

Die Verarbeitungsschritte in den UNICAR*agil*-Fahrzeugen, beginnend bei der Umfeldwahrnehmung durch die Sensoren bis hin zur Ansteuerung der Aktorik, sind von links nach rechts abgebildet. Die einzelnen Funktionen sind auf mehrere Ebenen aufgeteilt. Die Sensorik und die Aktorik bilden die unterste Ebene der auszuführenden Funktionen. Die Funktionen in der darüberliegenden Stabilisierungsebene gewährleisten ein sicheres Verhalten des Fahrzeugs auch bei Beeinträchtigungen oder einem Ausfall von Komponenten. In der darüberliegenden Führungsebene befinden sich die Funktionen für die Fahrzeugführung, wie beispielsweise die Trajektorienplanung. Die letzte Ebene fasst die Funktionen für die Navigation zusammen.[40a] Zusätzlich zu diesen Funktionen unterstützen externe Funktionen, im vorliegenden Fall die Cloud und die Leitwarte, das Fahrzeug.[39]

2.2.3 Dienste und Module

In diesem Abschnitt werden die bereits erwähnten Dienste und Module im Detail erläutert. Zur Übersicht der Wirkketten zwischen den Diensten bzw. Modulen dient die folgende Abbildung 2-5, welche die Systemarchitektur von UNICAR*agil* mit den relevanten Diensten und die Einordnung in die drei Ebenen *Großhirn*, *Stammhirn* und *Rückenmark* zeigt. Zu der Abbildung 2-5 ist anzumerken, dass die Sensormodule, die Cloud und die Leitwarte in dieser Abbildung ausgeblendet sind.

[39] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), S. 5.

[40] UNICARagil: UNICARagil news - 4. Ausgabe (2021), a: S. 27; b: S. 26.

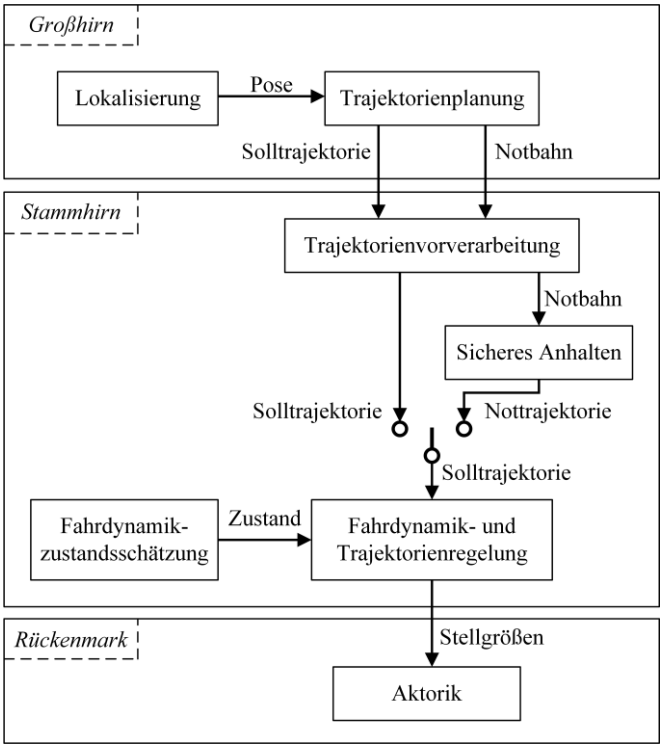

Abbildung 2-5: Systemarchitektur in UNICAR*agil* nach Gottschalg et al.[41]

2.2.3.1 Sensormodule

An den vier Ecken der UNICAR*agil*-Fahrzeuge sind identisch ausgestattete Sensormodule mit einem horizontalen Sichtbereich von ca. 270° je Sensormodul verbaut. Die vier Sensormodule bestehen aus *Radio detection and ranging* Sensorik (Radar-Sensorik), *Light detection and ranging* Sensorik (Lidar-Sensorik), monokularer und stereoskopischer Videosensorik sowie einer zugehörigen Recheneinheit zur Sensordatenverarbeitung. Es sind Redundanzen für die Wahrnehmungsaufgabe durch die unterschiedlichen Sensortypen und überlappenden Sichtbereiche gegeben. In jedem einzelnen Modul werden durch die Recheneinheit die Sensordaten verarbeitet und ein individuelles Umfeldmodell berechnet. Ein solches Umfeldmodell besteht aus einer Belegungsrasterkarte, einer sogenannte *Grid-Map*, welche in quadratische Zellen eingeteilt ist. Für die Zellen in der *Grid-Map* ist jeweils die Wahrscheinlichkeit der Belegung durch ein Objekt angegeben. Neben der *Grid-Map* besteht das Umfeldmodell auch aus einer Objektliste mit den Verkehrsteilnehmern, die sich in der Umgebung des Fahrzeugs befinden. Die Einträge in dieser Liste werden durch weitere Informationen zum Objekttyp, zu den Objektabmessungen, zur Position und zum dynamischen Bewegungszustand des Objekts ergänzt. Zusätzlich zur Berechnung der vier Umfeldmodelle findet in allen Sensormodulen eine videobasierte Lokalisierung des Fahrzeugs statt. Die vier Umfeldmodelle werden an das *Großhirn* weitergeleitet, welches dann die Modelle zu einem Gesamtumgebungsmodell fusioniert.[42,43,44]

[41] Eigene Darstellung nach Gottschalg, G. et al.: Inkonsistente Lokalisierungsdaten in modularen Systemen (2020), S. 3.

[42] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 9.

[43] UNICARagil: UNICARagil news - Halbzeitevent (2020), S. 81 ff.

[44] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), S. 24.

2.2.3.2 Cloud-Dienste

Die Cloud-Dienste erweitern die Wahrnehmung und sorgen für ein sichereres und effizienteres Verhalten der UNICAR*agil*-Fahrzeuge. Der erste Dienst in der Cloud beschreibt das Sammeln von Umfeldinformationen von verschiedenen Fahrzeugen, von Drohnen und aus Karten. Im nächsten Dienst, dem kollektiven Umfeldmodell, erfolgt die Kombination der gesammelten Daten zu Umfeldmodellen, Fahrzeugzuständen und zur Verhaltensplanung. Z.B. sind Objekte bei einigen Fahrzeugen verdeckt und bei anderen nicht. Mit dem kollektiven Umfeldmodell lassen sich potenzielle Gefahren frühzeitig erkennen, sodass diese in die Verhaltensplanung einbezogen werden. Der nachfolgende Dienst ist das kollektive Gedächtnis und dient dazu, aus den gesammelten Daten zu lernen, indem relevante Szenarien erkannt und Daten gesammelt werden. So wird beispielsweise die Erkennung von Objekten oder die Vorhersage von zukünftigem Verhalten erlernt und verbessert. Der letzte Dienst, das kollektive Verhalten, nutzt die Daten aus dem kollektiven Umfeldmodell und das erlernte Wissen aus dem kollektiven Gedächtnis, um passende Trajektorien für kritische Situationen zu entwickeln. Anzumerken ist, dass diese Trajektorien im Konzept weiterverwendet werden, jedoch dies noch nicht umgesetzt wurde. Auch das Sammeln der gefahrenen Solltrajektorien der Fahrzeuge ist im Konzept angedacht. Dies ist allerdings ebenfalls zum jetzigen Zeitpunkt nicht umgesetzt.[45,46a] Die Kommunikation zwischen Fahrzeugen und Cloud-Diensten läuft über das kommerzielle Mobilfunknetz.[47]

2.2.3.3 Leitwarte

Die Leitwarte ist ein externes Modul im Projekt UNICAR*agil*, welches ebenfalls mit dem Fahrzeug über das kommerzielle Mobilfunknetz verbunden ist. Die Leitwarte übernimmt Serviceaufgaben, wie z.B. die Kommunikation mit den Fahrgästen in Notfällen, bis hin zur Übernahme des fahrerlosen Fahrzeugs durch Servicepersonal in der Leitwarte. Bei einer gestörten Umgebungswahrnehmung oder einer unklaren Situationsinterpretation und generell in allen Situationen, in denen eine sichere Navigation durch die automatisierten Funktionen nicht sichergestellt ist, wird das Fahrzeug verlangsamt und kommt zum Stillstand. Danach greift aus der Ferne ein menschlicher Operator per Teleoperation ein. Wenn die automatisierten Funktionen, z.B. nach einer vorübergehenden Beeinträchtigung, wieder fähig sind, die Fahraufgabe zu erfüllen, übergibt der Operator die Führung des Fahrzeugs zurück an das System.[46b,48a,49] Durch die Erweiterung des Konzepts durch die Leitwarte wird die Zuverlässigkeit und Einsatzbereitschaft der UNICAR*agil*-Fahrzeuge erhöht.[48b]

---

[45] Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), S. 1552 f.

[46] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 100 f; b: S. 17.

[47] UNICARagil: UNICARagil news - 4. Ausgabe (2021), S. 25.

[48] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), a: S. 26 f; b: S. 26.

[49] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 5 f.

2.2.3.4 Lokalisierung

Die Lokalisierung ist der Dienst für die Posenbestimmung des automatisierten Fahrzeugs für die Trajektorienplanung im *Großhirn*. Die Bestimmung der Position und Orientierung ist erforderlich, damit das Fahrzeug sich auf der Karte wiederfindet. Die dafür notwendige Messung der Position und Orientierung erfolgt in der Fahrdynamikzustandsschätzung, welche im nächsten Abschnitt im Detail erläutert wird.[50]

2.2.3.5 Fahrdynamikzustandsschätzung

Die Aufgaben der Fahrdynamikzustandsschätzung (FZS) sind die Bestimmung und Bereitstellung des Ist-Zustands über die *ASOA*, welcher sich aus Position, Geschwindigkeit, Beschleunigung, Ausrichtung, dem Schwimmwinkel und weiteren Parametern zusammensetzt, sowie die Ausgabe von Qualitätskriterien zur Genauigkeit und Integrität. Die Integrität beschreibt die Zuverlässigkeit und Vertrauenswürdigkeit der Daten. Die Position und Orientierung werden anderen Diensten, beispielsweise der Lokalisierung, bereitgestellt und der vollständige Ist-Zustand wird z.B. der Fahrdynamik- und Trajektorienregelung zur Verfügung gestellt.[51a]

Die notwendigen Messdaten erhält die FZS von einem globalen Navigationssatellitensystem-Empfänger (*Global Navigation Satellite System* (GNSS)), von zwei inertialen Messeinheiten (*Inertial Measurement Unit* (IMU)) und von einer Odometrie-Sensorik an jedem Rad. Die Messdaten aus den Inertialsensoren geben Beschleunigungen und Drehraten aus. Die Odometrie-Sensorik übermittelt Drehraten und Lenkwinkel der Räder. Die Kombination dieser Messdaten mit den Satellitendaten ermöglicht eine genaue Positionsbestimmung. Die Fusionierung der Sensordaten erfolgt durch Filter, die auf präzisen Lokalisierungslagorithmen basieren.[50,51b,52a] Die Struktur der Sensordatenfusion der FZS ist in Abbildung 2-6 dargestellt.

Im Projekt UNICAR*agil* werden mehrere Datenfusionsfilter entwickelt, die aus verschiedenen Algorithmen bestehen und unterschiedliche Teilmengen der verfügbaren Sensordaten nutzen. Der Filter 1 wird an der TU Darmstadt entwickelt und verwendet die Daten von der ersten IMU, die Odometriedaten aller vier Räder und das GNSS-Signal. Die Filter 2a und 2b werden von zwei unabhängigen Entwicklergruppen erarbeitet und verwenden jeweils eine IMU, die Odometrie-Sensorik an einem Rad und das GNSS-Signal.[53] Mit der unabhängigen Entwicklung wird die Wahrscheinlichkeit des gleichzeitigen Auftretens von verborgenen Designfehlern minimiert.[52b] Anzumerken ist, dass während dieser Arbeit nur die Filter 1 und 2b vorhanden sind und beide auch noch nicht die Odometrie-Sensorik nutzen. Der nachfolgende Abstimmungsalgorithmus, der sogenannte *Voter*, entscheidet aufgrund der Qualitäts- und Integritätsinformationen der Filter über die Ausgabegrößen.[52b,54]

---

[50] UNICARagil: UNICARagil news - 4. Ausgabe (2021), S. 13.

[51] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 94 f; b: S. 80.

[52] Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), a: S. 1541 f; b: S. 1536.

[53] Gottschalg, G. et al.: Data Fusion of Redundant Fusion Filters (2021), S. 3760 f.

[54] Gottschalg, G. et al.: Concept for Sensor Fusion Algorithms (2020), S. 1 f.

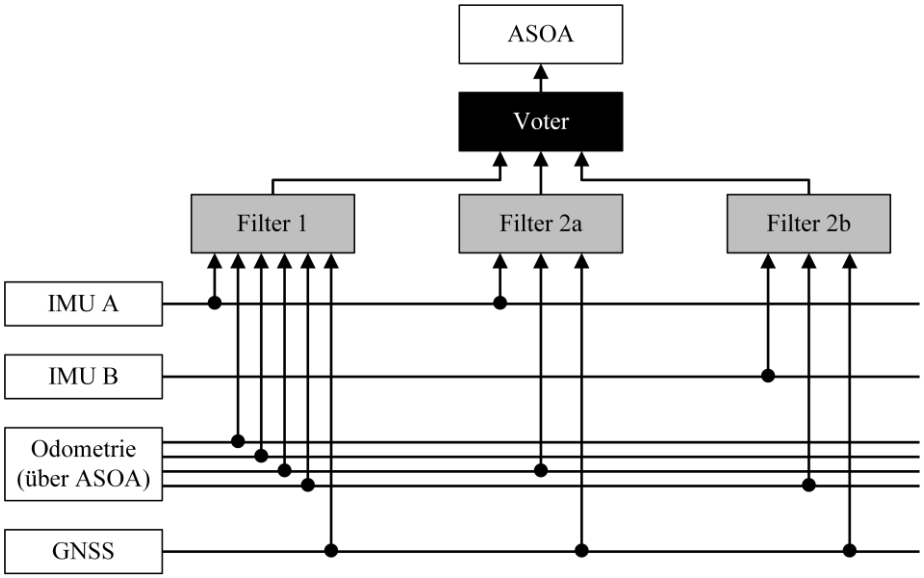

Abbildung 2-6: Struktur der Sensordatenfusion der FZS in UNICAR*agil* nach Buchholz et al.[55]

2.2.3.6 Verhaltens- und Trajektorienplanung

Der Verhaltens- und Trajektorienplaner (VTP) plant aufbauend auf dem Fahrzeugumgebungsmodell und der Objektliste, der aktuellen Pose, der Route sowie den Karteninformationen für den aktuellen Abschnitt der Route das Manöver und die zugehörige Trajektorie. Eine Trajektorie ist die für die nächsten Sekunden geplante Weg-Zeit-Kurve der Fahrzeugbewegung.[56] Neben der sogenannten Solltrajektorie wird auch eine Notbahn berechnet. Die Notbahn ist die Bahn, die in einen sicheren Stillstand führt. Auf die Notbahn wird im nachfolgenden Abschnitt eingegangen. Die Solltrajektorie und die Notbahn werden direkt nach Ausgabe an die Trajektorienvorverarbeitung (TVV) übermittelt. Dort erfolgt die Transformation der Sollpositionen der Trajektorie bzw. der Notbahn in ein lokales Koordinatensystem.[57] Die Systemstruktur des VTP setzt sich aus der Verhaltensgenerierung, der Vorverarbeitung und der abschließenden Planung und Berechnung zusammen.[57] Die Struktur ist in Abbildung 2-7 dargestellt und die drei Elemente im VTP werden im Folgenden näher erläutert.

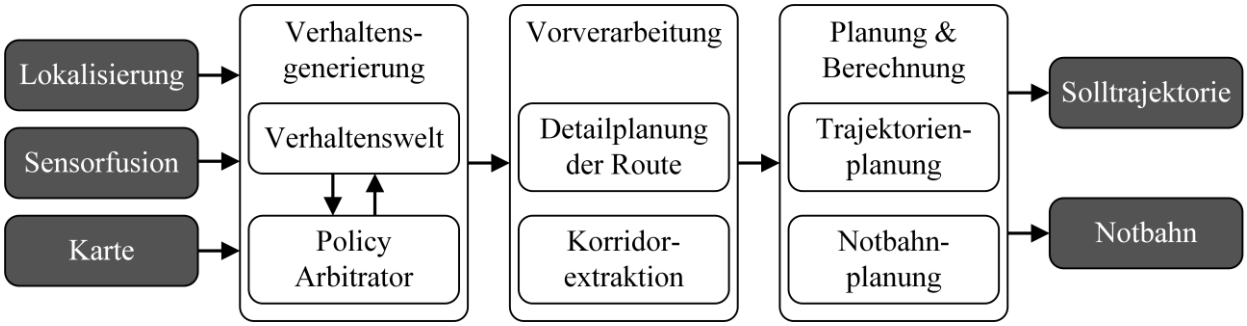

Abbildung 2-7: Systemstruktur des VTP nach Kamran et al.[58]

[55] Eigene Darstellung nach Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), S. 1536.

[56] UNICARagil: UNICARagil news - 4. Ausgabe (2021), S. 15.

[57] UNICARagil: UNICARagil news - Halbzeitevent (2020), S. 92 f.

[58] Eigene Darstellung nach Kamran, D.; Wang, L.: UNICARagil - Verhaltens- und Trajektorienplanung (2020), S. 92.

In der Verhaltensgenerierung wird die Situation analysiert und abhängig von der aktuellen Situation eine passende Verhaltensweise ausgewählt. Für die Auswahl wird der *Policy-Arbitrator* hinzugezogen. In diesem *Policy-Arbitrator* sind für alle Situationen spezifische Teilverhaltensregeln mit Aktivierungsbedingungen, die nur bei diesem Teilverhalten gelten, implementiert. Anschließend plant die Vorverarbeitung im Detail anhand eines befahrbaren Korridors eine Route, die die Kollisionsobjekte und Verkehrsregeln berücksichtigt. Im letzten Schritt erfolgt die Planung und Berechnung der Solltrajektorie und der Notbahn.[59a,60]

2.2.3.7 Sicheres Anhalten

Nach der Norm SAE J3016[61] sind bei automatisierten Fahrzeugen ab Automatisierungsstufe 3 Rückfallebenen in Form von maschinellen Systemen Voraussetzung, da ab Stufe 3 die dynamische Fahraufgabe vollständig vom System übernommen wird.[62a] Im Rahmen des Projekts UNICAR*agil* ist eine maschinelle Rückfallebene entwickelt worden, welche bei eingeschränkter Funktionalität entweder durch einen Ausfall oder durch funktionale Degradation das Fahrzeug in einen risikominimalen Zustand überführt. Diese Rückfallebene der fahrerlosen Fahrzeuge im Projekt UNICAR*agil* wird als *Sicheres Anhalten* bezeichnet. Bei Ausfall relevanter Fahrzeugkomponenten und der Nichterfüllung der Fahraufgabe wird diese Rückfallebene aktiviert. Das *Sichere Anhalten* ist ein risikominimales Anhaltemanöver, welches auf einer vorausberechneten Notbahn vom Verhaltens- und Trajektorienplaner basiert.[59b,62b] Eine schematische Skizze des risikominimalen Anhaltemanövers ist in Abbildung 2-8 dargestellt.

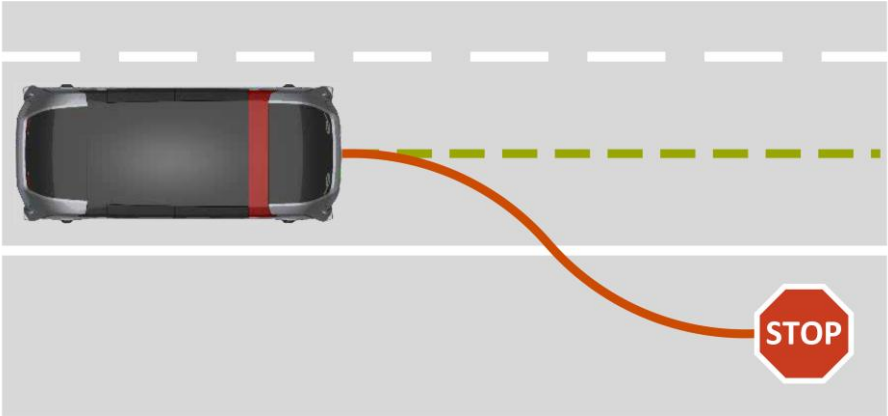

Abbildung 2-8: UNICAR*agil*-Fahrzeug mit einem örtlichen Solltrajektorienverlauf (grün - gestrichelt) und einem örtlichen Notbahnverlauf zu einem risikominimalen Ort (rot - durchgezogen) nach Ackermann et al.[63]

Die Planungsebene berechnet eine Notbahn und keine Nottrajektorie. Da bei auftretenden Kollisionsobjekten die Notwendigkeit besteht, die Fahrzeuggeschwindigkeit anzupassen, wird eine Nottrajektorie, welche den zeitlichen Verlauf der Fahrzeugposen beschreibt, ungültig. Die Notbahn umfasst

[59] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 93; b: S. 44.

[60] Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020), S. 1544 f.

[61] SAE International: SAE J3016: Taxonomy and Definitions (2021).

[62] Ackermann, S.; Winner, H.: Systemarchitektur beim Sicheren Anhalten (2020), a: S. 1; b: S. 5 ff.

[63] Eigene Darstellung nach Ackermann, S.; Winner, H.: UNICARagil - Sicheres Anhalten (2020), S. 44.

den örtlichen Verlauf der Sollposen bis zu einem risikominimalen Ort, der sich möglichst außerhalb des fließenden Verkehrs befindet, und ein ortsabhängiges Geschwindigkeitsprofil, welches zu Beginn bei der Geschwindigkeitsanpassung als Vorgabe für die Auslegungsgeschwindigkeit und danach für die Maximalgeschwindigkeit dient.[64,65a]

Für die Ausführung des risikominimalen Anhaltemanövers wird eine unabhängige Umfeldsensorik, die Plattformsensorik oder auch sekundäre Umfelderfassung, verwendet, welche die abgesicherte Fahrt durch einen vorgegebenen Korridor, jedoch nicht die Bahnplanung, ermöglicht. Die Plattformsensorik umfasst 24 Ultraschallsensoren, vier Kameras und zwei Radarsensoren. Die nachfolgende Überprüfung des Notbahnverlaufs auf Kollisionsobjekte wird in der Umfeldüberwachung durchgeführt. Dazu wird auf Grundlage der Daten der sekundären Umfelderfassung ein Umfeldmodell generiert und auf Basis des örtlichen Verlaufs der Notbahn und dem geplanten Geschwindigkeitsprofil die zukünftige Fahrzeugpose im Voraus berechnet. Anschließend wird mit den Fahrzeugabmessungen, einem Sicherheitsabstand und dem zeitlichen Verlauf der berechneten Fahrzeugposen ein örtlicher und zeitlicher Korridor beschrieben, der als *Fahrschlauch* bezeichnet wird. Mit den von der Umfelderfassung detektierten Objekten und deren Prädiktionen wird geprüft, ob örtliche und auch zeitliche Überlappungen mit dem berechneten *Fahrschlauch* und folglich Kollisionen vorliegen. Bei Detektion einer Kollision wird ein örtliches Geschwindigkeitsprofil geplant, welches die Kollision vermeidet. Nach der Umfeldüberwachung folgt die Trajektoriengenerierung als weiterer Baustein des S*icheren Anhaltens*. Die Trajektoriengenerierung berechnet die kollisionsfreien und geschwindigkeitsangepassten Solltrajektorien auf Basis der vorausberechneten Notbahn, dem dazugehörigen Geschwindigkeitsprofil und den detektierten Objekten. Diese berechnete Solltrajektorie weist den gleichen Aufbau wie die Solltrajektorie auf, die im *Großhirn* im Trajektorienplaner berechnet wird. Dies ermöglicht den Einsatz einer gemeinsamen Regelung.[64,65a]

Die Aktivierung des *Sicheren Anhaltens* erfolgt auf Grundlage der Selbstwahrnehmung, wenn diese eine Umschaltung auf die Rückfallebene für notwendig erachtet, sobald ein sicherer automatisierter Betrieb nicht mehr zu gewährleisten ist. Dies ist z.B. der Fall, wenn die Verbindung zum *Großhirn* abbricht und somit die primäre Umfelderfassung und die Verhaltens- und Trajektorienplanung wegfallen. Die Selbstwahrnehmung ist ein Teil des Sicherheitskonzepts der Fahrzeuge im Projekt UNICAR*agil*. Diese erstellt ein Gesamtbild der Fahrzeuggesundheit und überwacht somit das Fahrzeugverhalten.[65b] Beim Umschalten wird die Latenz minimiert, indem das *Sichere Anhalten* im „Hot-Standby"-Betrieb läuft.[65c]

2.2.3.8 Fahrdynamik- und Trajektorienregelung

In der Fahrdynamik- und Trajektorienregelung (FTR) werden die entweder vom *Großhirn*, vom *Sicheren Anhalten* oder von der Leitwarte eingehenden Solltrajektorien in Stellgrößen für die Aktorik umgesetzt. Der Trajektorienplaner und die FTR greifen auf Posen zurück, die auf zwei unterschiedlichen Lokalisierungsfunktionen basieren. Die FTR bekommt von der FZS den Ist-Zustand

[64] Ackermann, S.; Winner, H.: Systemarchitektur beim Sicheren Anhalten (2020), S. 5 ff.

[65] UNICARagil: UNICARagil news - Halbzeitevent (2020), a: S. 44 f; b: S. 41; c: S. 45.

einschließlich der Pose des Fahrzeugs übermittelt und die bei der Ermittlung der Solltrajektorie im Trajektorienplaner benötigte Pose über eine eigene Lokalisierung im *Großhirn*. Bei inkonsistenten Lokalisierungsinformationen über die aktuelle Pose des Fahrzeugs entsteht folglich ein unerwünschtes Systemverhalten aufgrund von scheinbaren Regelabweichungen, die nicht zu Regeleingriffen führen dürfen.[66a] Der im Projekt UNICAR*agil* verwendete Lösungsansatz basiert auf einer Offset-Bestimmung der ermittelten Pose, um inkonsistente Lokalisierungsdaten zwischen dem VTP und der FTR zu erlauben.[66b]

Eine weitere Herausforderung bei der Berechnung der Stellgrößen ist, dass jeder Aktor eine charakteristische Systemdynamik aufweist, die zu einer Zeitverzögerung bei der Umsetzung der Stellgrößen führt. Die erwarteten Verzugszeiten werden mit einer Vorsteuerung in der FTR ausgeglichen. Eine weitere Aufgabe der FTR ist die Schätzung der fahrdynamischen Ausführungsgrenzen des Fahrzeugs.[67]

2.2.3.9 Aktorik

Die Aktorik befindet sich im *Rückenmark*. Die vier identischen Dynamikmodule an jeder Fahrplattform eines UNICAR*agil*-Fahrzeugs empfangen die Stellbefehle vom *Stammhirn* und setzen die vorgegebenen Trajektorien um. Sie decken die Aufgaben des Lenkens, Antreibens und Bremsens vollständig ab. Neben den Aktuatoren sind die Dynamikmodule mit Steuergeräten ausgestattet. Die individuelle Steuerbarkeit jedes einzelnen Dynamikmoduls und somit auch der vier Räder und die Einstellung von Lenkwinkeln von bis zu 90° ermöglichen neuartige Fahrmanöver im Straßenverkehr, wie beispielsweise das in Abbildung 2-9 dargestellte seitliche Einparken oder auch das Wenden auf der Stelle.[68,69]

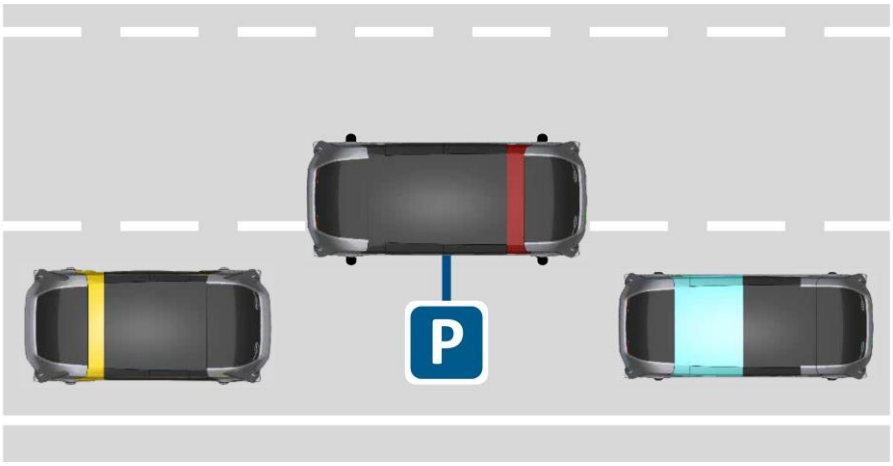

Abbildung 2-9: Seitliches Einparkmanöver (blau - durchgezogen) vom auto*SHUTTLE*

[66] Gottschalg, G. et al.: Inkonsistente Lokalisierungsdaten in modularen Systemen (2020), a: S. 3; b: S. 1.

[67] UNICARagil: UNICARagil news - Halbzeitevent (2020), S. 94.

[68] UNICARagil: UNICARagil news - 1. Ausgabe (2019), S. 8.

[69] Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018), S. 3.

2.3 Testmethoden in der Entwicklung im Automobilbereich

Die Entwicklung des zweiten Verfahrens, der Definition von Testfällen auf Grundlage von ermittelten Abhängigkeiten, erfordert die Einarbeitung in die Grundlagen des Testens, um ein systematisches Vorgehen bei der Definition der Testfälle und der anschließenden Testdurchführung sicherzustellen. Zuerst werden die für die vorliegende Arbeit notwendigen Begrifflichkeiten definiert und anschließend das Vorgehen eines allgemeinen Testprozesses vorgestellt. Abschließend werden die Herausforderungen beim Test automatisierter Fahrfunktionen sowie die Ansätze für den Test dieser beschrieben.

2.3.1 Terminologie

Im Folgenden werden Begriffsdefinitionen aus der Literatur eingeführt, um einen leichteren und domänenübergreifenden Austausch zu unterstützen.

2.3.1.1 Testen

Das allgemeine Ziel des Testens ist die Ausführung eines Systems oder einer Komponente unter spezifizierten Bedingungen, die Beobachtung der Ergebnisse während der Ausführung oder nachträglich bei einer Aufzeichnung und die abschließende Bewertung.[70a] Anhand von Tests werden die für den Abbruch von erforderlichen auszuführenden Funktionen verantwortlichen Defekte oder menschliche Aktionen, die das fehlerhafte Ergebnis erzeugen, aufgedeckt.[70b]

2.3.1.2 Testfall

Mit einem Testfall wird systematisch die Erfüllung der Anforderungen und des Testziels nachgewiesen. Mögliche Testziele sind beispielsweis die korrekte Implementierung, die Identifikation von Fehlern oder die Qualitätsprüfung.[70c,71]

2.3.1.3 Szene

Nach Ulbrich et al.[72a] beschreibt eine Szene eine Momentaufnahme des Umfelds und umfasst dynamische Elemente, die Szenerie und die Selbstrepräsentation aller Akteure und Beobachter. In einer Simulationsumgebung ist es möglich, eine allumfassende und objektive Szene darzustellen. Diese ist jedoch immer unvollständig, fehlerbehaftet und somit subjektiv.[72b]

2.3.1.4 Situation

Eine Situation beschreibt nach Ulbrich et al.[72c] die Gesamtheit der Umstände, die für die Auswahl passender Verhaltensmuster zu einem bestimmten Zeitpunkt zu berücksichtigen sind. Darüber hinaus umfasst eine Situation alle wesentlichen dynamischen Elemente, die relevante Szenerie, die relevante

[70] Bach, J.: Diss., Entwicklung und Test prädiktiver Fahrzeugregelungsfunktionen (2018), a: S. 42; b: S. 41 ff; c: S. 44.

[71] Steimle, M. et al.: Terminologie für den szenarienbasierten Testansatz (2018), S. 14.

[72] Ulbrich, S. et al.: Begriffe für das automatisierte Fahren (2015), a: S. 2; b: S. 2 f; c: S. 6.

Selbstrepräsentation, die notwendigen Ziele und Werte und die geforderten funktionsspezifischen Situationsaspekte.[73a] Da eine Situation aus einer Szene durch die Auswahl und Ableitung von Informationen mit bestimmten Zielen und Werten abgeleitet wird, ist eine Situation immer subjektiv.[73b]

2.3.1.5 Szenario

Für Ulbrich et al.[73c] ist ein Szenario die zeitliche Aneinanderreihung von Szenenelementen innerhalb einer Abfolge von Szenen, welche mit einer Startszene beginnt. Ein Szenario umfasst mindestens eine Szene, Aktionen und Ereignisse sowie Ziele und Werte.[73d] Szenarien weisen im Gegensatz zu Szenen eine Zeitspanne auf.[73c]

2.3.2 Allgemeiner Testprozess

Grundlage aller Testprozesse sind nach dem Standard IEEE 829[74] die folgenden drei Hauptphasen: *Testkonzept*, *Testspezifizierung* und *Testauswertung*. In der ersten Phase des Testprozesses werden Testkonzepte erstellt und die Planung der Tests durchgeführt. Bei dieser Vorbereitung werden zudem die konkreten Testziele definiert.[75a] In der zweiten Phase werden die Testfälle spezifiziert. Dazu werden die Anforderungen an die Szenarien und die gewünschten Eigenschaften ermittelt und in einem Lastenheft spezifiziert. Die Testspezifikation umfasst die Testfallableitung, die konkreten Testfälle sowie die Testobjekte und wird laufend erweitert und aktualisiert.[75b] Zu den Testspezifikationen gehören beispielsweise Schwellenwerte und Bewertungsmaßstäbe für Bewertungskriterien.[75c] Den spezifizierten Testfällen wird anschließend die benötigte Testinfrastruktur zugeordnet. Zu der Testinfrastruktur gehören die für die Durchführung der Testfälle notwendigen organisatorischen Bestandteile, wie beispielsweise die Testumgebung oder die Hardware- und Softwarekomponenten.[75d] Entsprechend der Testinstrumentierung werden die Testfälle aus der Testspezifikation durchgeführt. Die dritte Phase läuft parallel zur Durchführung der Testfälle oder bei einer Aufzeichnung der Daten erfolgt die Testbewertung nachträglich. Bei der Bewertung wird geprüft, ob die Testfälle die geforderten Bewertungskriterien erfüllen und sich das Testobjekt konform zu den spezifizierten Anforderungen verhält.[75e,76]

2.3.3 Testverfahren für automatisierte Fahrfunktionen

Der notwendige Sicherheitsnachweis für die Freigabe von automatisierten Fahrzeugen ab der Automatisierungsstufe 3 ist ein weiterhin ungelöstes Hindernis. Auf die Herausforderungen, die sich bei Absicherung automatisierter Fahrfunktionen ergeben, und auf neue Ansätze zur Bewältigung dieser wird im Folgenden eingegangen.

---

[73] Ulbrich, S. et al.: Begriffe für das automatisierte Fahren (2015), a: S. 8 f; b: S. 6; c: S. 9 f; d. S. 11.

[74] Institute of Electrical and Electronics Engineers: IEEE 829: Standard for Tests (2008).

[75] Steimle, M. et al.: Terminologie für den szenarienbasierten Testansatz (2018), a: S. 10 f; b: S. 13 f; c: S. 12; d: S. 15 f; e: S. 18.

[76] Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015), S. 445 f.

2.3.3.1 Herausforderungen beim Test automatisierter Fahrfunktionen

Das konventionelle Vorgehen in der Entwicklung sowie bei der Verifikation und Validierung von Fahrzeugen gelangt durch die zunehmende Komplexität der automatisierten Fahrfunktionen, die steigenden Haftungsrisiken hinsichtlich Sicherheit beim automatisierten Fahren und die komplexe Umgebung, mit der das System umgehen muss, an ihre Grenzen. Dementsprechend erhöht sich auch der benötigte Testumfang und Ansätze auf Basis von Realfahrten sind nicht mehr realisierbar. Infolgedessen werden neue strukturierte und systematische Ansätze zur Freigabe des Gesamtsystems benötigt.[77a,78]

2.3.3.2 Ansatz für den Test automatisierter Fahrfunktionen

Ein in der Literatur mehrfach beschriebener Testansatz für automatisierte Fahrfunktionen ist der szenariobasierte Ansatz. Schuldt et al.[79] beschreiben einen szenarienbasierten Testprozess mit einer Testfallgenerierung basierend auf einem 4-Ebenen-Modell. Bach et al.[80] entwickeln eine modellbasierte Szenariendarstellung für die Absicherung.[77b] Nach Wachenfeld et al.[81] ist der szenariobasierte Testansatz ein vielversprechender Ansatz zur Lösung der Herausforderungen beim Sicherheitsnachweis von automatisierten Fahrfunktionen und -systemen ab der Automatisierungsstufe 3. Aufgrund der Modularisierung des Systems ist eine Entkopplung der im Fokus stehenden Module von den anderen Modulen möglich und somit auch das Testen dieser entkoppelten Module.[82a,83]

Nach Bagschik et al.[84] sind szenariobasierte Testkonzepte in erster Linie für den Einsatz in Simulationsumgebungen geschaffen und basieren auf relevanten Betriebszenarien, welche variiert werden. Diese Szenarien werden klassischerweise in Expertengesprächen oder anhand von Testfahrten identifiziert. Dies bedeutet einen hohen Aufwand. Da die kritischen und herausfordernden Situationen und Szenarien seltener auftreten, ist ein Ansatz die Identifizierung der kritischen Szenarien, wodurch der Testaufwand für die Absicherung reduziert wird.[82b]

Für die Identifizierung der kritischen Szenarien sind in der Literatur verschiedene Ansätze vorhanden. Das Forschungsprojekt PEGASUS hat sich mit der Absicherung hochautomatisierter Fahrfunktionen beschäftigt und einen Ansatz zur Szenariodefinition mit einem 6-Ebenen Modell entwickelt.[85] Bagschik et al.[84] haben einen wissensbasierten Ansatz für die Szenariengenerierung für Betriebsszenarien auf deutschen Autobahnen entwickelt. Neben der wissensbasierten Szenariogenerierung existieren in

---

[77] Bagschik, G. et al.: Szenarien für automatisierte Fahrzeuge (2017), a: S. 1; b: S. 2.

[78] Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015), S. 440.

[79] Schuldt, F. et al.: Effiziente systematische szenarienbasierte Testgenerierung (2013), S. 6 f.

[80] Bach, J. et al.: Model based scenario specification (2016), S. 1.

[81] Wachenfeld, W. et al.: Objective Identification of Scenarios (2016), S. 4.

[82] Lippert, M. et al.: Bestehens- & Versagenskriterien für Tests (2020), a: S. 1; b: S. 4.

[83] Steimle, M. et al.: Terminologie für den szenarienbasierten Testansatz (2018), S. 10.

[84] Bagschik, G. et al.: Wissensbasierte Szenariengenerierung für Betriebsszenarien (2018), S. 1.

[85] BMWI: PEGASUS Method (2019), S. 4 ff.

der Literatur Ansätze für eine expertenbasierte, eine anforderungsbasierte und eine risikobasierte Szenariogenerierung.[86,87,88]

Neben diesem szenariobasierten Ansatz existiert der im Projekt UNICAR*agil* verfolgte Ansatz, das Gesamtsystem eines automatisierten Fahrzeugs in Module aufzuteilen und eine Absicherung dieser unabhängigen Module vorzunehmen. So kann der Testaufwand ebenfalls reduziert.[89] Voraussetzung für individuelle Modultests ist die Identifizierung der Abhängigkeiten zwischen den Modulen und zur Umgebung.[90]

Das Ziel neuer Testansätze für die Absicherung und Freigabe von automatisierten Fahrfunktionen ist daher zum einen, den Fokus auf die relevanten Szenarien und Situation zu legen. Zum anderen wird das Ziel verfolgt, herkömmliche System- und Integrationstests aufgrund des benötigten enormen Aufwands mit Modultests zu umgehen.

[86] Bagschik, G. et al.: Ontology based Scene Creation (2018), S. 1.

[87] Althoff, M.; Lutz, S.: Generation of Safety-Critical Test Scenarios (2018), S. 1.

[88] Abbas, H. et al.: Requirement and risk-based scenarios (2017), S. 1 ff.

[89] Klamann, B.; Lippert, M.: UNICARagil - Modulare Absicherung (2020), S. 1.

[90] Bach, J.: Diss., Entwicklung und Test prädiktiver Fahrzeugregelungsfunktionen (2018), S. 2.

3 Entwicklung des ersten Verfahrens: Ermittlung und Darstellung von Abhängigkeiten

In diesem Kapitel wird die Entwicklung eines Verfahrens zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung beschrieben. Das ermittelte Verfahren wird dann im nachfolgenden Kapitel auf das Projekt UNICAR*agil* angewendet, um die Abhängigkeiten zwischen den Modulen und zur Umgebung zu identifizieren.

In Abschnitt 3.1 werden zunächst die Anforderungen an das zu entwickelnde Verfahren definiert. Abschnitt 3.2 gibt anschließend einen Überblick über den Stand der Forschung im Bereich der Verfahren zur Ermittlung und Darstellung von Abhängigkeiten mit Fokus auf funktionale Abhängigkeiten. Die Verifizierung der aus der Literatur in Frage kommenden Verfahren anhand der in Abschnitt 3.1 definierten Anforderungen und die Entscheidung für ein Verfahren aus dem Lösungsraum erfolgen in Abschnitt 3.3. Anhand dieser Auswahl wird in Abschnitt 3.4 das abgeleitete Verfahren mit den entsprechenden Anpassungen für die gewünschte Anwendung beschrieben.

3.1 Anforderungsanalyse – Erstes Verfahren

Zu Beginn der Entwicklung der Methodik für das erste Verfahren werden im folgenden Abschnitt Anforderungen an das zu entwickelnde Verfahren für die Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung definiert. Diese ergeben sich einerseits aus dem Projekt UNICAR*agil* und andererseits aus dem Verwendungszweck dieses Verfahrens. Die Anforderungen dienen dazu, den Lösungsraum möglicher Verfahren in der Literatur einzugrenzen und einen bewertenden Vergleich dieser durchzuführen.

Im Folgenden werden zwei Arten von Anforderungen verwendet: Festforderungen und Wünsche. Eine Festforderung beschreibt eine Anforderung, welche das Verfahren unter allen Umständen zu erfüllen hat. Erfüllt ein Verfahren eine Festforderung nicht, wird sie ausgeschlossen. Ein Wunsch beschreibt eine Anforderung, welche nach Möglichkeit berücksichtigt wird, wobei jedoch die Erfüllung nicht unbedingt notwendig ist.

Ausgangspunkt der zu entwickelnden Methodik ist das modulare System im Projekt UNICAR*agil,* welches sich über alle Architekturebenen erstreckt. Daraus ergibt sich die **Festforderung**

1. ***Das Verfahren eignet sich für eine große Anzahl an Abhängigkeiten.***

Da das Gesamtsystem eines automatisierten Fahrzeugs eine Vielzahl an Schnittstellen und wechselwirkenden Modulen aufweist, ist die Zahl der vorliegenden Abhängigkeiten ebenfalls hoch. Bei der Recherche ist zu beachten, dass das dort beschriebene Anwendungsgebiet auf automatisierte Fahrzeuge hinsichtlich der Anzahl an Schnittstellen übertragbar ist, sodass es keine Beschränkungen bei der Anzahl der zu identifizierenden Abhängigkeiten gibt.

Weiterhin ergibt sich aus der Architektur von UNICAR*agil*-Fahrzeugen die **Festforderung**

2. ***Das Verfahren weist einen hohen Detaillierungsgrad bei der Ermittlung und Darstellung auf.***

Das Ziel, die Durchführung von Modultests anstatt von Integrations- und Systemtests, setzt die Identifizierung aller Abhängigkeiten zwischen den Modulen und zur Umgebung voraus. Da bei automatisierten Fahrzeugen komplexe Architekturen sowie Wechselwirkungen zwischen Komponenten vorliegen, ist eine detaillierte Ermittlung und Darstellung erforderlich.

Aufgrund der *ASOA* werden die modularen Dienste in Abhängigkeit des Betriebsmodus dynamisch verschaltet, sodass eine dynamische Rekonfiguration des Systems vorliegt. Daher wird die **Festforderung**

3. ***Das Verfahren eignet sich für die Ermittlung und Darstellung sowohl von statischen als auch von dynamischen Abhängigkeiten***

formuliert. Hierbei handelt es sich um eine Festforderung, da neben den statischen Abhängigkeiten die dynamischen Abhängigkeiten ebenfalls ein Bestandteil im Gesamtsystem sind.

Die nächste **Festforderung** basiert auf dem grundlegenden Gesamtaufbau mit den verschiedenen Architektursichten im Projekt UNICAR*agil* und lautet

4. ***Das Verfahren ermöglicht die Ermittlung von Abhängigkeiten auf verschiedenen Systemebenen.***

Durch die Integration der *ASOA* auf dem *Großhirn*, dem *Stammhirn* und dem *Rückenmark* wird eine ebenen- und hardwareübergreifende Kommunikation ermöglicht.

Neben den offensichtlichen und direkten Abhängigkeiten existieren auch Abhängigkeiten, die aufgrund der Interpretation des Empfängers entstehen. Daher wird die **Festforderung**

5. ***Das Verfahren eignet sich für die Ermittlung sowohl von direkten als auch von indirekten Abhängigkeiten***

aufgestellt. Offensichtliche Abhängigkeiten sind beispielsweise die Übertragung von Daten. Neben der reinen Datenübertragung werden auch unterschwellige Informationen übertragen, welche vom Empfänger subjektiv interpretiert werden. Diese indirekten Abhängigkeiten sind ebenfalls relevant für eine individuelle Absicherung der einzelnen Module. Ein Beispiel für unterschwellig übertragene Informationen ist, dass eine Hecke und eine Mauer in erster Linie nicht durchfahrbare Objekte sind, gleichzeitig aber die Hecke gegebenenfalls eine höhere Wahrscheinlichkeit aufweist, doch durchfahrbar zu sein.

Ein Ziel im Projekt UNICAR*agil* ist die Entwicklung weiterer Methoden für den Sicherheitsnachweis von automatisierten Fahrzeugen. Die in dieser Arbeit verfolgte Methode ist die individuelle Absicherung. Grundlage dafür ist die Identifizierung der offensichtlichen, aber auch der verborgenen Abhängigkeiten. Daraus ergibt sich die **Festforderung**

6. ***Das Verfahren ermöglicht die Ermittlung von unbekannten Abhängigkeiten.***

Diese Anforderung wird als Festforderung definiert, da eine Herausforderung bei der individuellen Absicherung darin besteht, besonders die unbekannten Abhängigkeiten aufzudecken.

Aufgrund der hohen Anzahl an vorliegenden Abhängigkeiten wird zur Übersichtlichkeit eine Kategorisierung der Abhängigkeiten angestrebt und daraus ergibt sich die **Festforderung**

7. ***Das Verfahren eignet sich für die Ermittlung und Darstellung von verschiedenen Kategorien von Abhängigkeiten.***

Besonders hinsichtlich der Darstellung ist eine Unterscheidung zwischen verschiedenen Kategorien notwendig. Dies erleichtert dem Benutzer zudem die anschließende Definition von Testfällen.

Die nachfolgende **Festforderung** ergibt sich aus der Aufgabenstellung dieser Arbeit und lautet

8. ***Das Verfahren basiert auf einer grafischen Modellierungssprache.***

Abhängigkeiten beziehen sich im Allgemeinen auf Komponenten, Werte oder benötigte Ressourcen.[91] Eine grafische Darstellung dieser Elemente hilft daher beim Verständnis der verschiedenen Abhängigkeiten.

Da die Modellierungssprache *Unified Modeling Language* (*UML*) eine Standard-Notation ist und somit bei der Nachvollziehbarkeit der Ermittlung und Darstellung von Abhängigkeiten unterstützt, besteht der **Wunsch**

9. ***Die grafische Modellierungssprache des Verfahrens ist UML.***

Da die *ASOA* die nachträgliche Integration neuer Softwarekomponenten erlaubt, die Identifizierung der Abhängigkeiten ein tiefes Systemverständnis der einzelnen Module erfordert, welches durch beispielsweise Expertengespräche mit den Entwicklern der Komponenten erweitert wird und im Nachhinein durchgeführte Testfälle neue Erkenntnisse über unbekannte Abhängigkeiten aufdecken, ergibt sich die **Festforderung**

10. ***Das Verfahren eignet sich für einen iterativen Prozess und eine nachträgliche Ermittlung weiterer Abhängigkeiten, um neue Erkenntnisse aus beispielsweise Expertenbefragungen oder durchgeführten Testfällen zu berücksichtigen.***

Nach der Ermittlung und Darstellung von Abhängigkeiten folgt die Definition und Durchführung von Testfällen. Daher wird der **Wunsch**

11. ***Die Darstellung eignet sich für eine nachfolgende Testfallgenerierung***

formuliert. Die Testfälle dienen dazu, die Abhängigkeiten aufzuzeigen, die mit dem Verfahren zuvor identifiziert wurden, obwohl sie vorher nicht offensichtlich waren. Daher ist von Vorteil, wenn sich aus der Darstellung der Abhängigkeiten direkt Testfälle generieren lassen. Es ist jedoch auch denkbar, dass die Darstellung in eine weitere überführt wird, um anhand dieser die Testfälle zu definieren.

Allgemein ergibt sich zudem folgender **Wunsch**

12. ***Das Verfahren weist eine geringe Komplexität auf, sodass eine leichte Nachvollziehbarkeit und ein geringer Aufwand bei der Einarbeitung und Fortführung gegeben sind.***

[91] Arias, C. T. et al.: Dependency Analysis Solutions (2011), S. 556.

Dieser Wunsch folgt daher, dass der Schwerpunkt dieser Arbeit darin liegt, ein Verfahren zu entwickeln und für eine Auswahl an Modulen die Ermittlung und Darstellung der Abhängigkeiten im Projekt UNICAR*agil* durchzuführen. Das Ziel der Entwicklung des Verfahrens ist, dass das Vorgehen übertragbar ist und nach Beendigung dieser Arbeit die Identifizierung von Abhängigkeiten im Projekt UNICAR*agil* fortgeführt wird.

Alle genannten Anforderungen für das erste Verfahren sind in Tabelle 3-1 zusammengefasst.

Tabelle 3-1: Anforderungen an das Verfahren zur Ermittlung und Darstellung von Abhängigkeiten

	Anforderung	**Art**
1.	Das Verfahren eignet sich für eine große Anzahl an Abhängigkeiten.	Festforderung
2.	Das Verfahren weist einen hohen Detaillierungsgrad bei der Ermittlung und Darstellung auf.	Festforderung
3.	Das Verfahren eignet sich für die Ermittlung und Darstellung sowohl von statischen als auch von dynamischen Abhängigkeiten.	Festforderung
4.	Das Verfahren ermöglicht die Ermittlung von Abhängigkeiten auf verschiedenen Systemebenen.	Festforderung
5.	Das Verfahren eignet sich für die Ermittlung sowohl von direkten als auch von indirekten Abhängigkeiten.	Festforderung
6.	Das Verfahren ermöglicht die Ermittlung von unbekannten Abhängigkeiten.	Festforderung
7.	Das Verfahren eignet sich für die Ermittlung und Darstellung von verschiedenen Kategorien von Abhängigkeiten.	Festforderung
8.	Das Verfahren basiert auf einer grafischen Modellierungssprache.	Festforderung
9.	Die grafische Modellierungssprache des Verfahrens ist *UML*.	Wunsch
10.	Das Verfahren eignet sich für einen iterativen Prozess und eine nachträgliche Ermittlung weiterer Abhängigkeiten, um neue Erkenntnisse aus beispielsweise Expertenbefragungen oder durchgeführten Testfällen zu berücksichtigen.	Festforderung
11.	Die Darstellung eignet sich für eine nachfolgende Testfallgenerierung.	Wunsch
12.	Das Verfahren weist eine geringe Komplexität auf, sodass eine leichte Nachvollziehbarkeit und ein geringer Aufwand bei der Einarbeitung und Fortführung gegeben sind.	Wunsch

3.2 Stand der Forschung – Erstes Verfahren

In diesem Abschnitt wird der Stand der Forschung im Bereich der Verfahren zur Ermittlung und Darstellung von Abhängigkeiten vorgestellt. Dabei wird der Schwerpunkt auf drei vielversprechende Methoden für die Anwendung in dieser Arbeit, die aus dem Bereich der Unfall- und Risikoanalyse kommen, gelegt. Die Aufgabe der Verfahren ist die Identifizierung von Abhängigkeiten mit dem Fokus auf funktionale Abhängigkeiten. Nach Strode und Huff[92] wird eine Abhängigkeit als Eigenschaft definiert, die eintritt, wenn der Fortschritt einer Handlung vom rechtzeitigen Ergebnis einer früheren Handlung oder dem Vorhandensein einer bestimmten Komponente oder Ressource abhängt.[92]

[92] Strode, D. E.; Huff, S. L.: Taxonomy of Dependencies (2012), S. 1.

3.2.1 Bayesian Network

Das Bayessche Netz (*Bayesian Network* (BN)) ist ein probabilistisches grafisches Modell und ein in der Literatur genutztes Verfahren zur Identifizierung von Abhängigkeiten und zur Zuverlässigkeitsanalyse. Ein BN weist Eigenschaften auf, welche zur Modellierung von Abhängigkeiten in komplexen Netzen passend sind.[93a] Mithilfe der bedingten Wahrscheinlichkeit und des *Bayes'schen Theorems* wird ein BN strukturiert. Durch diese Struktur werden die Abhängigkeiten zwischen den Systemkomponenten erfasst. Darüber hinaus werden in einem BN Risiken quantifiziert und Unsicherheiten modelliert.[93b] Die Abhängigkeiten werden als gerichtete Kanten in einem azyklischen Graphen dargestellt.[94] Zu den Anwendungsbereichen dieser Methode gehören die Systemtechnik, die Zuverlässigkeitstechnik und die künstliche Intelligenz, welche z.B. medizinische Diagnosen stellt.[93c]

3.2.2 Functional Resonance Analysis Method

Die Methode der funktionellen Resonanzanalyse (*Functional Resonance Analysis Method* (*FRAM*)) ist ein weiteres Verfahren in der Literatur, welches ein Modellierungswerkzeug darstellt und die Systemarchitektur untersucht.[95a] *FRAM* modelliert das System auf funktionaler Ebene und identifiziert Abhängigkeiten zwischen den einzelnen Systemelementen.[96]

Das *FRAM*-Modell wird in mehreren Iterationen erstellt und das Vorgehen innerhalb einer Iteration setzt sich aus mehreren Schritten zusammen.[97] Der erste Schritt ist die Identifizierung und Beschreibung der Funktionen. Funktionen stellen Handlungen oder Aktivitäten dar, die erforderlich sind, um ein bestimmtes Ergebnis zu erzielen. Dieses Hauptelement des *FRAM*-Modells ist ein Sechseck mit jeweils einer Erweiterung an den Ecken und ist in Abbildung 3-1 dargestellt. Diese sechs Aspekte sind die Zeit, die Steuerung, der Ausgang, die Ressource, die Vorbedingung und der Eingang und definieren die Funktion.[95b] Im zweiten Schritt wird die Variabilität identifiziert. Die zeitliche Variabilität beschreibt, ob der Output pünktlich, innerhalb eines akzeptablen Zeitrahmens, zu früh oder zu spät ist. Die qualitätsbezogene Variabilität definiert, ob der Output den erwarteten Standards entspricht oder außerhalb der erwarteten Standards liegt. Die Aggregation der Variabilität ist der dritte Schritt.[98]

Das Modellierungswerkzeug *FRAM* ist sehr flexibel und offen gestaltet und ermöglicht folglich die Modellierung und Analyse in vielen verschiedenen Bereichen.[98] Anwendungsbereiche der *FRAM*-Methode sind z.B. das Gesundheitswesen, die Luftfahrt, die Schifffahrt und der Eisenbahnsektor.[99]

---

[93] Hosseini, S.; Ivanov, D.: Bayesian networks for risk analysis (2020), a: S. 2; b: S. 3; c: S. 32.

[94] Kammouh, O. et al.: Use of Bayesian networks (2020), S. 5 f.

[95] Ferreira, P. N.; Cañas, J. J.: FRAM and Impacts of Automation (2019), a: S. 537; b: S. 537 f.

[96] Grabbe, N. et al.: Safety of automated driving: FRAM (2020), S. 7.

[97] Hollnagel, E. et al.: The Functional Resonance Analysis Method (2014), S. 48.

[98] Ferreira, P. N.; Cañas, J. J.: FRAM and Impacts of Automation (2019), S. 538.

[99] Salehi, V. et al.: Modeling Complex Socio-Technical Systems (2021), S. 22.

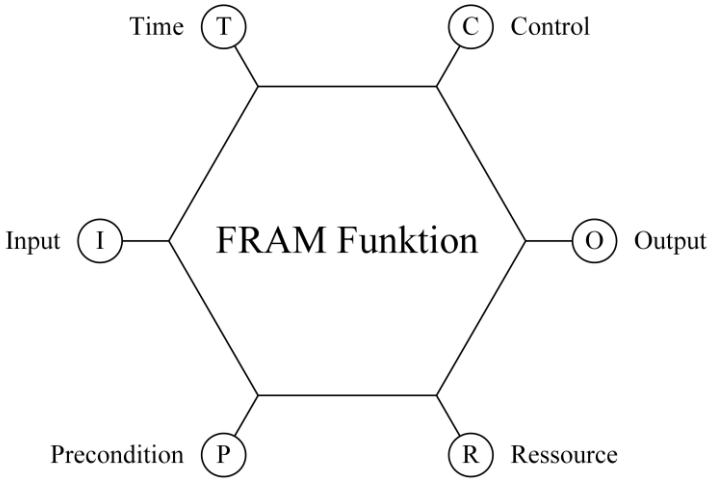

Abbildung 3-1: Struktur einer *FRAM*-Funktion nach Carvalho et al.[100]

In Grabbe et. al[101a] wird *FRAM* als Risikobewertungsmethode im Entwicklungsprozess hochautomatisierter Fahrzeuge untersucht. Die verfolgten Ziele sind dabei die Ableitung von Empfehlungen für das Systemdesign und der Gewinn an Erkenntnissen zur Reduzierung des Validierungsaufwandes, indem Abhängigkeiten zwischen den Systemelementen identifiziert werden.[101b]

3.2.3 Infrastructure-Resilience-oriented Modelling Language

Die Infrastruktur-Resilienz-orientierte Modellierungssprache (*Infrastructure-Resilience-oriented Modelling Language* (*IRML*)) ist ein Werkzeug zur Modellierung von Infrastrukturen und ihren Verbindungen. Dabei werden die funktionalen Abhängigkeiten zwischen den Systemkomponenten und Funktionen identifiziert. Die *IRML*-Methode wird bei kritischen Infrastrukturen, wie beispielsweise Stromnetzen, Telekommunikationsnetzen oder der Bereitstellung von Krankenwagen, mit dem Ziel der Stärkung der Widerstandsfähigkeit und des Erkenntnisgewinns bei Systemfehlfunktion angewendet.[102]

In jeder Infrastruktur sind physische Objekte, ihre Verbindungen, abstrakte Dienste bzw. Funktionen, ihre Zustände und organisatorische Gesichtspunkte, wie z.B. Richtlinien, vorhanden. Zudem handelt es sich bei den Architekturen von Infrastrukturen nicht um eigenständige und statische Einheiten, sondern um Aggregationen von Modulen, die in Beziehung zueinander stehen und sich zusätzlich entsprechend der Betriebsszenarien verändern.[102b]

Der erste Schritt der *IRML*-Methode ist die Modellierung des *IRML*-Modells. Dazu stehen drei Komponenten zur Verfügung: Dienste (dargestellt als Ellipsen), Domänen (dargestellt als abgerundete Rechtecke) und Ressourcen (dargestellt als Zylinder). Dienste repräsentieren bei der *IRML*-Methode Aktivitäten, die benötigt werden, um ein Ergebnis zu erzielen. Domänen beschreiben Phänomene, die in der Analyse als Einheit behandelt und dargestellt werden. Beispiele für Domänen sind z.B.

[100] Eigene Darstellung nach Carvalho, E. A. de et al.: FRAM and Complex Systems (2021), S. 71.

[101] Grabbe, N. et al.: Safety of automated driving: FRAM (2020), a: S. 1; b: S. 20.

[102] Silva, A.; Filippini, R.: Infrastructure Resilience-oriented Modelling Language (2011), a: S. 3; b: S. 5.

medizinisches Personal (organisatorisch), eine Pforte (physisch) und eine Pfortensteuerung (abstrakt). Die letzte Komponente, Ressourcen, sind Mengen, die gespeichert und verbraucht werden. Für die Verbindung der drei Komponenten sind zwei Verbindungsarten vorhanden. Es gibt Datenflüsse und Kontrollbeziehungen. Datenflüsse werden mittels Pfeils in Richtung des Empfängers dargestellt. Eine Kontrollbeziehung ist gekennzeichnet durch einen ausgefüllten Kreis auf der Kontrollseite und einen leeren Kreis auf der kontrollierten Seite.[103]

In Abbildung 3-2 ist die Modellierung eines vereinfachten Gasförderungssystems mit der *IRML*-Methode dargestellt.

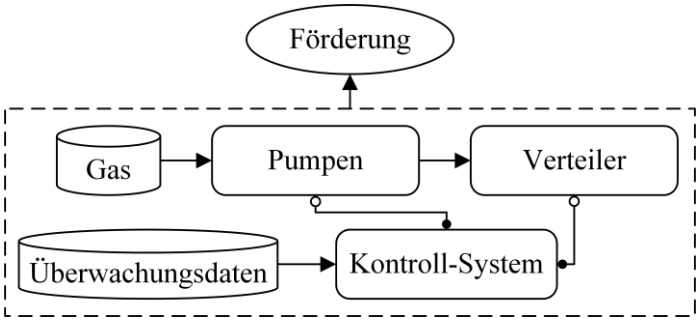

Abbildung 3-2: *IRML*-Modell am Beispiel eines Gasförderungssystems nach Hu et al.[104]

Im zweiten Teil der Methode werden anschließend an die Modellierung die Eigenschaften Ausfallsicherheit und die Fähigkeit, Störungen zu widerstehen und sich zu erholen, untersucht. Dazu wird aus dem *IRML*-Modell ein Referenzmodell abgeleitet, welches aus einem gerichteten Graphen mit Knoten und Verknüpfungen besteht.[105]

3.3 Bewertung des Lösungsraums – Erstes Verfahren

Für die Beurteilung der drei Verfahren aus der Literatur werden die Methoden anhand der Anforderungen aus Abschnitt 3.1 evaluiert. Anschließend wird die Auswahl, welche Methode an die gewünschte Anwendung angepasst und dann in Kapitel 4 angewendet wird, beschrieben.

3.3.1 Verifizierung der Verfahren

Im Folgenden werden die drei Methoden aus der Literatur beurteilt und dazu hinsichtlich der Erfüllung der definierten Festforderungen untersucht.

3.3.1.1 Bayesian Network

Die Verwendung von BN ist auf statische Systeme beschränkt, die sich zudem in einem Gleichgewichtszustand befinden.[106] Somit ist die in Abschnitt 3.1 aufgestellte Festforderung (3.), welche die

[103] Silva, A.; Filippini, R.: Infrastructure Resilience-oriented Modelling Language (2011), S. 10 f.

[104] Eigene Darstellung nach Hu, J. et al.: Dynamic Resilience of LNG System (2021), S. 21.

[105] Filippini, R.; Silva, A.: Modeling Framework for Networked Systems-of-Systems (2014), S. 84 f.

[106] Kammouh, O. et al.: Use of Bayesian networks (2020), S. 2.

Eignung des Verfahrens für statische und dynamische Abhängigkeiten fordert, aus Tabelle 3-1 nicht erfüllt. Ein BN ist zudem ein gerichteter azyklischer Graph. Dies bedeutet, dass, wenn eine Variable von einer anderen Variablen abhängt, andersherum keine Abhängigkeit vorliegt. Dementsprechend sind Rückkopplungsschleifen nicht realisierbar.[107] Folglich wird die Festforderung (10.), die Eignung des Verfahrens für einen iterativen Prozess, aus Abschnitt 3.1 nicht erfüllt. Aus diesen Gründen wird dieses Verfahren in dieser Arbeit nicht weiterverfolgt.

3.3.1.2 Functional Resonance Analysis Method

Die Identifikation der Funktionen und ihrer jeweiligen Aspekte ist bei der Erstellung eines groß angelegten *FRAM*-Modells sehr zeitaufwendig.[108,109] Bei der Ermittlung und Darstellung von Abhängigkeiten im Rahmen des Projekts UNICAR*agil* wird ein größer angelegtes Modell aufgrund der Anzahl der Module und Schnittstellen sowie der verschiedenen Auslöser für Zustands- und Verhaltensänderungen erstellt.

Es gibt eine Unterstützung bei der Modellierung mit *FRAM* durch die Software *FRAM Model Visualiser*. Bei einer Anzahl von ca. 50 Funktionen kommt es zu zeitlichen Verzögerungen bei der Software.[110] Daher bleibt der erhöhte Aufwand bei größeren Systemen wie im Projekt UNICAR*agil*. Dies ist der einzige negative Aspekt dieser Methode, denn sonst erfüllt das Modellierungswerkzeug *FRAM* alle Festforderungen aus Tabelle 3-1.

3.3.1.3 Infrastructure-Resilience-oriented Modelling Language

Beim Umgang der untersuchten Eigenschaften, der untersuchten Ausfallsicherheit und des analysierten Verhaltens existieren hinsichtlich Unsicherheiten und Belastbarkeit Schwierigkeiten mit dem Referenzmodell.[109] Da jedoch für das zu entwickelnde Verfahren das Vorgehen der *IRML*-Methode bis zum Modell ausreicht und der Rest der Methode vernachlässigt wird, liegen in Bezug auf dieses Problem keine Einschränkungen vor. Die Methode erfüllt alle Festforderungen aus Tabelle 3-1.

3.3.2 Auswahl der Methode

Da *FRAM* und die *IRML*-Methode beide die Festforderungen erfüllen, erfolgt anhand der Wünsche aus Tabelle 3-1 die Auswahl des Verfahrens. Es gibt drei Wünsche, die in Abschnitt 3.1 definiert werden.

[107] Kammouh, O. et al.: Use of Bayesian networks (2020), S. 32.

[108] Salehi, V. et al.: Modeling Complex Socio-Technical Systems (2021), S. 22.

[109] Hu, J. et al.: Dynamic Resilience of LNG System (2021), S. 3.

[110] Grabbe, N.: FRAM and Automated Driving (2019), S. 31.

Der erste Wunsch ist, dass die grafische Modellierungssprache des Verfahrens *UML* ist. *IRML* ist bereits eine grafische Modellierungssprache und eine Anpassung an *UML* ist daher einfacher umsetzbar als bei *FRAM*.[111a] *FRAM* basiert auf einer eigenen Darstellungsweise in Form von Sechsecken.[112]

Der zweite Wunsch ist, dass die Darstellung sich für eine nachfolgende Testfallgenerierung eignet. Dies ist vor der Entwicklung des zweiten Verfahrens zur Definition von Testfällen schwer zu beurteilen. Da die Darstellung bei *FRAM* und des *IRML*-Modells ähnlich vom Aufbau sind, werden gleichermaßen Anpassungen notwendig.

Der letzte Wunsch ist, dass das Verfahren eine geringe Komplexität aufweist, sodass eine leichte Nachvollziehbarkeit und ein geringer Aufwand bei der Einarbeitung und Fortführung gegeben sind. Da bei beiden Methoden keine computergestützten Hilfsmittel zur Verfügung stehen, liegt kein Vorteil für eine der beiden Methoden vor. Die *IRML*-Methode und das in dieser Arbeit zu entwickelnde Verfahren werden beide auf ähnliche Architekturen angewendet. Bei der *IRML*-Methode handelt es sich um Infrastrukturarchitekturen, die durch entsprechende graphische Elemente bei der Modellierung abgebildet werden.[111b] Dadurch ergibt sich für das zu entwickelnde Verfahren auf Grundlage der *IRML*-Methode eine einfachere Nachvollziehbarkeit und folglich auch ein geringerer Aufwand anstatt auf Basis von *FRAM*.

Auf Grundlage der leichteren Umsetzbarkeit bei der Anpassung an *UML*, die einfachere Nachvollziehbarkeit und der geringere Aufwand wird die *IRML*-Methode als Grundlage für das zu entwickelnde Verfahren zur Ermittlung und Darstellung von Abhängigkeiten ausgewählt.

3.4 Methodik – Erstes Verfahren

Zuerst werden die erforderlichen Anpassungen an die zuvor ausgewählte Methode für die gewünschte Anwendung beschrieben. Danach erfolgt eine ausführliche Beschreibung des angepassten Verfahrens zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung.

3.4.1 Anpassung an die Anwendung

Bei der *IRML*-Methode sind die drei Komponenten der Modellierung der Dienst, die Domäne und die Ressource. Da der Begriff „Dienst“ bereits im Projekt UNICAR*agil* im Zusammenhang mit der dienstorientierten Architektur verwendet wird, wird im Folgenden bei dem entwickelten Verfahren zur Ermittlung und Darstellung von Abhängigkeiten, welches auf der *IRML*-Methode basiert, der Begriff „Funktion“ statt „Dienst“ verwendet.

Der Wunsch nach *UML* als grafische Modellierungssprache wird umgesetzt und die Komponenten, die der Notation noch nicht entsprechen, werden angepasst. Die angepassten Komponenten sind in Abbildung 3-3 dargestellt.

[111] Silva, A.; Filippini, R.: Infrastructure Resilience-oriented Modelling Language (2011), a: S. 2; b: S. 10 f.

[112] Ferreira, P. N.; Cañas, J. J.: FRAM and Impacts of Automation (2019), S. 537 f.

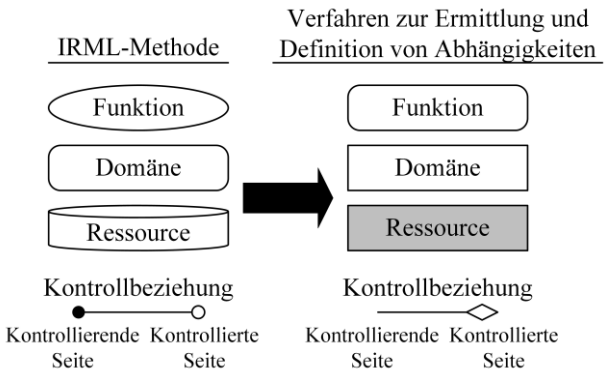

Abbildung 3-3: Anpassungen an die Modellierungssprache *UML*

3.4.2 Verfahren zur Ermittlung und Darstellung von Abhängigkeiten

Das nachfolgend beschriebene Verfahren zur Ermittlung und Darstellung von Abhängigkeiten basiert auf dem ersten Teil der *IRML*-Methode und ist ein iterativer Prozess. Zu Beginn erfolgen die Systemmodellierung und der Aufbau eines Abhängigkeitsnetzes. Dabei werden folgende Zwischenschritte nacheinander abgearbeitet. Der erste Zwischenschritt ist die Identifizierung und das Aufzählen der beteiligten Funktionen. Die einzelnen Funktionen werden anschließend in das Netz aufgenommen und dienen als Startpunkte des weiteren Vorgehens.[113a] Im zweiten Zwischenschritt wird jede Funktion dahingehend untersucht, welche Domänen und Ressourcen für die Erfüllung der jeweiligen Funktion erforderlich sind.[113b] Im dritten Zwischenschritt erfolgt nacheinander eine Einzelbetrachtung der Funktionen mit ihren Domänen und Ressourcen, um das Verhältnis der Domänen und Ressourcen untereinander zu bestimmen. Zu dieser Identifizierung wird die folgende Frage gestellt: „Liegt ein Datenfluss oder eine Kontrollbeziehung vor?“[113b]

Bei einem vorliegenden Datenfluss werden die Fragen unter **1.** und bei einer vorliegenden Kontrollbeziehung werden die Fragen unter **2.** gestellt.[113c]

1. „Welches Ziel wird beim Datenfluss verfolgt?“

Bei zwei Domänen wird zusätzlich die Frage gestellt: „Welche Domäne liefert die Daten und welche liest die Daten?“

Bei einer Ressource und einer Domäne wird zusätzlich die Frage gestellt: „Stellt die Domäne die Materialien bereit oder nutzt die Domäne die in der Ressource gespeicherten Materialien?“

Dieser Datenfluss wird mit einem Pfeil in Richtung des Empfängers der Daten eingezeichnet.

[113] Silva, A.; Filippini, R.: Infrastructure Resilience-oriented Modelling Language (2011), a: S. 18 f; b: S. 19; c: S. 23 f.

2. „Welches Ziel verfolgt die kontrollierende Seite und welches die kontrollierte Seite?“

   Bei zwei Domänen wird zusätzlich die Frage gestellt: „Welche Domäne kontrolliert die andere?“

   Bei einer Ressource und einer Domäne wird zusätzlich die Frage gestellt: „Welche Seite ist für das Hinzufügen, Beseitigen und Aktualisieren der Elemente im Speicher zuständig?“

   Diese Kontrollbeziehung wird mit einem leeren rautenförmigen Pfeil eingezeichnet. Die Raute ist auf der Seite der kontrollierten Seite.

Nach der Identifizierung und der Einzeichnung der Verhältnisse der Domänen und Ressourcen untereinander für alle Funktionen werden Grenzlinien um die jeweils für eine Funktion relevanten Domänen und Ressourcen gezogen, sodass Teilsysteme entstehen. Die Funktionen werden mit ihrem jeweiligen Teilsystem in Form eines Pfeils in Richtung der Funktion verbunden.[114]

Im vierten Zwischenschritt werden die Teilsysteme dahingegen hinterfragt, ob diese von einer oder mehreren Funktionen abhängig sind. Wenn eine Abhängigkeit zwischen einem Teilsystem und einer neu gefundenen Funktion besteht, löst dies eine weitere Suche nach neuen Domänen und Ressourcen aus, die möglicherweise wieder von anderen Funktionen abhängig sind. Dazu wird nach einer neu gefundenen Funktion wieder beim ersten Zwischenschritt der Systemmodellierung und dem Aufbau des Abhängigkeitsnetzes begonnen. Diese Kette lässt sich soweit fortsetzen, bis vom Modellierer die Entscheidung getroffen wird, welche Funktionen nicht mehr aufgenommen werden.[114]

Das Ergebnis dieses ersten Schritts des Verfahrens ist die Darstellung der Abhängigkeiten in Form eines Abhängigkeitsnetzes. Im nachfolgenden Schritt erfolgt anhand des Abhängigkeitsnetzes die Definition verschiedener Abhängigkeitskategorien. Mit den Abhängigkeitskategorien wird anschließend ein Steckbrief für die Abhängigkeiten entworfen. Dieser Abhängigkeitssteckbrief ist eine Übersicht über die Merkmale und Eigenschaften der Abhängigkeiten.

Die aus dem Abhängigkeitsnetz herausgelesenen Abhängigkeiten werden im nächsten Schritt in einer Liste notiert und jeder Abhängigkeit wird eine Abhängigkeitskategorie zugewiesen. Diese Liste wird an den Aufbau des Abhängigkeitssteckbriefs angepasst und die neuen relevanten Angaben zu den Abhängigkeiten ergänzt. Gleichzeitig werden die rautenförmigen Pfeile im Abhängigkeitsnetz an den definierten Typ von Pfeil der jeweiligen Abhängigkeitskategorie angepasst. So ist eine Unterscheidung bei der Modellierung und dem Aufbau des Netzes am Anfang jeder Iteration zwischen den bereits identifizierten und kategorisierten Abhängigkeiten zu den gerade aufgedeckten Abhängigkeiten möglich. Mit diesem Zwischenergebnis werden im nächsten Schritt Expertenbefragungen durchgeführt. Diese Interviews dienen dazu, weitere Erkenntnisse über die Module, deren Schnittstellen und auftretende Zustandsänderungen, die Verhaltensänderungen in dem jeweiligen Modul auslösen, zu erhalten. Auf das Vorgehen bei den Befragungen wird in Abschnitt 4.2.1 eingegangen.

[114] Silva, A.; Filippini, R.: Infrastructure Resilience-oriented Modelling Language (2011), S. 19 f.

Die neuen Erkenntnisse werden in einer zweiten Iteration in das Abhängigkeitsnetz eingebaut, um weitere Abhängigkeiten zu identifizieren. Dazu werden die oben beschriebenen Zwischenschritte zur Systemmodellierung und zum Aufbau des Abhängigkeitsnetzes durchgeführt. Die in dieser Iteration neu ermittelten Abhängigkeiten werden in die bestehende Abhängigkeitsliste mit dem Aufbau des Abhängigkeitssteckbriefs aufgenommen und ihnen ebenfalls eine Abhängigkeitskategorie zugewiesen. Danach folgt die Anpassung der neuen Kontrollbeziehungen im Abhängigkeitsnetz an die jeweilige Abhängigkeitskategorie. Damit ist die Einarbeitung der neuen Erkenntnisse durch die erste Expertenbefragung erfolgt und das weitere Vorgehen ist dem Modellierer überlassen. Weitere Expertenbefragungen sind möglich. Jedenfalls ist auch das Verfahren zur Definition von Testfällen auf Grundlage der bis zu diesem Zeitpunkt ermittelten Abhängigkeiten durchführbar. Über diesen Weg besteht die Möglichkeit, weitere unbekannte Abhängigkeiten aufzudecken und diese anschließend in einer weiteren Iteration in das Abhängigkeitsnetz einzubinden. Die gesamte Methodik zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung automatisierter Fahrfunktionen ist zusammengefasst in Abbildung 3-4 dargestellt.

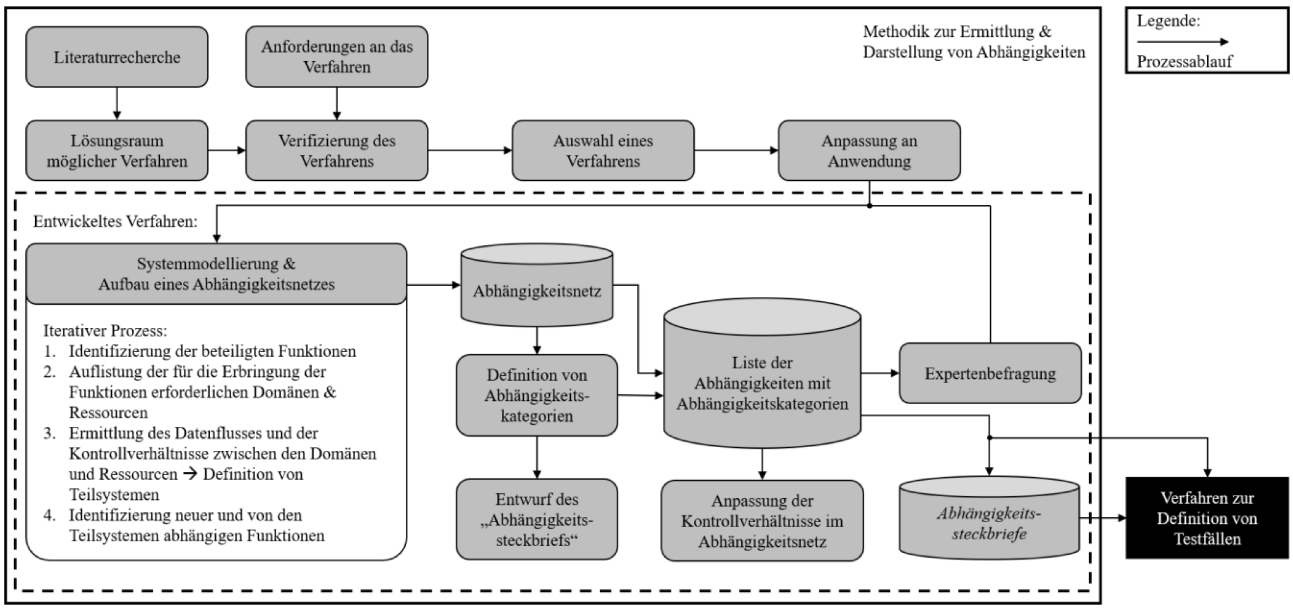

Abbildung 3-4: Methodik zur Ermittlung und Darstellung von Abhängigkeiten

4 Anwendung des ersten Verfahrens: Ermittlung und Darstellung von Abhängigkeiten im Rahmen des Projekts UNICAR*agil*

In diesem Kapitel wird das zuvor in Kapitel 3 entwickelte Verfahren zur Ermittlung und Darstellung von Abhängigkeiten auf das im Projekt UNICAR*agil* entwickelte Gesamtsystem angewendet. In Abschnitt 4.1 werden anhand von Beispielen die Systemmodellierung und der Aufbau des Abhängigkeitsnetzes erläutert. Abschnitt 4.2 beschreibt die drei durchgeführten Expertenbefragungen und die gewonnenen Erkenntnisse. Die Ergebnisse aus der Anwendung des Verfahrens werden in Abschnitt 4.3 vorgestellt.

4.1 Systemmodellierung und Aufbau des Abhängigkeitsnetzes

Der erste Schritt des Verfahrens ist die Systemmodellierung und der Aufbau des Abhängigkeitsnetzes. Die Zwischenschritte sind in Abschnitt 3.4.2 im Detail beschrieben. Neben der Literatur zum Projekt UNICAR*agil* dient das webbasierte Architektur Tool[115] vom Projekt UNICAR*agil* als weitere Informationsquelle für die verschiedenen Module und deren Schnittstellen.

Im Vorhinein der Systemmodellierung ist anzumerken, dass die Ermittlung und Darstellung von Abhängigkeiten mit Fokus auf ausgewählte Module erfolgt ist. Die folgenden Module sind die Fahrdynamikzustandsschätzung (FZS), der Verhaltens- und Trajektorienplaner (VTP), die Fahrschlauchabsicherung (FSA) und die Fahrdynamik- und Trajektorienregelung (FTR). Die Ermittlung und Darstellung von Abhängigkeiten erfolgte für die Trajektorienvorverarbeitung, die Plattfomsensorik und das Dynamikmodul detaillierter als für die anderen Module. Diese sind mit ihren Solldatenschnittstellen lediglich begrenzt modelliert. Eine weitere Anmerkung ist, dass in den überwiegenden Modulen zu Beginn Plausibilitäts- und Altersprüfungen durchgeführt werden, diese jedoch bei der Systemmodellierung und dem Aufbau des Abhängigkeitsnetzes vernachlässigt werden.

Zur Unterstützung der Nachvollziehbarkeit sind die folgenden vier Zwischenschritte bei der Systemmodellierung und dem Aufbau des Abhängigkeitsnetzes um Abbildungen und Erläuterungen aus dem Entwicklungsprozess ergänzt. Diese Abbildungen stellen den Zwischenstand der ersten Iteration dar. Die Legende für Abbildung 4-2 bis Abbildung 4-5 ist in Abbildung 4-1 dargestellt.

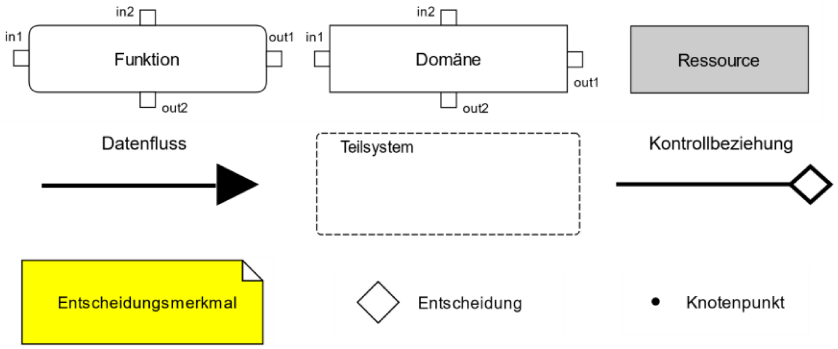

Abbildung 4-1: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - Legende

[115] UNICARagil Architektur Tool.

4.1.1 Identifizierung der beteiligten Funktionen

Der erste Zwischenschritt ist die Identifizierung und die Aufzählung der beteiligten Funktionen. Beim Start der Anwendung des entwickelten Verfahrens zur Ermittlung und Darstellung von Abhängigkeiten wird mit den Modulen begonnen, auf denen der Fokus in dieser Arbeit liegt. Die Funktion der FZS ist die Bestimmung des Ist-Zustands der Fahrzeugbewegung. Bei dem VTP sind die Funktionen zum einen die Planung einer Weg-Zeit Kurve, der Solltrajektorie, und zum anderen die Planung einer Bewegung in den sicheren Stand, der Notbahn. Die FSA besitzt die Funktion der Ausführung eines risikominimalen Anhaltemanövers. Bei der FTR werden zwei Funktionen ausgeführt. Die erste Funktion ist die Umsetzung der Solltrajektorie und die Generierung von Stellgrößen. Die Schätzung der fahrdynamischen Grenzen ist die zweite Funktion. Diese sechs Funktionen dienen als Startpunkt der Systemmodellierung und dem Aufbau des Abhängigkeitsnetzes. Dieser erste Zwischenschritt ist in Abbildung 4-2 veranschaulicht.

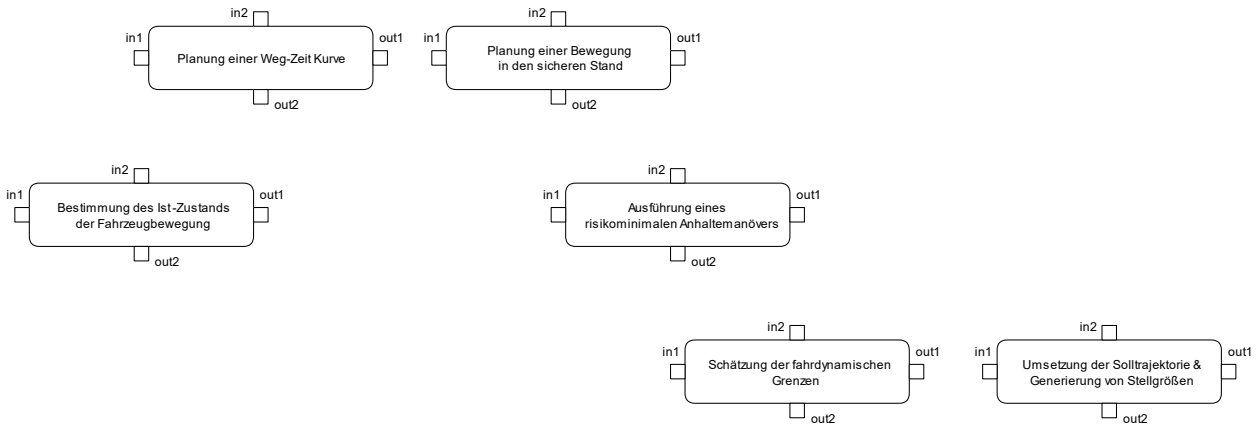

Abbildung 4-2: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 1. Zwischenschritt

4.1.2 Auflistung der erforderlichen Domänen und Ressourcen

Im zweiten Zwischenschritt werden für die sechs zuvor identifizierten Funktionen die Domänen und Ressourcen aufgelistet, die zur Erbringung der jeweiligen Funktion erforderlich sind. Für die Funktionen der FTR ist dieser Zwischenschritt in Abbildung 4-3 dargestellt und wird im Folgenden näher erläutert. Die nachfolgenden zwei Zwischenschritte werden auch anhand der FTR dargestellt. Das Vorgehen der Zwischenschritte zwei bis vier ist für alle Funktionen identisch.

Für die Erbringung der beiden Funktionen der FTR werden in der ersten Iteration der Regler, die folgenden Domänen und Ressourcen betrachtet. Domänen repräsentieren Phänomene, die als Einheit verstanden werden. Eine Einheit, die erforderlich für die Funktionen der FTR ist, sind die Offsetberechnung und -korrektur. Diese dient der Vermeidung von unerwünschten Effekten auf die FTR durch inkonsistente Lokalisierungsdaten zwischen dem VTP und der FTR und gibt die korrigierte Ist-Pose aus.

Eine weitere Domäne ist die Verzugszeit der Aktorik. Die im Antriebsstrang des Fahrzeugs eingesetzte Aktorik unterliegt Verzögerungszeiten, da Sollwertbefehle dort mit einem bestimmten

dynamischen Verhalten umgesetzt werden. Der Zielzustand innerhalb einer neuen Trajektorie wird daher nicht sofort erreicht. Die Fahrdynamik ist ein weiteres Phänomen innerhalb der FTR, welche beispielsweise durch die Lenkgeschwindigkeits- und Beschleunigungsgrenzen beeinflusst wird. Analog beeinflussen die Aktuatorgrenzen ebenfalls die Funktionsweise der FTR. Darüber hinaus sind das Straßenprofil und der Reibkoeffizient weitere Einflussfaktoren und werden folglich als Domänen aufgeführt. Der Regler ist eine selbstständige Einheit und berechnet den Kraftbedarf zur Beseitigung der Regelabweichung. Daher wird dieser ebenfalls als Domäne eingeführt. Eine weitere Domäne, die für die Erbringung der Funktionen notwendig ist, ist die zur Verfügung gestellte Rechenleistung vom Steuergerät auf dem *Stammhirn*.

Neben diesen aufgeführten Domänen sind folgende Ressourcen für die Erfüllung der beiden Funktionen der FTR notwendig. Ressourcen sind Mengen, die gespeichert und verbraucht werden. Die korrigierten Ist-Posen stammen aus der Offsetberechnung und -korrektur und werden für die Regelung verwendet. Darüber hinaus sind weitere Ressourcen notwendig. Die Solltrajektorie, der Fahrzeugzustand, die Uhrzeit, der Ladestand der Batterien sowie der Zustand der Dynamikmodule sind erforderliche, von außen bereitgestellte Ressourcen für die beiden Funktionen der FTR.

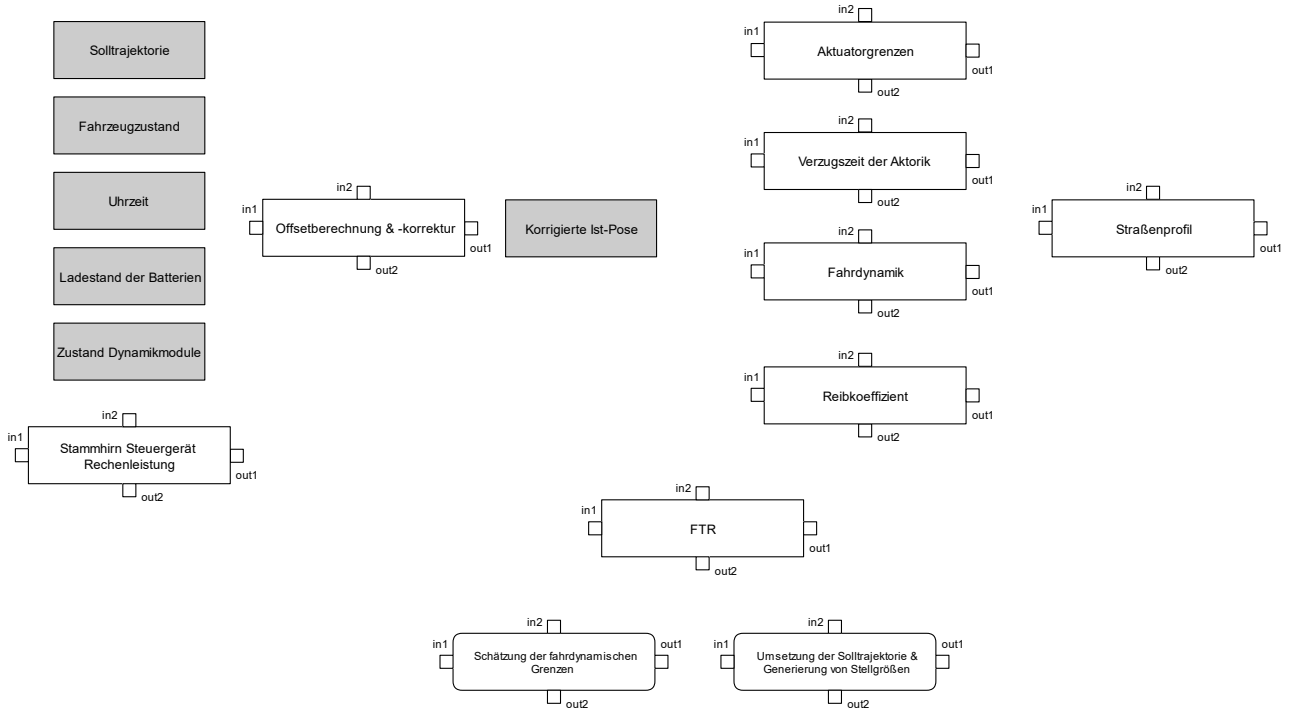

Abbildung 4-3: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 2. Zwischenschritt (Ausschnitt FTR)

4.1.3 Ermittlung der Datenflüsse und Kontrollbeziehungen und Teilsystemdefinition

Der dritte Zwischenschritt dient der Bestimmung der Verhältnisse der Domänen und Ressourcen untereinander. Dazu werden die in Abschnitt 3.4.2 definierten Fragen nacheinander für jede Komponente beantwortet und so die Abbildung 4-3 schrittweise um Datenflüsse und Kontrollbeziehungen erweitert. Abschließend wird eine Grenzlinie um diese Domänen und Ressourcen gezogen. Dieses Teilsystem aus Domänen und Ressourcen ist somit relevant für die Erbringung der jeweiligen Funktionen.

In Abbildung 4-4 ist der Ausschnitt für die FTR als Teilsystem mit der Erweiterung um die Datenflüsse und Kontrollbeziehungen dargestellt. Nachfolgend ist ein Beispiel für einen Datenfluss zwischen einer Domäne und einer Ressource aufgeführt. Die Domäne Offsetberechnung und -korrektur stellt die korrigierten Ist-Posen als Ressource zur Verfügung. Die Daten von dieser Ressource werden anschließend vom Regler gelesen. Daneben sind in dem Teilsystem mehrere Kontrollbeziehungen vorhanden. Das Straßenprofil kontrolliert den Reibkoeffizienten und auch die Fahrdynamik. Diese beiden Domänen wiederum kontrollieren den Regler. Nach der Ermittlung der Datenflüsse und Kontrollbeziehungen wird wie in Abbildung 4-4 eine Grenzlinie um die Domänen und Ressourcen gezogen, sodass deutlich wird, welche relevant für die Erbringung der beiden Funktionen Umsetzung der Solltrajektorie und Generierung von Stellgrößen und Schätzung der fahrdynamischen Grenzen sind.

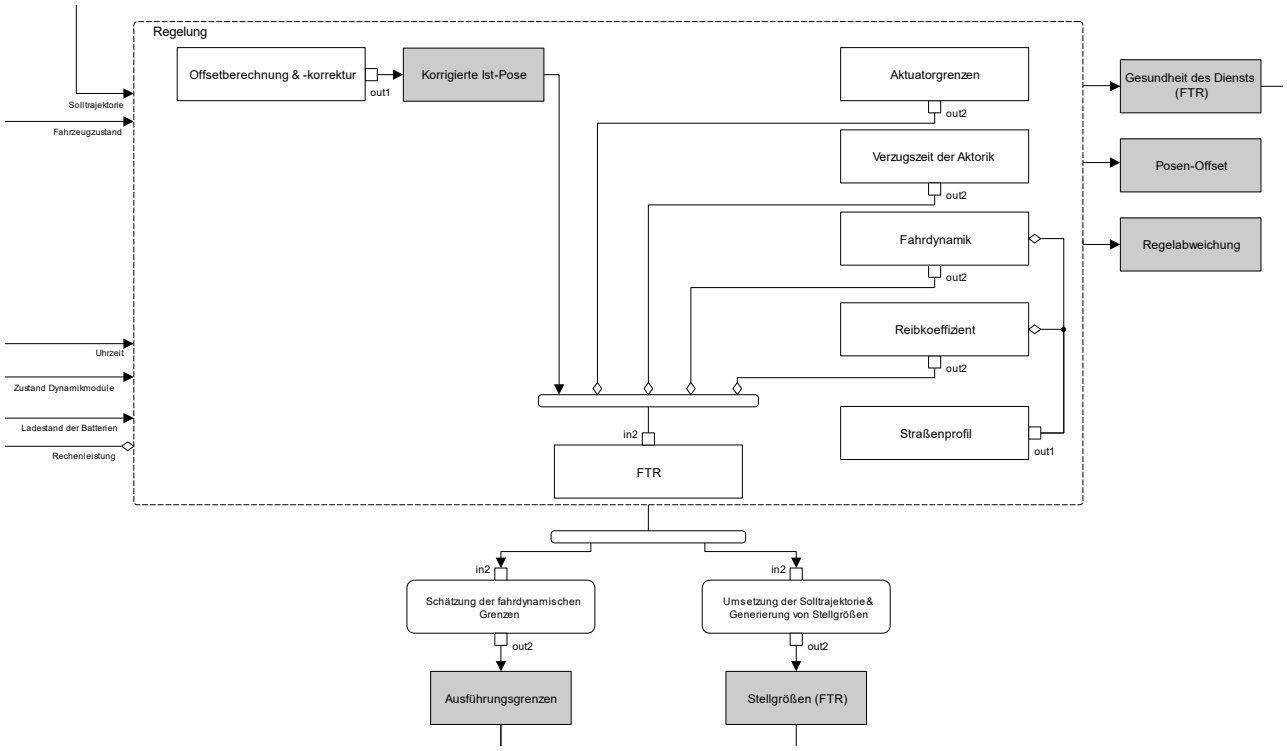

Abbildung 4-4: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 3. Zwischenschritt (Ausschnitt FTR)

4.1.4 Identifizierung neuer und von den Teilsystemen abhängigen Funktionen

Im letzten Zwischenschritt werden die Teilsysteme dahingegen untersucht, ob diese von weiteren Funktionen abhängig sind. Am Beispiel der FTR lässt sich eine solche Abhängigkeit zeigen. In Abhängigkeit vom Betriebsmodus verarbeitet die FTR die Solltrajektorien von jeweils unterschiedlichen Funktionen. Die FTR erhält dabei keine Kenntnis über die Quelle der empfangenen Trajektorien. Beim Betriebsmodus *Sicheres Anhalten* werden die in der Fahrschlauchabsicherung berechneten Nottrajektorien in der FTR verarbeitet. Wenn dieser Betriebsmodus nicht vorliegt, verarbeitet die FTR die von der Trajektorienvorverarbeitung kommenden Solltrajektorien. Die Ergänzung der beiden abhängigen Funktionen ist in Abbildung 4-5 dargestellt.

Mit diesem Zwischenschritt ist der Aufbau des Abhängigkeitsnetzes für die erste Iteration abgeschlossen. Es folgt lediglich die Anpassung der Kontrollverhältnisse an den Pfeiltyp der entsprechenden Abhängigkeitskategorie. Das Netz ist die Grundlage für die nachfolgenden Schritte: der Definition der Abhängigkeitskategorien, dem Entwurf des Abhängigkeitssteckbriefs sowie der Erstellung der Abhängigkeitsliste. Anschließend erfolgt die Expertenbefragung.

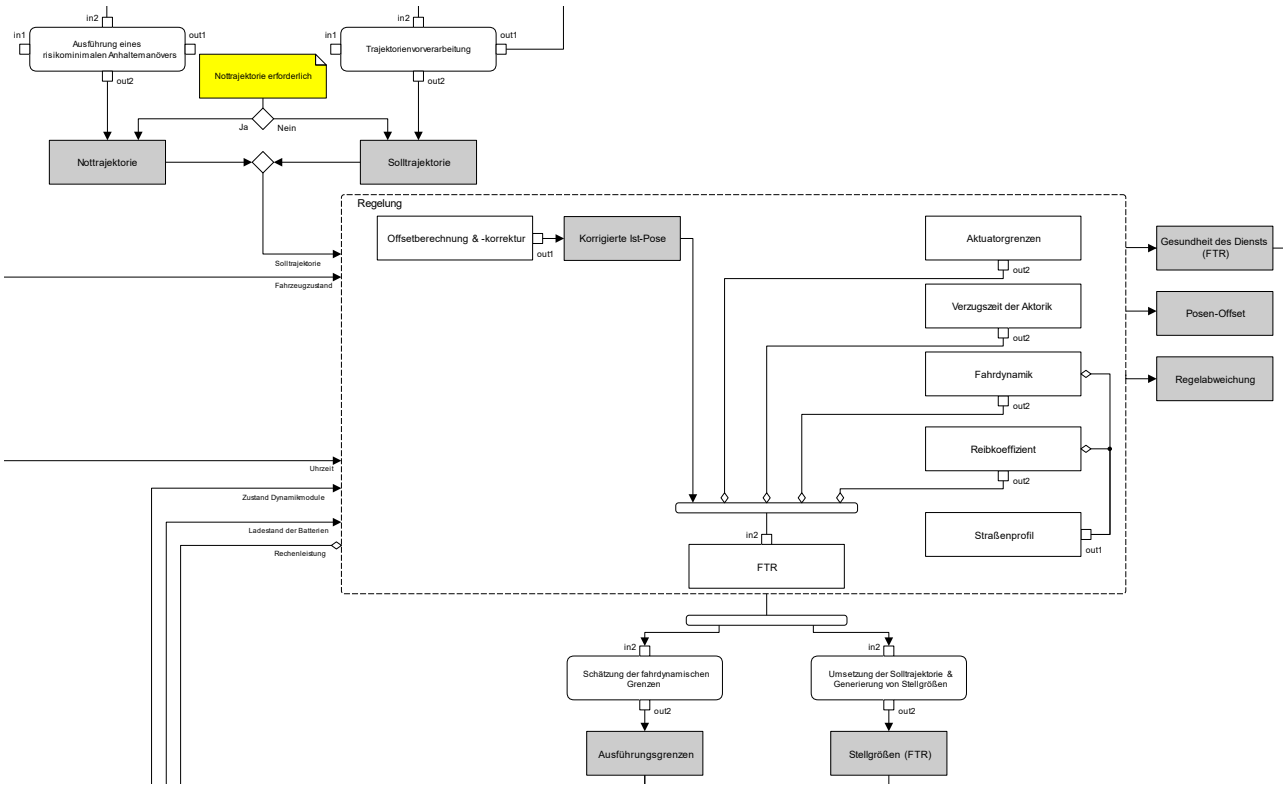

Abbildung 4-5: Systemmodellierung und Aufbau des Abhängigkeitsnetzes - 4. Zwischenschritt (Ausschnitt FTR)

4.2 Expertenbefragung

Neben der Einarbeitung in die Literatur zum Projekt UNICAR*agil* und die Analyse der Simulationsmodelle sind die Gespräche mit Experten ein Bestandteil, um Erkenntnisse über die Module, deren Schnittstellen und mögliche Auslöser von auftretenden Zustands- und Verhaltensänderungen und somit auch Hinweise auf weitere Abhängigkeiten zu erhalten. Zuerst wird das Vorgehen vorgestellt und anschließend werden die gewonnenen Erkenntnisse beschrieben.

4.2.1 Vorgehensweise bei den Gesprächen

Der erste Teil der Expertenbefragung hat eine vorgegebene Struktur und der zweite Teil dient dem freien Austausch und der Diskussion. Im ersten Teil wird die Aufgabenstellung mit dem Fokus auf das Verfahren zur Ermittlung und Darstellung von Abhängigkeiten eingeführt und das Ziel der Expertenbefragung in den Rahmen der Aufgabenstellung eingeordnet. Darüber hinaus werden das bisherige Vorgehen sowie die zum Zeitpunkt der ersten Iteration definierten Abhängigkeitskategorien mit Beispielen vorgestellt, damit die Experten ein Verständnis dafür bekommen, welche Abhängigkeiten gesucht werden. Für den Rest des Gesprächs, den zweiten Teil, wird der Schwerpunkt auf das vom Experten mitentwickelte Modul gelegt. Die gemeinsame Diskussion bietet die Möglichkeit, konkrete Situationen, die Verhaltens- und Zustandsänderungen auslösen, zu erarbeiten und gegebenenfalls auch bestehende Abhängigkeiten aufzudecken.

4.2.2 Erkenntnisse aus den Gesprächen

Die Durchführung von drei Expertengesprächen mit Doktoranden von der Technischen Universität Darmstadt haben die erste Iteration des entwickelten Verfahrens abgeschlossen. Die Dauer der Expertenbefragung betrug jeweils 60 Minuten.

Im ersten Gespräch stand die FTR mit den dort überwachten Zustandsvariablen sowie den dort möglichen auftretenden Verhaltens- und Zustandsänderungen im Mittelpunkt der Diskussion. Solche Zustandsänderungen in Signalen sind z.B. der komplette Ausfall eines Signals, ein wechselndes Signal oder ein Sprung in einem Signal. Ein solcher Sprung entsteht beispielsweise, wenn plötzlich auf Systemebene ein Objekt, z.B. ein Mensch, erkannt wird, da sich die Solltrajektorie plötzlich ändert. Weitere Auslöser von Zustands- und Verhaltensänderungen in der FTR sind der Kurswinkel bei Kreisfahrten oder engem Abbiegen und Inkonsistenzen in der Solltrajektorie. Eine weitere relevante Erkenntnis ist, dass ein Sollschwimmwinkel von 90° eine Abfrage im Softwarecode der FTR auslöst und ein anderer Teil vom Code aktiviert wird und abläuft. Ein Sollschwimmwinkel von 90° ist der Auslöser für das seitliche Parkmanöver. In der Diskussion ist auch ein Phänomen während Testfahrten mit einem der UNICAR*agil*-Fahrzeuge angesprochen worden. Bei diesem Phänomen handelt es sich um ein Ruckeln, welches durch einen sprungförmigen Verlauf der Drehmomentgrenzen hervorgerufen wird. Die Erforschung des Auslösers für diese Verhaltensänderung ist dann anschließend anhand des Abhängigkeitsnetzes erfolgt.

Beim zweiten Gespräch lag der Fokus auf der FSA. Zuerst wurden die funktionalen Einschränkungen der primären Fahrfunktion besprochen, welche diese Rückfallebene auslösen, und danach wurde auf mögliche Auslösungszenarien eingegangen, welche Zustands- und Verhaltensänderungen zur Folge haben. Auslöser sind z.B. ein fehlender Sensor- oder Karteneingang, Softwareabstürze in der Verarbeitungskette, ein Spannungs- oder Netzwerkausfall am Steuergerät und die Unterbrechung der Verbindung von Daten. Neben diesen Auslösern sind im Softwarecode mehrere Trigger mit unterschiedlichen Auslösungsbedingungen vorhanden. Ein Trigger startet alle 100 ms die Fusionierung der Objektlisten. Ein weiterer Trigger löst bei einer neuen Notbahn die Interpolation der Notbahn aus. Bei der Trajektoriengenerierung ist ebenfalls ein Trigger eingesetzt. Dieser löst bei einer neuen Objektliste, einer neuen Notbahn oder spätestens alle 100 ms die Generierung der Trajektorie aus.

Im letzten Gespräch waren die Themen die FZS sowie die Zustands- und Verhaltensänderungen aufgrund von dortigen Abhängigkeiten. Ein Szenario für die FZS ist die Abschattung des GNSS, z.B. eine Fahrt durch den Tunnel. Da die Inertial- und die Odometriesensorik bei diesem Szenario keine absolute Position ausgeben, kommt es zu einem Drift in der geschätzten Pose von der FZS. Die Abweichung ist longitudinal nahezu linear und lateral quadratisch. Bei der Ausfahrt aus dem Tunnel erfolgt ein Sprung in der geschätzten Pose der FZS.

In jedem der drei Expertengespräche haben die Experten den genauen Ablauf ihrer Module vorgestellt. Die dabei neu gewonnenen Informationen sowie Erkenntnisse über auftretende Zustands- und Verhaltensänderungen und Hinweise zu Abhängigkeiten aus den Diskussionen wurden in einer zweiten Iteration in das System modelliert und das Abhängigkeitsnetz erweitert.

4.3 Ergebnisse – Erstes Verfahren

Im Folgenden werden die Ergebnisse der Anwendung des ersten Verfahrens nach zwei durchgeführten Iterationen vorgestellt. Dazu gehören die definierten Abhängigkeitskategorien, das aufgebaute Abhängigkeitsnetz, der entwickelte Abhängigkeitssteckbrief und die ermittelten Abhängigkeiten. Die Bewertung dieser Ergebnisse schließt das Kapitel 4 ab.

4.3.1 Abhängigkeitskategorien

Anhand der Abhängigkeiten, die in Form eines Abhängigkeitsnetzes vorliegen, werden verschiedene Abhängigkeitskategorien definiert. Diese Definition basiert auf den Abhängigkeitstypen nach Pohl[116], Dahlstedt et al.[117] und Zhang et al.[118]. In Tabelle 4-1 sind die definierten Abhängigkeitskategorien mit ihrer jeweiligen Definition und Beispielen aufgelistet.

Tabelle 4-1: Definition der Abhängigkeitskategorien und Beispiele für UNICAR*agil*

Datenbasierte Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die eine Übertragung von Daten repräsentiert. (z.B. Sollschnittstellen der Signale)
Informationsbasierte Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die eine Übertragung von unterschwelligen Informationen, die vom Empfänger subjektiv interpretiert werden, repräsentiert. (z.B. Notbahn, Sollschwimmwinkel von 90°)
Inter-Datenbasierte Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die bei der Wechselwirkung von verschiedenen Daten relevant ist. (z.B. Konsistenz in den Eingangssignalen)
Datentransferbasierte Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die auf Einflüsse beim Austausch von Daten basiert. (z.B. Verbindung zu Sensorschnittstellen)
Systemdesignbasierte Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die auf Einflussfaktoren des vorliegenden Systems basiert. Bei diesen Einflussfaktoren handelt es sich um Einflüsse aufgrund der Eigenschaften und technischen Details der gewählten Komponenten wie der Aktorik, der Sensorik oder auch des Gesamtfahrzeugs. (z.B. Verzugszeit der Aktorik, Sensitivität der Sensoren)
Ressourcenbasierte Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die notwendige Mittel und Merkmale für das System oder einzelne Module bereitstellt. (z.B. Rechenleistung, Internetverbindung)
Codebasierte Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die im Softwarecode definiert ist. (z.B. Sollschwimmwinkel, Trigger)
Systemunabhängige Abhängigkeit	Beschreibt eine Beziehung zwischen zwei Schnittstellen, die die Einwirkung von äußeren Einflussfaktoren auf das System oder einzelne Module berücksichtigt. (z.B. Straßenprofil, *Policy Arbitrator*)

Die datenbasierten Abhängigkeiten basieren auf den Datenflüssen, die durch die Sollschnittstellen definiert sind. Diese Abhängigkeitskategorie wird durch die informationsbasierten Abhängigkeiten ergänzt. Diese Kategorie existiert parallel zu den datenbasierten Abhängigkeiten, denn diese beschreibt den Datenfluss von unterschwelligen bzw. nicht offensichtlichen Informationen im System.

[116] Pohl, K.: Process-centered requirements engineering (1996).

[117] Dahlstedt, Å. G.; Persson, A.: Requirements Interdependencies and Interdependency Types (2005), S. 100 ff.

[118] Zhang, H. et al.: Dependencies in Software Requirements (2014), S. 10 f.

Eine Überlappung bei den datenbasierten und informationsbasierten Abhängigkeiten liegt zum einen darin begründet, dass neben der Übermittlung von Daten auch eine Interpretation dieser Daten vom Empfänger erfolgt und weitere Informationen aus den Daten gewonnen werden, die nicht offensichtlich übertragen werden. Zum anderen ist ein Grund für eine solche Überlappung fehlerhaft spezifizierte Sollschnittstellen, welche gegebenenfalls nicht alle notwendigen Informationen übertragen. Neben diesen auf einer Datenübertragung basierenden Abhängigkeiten existieren im Projekt UNICAR*agil* zudem die auf Kontrollbeziehungen basierenden Abhängigkeiten. Da die Merkmale dieser Kontrollbeziehung und der Kontext der jeweiligen sehr verschieden sind, erfolgt die Einteilung in sechs Kategorien, die inter-datenbasierten, datentransferbasierten, systemdesignbasierten, ressourcenbasierten, codebasierten und systemunabhängigen Abhängigkeiten. Diese insgesamt acht definierten Abhängigkeitskategorien decken somit die im Projekt UNICAR*agil* vorhandenen Abhängigkeiten vollständig ab und sind in Abbildung 4-6 veranschaulicht.

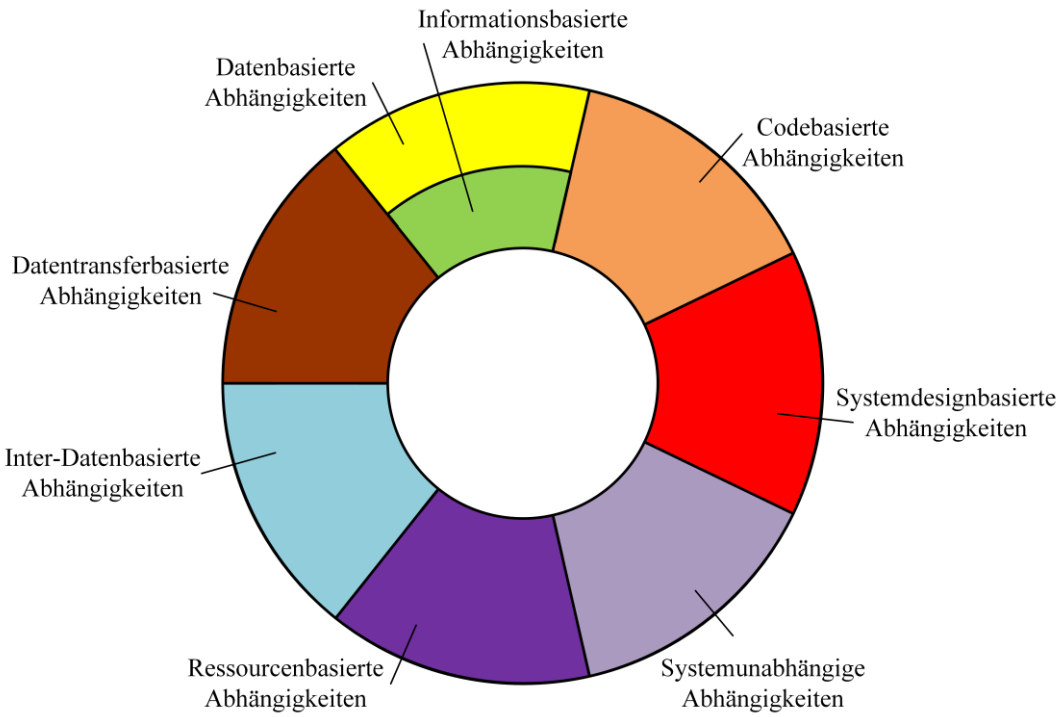

Abbildung 4-6: Vollständigkeit der definierten Abhängigkeitskategorien in UNICAR*agil*

4.3.2 Abhängigkeitsnetz

Das aufgebaute Abhängigkeitsnetz ist aufgrund der Dimensionen nicht vollständig in dieser Ausarbeitung abgebildet. Die Systemmodellierung und der Aufbau des Abhängigkeitsnetzes sind mit der Software *yED Graph Editor*[119] erfolgt. Dies ist ein kostenloses Visualisierungsprogramm zum Erstellen von Diagrammen. Die Datei mit dem entwickelten Abhängigkeitsnetz wird mit der Ausarbeitung abgegeben.

[119] yWorks GmbH: yEd.

In Abschnitt 4.1 sind die Systemmodellierung und der Aufbau des Abhängigkeitsnetzes anhand der FTR während der ersten Iteration beschrieben. Nach den neuen Erkenntnissen aus den Expertengesprächen erfolgten in einer zweiten Iteration die Fortführung der Systemmodellierung und die Erweiterung des Abhängigkeitsnetzes sowie die Anpassung der rautenförmigen Pfeile an den definierten Typ von Pfeil der jeweiligen Abhängigkeitskategorie. Der Ausschnitt für die FTR aus dem resultierenden Abhängigkeitsnetz ist in Abbildung 4-7 dargestellt. Die dazugehörige Legende ist in Abbildung 4-8 dargestellt.

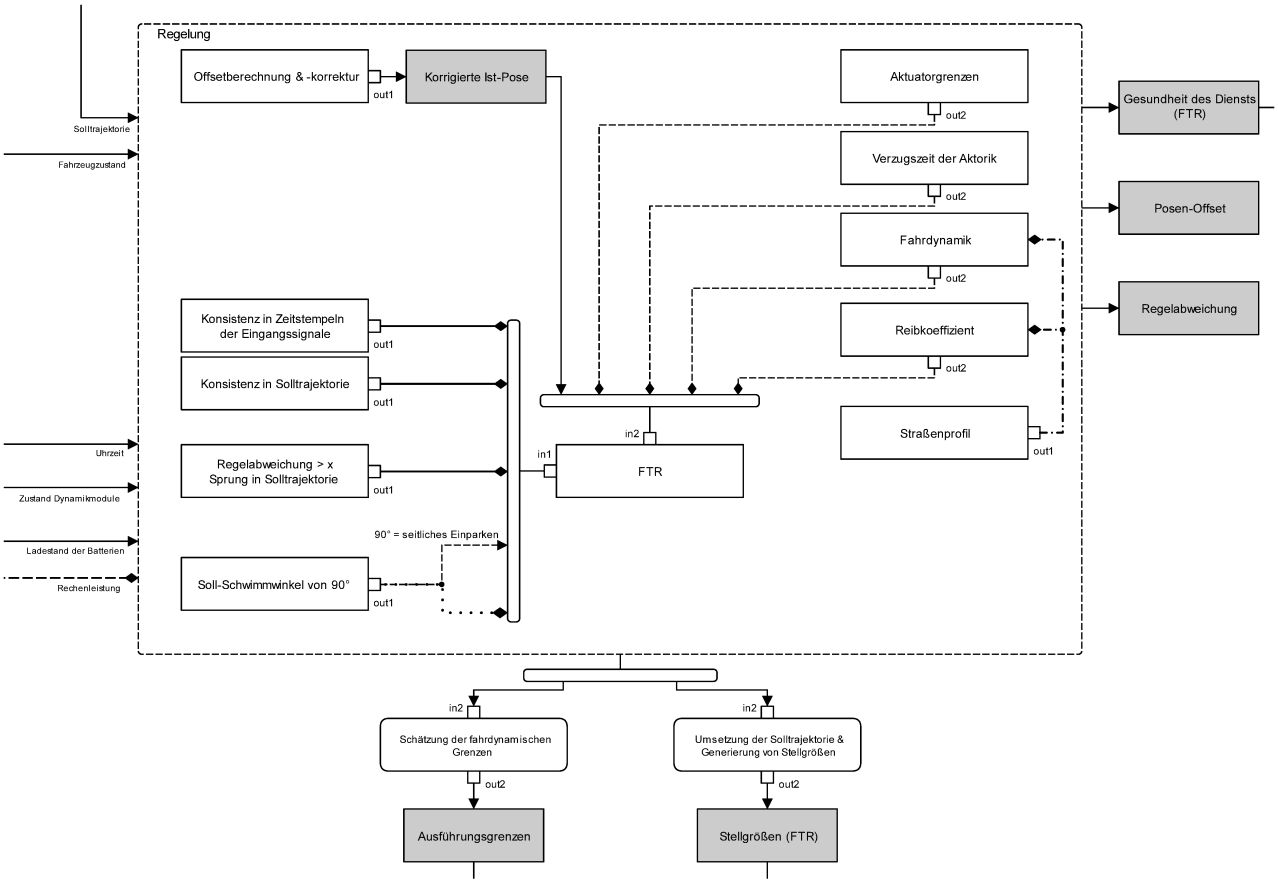

Abbildung 4-7: Abhängigkeitsnetz (Ausschnitt FTR)

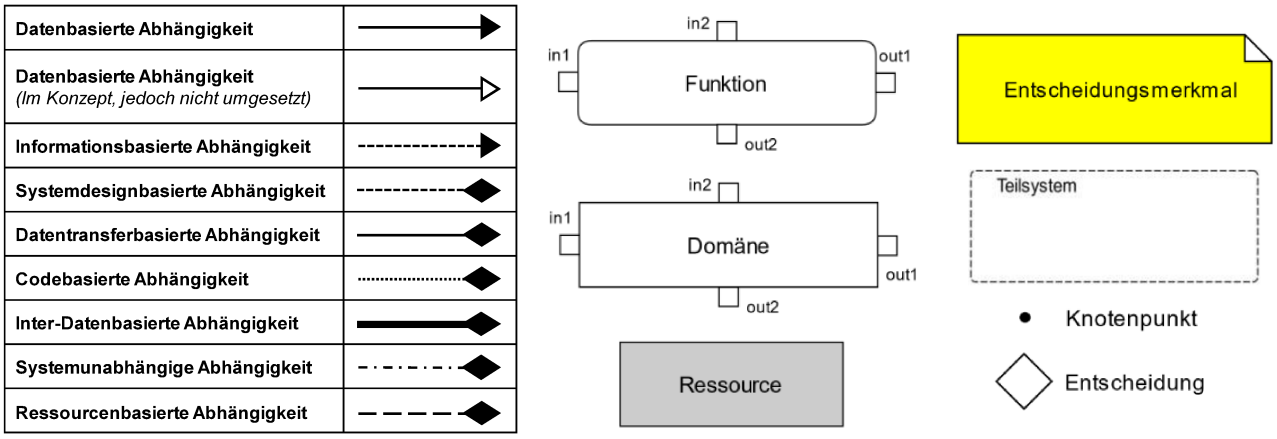

Abbildung 4-8: Legende für Abhängigkeitsnetz

Eine Domäne, die aus den neu gewonnenen Erkenntnissen stammt und auch in weiteren Modulen, wie beispielsweise der FZS und der FSA, modelliert ist, ist die Konsistenz in den Zeitstempeln der Eingangssignale. Diese ist relevant, damit kein unerwünschtes Systemverhalten auftritt. Eine weitere Voraussetzung für die Erbringung der Funktionen der FTR ist die Konsistenz in der Solltrajektorie. In dem durchgeführten Expertengespräch mit dem Fokus auf die FTR ist der Sollwinkel von 90° als weiterer Auslöser von Zustands- und Verhaltensänderungen identifiziert und daher als weitere Domäne hinzugefügt worden.

Der Auslöser für das zudem im Expertengespräch mit Fokus auf die FTR angesprochene Ruckeln, welches auf einen sprungförmigen Verlauf der Drehmomentgrenzen zurückzuführen ist, ist anhand des Abhängigkeitsnetzes analysiert worden. Dabei handelt es sich um über mehrere Module verkettete Abhängigkeiten. Der sprungförmige Verlauf der Drehmomentgrenzen wird durch eine wechselnde Momentenvorgabe in der FTR ausgelöst. Eine mögliche Erklärung ist, dass die wechselnde Momentenvorgabe durch eine wechselnde Trajektorie im VTP hervorgerufen wird. Diese wechselnde Trajektorie entsteht, wenn ein Objekt durch das Sensormodul kurzzeitig erkannt wird, dann für kurze Zeit nicht und danach die Erkennung des Objekts erneut ausgelöst wird. Demzufolge wird die Regelabweichung, die zu einem Sprung in der Solltrajektorie führt, als ein Auslöser für Änderungen in der FTR erkannt und daher hier als weitere Domäne berücksichtigt.

Die Anpassung der Kontrollbeziehungen der fünf Domänen auf der rechten Seite in Abbildung 4-7aus der ersten Iteration an die definierten Abhängigkeitskategorien sieht wie folgt aus. Bei den Aktuatorgrenzen, der Verzugszeit der Aktorik, der Fahrdynamik und dem Reibkoeffizienten handelt es sich um systemdesignbasierte Abhängigkeiten, da es sich bei diesen Phänomenen um Einflussfaktoren aufgrund des vorliegenden Systems handelt. Das Straßenprofil beschreibt eine systemunabhängige Abhängigkeit.

Bei den in der zweiten Iteration dazugekommenen Kontrollbeziehungen werden ebenfalls Anpassungen an den Pfeiltyp der Abhängigkeitskategorie vorgenommen. Bei dem Sollschwimmwinkel von 90° liegen zwei Kontrollbeziehungen vor. Zum einen wird im Softwarecode ein anderer Codeabschnitt aktiviert und zum anderen wird aus dem Eingangssignal bei einem Sollwinkel von 90° ein durchzuführendes seitliches Parkmanöver interpretiert. Infolgedessen liegen eine codebasierte und eine informationsbasierte Abhängigkeit vor. Bei den anderen drei beschriebenen Domänen, der Konsistenz in den Zeitstempeln der Eingangssignale, der Konsistenz in der Solltrajektorie sowie der Regelabweichung, die zu einem Sprung in der Solltrajektorie führt, handelt es sich um inter-datenbasierte Abhängigkeiten.

Der Fokus bei der Ermittlung und Darstellung von Abhängigkeiten liegt in dieser Arbeit neben der FTR auf der FZS, dem VTP sowie der FSA. Die Ausschnitte aus dem Abhängigkeitsnetz für diese drei Module befinden sich im Anhang (für die FZS siehe Abbildung A-1, für den VTP siehe Abbildung A-2 und für die FSA siehe Abbildung A-5). Die Ausschnitte der TVV, der Plattformsensorik und des Dynamikmoduls sind ebenfalls im Anhang hinterlegt (für die TVV siehe Abbildung A-3, für die Plattformsensorik siehe Abbildung A-4 und für das Dynamikmodul siehe Abbildung A-6)

4.3.3 Abhängigkeitssteckbrief

Die Darstellung der Abhängigkeiten in Form eines Abhängigkeitsnetzes eignet sich für eine nachfolgende Testfallgenerierung nicht. Die Abhängigkeiten sind über das ganze Netz verstreut. Zudem ist die Möglichkeit, weitere Eigenschaften oder auch Informationen zu Testfällen zu ergänzen, nicht gegeben.

Daher erfolgt nach dem Aufbau des Abhängigkeitsnetzes und der Definition der Abhängigkeitskategorien der Entwurf eines sogenannten Abhängigkeitssteckbriefs. Dieser Steckbrief wird für jede Abhängigkeit erstellt und liefert eine Übersicht über die Merkmale und Eigenschaften der jeweiligen Abhängigkeit. Der entworfene Abhängigkeitssteckbrief ist in Abbildung 4-9 dargestellt.

Die ID ist eine Identifikationsnummer, die für jede Abhängigkeit unterschiedlich ist. Infolgedessen ist eine eindeutige Zuweisung der Abhängigkeiten gegeben. Der Name liefert Informationen, um welche Abhängigkeit es sich handelt. Die Angabe der beteiligten Module dient der Orientierung, wo im System sich die Abhängigkeit befindet und zwischen welchen Modulen bzw. innerhalb welches Moduls diese Abhängigkeit besteht. Bei der Kategorie wird die für die Abhängigkeit zutreffende Abhängigkeitskategorie angegeben. Die abschließende Beschreibung bietet Platz für die Angabe weiterer Details oder relevante Anmerkungen zu der Abhängigkeit wie z.B., dass die Abhängigkeit im Konzept besteht, aber nicht umgesetzt wurde. Der Abhängigkeitssteckbrief bietet zudem Platz für weitere Angaben, beispielsweise Anforderungen an nachfolgende Testfälle.

Abhängigkeitssteckbrief

- ID:
- Name:
- Beteiligte Module:
- Kategorie:
- Beschreibung:

Abbildung 4-9: Entwickelter Entwurf des Abhängigkeitssteckbriefs

Eine Erstellung des Steckbriefs für jede Abhängigkeit führt zu vielen Einzeldokumenten, da allein für die Module FZS, VTP, FSA und FTR insgesamt über 100 Abhängigkeiten identifiziert wurden. Dementsprechend wird die erstellte Liste mit den Abhängigkeiten an den Aufbau des Abhängigkeitssteckbriefs angepasst.

4.3.4 Ermittelte Abhängigkeiten

Der Ausschnitt aus der erstellten Liste mit den in der Systemarchitektur im Projekt UNICAR*agil* ermittelten Abhängigkeiten für die FTR folgt in Tabelle 4-2. Die darüber hinaus identifizierten Abhängigkeiten sind in Tabellen für die jeweiligen Module im Anhang aufgelistet (FZS siehe Tabelle A-1, VTP siehe Tabelle A-2, TVV siehe Tabelle A-3, Plattformsensorik siehe Tabelle A-4, FSA siehe Tabelle A-5 und Dynamikmodul siehe Tabelle A-6). Bei der Erstellung der Abhängigkeitsliste sind weitere Abkürzungen zur Übersichtlichkeit eingeführt worden. Dieses zusätzliche Abkürzungsverzeichnis ist nachfolgend aufgeführt.

Abkürzungsverzeichnis für die Abhängigkeitsliste:

BMM	Betriebsmodusmanagement
BTM	Batteriemanagement
CLO	Cloud
DLTWS	Direkte Leitwartensteuerung
DYM	Dynamikmodule
ECU	Electronic Control Unit
ENU	East North UP (Kartesische Koordinaten)
FSA	Fahrschlauchabsicherung
FTR	Fahrdynamik- und Trajektorienregelung
FZS	Fahrdynamikzustandsschätzung
GCS	Geographic Coordinate System (Kugelkoordinaten)
LOF	Lokalisierungsfusion
LTE	Long Term Evolution-Modem
LTW	Leitwarte
PTS	Plattformsensorik
STG	Steuergerät
TSLTW	Trajektoriensteuerung Leitwarte
TVV	Trajektorienvorverarbeitung
UGM	Umgebungsmodell
VLOF	Videobasierte Lokalisierungsfusion
VTP	Verhaltens- und Trajektorienplaner

Da die in der folgenden Tabelle aufgeführten Abhängigkeiten der FTR bereits bei der Beschreibung der Zwischenschritte für die Systemmodellierung und den Aufbau des Abhängigkeitsnetzes in Abschnitt 4.1 und des Abhängigkeitsnetzes in Abschnitt 4.3.2 erläutert sind, wird hier auf eine erneute Erklärung der Abhängigkeiten verzichtet.

Tabelle 4-2: Ermittelte Abhängigkeiten – Fahrdynamik- und Trajektorienregelung

ID	Abhängigkeit	Beteiligte Module	Gruppe	Beschreibung
TVV_FTR_001	Solltrajektorie von der TVV	TVV - FTR	Datenbasierte Abhängigkeit	
FSA_FTR_002	Nottrajektorie aus der FSA	FSA - FTR	Datenbasierte Abhängigkeit	
FZS_FTR_003	Fahrzeugzustand	FZS - FTR	Datenbasierte Abhängigkeit	
FTR_FTR_004	Uhrzeit	FTR	Datenbasierte Abhängigkeit	
DYM_FTR_005	Zustand Dynamikmodule	DYM - FTR	Datenbasierte Abhängigkeit	
BTM_FTR_006	Ladestand der Batterie	BTM - FTR	Datenbasierte Abhängigkeit	
FTR_FTR_007	Korrigierte Ist-Pose aus Off-setberechnung & -korrektur	FTR - FTR	Datenbasierte Abhängigkeit	
FTR_FTR_008	Daten aus FTR für Schätzung der fahrdynamischen Grenzen	FTR - FTR	Datenbasierte Abhängigkeit	
FTR_FTR_009	Daten aus FTR für Umsetzung der Solltrajektorie & Generie-rung von Stellgrößen	FTR - FTR	Datenbasierte Abhängigkeit	
FTR_FTR_010	Sollschwimmwinkel von 90° wird als seitliches Einparkma-növer interpretiert	FTR - FTR	Informationsbasierte Abhängigkeit	
FTR_FTR_011	Konsistenz in Zeitstempeln der Eingangssignale für FTR	FTR - FTR	Inter-Datenbasierte Abhängigkeit	
FTR_FTR_012	Konsistenz in Solltrajektorie für FTR	FTR - FTR	Inter-Datenbasierte Abhängigkeit	
FTR_FTR_013	Regelabweichung > x Sprung in Solltrajektorie	FTR - FTR	Inter-Datenbasierte Abhängigkeit	
FTR_FTR_014	Verzugszeit der Aktorik	FTR - FTR	Systemdesignbasierte Abhängigkeit	
FTR_FTR_015	Fahrdynamik	FTR - FTR	Systemdesignbasierte Abhängigkeit	
FTR_FTR_016	Aktuatorgrenzen	FTR - FTR	Systemdesignbasierte Abhängigkeit	
FTR_FTR_017	Reibkoeffizient	FTR - FTR	Systemdesignbasierte Abhängigkeit	
STG_FTR_018	Rechenleistung	Stammhirn Steu-ergerät - FTR	Ressourcenbasierte Abhängigkeit	
FTR_FTR_019	Sollschwimmwinkel von 90°	FTR - FTR	Codebasierte Abhängigkeit	
FTR_FTR_020	Straßenprofil	FTR - FTR	Systemunabhängige Abhängigkeit	

4.3.5 Bewertung der Ergebnisse – Erstes Verfahren

Zum Schluss erfolgt die Bewertung der Ergebnisse aus der Anwendung des ersten Verfahrens. Dafür wird das Verfahren anhand der Festforderungen und Wünsche aus der Anforderungsliste in Abschnitt 3.1 bewertet und somit abschließend das entwickelte Verfahren zur Ermittlung und Darstellung von Abhängigkeiten verifiziert. In Abschnitt 7.1 folgt dann die Diskussion der Einflussfaktoren auf das Verfahren.

Die Anwendung des Verfahrens auf die Systemarchitektur eines UNICAR*agil*-Fahrzeugs zeigt, dass das Verfahren sich für eine große Anzahl an Abhängigkeiten eignet. Bei der Ermittlung der Abhängigkeiten für die FZS, den VTP, die FSA und die FTR liegen bei den über 100 Abhängigkeiten keine Einschränkungen vor. Lediglich ist anzumerken, dass bei der Systemmodellierung und dem Aufbau des Abhängigkeitsnetzes auf ausreichend Platz zwischen den Modulen zu achten ist und bei der Arbeitsweise strukturiert und geordnet vorzugehen ist, um nicht die Übersicht im aufgebauten Abhängigkeitsnetz zu verlieren. Folglich ist die erste Festforderung erfüllt.

Bei der Anwendung des Verfahrens in der Systemarchitektur vom Projekt UNICAR*agil* ist zudem festzustellen, dass das Verfahren einen hohen Detaillierungsgrad aufweist. Die einzelnen Elemente der Architektur, die Auslöser für Zustands- und Verhaltensänderungen sowie die Verbindungen innerhalb des Systems sind mit dem Modellierungswerkzeug der *IRML*-Methode darstellbar. Aus diesem Grund ist diese Festforderung ebenfalls erfüllt.

Mit der Einführung von Entscheidungsmerkmalen und Entscheidungsknoten ist die Ermittlung und Darstellung nicht nur von statischen, sondern auch von dynamischen Abhängigkeiten möglich. So lassen sich beispielweise die dynamischen Abhängigkeiten je nach Betriebsmodus aufgrund der dynamischen Verschaltung durch den Orchestrator abbilden. Aus diesem Grund ist die dritte Festforderung erfüllt.

Durch die Anwendung des Verfahrens wird die nächste Festforderung geprüft. Die Abhängigkeiten z.B. auf der Hardware-, der Softwareebene sowie der funktionalen Ebene lassen sich identifizieren und demzufolge ist bestätigt, dass sich das Verfahren auch für die Ermittlung von Abhängigkeiten auf verschiedenen Architektursichten eignet. Demnach ist die vierte Festforderung ebenso erfüllt.

In der fünften Festforderung wird gefordert, dass sich das Verfahren für die Ermittlung sowohl von direkten als auch von indirekten Abhängigkeiten eignet. Nach der Anwendung des Verfahrens ist anzumerken, dass direkte Abhängigkeiten kein Problem bei der Ermittlung darstellen, jedoch die Ermittlung von indirekten Abhängigkeiten bei zwei durchgeführten Iterationen sich als schwierig herausgestellt hat. Mit den indirekten Abhängigkeiten sind Abhängigkeiten gemeint, die bei einer Übertragung eines Datenflusses vorkommen und durch die Interpretation der übermittelten Daten durch den Empfänger auftreten. Infolgedessen werden unterschwellige Informationen mit den Daten übermittelt. Ein Beispiel aus dieser Arbeit ist z.B. die Notbahn, die von der FSA als befahrbare Bahn interpretiert wird. Diese Abhängigkeiten werden in dieser Arbeit als informationsbasierte Abhängigkeiten definiert. Generell sind direkte und auch indirekte Abhängigkeiten mit diesem Verfahren identifizierbar und daher ist diese Festforderung ebenfalls erfüllt. Auf die indirekten Abhängigkeiten und

einen Ansatz zur Verbesserung des Verfahrens hinsichtlich dieser Unzugänglichkeit wird in Kapitel 7 näher eingegangen.

Die Anwendung des Verfahrens auf die Systemarchitektur hat zudem gezeigt, inwiefern sich unbekannte, also im Vorhinein nicht offensichtliche Abhängigkeiten ermitteln lassen. In der ersten Iteration wird angefangen, das System zu modellieren und das Abhängigkeitsnetz aufzubauen mit Fokus auf die Solldatenschnittstellen und den dort vorliegenden Abhängigkeiten. Um neben den Erkenntnissen aus Expertengesprächen in einer zweiten Iteration die unbekannten Abhängigkeiten zu identifizieren, ist folgende Frage eine Hilfestellung: „Was kann funktional schiefgehen?“ Dazu bietet es sich an, diese Frage für jedes Modul einzeln zu beantworten. Mögliche Ursachen sind beispielsweise, dass ein Objekt nicht erkannt wird, falsch detektiert wird oder ein kompletter Ausfall vorliegt, da die Übertragung der Daten unterbrochen wird, weil z.B. ein elektronisches Bauteil aufgrund von Verschleiß versagt. Mit dieser Fragestellung sind beispielsweise die Abhängigkeiten, die zum sprungförmigen Verlauf in den Aktuatorgrenzen führen, ermittelt worden. Folglich ist diese Festforderung ebenfalls erfüllt.

Bei der Anwendung des Verfahrens auf die Systemarchitektur vom Projekt UNICAR*agil* hat sich gezeigt, dass neben den Datenflüssen verschiedene Kontrollbeziehungen vorliegen, welche sich in unterschiedliche Kategorien von Abhängigkeiten unterteilen lassen. Bei der Darstellung werden unterschiedliche Pfeiltypen eingeführt, um eine Unterscheidung der Kategorien zu ermöglichen und um schließlich die Übersicht zu behalten. Dementsprechend ist die siebte Festforderung erfüllt.

Das entwickelte Verfahren basiert auf der *IRML*-Methode. *IRML* ist eine grafische Modellierungssprache, entwickelt für kritische Infrastrukturen. Daher ist die Festforderung, dass das Verfahren auf einer grafischen Modellierungssprache basiert, erfüllt. Bei dem entwickelten Verfahren sind zusätzlich Anpassungen vorgenommen worden, um zudem den Wunsch, dass die grafische Modellierungssprache des Verfahrens *UML* ist, zu erfüllen.

Die letzte Festforderung an das Verfahren ist, dass sich dieses für einen iterativen Prozess und eine nachträgliche Ermittlung weiterer Abhängigkeiten eignet. Die Anwendung des Verfahrens hat gezeigt, dass ein iteratives Vorgehen umsetzbar und auch empfohlen ist, um alle Abhängigkeiten zu identifizieren. Somit ist diese Festforderung auch erfüllt.

Um den zweiten Wunsch aus der Anforderungsliste, dass sich die Darstellung für eine nachfolgende Testfallgenerierung eignet, zu erfüllen, ist eine weitere Darstellungsform neben dem Abhängigkeitsnetz entwickelt worden. Die Liste mit dem Aufbau des Abhängigkeitssteckbriefs ermöglicht eine Erweiterung um zusätzliche Angaben, beispielsweise Anforderungen an nachfolgende Testfälle oder Beispiele für Szenarien aus den Expertengesprächen. Nach der Definition und Durchführung lässt sich die Erfüllung dieses Wunsches endgültig bewerten.

Nach der Anwendung des Verfahrens ist die Bewertung des letzten Wunsches hinsichtlich der Komplexität, Nachvollziehbarkeit sowie dem Aufwand bei der Einarbeitung und Fortführung des Verfahrens durchführbar. Aufgrund der Anpassung der Modellierungssprache an den Standard *UML* und der Ähnlichkeit bei den Anwendungsgebieten der *IRML*-Methode und des hier entwickelten Verfahrens ist eine Basis für eine leichte Einarbeitung und Fortführung gegeben. Im Detail lässt sich dieser

Wunsch erst im Nachhinein an diese Arbeit bewerten. Dennoch wird der Wunsch als erfüllt angesehen, da das Ziel dieses Wunsches angestrebt wurde.

Zusammenfassend lässt sich sagen, dass alle Festforderungen aus Tabelle 3-1 von dem entwickelten Verfahren erfüllt werden und auch die drei Wünsche, soweit es zum jetzigen Zeitpunkt bewertbar ist, ebenfalls erfüllt werden. Folglich ist festzustellen, dass das entwickelte Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung eine erfolgversprechende Methode zur Lösung der Aufgabenstellung darstellt.

5 Entwicklung des zweiten Verfahrens: Definition von Testfällen

Das folgende Kapitel beschreibt die Entwicklung eines Verfahrens zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten. Auf Basis des in diesem Kapitel entwickelten Verfahrens und den in Kapitel 4 identifizierten Abhängigkeiten in der Systemarchitektur vom Projekt UNICAR*agil* werden in Kapitel 6 zugehörige Testfälle ermittelt, durchgeführt und abschließend bewertet.

Zunächst werden in Abschnitt 5.1 die Anforderungen an das zu entwickelnde Verfahren definiert. In Abschnitt 5.2 wird ein Überblick über den Stand der Forschung bei der Testfallgenerierung gegeben. Anhand der in Abschnitt 5.1 definierten Anforderungen folgen in Abschnitt 5.3 die Bewertung der recherchierten Ansätze und die Entscheidung für ein Verfahren aus dem Lösungsraum. Im Anschluss daran wird anhand des ausgewählten Verfahrens und der entsprechenden Anpassungen das abgeleitete Verfahren zur Definition von Testfällen beschrieben.

5.1 Anforderungsanalyse – Zweites Verfahren

Im folgenden Abschnitt werden als erster Schritt der Verfahrensentwicklung Anforderungen an das zu entwickelnde Verfahren zur Definition der Testfälle auf Grundlage ermittelter Abhängigkeiten definiert. Die Anforderungen resultieren aus der Einbeziehung der ermittelten Abhängigkeiten, der Verwendung des szenariobasierten Testansatzes für automatisierte Fahrfunktionen sowie der bestehenden Simulationsumgebung, die verwendet wird. Für die Eingrenzung des Lösungsraums der in der Literatur vorhandenen Verfahren, für die Verifizierung sowie für die anschließende Wahl eines Verfahrens werden die Anforderungen hinzugezogen.

Im Folgenden werden zwei Arten von Anforderungen verwendet: Festforderungen und Wünsche. Die Definition dieser beiden Anforderungstypen ist in Abschnitt 3.1 beschrieben.

Ausgangspunkt dieses zu entwickelnden Verfahrens sind die zuvor ermittelten Abhängigkeiten. Daraus ergibt sich die **Festforderung**

1. ***Die definierten Testfälle zeigen die mit dem vorherigen Verfahren identifizierten Abhängigkeiten auf.***

Das Ziel der Definition und Durchführung der Testfälle ist in erster Linie das Aufzeigen der identifizierten Abhängigkeiten, obwohl sie vorher nicht offensichtlich waren, und damit die Validierung des Verfahrens zur Ermittlung und Darstellung von Abhängigkeiten.

Weiterhin ergibt sich aus den zuvor ermittelten Abhängigkeiten die **Festforderung**

2. ***Das Verfahren eignet sich für Abhängigkeiten verschiedener Kategorien.***

Aufgrund der Systemarchitektur vom Projekt UNICAR*agil* existieren ebenen- und hardwareübergreifende Abhängigkeiten. Folglich handelt es sich bei den Abhängigkeiten um unterschiedliche Typen.

Neben dem vordergründigen Ziel, das Aufzeigen der identifizierten Abhängigkeiten, beschreibt die **Festforderung**

3. ***Das Verfahren eignet sich zur Aufdeckung von weiteren unbekannten Abhängigkeiten***

ein weiteres Ziel dieses Verfahrens. Für die individuelle Absicherung der Module ist die Aufdeckung unbekannter Abhängigkeiten ebenfalls von hoher Relevanz, da sonst die angestrebten Modultests nicht umsetzbar sind.

Die nächste **Festforderung** basiert auf dem szenariobasierten Testansatz für automatisierte Fahrfunktionen und lautet

4. ***Die Definition von Testfällen basiert auf einzelnen Szenarien.***

Diese Anforderung geht auf die Herausforderungen beim notwendigen Sicherheitsnachweis für automatisierte Fahrfunktionen und -systeme ab der Automatisierungsstufe 3 zurück. Ein Ansatz zur Lösung ist der szenariobasierte Testansatz.

Die **Festforderung**

5. ***Das Verfahren basiert auf verschiedenen Arten der Szenariogenerierung***

ergänzt die zuvor aufgestellte Anforderung. Beim szenariobasierten Testansatz existieren mehrere Möglichkeiten der Szenariogenerierung in der Literatur. Diese sind z.B. die expertenbasierte, die anforderungsbasierte, die risikobasierte und die wissensbasierte Szenariogenerierung.[120,121,122] Die Verwendung verschiedener Konzepte zielt darauf ab, dass, angepasst an die identifizierten Abhängigkeiten, Testfälle definiert werden.

Da zusätzlich zur Definition der Testfälle auch die Durchführung dieser angestrebt wird, wird der **Wunsch**

6. ***Die mit dem Verfahren definierten Testfälle sind in der Simulationsumgebung und mit den bestehenden Simulationsmodellen umsetzbar***

formuliert. Da zum aktuellen Stand dieser Arbeit einzelne Komponenten in der Simulation nicht oder nicht vollständig umgesetzt sind, ist die Umsetzung aller definierten Testfälle nicht gewährleistet.

Aufgrund des iterativen Prozesses bei dem Verfahren zur Ermittlung und Darstellung von Abhängigkeiten ergibt sich die **Festforderung**

7. ***Das Verfahren eignet sich für einen iterativen Prozess und eine nachträgliche Definition weiterer Testfälle, um neue Erkenntnisse zu berücksichtigen.***

Mit einem ebenfalls iterativen Prozess bei der Definition der Testfälle wird z.B. ermöglicht, die in einer zweiten Iteration ermittelten Abhängigkeiten in Form von Testfällen aufzuzeigen und gegebenenfalls auch noch weitere unbekannte Abhängigkeiten aufzudecken.

Wie bei dem ersten Verfahren wird auch für das zweite Verfahren der **Wunsch**

---

[120] Bagschik, G. et al.: Ontology based Scene Creation (2018), S. 1.

[121] Althoff, M.; Lutz, S.: Generation of Safety-Critical Test Scenarios (2018), S. 1.

[122] Abbas, H. et al.: Requirement and risk-based scenarios (2017), S. 1 ff.

8. ***Das Verfahren weist eine geringe Komplexität auf, sodass eine leichte Nachvollziehbarkeit und ein geringer Aufwand bei der Einarbeitung und Fortführung gegeben sind***

formuliert. Demnach liegt bei diesem Verfahren der Schwerpunkt ebenfalls auf der Entwicklung einer Methodik und einer beispielhaften Durchführung von einzelnen Testfällen zur Validierung der gesamten entwickelten Methodik.

Alle genannten Anforderungen für das zweite Verfahren sind in Tabelle 5-1 zusammengefasst.

Tabelle 5-1: Anforderungen an das Verfahren zur Definition von Testfällen

	Anforderung	**Art**
1.	Die definierten Testfälle zeigen die mit dem vorherigen Verfahren identifizierten Abhängigkeiten auf.	Festforderung
2.	Das Verfahren eignet sich für Abhängigkeiten verschiedener Kategorien.	Festforderung
3.	Das Verfahren eignet sich zur Aufdeckung von weiteren unbekannten Abhängigkeiten.	Festforderung
4.	Die Definition von Testfällen basiert auf einzelnen Szenarien.	Festforderung
5.	Das Verfahren basiert auf verschiedenen Arten der Szenariogenerierung.	Festforderung
6.	Die mit dem Verfahren definierten Testfälle sind in der Simulationsumgebung und mit den bestehenden Simulationsmodellen umsetzbar.	Wunsch
7.	Das Verfahren eignet sich für einen iterativen Prozess und eine nachträgliche Definition weiterer Testfälle, um neue Erkenntnisse zu berücksichtigen.	Festforderung
8.	Das Verfahren weist eine geringe Komplexität auf, sodass eine leichte Nachvollziehbarkeit und ein geringer Aufwand bei der Einarbeitung und Fortführung gegeben sind.	Wunsch

5.2 Stand der Forschung – Zweites Verfahren

Im Folgenden wird ein Überblick über den Stand der Forschung bei der Szenariogenerierung gegeben. Wie bereits in den Grundlagen in Abschnitt 2.3.3 beschrieben, ist nach Wachenfeld et al.[123] der szenariobasierte Testansatz ein erfolgversprechender Ansatz zur Lösung der Herausforderungen beim Sicherheitsnachweis für die Freigabe von automatisierten Fahrsystemen ab der Automatisierungsstufe 3.[124] Dieser Testansatz wird nach Pfeffer[125] und auch Bagschik et al.[126] ebenfalls für den Test von automatisierten Fahrfunktionen und -systemen vorgeschlagen. Daher wird als Grundlage für das zu entwickelnde Verfahren zur Definition von Testfällen der szenariobasierte Testansatz verwendet. Nachfolgend liegt der Fokus dementsprechend auf dem szenariobasierten Testansatz und verschiedenen Ansätzen aus der Literatur zur Szenariogenerierung. Die Begriffserklärung der wesentlichen Bestandteile, wie das Szenario, die Szene und die Situation, für diesen Testansatz ist in Abschnitt 2.3.1 erfolgt.

[123] Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015), S. 4.

[124] Lippert, M. et al.: Bestehens- & Versagenskriterien für Tests (2020), S. 4.

[125] Pfeffer, R.: Diss., Funktionale Absicherung hochautomatisierter Fahrfunktionen (2020), S. 52.

[126] Bagschik, G. et al.: Wissensbasierte Szenariengenerierung für Betriebsszenarien (2018), S. 1.

5.2.1 Szenariogenerierung

In diesem Abschnitt wird der Stand der Forschung im Bereich der Szenariogenerierung für den szenariobasierten Testansatz vorgestellt. Der Schwerpunkt wird dabei auf vier Ansätze gelegt, da diese in der Literatur im Zusammenhang für den Test automatisierter Fahrfunktionen und -systeme Anwendung finden.

5.2.1.1 Expertenbasierte Szenariogenerierung

Die expertenbasierte Szenariogenerierung nutzt das Wissen von Experten, um auf relevante und notwendige Szenarien, die z.B. Unfälle verursachen oder zu Fehlern führen, zu schließen.[127] Diese expertenbasierte Szenariogenerierung bietet den Vorteil, mithilfe der Expertise von Experten Szenarien einschließlich der Szenen und enthaltene Elemente zu identifizieren, die schwer zu greifen und zu erkennen sind.[128a,129]

Ein Nachteil dieses Ansatzes zur Szenariogenerierung ist der hohe und manuelle Aufwand, der mit der Durchführung von Expertengesprächen verbunden ist. Des Weiteren ist bei einer expertenbasierten Szenariogenerierung zu beachten, dass der Grad der Detailliertheit der beschriebenen Szenarien vom Experten abhängt und sich je nach Szenario sehr unterscheidet.[128] Zudem ist anzumerken, dass es Limitationen bei der Anzahl der Szenarien gibt, da das den Experten zur Verfügung stehende Expertenwissen nicht für alle Szenarien reicht.[127]

5.2.1.2 Anforderungsbasierte Szenariogenerierung

Die anforderungsbasierte Szenariogenerierung nutzt Anforderungen, die an das Testobjekt gestellt werden, um Szenarien aufzubauen.[128b] Dazu wird zu Beginn eine Anforderungsanalyse durchgeführt. Bei diesen Anforderungen handelt es sich um technische Anforderungen, die physikalisch messbar sind, oder um funktionale Anforderungen. Unter einer technischen Anforderung wird beispielsweise der Abstand in Metern zu anderen Verkehrsteilnehmern verstanden. Bei einer funktionalen Anforderung wird lediglich ein Sicherheitsabstand zu den anderen Teilnehmern gefordert.[128c]

5.2.1.3 Risikobasierte Szenariogenerierung

Bei der risikobasierten Szenariogenerierung wird der Fokus bei der Analyse auf die kritischen Szenarien gelegt, die Risiken aufgrund der automatisierten Fahrfunktionen und -systeme darstellen.[127] Die Berechnung des Risikos ist die notwendige Voraussetzung und bildet den Anfang der Generierung der Szenarien. In der Literatur existieren diverse Ansätze zur Risikoermittlung. Bei einem Ansatz werden beispielsweise die Fahranforderungen und folglich das Risiko der Gefährdung des Testobjekts und gegebenenfalls der Testbeteiligten geschätzt.[130]

---

[127] Bagschik, G. et al.: Wissensbasierte Szenariengenerierung für Betriebsszenarien (2018), S. 1.

[128] Menzel, T. et al.: Scenarios for Test and Validation (2018), a: S. 2; b: S. 1 f; c: S. 3.

[129] Pfeffer, R.: Diss., Funktionale Absicherung hochautomatisierter Fahrfunktionen (2020), S. 93.

[130] BMWI: PEGASUS Method (2019), S. 15.

5.2.1.4 Wissensbasierte Szenariogenerierung

Die wissensbasierte Szenariogenerierung stützt sich auf die Verwendung von Wissen in Form von Gesetzen, Normen, Richtlinien oder verwendet die gewonnene Wissensbasis, welche z.B. mit der ontologiebasierten, der risikobasierten oder der expertenbasierten Szenariogenerierung aufgebaut wird.[131a,132a] Die Szenarien werden durch die Variation von Merkmalen und Limitationen der gegebenen Wissensbasis generiert.[132b] Vorteil dieses Ansatzes ist die große Vielfalt bei der Erstellung von Szenarien.[133a] Jedoch liegen Einschränkungen bei der Verwendung des Hintergrundwissens vor, da dieses Wissen vereinfacht und reduziert modelliert wird.[132c]

5.3 Bewertung des Lösungsraums und Methodik – Zweites Verfahren

Zur Beurteilung des szenariobasierten Testansatzes und der vier Ansätze zur Szenariogenerierung werden diese anhand der Anforderungen aus Abschnitt 5.1 evaluiert. Anschließend wird das Verfahren zur Definition von Testfällen mit dem ausgewählten Testkonzept beschrieben.

5.3.1 Verifizierung des Testprozesses

Der szenariobasierte Testansatz ermöglicht die Definition von Testfällen auf Basis von Szenarien für die zuvor identifizierten Abhängigkeiten. Zudem erfüllt dieser die aufgestellten Festforderungen. Die abschließende Verifizierung erfolgt nach der Anwendung des hier entwickelten Verfahrens in Kapitel 6.

In dieser Arbeit werden bereits zur Ermittlung und Darstellung der Abhängigkeiten in der Systemarchitektur vom Projekt UNICAR*agil* Expertenbefragungen durchgeführt. Bei diesen Gesprächen werden in Diskussionen mögliche Auslöser von Zustands- und Verhaltensänderungen analysiert und kritische Szenarien herausgearbeitet. Eine expertenbasierte Szenariogenerierung ist daher als ein Teil der Testfallgenerierung anzustreben, da darüber hinaus kein zusätzlicher Aufwand mit der Durchführung von Expertengesprächen verbunden ist.[132d] Demzufolge lässt sich der Vorteil dieses Ansatzes nutzen, Szenarien zu identifizieren, die schwer zu greifen und zu erkennen sind.[134] Der Nachteil, dass aufgrund der begrenzten Expertise die Anzahl der Szenarien limitiert ist, wird mit einem zweiten Ansatz zur Szenariogenerierung aufgehoben.[131b]

Die anforderungsbasierte Szenariogenerierung stellt einen zweiten Ansatz dar, um anhand zuvor ermittelter Abhängigkeiten Szenarien und schließlich Testfälle zu definieren. Dazu werden die einzelnen Abhängigkeiten analysiert und die Anforderungen an das Testobjekt herausgearbeitet. Die Definition von Anforderungen für jede einzelne Abhängigkeit erlaubt, dass die entwickelten Szenarien die Abhängigkeiten aufzeigen.[133b]

---

[131] BMWI: PEGASUS Method (2019), a: S. 7; b: S. 15.

[132] Bagschik, G. et al.: Wissensbasierte Szenariengenerierung für Betriebsszenarien (2018), a: S. 4; b: S. 5; c: S. 13; d: S. 1.

[133] Menzel, T. et al.: Scenarios for Test and Validation (2018), a: S. 7; b: S. 1 ff.

[134] Pfeffer, R.: Diss., Funktionale Absicherung hochautomatisierter Fahrfunktionen (2020), S. 93.

Die zuvor vorgestellten Ansätze zur risikobasierten und wissensbasierten Szenariogenerierung eignen sich nicht für die in dieser Arbeit gewünschte Anwendung. Für die risikobasierte Szenariogenerierung ist die Abschätzung des vorliegenden Risikos erforderlich.[135a] Für eine einzelne Anwendung der Berechnung bei jeder identifizierten Abhängigkeit ist dieses Vorgehen daher ungeeignet. Bei der wissensbasierten Szenariogenerierung wird weiteres Hintergrundwissen aus Richtlinien oder aus bereits durchgeführten Ansätzen zur Szenariogenerierung vorausgesetzt.[135b]

Da der szenariobasierte Testansatz die Anforderungen aus Tabelle 5-1 erfüllt und sich die expertenbasierte sowie die anforderungsbasierte Szenariogenerierung für diese Anwendung eignet, werden diese Elemente des Testprozesses für das zu entwickelnde Verfahren zur Definition von Testfällen ausgewählt.

5.3.2 Verfahren zur Definition von Testfällen

Das im Folgenden beschriebene Verfahren zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten beruht auf einem szenariobasierten Testansatz. Bei dem Verfahren liegt ebenfalls ein iteratives Vorgehen vor.

Für dieses Verfahren werden definierten Abhängigkeitskategorien sowie die Liste der ermittelten Abhängigkeiten mit dem Aufbau des Abhängigkeitssteckbriefs aus dem ersten Verfahren benötigt. Zu Beginn des Verfahrens erfolgt anhand der acht definierten Abhängigkeitskategorien die Gruppierung der Abhängigkeiten in acht Testfallgruppen. Diese dienen der Übersicht und ermöglichen ein systematisches Vorgehen bei der Definition der Testfälle.

Nach dieser Vorbereitung beginnt der iterative Prozess. Zur Testfallgenerierung wird mit einer Einzelbetrachtung der Abhängigkeiten der ersten Testfallgruppe begonnen. Danach erfolgt die Generierung der Testfälle für die zweite Testfallgruppe und so weiter. Bei jeder Abhängigkeit wird geprüft, ob in den bereits durchgeführten Expertenbefragungen kritische Szenarien, die Zustands- und Verhaltensänderungen verursachen, bereits für diese Abhängigkeit identifiziert wurden. Diese Expertise ermöglicht bereits für einen Teil der Abhängigkeiten die Identifizierung von Szenarien einschließlich der Szenen und enthaltenen Elemente. Da mit Expertenwissen die Anzahl der zu identifizierbaren Szenarien limitiert ist, wird bei jeder Abhängigkeit darüber hinaus der anforderungsbasierte Ansatz zur Szenariogenerierung angewendet. Dafür wird die erstellte Liste mit den ermittelten Abhängigkeiten um Anforderungen erweitert. Hierbei handelt es sich um Anforderungen, die aufgrund der vorliegenden Abhängigkeit an das Testobjekt gestellt werden. Die expertenbasierten Szenarien werden ebenfalls bei der jeweiligen Abhängigkeit in die Liste aufgenommen.

Wenn diese Testfallgenerierung für eine entsprechende Anzahl an Abhängigkeiten pro Testfallgruppe erfolgt ist, werden für jeden Testfall das Testziel und die Bewertungskriterien ergänzt. Die definierten Testfälle werden in der bestehenden Simulationsumgebung CarMaker[136] umgesetzt. Die Bewertung der Ergebnisse der durchgeführten Testfälle erfolgt anhand des jeweiligen Testziels und der zuvor

[135] BMWI: PEGASUS Method (2019), a: S. 15; b: S. 7.

[136] IPG Automotive GmbH: CarMaker.

festgelegten Kriterien. Zusätzlich werden die durchgeführten Testfälle dahingehend analysiert, ob noch weitere unbekannte Abhängigkeiten vorliegen.

Unbekannte Abhängigkeiten werden anschließend an das erste Verfahren weitergeleitet und dort erfolgt in einer weiteren Iteration die Analyse der Abhängigkeiten mittels der Systemmodellierung und des Aufbaus des Abhängigkeitsnetzes. In Abhängigkeit von der Bewertung der Testergebnisse werden einzelne Testfälle in einer zweiten Iteration verfeinert. Auch die Testfallgenerierung von weiteren Abhängigkeiten ist in einer zweiten Iteration durchführbar.

In Abbildung 5-1 ist die Methodik zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten zusammengefasst dargestellt.

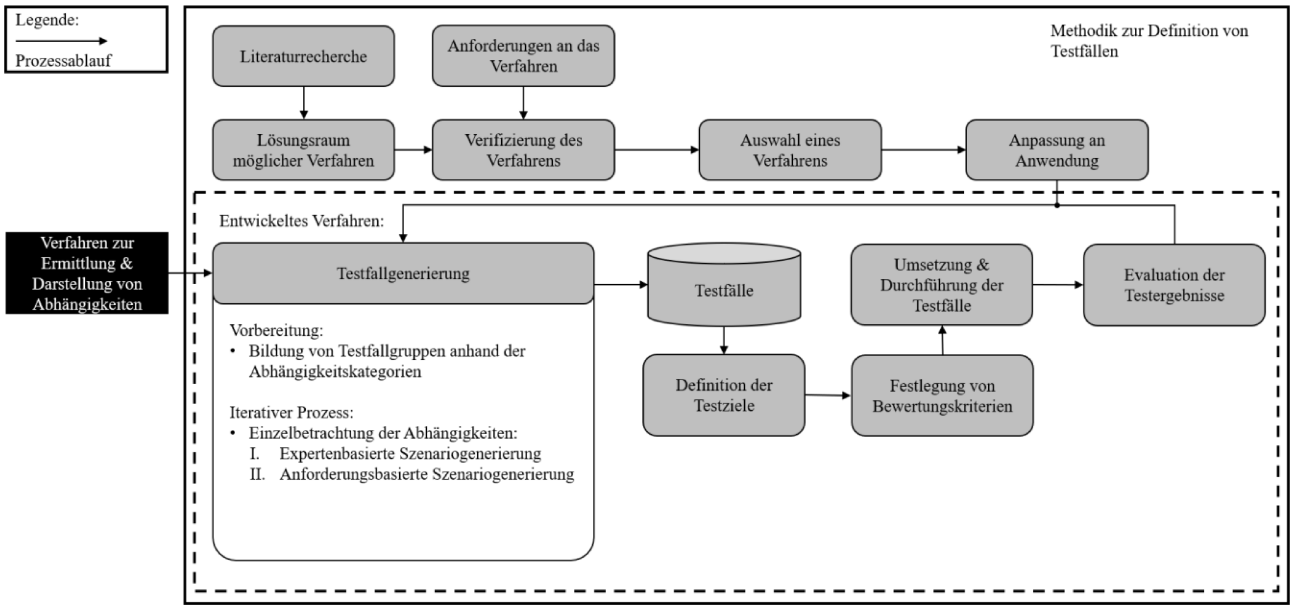

Abbildung 5-1: Methodik zur Definition von Testfällen

6 Anwendung des zweiten Verfahrens: Definition von Testfällen im Rahmen des Projekts UNICAR*agil*

In diesem Kapitel wird das zuvor in Kapitel 5 entwickelte Verfahren zur Definition von Testfällen auf die in Kapitel 4 ermittelten Abhängigkeiten angewendet. Das verfolgte Ziel der Definition und Durchführung von Testfällen ist, die zuvor im ersten Verfahren identifizierten Abhängigkeiten zu bestätigten, obwohl sie im Vorhinein nicht offensichtlich waren. Außerdem ermöglichen die Definition und Durchführung von Testfällen die Aufdeckung von weiteren Abhängigkeiten.

In Abschnitt 6.1 werden mit Hilfe der anforderungs- und expertenbasierten Szenariogenerierung Testszenarien definiert. Die Ergebnisse der Anwendung des Verfahrens zur Definition von Testfällen sowie der durchgeführten Testfälle werden in Abschnitt 6.2 beschrieben.

6.1 Definition der Testszenarien

Der Fokus bei der Definition der Testszenarien liegt auf den codebasierten Abhängigkeiten (Testfallgruppe I), den systemunabhängigen Abhängigkeiten (Testfallgruppe II), den systemdesignbasierten Abhängigkeiten (Testfallgruppe III), den inter-datenbasierten Abhängigkeiten (Testfallgruppe IV) und den datentransferbasierten Abhängigkeiten (Testfallgruppe V). Die ressourcenbasierten und datenbasierten Abhängigkeiten sind hinsichtlich der Identifizierung von unbekannten Abhängigkeiten in erster Linie für die modulare Absicherung nicht relevant, da diese Abhängigkeiten erforderlich für die Erbringung der Funktionen sind und zudem offensichtliche Abhängigkeiten sind. Die informationsbasierten Abhängigkeiten werden hier ebenfalls vernachlässigt, da diese die Übertragung von unterschwelligen Informationen, die vom Empfänger subjektiv interpretiert werden, beschreiben. Folglich tragen sie nicht zu relevanten Verhaltensänderungen bei.

Im Folgenden werden für die Testfallgruppen I bis V auf Basis der anforderungs- und expertenbasierten Szenariogenerierung ein bis zwei Szenarien entwickelt und beschrieben. Diese Beschreibungen enthalten darüber hinaus das Testziel und Bewertungskriterien für das jeweilige Testszenario.

6.1.1 Testfallgruppe I – Codebasierte Abhängigkeiten

Diese Testfallgruppe besteht aus den codebasierten Abhängigkeiten, welche eine Beziehung zwischen zwei Schnittstellen im Softwarecode festlegen. Für zwei identifizierte codebasierte Abhängigkeiten werden Testfälle definiert.

6.1.1.1 Szenario 1 – „Seitliches Parkmanöver“

Im Softwarecode der FTR wird mit einem Sollschwimmwinkel von 90° (ID: *FTR_FTR_019*) die Ausführung des für das seitliche Parkmanöver relevanten Codes ausgelöst. Das Testziel dieses Szenarios ist die Überprüfung der Abfrage sowie der nachfolgenden Umschaltung im Softwarecode beim Wechsel von einer Geradeausfahrt zu einem seitlichen Parkmanöver und anschließend wieder zurück zu einer Geradeausfahrt.

Das dazu entwickelte Szenario ist eine Geradeausfahrt zu der Position des seitlichen Parkmanövers. Dort erfolgt das seitliche Einparken des Fahrzeugs. Nach dem Einparkvorgang erfolgt eine Weiterfahrt auf dem gleichen Fahrstreifen. Dieses Szenario ist schematisch in Abbildung 6-1 dargestellt. Dieser Testfall basiert auf den Erkenntnissen aus den Expertengesprächen und wurde dort bereits im Detail ausgeführt, sodass dieser Testfall aus einer expertenbasierte Szenariogenerierung hervorgeht.

Die Bewertungskriterien für diesen Testfall lauten

1. Eine Längsbeschleunigung findet statt und die innere und die äußere laterale Grenze des Fahrstreifens werden nicht verletzt.
2. Während der Längsbeschleunigung wird ein seitliches Einparkmanöver erkannt und in Form einer Querbeschleunigung durchgeführt. Dabei wird die äußere laterale Grenze des Fahrstreifens nicht verletzt.
3. Das Ausparkmanöver in Fahrtrichtung wird erkannt und in Form einer Längsbeschleunigung durchgeführt. Dabei werden die innere und die äußere laterale Grenze des Fahrstreifens nicht verletzt.

Dieser Testfall ist zum Zeitpunkt der Durchführung der Testfälle nicht wie beschrieben umsetzbar. Seitliche Parkmanöver sind in der Simulationsumgebung implementiert. Jedoch handelt es sich dabei um ein Einzelmanöver, das über einen eigenen *parking*-Planer durchgeführt wird. Die Umsetzung eines solchen zusammengesetzten Manövers erfordert seitens der Entwickler zusätzliche Unterstützung.

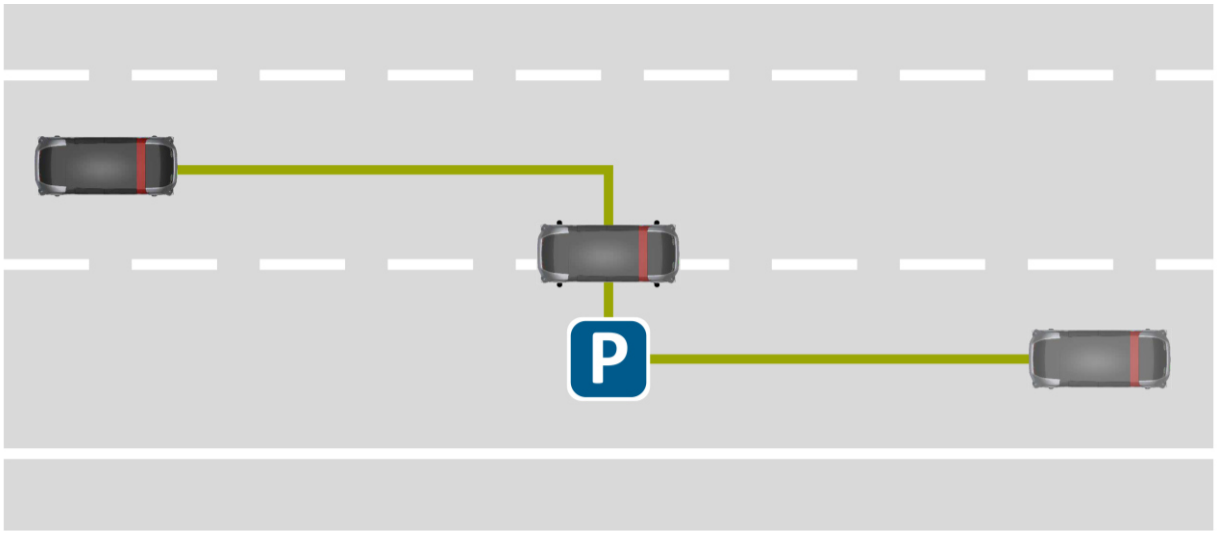

Abbildung 6-1: Szenario 1 – „Seitliches Parkmanöver“

6.1.1.2 Szenario 2 – „Trigger Auslösung beim *Sicheren Anhalten*“

Eine weitere codebasierte Abhängigkeit ist der Trigger für die Trajektoriengenerierung (ID: *FSA_FSA_027*) in der FSA. Dieser im Softwarecode implementierte Trigger löst bei einer neuen Objektliste, einer neuen Notbahn oder spätestens alle 100 ms die Generierung der Trajektorie aus. Das

Testziel dieses Szenarios ist die Überprüfung des Triggers, wie dieser auf eine wiederholte Auslösung innerhalb von 1 ms reagiert.

Bei dem dafür entwickelten Szenario wird während des *Sicheren Anhaltens* eine neue Objektliste und mit Abstand von 1 ms eine weitere Objektliste verschickt. Die erste Objektliste löst den Trigger für die Trajektoriengenerierung einer neuen Nottrajektorie aus und nach 1 ms wird die Generierung durch die zweite Objektliste erneut ausgelöst. Dieser Testfall basiert auf den Erkenntnissen aus den Expertengesprächen. Daher handelt es sich um eine expertenbasierte Szenariogenerierung.

Das Bewertungskriterium für diesen Testfall lautet

1. Das risikominimale Anhaltemanöver wird durchgeführt und die Fahrzeugbewegung endet an einem risikominimalen Ort.

Dieser Testfall wird nicht umgesetzt, da der Fokus bei der Durchführung der Testfälle auf dem anderen Testszenario der Testfallgruppe I liegt.

6.1.2 Testfallgruppe II – Systemunabhängige Abhängigkeiten

Diese Testfallgruppe besteht aus den systemunabhängigen Abhängigkeiten, welche die Einwirkung von äußeren Einflussfaktoren auf das System oder einzelner Module berücksichtigen. Für zwei identifizierte systemunabhängige Abhängigkeiten werden Testfälle definiert.

6.1.2.1 Szenario 3 – „Fahrt über Bodenwellen“

Die in diesem Szenario untersuchte Abhängigkeit ist das Straßenprofil (ID: *FTR_FTR_020*). Eine Bodenwelle im Straßenprofil beeinflusst die Fahrdynamik des Fahrzeugs und den Reibkoeffizienten aufgrund der veränderten Reibung zwischen den Reifen und dem Belag der Bodenwelle. Das Testziel dieses Szenarios ist die Überprüfung, was für Verhaltensänderungen durch eine Bodenwelle aufgrund einer veränderten Fahrdynamik und einem abweichenden Reibkoeffizienten in der FTR ausgelöst werden.

Das dazu entwickelte Szenario ist eine Geradeausfahrt mit zwei unterschiedlichen Bodenwellen, die zu überfahren sind. Dieses Szenario ist schematisch in Abbildung 6-2 dargestellt ist. Dieser Testfall basiert auf der anforderungsbasierten Szenariogenerierung.

Das Bewertungskriterium für diesen Testfall lautet

1. Eine Längsbeschleunigung des Fahrzeugs findet statt und die innere und die äußere laterale Grenze des Fahrstreifens werden nicht verletzt.

Dieser Testfall wird in der Simulationsumgebung umgesetzt und durchgeführt. Die erste Bodenwelle erstreckt sich 5,0 m in longitudinale Richtung und ist 0,1 m hoch und die zweite Bodenwelle umfasst 0,4 m in longitudinale Richtung und ist 0,05 m hoch.

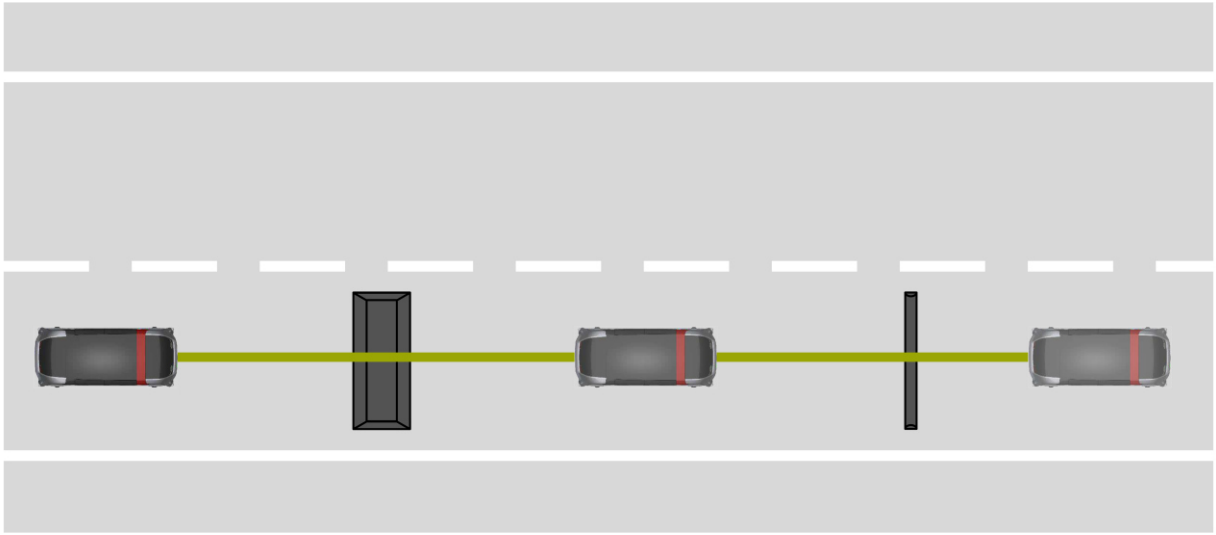

Abbildung 6-2: Szenario 3 – „Fahrt über Bodenwellen“

6.1.2.2 Szenario 4 – „Wechsel von trockenem zu nassem Straßenbelag“

Die in diesem Szenario untersuchte Abhängigkeit ist ebenfalls das Straßenprofil (ID: *FTR_FTR_020*). Im Gegensatz zum Szenario 3 wird lediglich der Einfluss des Reibkoeffizienten auf die FTR durch den Wechsel von einem trockenem zu einem nassem Straßenbelag untersucht. Das Testziel dieses Szenarios ist die Überprüfung, was für Verhaltensänderungen durch unterschiedliche Fahrbahneigenschaften und dementsprechend sich ändernde Reibkoeffizienten in der FTR ausgelöst werden.

Das dazu entwickelten Szenario ist eine Geradeausfahrt mit einem Wechsel des Reibkoeffizienten. Dieser Testfall basiert auf der anforderungsbasierten Szenariogenerierung.

Das Bewertungskriterium für diesen Testfall lautet

1. Eine Längsbeschleunigung des Fahrzeugs findet statt und die innere und die äußere laterale Grenze des Fahrstreifens werden nicht verletzt.

Dieser Testfall wird nicht umgesetzt, da der Fokus bei der Durchführung der Testfälle auf dem anderen Testszenario der Testfallgruppe II liegt.

6.1.3 Testfallgruppe III – Systemdesignbasierte Abhängigkeiten

Diese Testfallgruppe besteht aus den systemdesignbasierten Abhängigkeiten, welche die Einwirkung von Einflüssen aufgrund der Eigenschaften und technischen Details berücksichtigen. Für zwei identifizierte systemdesignbasierte Abhängigkeiten werden Testfälle definiert.

6.1.3.1 Szenario 5 – „Scharfe Kurve“

Die in diesem Szenario untersuchte Abhängigkeit ist die Fahrdynamik (ID: *FTR_FTR_015*). Eine enge Kurve beeinflusst die Fahrdynamik des Fahrzeugs. Das Testziel dieses Szenarios ist die Überprüfung, was für Verhaltens- und Zustandsänderungen durch eine enge Kurve aufgrund einer veränderten Fahrdynamik in der FTR ausgelöst werden.

Das dazu entwickelte Szenario ist eine Geradeausfahrt, an die sich eine enge Kurve anschließt und dann in einer Geradeausfahrt endet. In Abbildung 6-3 ist dieses Szenario schematisch dargestellt. Dieser Testfall basiert auf den Erkenntnissen aus den Expertengesprächen und wurde dort bereits kurz erwähnt, sodass es sich bei diesem Testfall um eine expertenbasierte Szenariogenerierung handelt.

Die Bewertungskriterien für diesen Testfall lauten

1. Bei der Längsbeschleunigung des Fahrzeugs werden die innere und die äußere laterale Grenze nicht verletzt.
2. Bei der Querbeschleunigung des Fahrzeugs werden die innere und die äußere laterale Grenze nicht verletzt.

Dieser Testfall wird in der Simulationsumgebung umgesetzt.

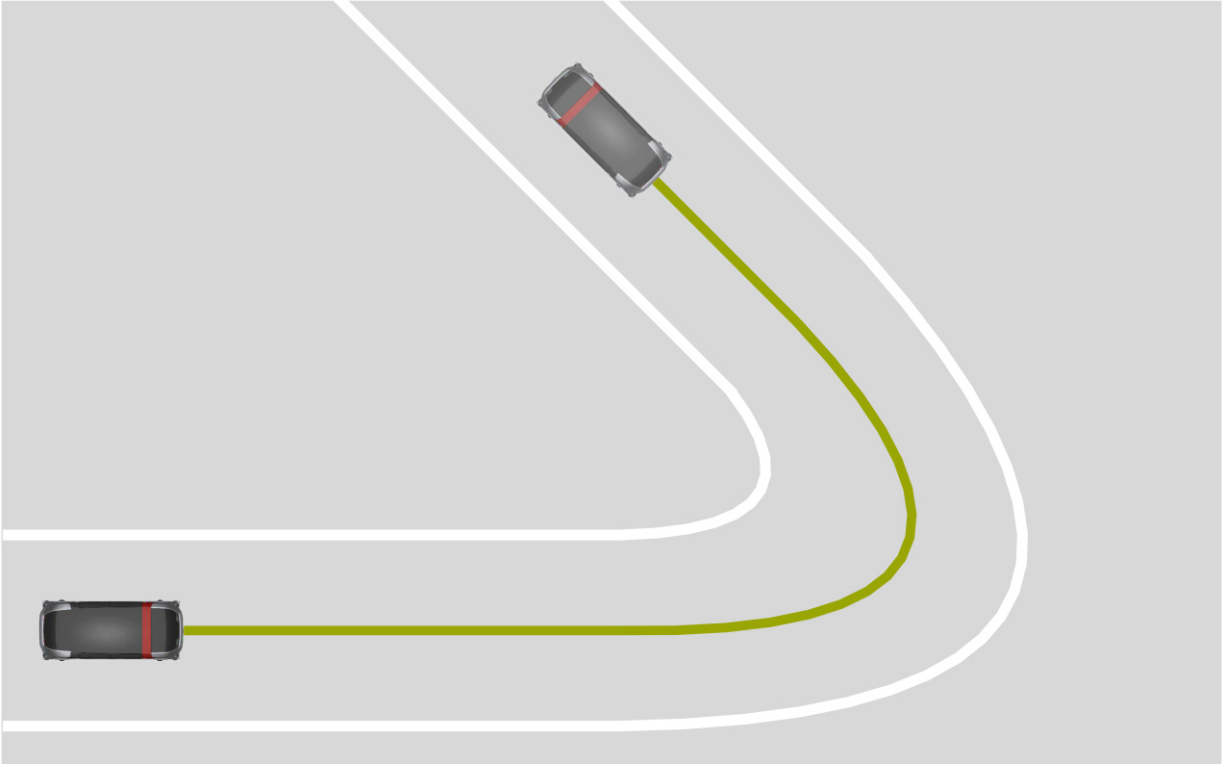

Abbildung 6-3: Szenario 5 – „Scharfe Kurve“

6.1.3.2 Szenario 6 – „Ruckeln“

Zustandsänderungen in Signalen sind z.B. der komplette Ausfall eines Signals, ein wechselndes Signal oder ein Sprung in einem Signal. Ein wechselndes Signal liegt beispielsweise vor, wenn ein Objekt kurze Zeit erkannt wird, dann für kurze Zeit nicht und abschließend die Erkennung des Objekts wieder erfolgt. Daher folgt die Berechnung einer wechselnden Trajektorie, welche wiederum eine wechselnde Momentenvorgabe zur Folge hat. Infolgedessen entsteht ein sprungförmiger Verlauf in den Aktuatorgrenzen und das UNICAR*agil*-Fahrzeug fängt an zu ruckeln. Bei diesem Szenario handelt es sich um mehrere miteinander verknüpfte Abhängigkeiten über verschiedene Module hinweg.

Das Testziel dieses Szenarios ist die Überprüfung der Verkettung dieser Abhängigkeiten. Diese Verkettung beginnt bei den Sensormodulen, führt über den VTP zur FTR und abschließend zu den Dynamikmodulen.

Das dazu entwickelte Szenario ist eine Geradeausfahrt mit einer wechselnden Objekterkennung. Dieser Testfall basiert auf den Erkenntnissen aus den Expertengesprächen und wurde dort bereits kurz erwähnt. Anschließend wurden anhand des Abhängigkeitsnetzes die Abhängigkeiten mit den Anforderungen erarbeitet, sodass es sich bei diesem Testfall um eine Mischung aus experten- und anforderungsbasierter Szenariogenerierung handelt.

Das Bewertungskriterium für diesen Testfall lautet

1. Eine Längsbeschleunigung des Fahrzeugs ohne Ruckeln findet statt und die innere und die äußere laterale Grenze des Fahrstreifens werden nicht verletzt.

Dieser Testfall wird nicht umgesetzt, da der Fokus bei der Durchführung der Testfälle auf dem anderen Testszenario der Testfallgruppe III liegt.

6.1.4 Testfallgruppe IV – Inter-Datenbasierte Abhängigkeiten

Diese Testfallgruppe besteht aus den inter-datenbasierten Abhängigkeiten, welche Beziehungen beschreiben, die bei der Wechselwirkung von verschiedenen Daten relevant sind. Für eine identifizierte inter-datenbasierte Abhängigkeit wird nachfolgend ein Testfall definiert.

6.1.4.1 Szenario 7 – „180° Wendung“

Die in diesem Szenario untersuchte Abhängigkeit ist die Fahrdynamik (ID: *FTR_FTR_015*) bei einer 180° Wendung des Fahrzeugs. Ein solches Wendemanöver beeinflusst die Fahrdynamik des Fahrzeugs und stellt eine Herausforderung für die Lokalisierung und die Posen-Bestimmung in der FZS dar. Das Testziel dieses Szenarios ist die Überprüfung, was für Verhaltens- und Zustandsänderungen durch eine 180° Wendung aufgrund einer sich ändernden Fahrdynamik in der FTR ausgelöst werden und ob die FZS nach einem solchen Manöver den Ist-Zustand des Fahrzeugs korrekt ermittelt.

Das dazu entwickelte Szenario ist eine Geradeausfahrt, eine anschließende 180° Wendung und eine abschließende Geradeausfahrt in die entgegengesetzte Richtung als bei der Hinfahrt. Dieses Szenario ist schematisch in Abbildung 6-4 dargestellt. Dieser Testfall basiert auf den Erkenntnissen aus den Expertengesprächen und wurde angesprochen, sodass es sich bei diesem Testfall um eine expertenbasierte Szenariogenerierung handelt.

Die Bewertungskriterien für diesen Testfall lauten

1. Bei der Längsbeschleunigung des Fahrzeugs werden die innere und die äußere laterale Grenze nicht verletzt.
2. Bei der Querbeschleunigung des Fahrzeugs werden die innere und die äußere laterale Grenze nicht verletzt.

Dieser Testfall wird in der Simulationsumgebung umgesetzt.

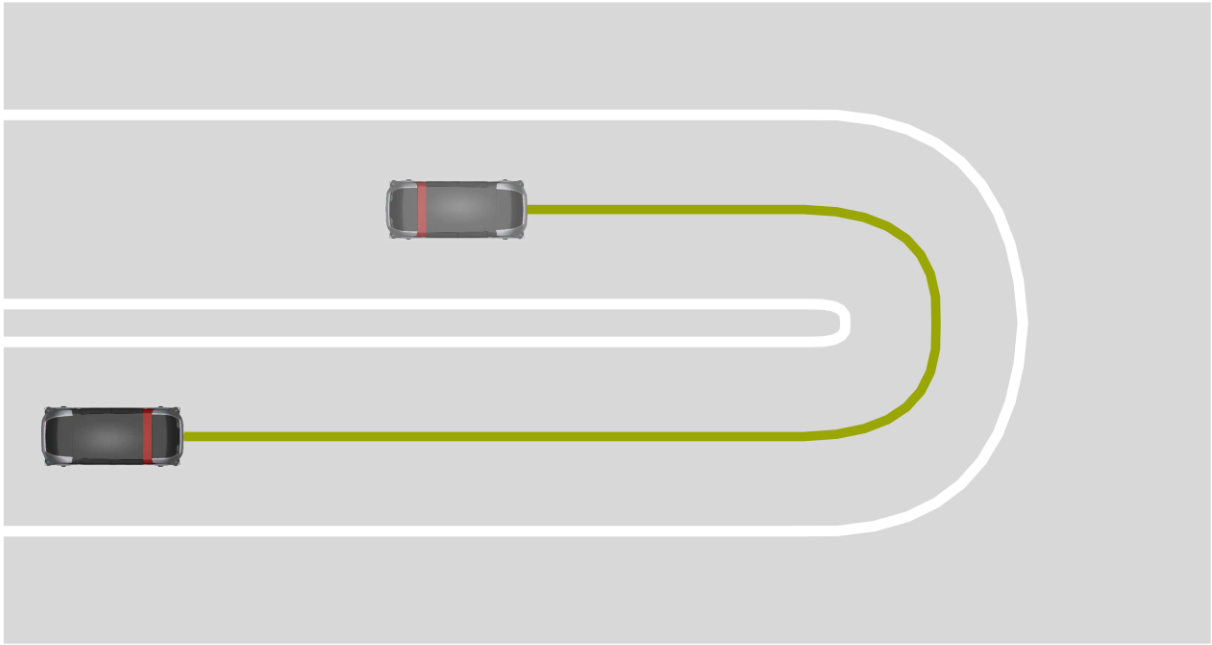

Abbildung 6-4: Szenario 7 – „180° Wendung“

6.1.5 Testfallgruppe V – Datentransferbasierte Abhängigkeiten

Diese Testfallgruppe besteht aus den datentransferbasierten Abhängigkeiten, welche eine Beziehung beschreiben, die auf Einflüssen beim Austausch von Daten basiert. Für eine datentransferbasierte Abhängigkeit wird im Folgenden ein Testfall definiert.

6.1.5.1 Szenario 8 – „Tunnelfahrt“

Die in diesem Szenario untersuchte Abhängigkeit ist die Verbindung zwischen dem Satelliten und dem GNSS-Empfänger (ID: *FZS_FZS_017*). Bei einer Tunnelfahrt bricht die Verbindung zwischen dem Satelliten und dem GNSS-Empfänger ab. Dies löst einen Drift in der Pose aus und führt somit zu einer falschen Position und einem Abkommen des Fahrzeugs. Die Möglichkeiten bei der Datenübertragung sind, dass entweder keine Daten, die Daten zu früh oder zu spät ankommen oder ein schwankendes Signal vorliegt. Bei einer Aneinanderreihung von Tunnelabschnitten mit etwas Abstand zueinander verursacht dies ein schwankendes Signal für den GNSS-Empfänger. Das Testziel dieses Szenarios ist die Überprüfung, was ein wechselndes Signal für Zustands- und Verhaltensänderungen in der FZS und der nachfolgenden FTR, die den Ist-Zustand von der FZS erhält, hervorruft. Zudem ist von Interesse, ob bei wiederholter Störung sich Hinweise in Bezug auf noch unbekannte Abhängigkeiten ergeben.

Das dazu entwickelte Szenario ist eine Geradeausfahrt durch zwei Tunnelabschnitte. Dieses Szenario ist schematisch in Abbildung 6-5 dargestellt.

Dieser Testfall basiert auf den Erkenntnissen aus den Expertengesprächen. Dieser wurde dort bereits kurz erwähnt, sodass es sich bei diesem Testfall um eine expertenbasierte Szenariogenerierung handelt.

Die Bewertungskriterien für diesen Testfall lauten

1. Bei der Längsbeschleunigung des Fahrzeugs werden die innere und die äußere laterale Grenze nicht verletzt.
2. Bei einer Querbeschleunigung des Fahrzeugs werden die innere und die äußere laterale Grenze nicht verletzt.

Dieser Testfall ist zum Zeitpunkt der Durchführung der Testfälle nicht wie beschrieben umsetzbar, da in dem derzeit genutzten Verzeichnis die von der TU Darmstadt entwickelte FZS nicht implementiert ist, sondern eine Zustandsschätzung von CarMaker verwendet wird. Die FZS befindet sich in einem anderen Verzeichnis, in dem der VTP nicht implementiert ist. An der Integrierung der FZS in das Verzeichnis mit dem VTP wird zurzeit gearbeitet. Sobald dies erfolgt ist, ist die Umsetzung dieses Testfalls möglich.

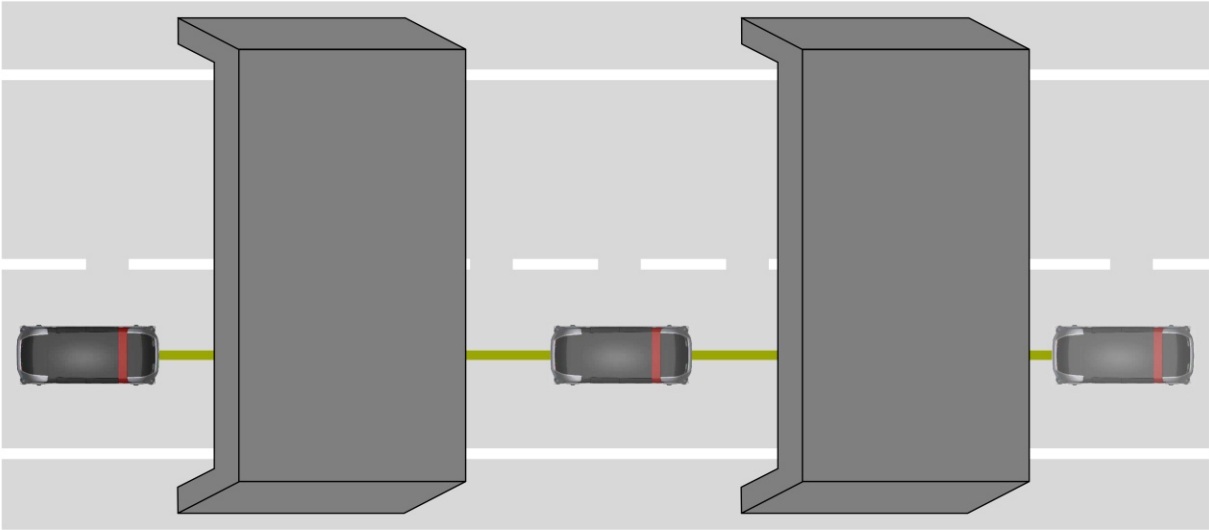

Abbildung 6-5: Szenario 8 – „Tunnelfahrt“

6.2 Ergebnisse – Zweites Verfahren

Die Ergebnisse der Anwendung des zweiten Verfahrens werden im Folgenden vorgestellt. Zuerst erfolgt die Bewertung der drei in der Simulationsumgebung durchgeführten Testfälle. Zum Abschluss dieses Kapitels wird das entwickelte Verfahren zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten anhand der definierten Festforderungen und Wünsche bewertet.

6.2.1 Bewertung der Testfälle

Drei der in Abschnitt 6.1 beschriebenen Testszenarien sind in der bestehenden Simulationsumgebung umgesetzt und durchgeführt worden. Bei zwei weiteren Testszenarien war die Umsetzung geplant, jedoch haben sich im Laufe der Umsetzung Unzugänglichkeiten herausgestellt. Dabei handelt es sich um das erste Szenario, das seitliche Parkmanöver, und das achte Szenario, die Tunnelfahrt. Auf die genaue Problematik wird in Abschnitt 6.1 eingegangen.

6.2.1.1 Testfall 1 – „Fahrt über Bodenwellen“

Beim ersten Testfall, der Fahrt über zwei Bodenwellen, sind zwei Effekte zu beobachten. Zum einen ändert sich der Reibkoeffizient beim Überqueren der Bodenwellen und zum anderen verändert sich der Sensorsichtbereich aufgrund der Bewegung des Fahrzeugs. Bei dem zweiten Einfluss auf die Sensormodule von CarMaker ist anzumerken, dass dieser keine Berücksichtig findet, da der veränderte Sensorsichtbereich in der Simulationsumgebung nicht valide darstellbar ist.

Das Testziel bei diesem Testfall ist die Überprüfung möglicher Verhaltensänderungen in der FTR aufgrund einer veränderten Fahrdynamik und einem abweichenden Reibkoeffizienten, die durch die Bodenwellen ausgelöst werden. Bei der Durchführung ist zu beobachten, dass die Bodenwellen keinen Auslöser für Verhaltens- oder Zustandsänderungen darstellen und in der Simulation einfach überfahren werden. Das aufgestellte Bewertungskriterium für diesen Testfall ist ebenfalls erfüllt. Eine kontinuierliche Längsbeschleunigung des Fahrzeugs über die gesamte Geradeausfahrt und die zwei unterschiedlichen Bodenwellen findet statt und gleichzeitig werden die innere und die äußere laterale Grenze des Fahrstreifens nicht verletzt.

Daher lässt sich bei diesem Testfall zusammenfassen, dass dieser Test erfüllt ist und eine Bodenwelle mit einem veränderten Reibkoeffizienten kein Auslöser für Verhaltensänderungen ist. Zudem wurden keine noch unbekannten Abhängigkeiten aufgedeckt.

6.2.1.2 Testfall 2 – „Scharfe Kurve“

Der zweite Testfall, die scharfe Kurve, untersucht den Einfluss der Fahrdynamik. Das in diesem Testfall verfolgte Testziel ist die Überprüfung möglicher Verhaltens- und Zustandsänderungen in der FTR durch die veränderte Fahrdynamik aufgrund der engen Kurve.

Es ist bei der Durchführung des Testfalls zu beobachten, dass die enge Kurve keinen Auslöser für Verhaltens- oder Zustandsänderungen darstellt. In der Simulation fährt das Fahrzeug die enge Kurve ab. Die aufgestellten Bewertungskriterien für diesen Testfall sind nach der Beobachtung des Testszenarios erfüllt. Die innere und die äußere laterale Grenze bei der Längsbeschleunigung des Fahrzeugs werden nicht verletzt. Ebenso werden die innere und die äußere laterale Grenze bei der Querbeschleunigung des Fahrzeugs nicht verletzt.

Zusammenfassend ist bei diesem Testfall zu erwähnen, dass dieser Test ebenfalls erfüllt ist und die enge Kurve aufgrund der Fahrdynamik in diesem Testszenario keine Verhaltensänderungen ausgelöst hat. Weitere unbekannte Abhängigkeiten wurden auch nicht aufgedeckt.

6.2.1.3 Testfall 3 – „180° Wendung“

Bei dem letzten durchgeführten Testfall, der 180° Wendung, handelt es sich um eine Herausforderung für die Lokalisierung in der FZS und aufgrund einer veränderten Fahrdynamik für die FTR. Da wie bereits angesprochen in dem verwendeten Verzeichnis die FZS nicht implementiert ist, sondern eine Zustandsschätzung von CarMaker, ist keine Aussage zur Ermittlung des Ist-Zustands in der FZS bei diesem Testszenario möglich. Folglich vereinfacht sich das Testziel und beschreibt lediglich die

Überprüfung hinsichtlich Verhaltens- und Zustandsänderungen in der FTR aufgrund einer sich ändernden Fahrdynamik.

Während der Simulation sind hinsichtlich einer Verhaltens- oder Zustandsänderung keine Auffälligkeiten zu beobachten. Jedoch ist nach dem Beenden des Wendemanövers eine unbekannte Abhängigkeit identifiziert worden. Bei der abschließenden Geradeausfahrt nach der 180° Wendung hängt das Verhalten des Fahrzeugs von den definierten Lanelets in der Karte ab. Je nachdem wie der Fahrstreifen modelliert ist, fährt das Fahrzeug bis zum Ende der vorgegebenen Route oder stoppt vorzeitig. Da die Kartengeometrie (ID: *VTP_VTP_007*) als datenbasierte Abhängigkeit beim VTP vorliegt, handelt es sich bei dieser um eine informationsbasierte Abhängigkeit. Neben den offensichtlichen Daten erfolgt eine Interpretation von Seiten des VTP, sodass unterschwellig zusätzliche Informationen ankommen und diese Verhaltensänderungen auslösen.

Die aufgestellten Bewertungskriterien für diesen Testfall sind zum einen die Nicht-Verletzung der inneren und der äußeren lateralen Grenze bei der Längsbeschleunigung und zum anderen die Nicht-Verletzung der inneren und der äußeren lateralen Grenze bei der Querbeschleunigung des Fahrzeugs.

Zusammenfassend ist bei diesem Testfall zu notieren, dass dieser Test ebenfalls erfüllt ist und die 180°Wendung aufgrund der Fahrdynamik in diesem Testszenario keine Verhaltensänderungen ausgelöst hat. Zudem wurde eine unbekannte Abhängigkeit aufgedeckt.

6.2.2 Bewertung der Ergebnisse – Zweites Verfahren

Abschließend an die Anwendung des entwickelten Verfahrens zur Definition von Testfällen erfolgt die Verifizierung anhand der Festforderungen und Wünsche aus der Anforderungsliste in Tabelle 5-1. Die Diskussion der Einflussfaktoren auf dieses Verfahren folgt in Abschnitt 7.2.

Die Anwendung des Verfahrens auf die mit dem vorherigen Verfahren ermittelten Abhängigkeiten zeigt, dass die definierten Testfälle diese Abhängigkeiten aufzeigen. Aus diesem Grund ist die erste Festforderung erfüllt.

Zudem hat die Anwendung des Verfahrens verdeutlicht, dass auch die Definition von Testfällen auf Grundlage von Abhängigkeiten verschiedener Kategorien realisierbar ist. Bei den drei durchgeführten Testfällen handelt es sich um Abhängigkeiten dreier unterschiedlicher Kategorien. Der Testfall „Fahrt über Bodenwellen“ basiert auf einer systemunabhängigen Abhängigkeit. Der zweite Testfall „Scharfe Kurve“ stammt von einer systemdesignbasierten Abhängigkeit. Bei dem dritten Testfall „180° Wendung“ handelt es sich um eine inter-datenbasierte Abhängigkeit. Folglich ist auch die zweite Festforderung erfüllt.

In der dritten Festforderung wird gefordert, dass sich das Verfahren auch zur Aufdeckung von weiteren unbekannten Abhängigkeiten eignet. Bei dem dritten Testfall wurde eine unbekannte Abhängigkeit identifiziert. Dabei handelt es sich um eine informationsbasierte Abhängigkeit. Demnach ist diese Festforderung ebenfalls erfüllt.

Die vierte Festforderung ist erfüllt, da ein szenariobasierter Testansatz zur Generierung von Testfällen die Grundlage dieses Verfahrens bildet. Die Entscheidung für diesen Testansatz rührt daher, dass

mehrfach in der Literatur dieser Ansatz zur Lösung der Herausforderungen beim Sicherheitsnachweis für die Freigabe von automatisierten Fahrsystemen ab der Automatisierungsstufe 3 vorgeschlagen wird.

Die Szenariogenerierung bei diesem Verfahren erfolgt anhand des expertenbasierten und des anforderungsbasierten Ansatzes. Dementsprechend ist die fünfte Festforderung ebenfalls erfüllt.

Der Wunsch, dass die definierten Testfälle in der Simulationsumgebung und mit den bestehenden Simulationsmodellen umsetzbar sind, ist teilweise erfüllt. Dies wird einmal durch den aktuellen Stand vom VTP und dessen eingeschränkten Fähigkeiten beeinflusst. Die fehlende FZS im Verzeichnis vom VTP ist ein anderer Punkt, der die Unzugänglichkeiten bei der Wunscherfüllung verursacht.

Die letzte Festforderung an das Verfahren ist, dass sich das Verfahren für einen iterativen Prozess und eine nachträgliche Definition weiterer Testfälle eignet. Dies ist gegeben, da die Testfallgenerierung einzeln für die Abhängigkeiten erfolgt. Aus diesem Grund ist auch die sechste Festforderung erfüllt.

Nach der Definition und Durchführung der Testfälle lässt sich der letzte Wunsch hinsichtlich der Komplexität, Nachvollziehbarkeit sowie des Aufwands bei der Einarbeitung und Fortführung des Verfahrens nicht vollständig bewerten. Bei dem entwickelten Verfahren wird das Ziel angestrebt, diesen Wunsch zu erfüllen. Daher wird dieser Wunsch hier als erfüllt angesehen.

Zusammenfassend ist anzumerken, dass alle Festforderungen und die beiden Wünsche aus der Tabelle 5-1 von dem entwickelten Verfahren zur Definition von Testfällen erfüllt werden. Mit diesem entwickelten Verfahren werden Testfälle abgeleitet, die eine Bestätigung für die identifizierten und vorher nicht offensichtlichen Abhängigkeiten darstellen. Außerdem ermöglichen die Definition und Durchführung von Testfällen die Aufdeckung von weiteren unbekannten Abhängigkeiten. Abschließend ist zu erwähnen, dass der zweite Wunsch aus Tabelle 3-1, die Eignung der Darstellung für eine nachfolgende Testfallgenerierung an das erste Verfahren, erfüllt ist.

7 Diskussion der Ergebnisse

Im Rahmen dieser Arbeit wird im ersten Schritt ein Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung entwickelt. Im zweiten Schritt wird ein Verfahren zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten erarbeitet. Anhand der Systemarchitektur im Projekt UNICAR*agil* wird die Anwendbarkeit dieser entwickelten Verfahren demonstriert. Die Identifizierung der Abhängigkeiten erfolgt mit Fokus auf die Fahrdynamikzustandsschätzung, den Verhaltens- und Trajektorienplaner, die Fahrschlauchabsicherung sowie die Fahrdynamik- und Trajektorienregelung. Mit der Definition und Durchführung von Testfällen werden anschließend die Ziele verfolgt, die identifizierten Abhängigkeiten zu bestätigten, obwohl sie im Vorhinein nicht offensichtlich waren, und weitere unbekannte Abhängigkeiten aufzudecken.

Das Ergebnis des ersten Verfahrens ist ein aufgebautes Abhängigkeitsnetz, woraus eine Liste mit den ermittelten Abhängigkeiten erstellt und Abhängigkeitskategorien zur Einteilung der Vielzahl an Abhängigkeiten definiert werden. Die Ergebnisse des zweiten Verfahrens sind teils theoretisch beschriebene und teils durchgeführte Testfälle, basierend auf einer expertenbasierten und anforderungsbasierten Szenariogenerierung für einen Teil der ermittelten Abhängigkeiten. Anhand dieser Ergebnisse und der Bewertung der Verfahren, basierend auf den jeweiligen zu Beginn erstellten Anforderungslisten, wird deutlich, dass die entwickelten Verfahren einen erfolgversprechenden Ansatz hinsichtlich der Berücksichtigung von Abhängigkeiten zu anderen Modulen für eine individuelle Erprobung darstellen.

Im Abschnitt 7.1 wird auf die Einflussfaktoren, die auf das erste Verfahren wirken, eingegangen. Die Einflussfaktoren für das zweite Verfahren werden in Abschnitt 7.2 besprochen.

7.1 Einflüsse und Diskussion – Erstes Verfahren

Das entwickelte Verfahren zur Ermittlung und Darstellung von Abhängigkeiten basiert auf der *IRML*-Methode. Neben dieser Methode wurden in dieser Arbeit noch *FRAM* und BN als mögliche Ansätze analysiert. Das BN erfüllt einen Teil der definierten Festforderungen nicht und wurde daher nicht weiterverfolgt. Dagegen bietet *FRAM* wie die *IRML*-Methode ein Modellierungswerkzeug zur Ermittlung und Darstellung von Abhängigkeiten. Die Wahl auf die *IRML*-Methode fällt aufgrund der einfacheren Nachvollziehbarkeit und dem geringeren Aufwand, was durch die Ähnlichkeit der verwendeten Architekturen erreicht wird. Jedoch bietet *FRAM* ebenfalls das Potential, funktionale Abhängigkeiten zu ermitteln. In Grabbe et al.[137] wird diese Methode zur Risikobewertung im Entwicklungsprozess hochautomatisierter Fahrzeuge verwendet. Folglich stellt die Anwendung von *FRAM* eine an diese Arbeit anknüpfende mögliche Untersuchung dar.

Die Anwendung des Verfahrens zur Ermittlung und Darstellung von Abhängigkeiten zeigt auf, dass Dopplungen von Abhängigkeiten in den Abhängigkeitskategorien der datenbasierten und informationsbasierten Abhängigkeiten auftreten. Die datenbasierten Abhängigkeiten basieren auf den

[137] Grabbe, N. et al.: Safety of automated driving: FRAM (2020), S. 1.

Datenflüssen und die informationsbasierten Abhängigkeiten beschreiben den Datenfluss von unterschwelligen Informationen im System. Diese Dopplung wird durch denselben Datenfluss hervorgerufen, der zum einen die Daten übermittelt und zum anderen auch eine Interpretation dieser Daten zulässt.

Anhand der Ergebnisse durch die Anwendung wird deutlich, dass das ausgewählte Modellierungswerkzeug einen systematischen Ansatz zur Ermittlung von funktionalen Abhängigkeiten auch in Architekturen von automatisierten Fahrfunktionen und -systemen darstellt. Die Ergebnisse des ersten Verfahrens zeigen zudem, dass das entwickelte Verfahren bei der Identifizierung von unterschwelligen Informationen, also indirekten Abhängigkeiten, Unzugänglichkeiten aufweist. Ein Lösungsansatz zur Behebung und zur Verbesserung ist die Durchführung einer Iteration mit dem Schwerpunkt auf den Datenflüssen, um übertragene unterschwellige Informationen zu identifizieren.

7.2 Einflüsse und Diskussion – Zweites Verfahren

Das entwickelte Verfahren zur Definition von Testfällen basiert auf einem szenariobasierten Testansatz. Die Szenarien werden in dieser Arbeit expertenbasiert und anforderungsbasiert generiert. Ausschlaggebend für die Entscheidung für diese beiden Ansätze war die Anwendbarkeit in dieser Arbeit.

Bei der Anwendung des Verfahrens zur Definition von Testfällen werden die Einflüsse durch die Limitationen in der Simulationsumgebung ersichtlich. Die Einflüsse ergeben sich hauptsächlich aus dem aktuellen Stand der Simulationsmodelle. Ein Einflussfaktor ist beispielsweise, dass der VTP in der Simulationsumgebung zum jetzigen Zeitpunkt nur eingeschränkte Fähigkeiten aufweist und daher die Entwicklung einiger Szenarien nicht umsetzbar ist. Darüber hinaus ist ein weiterer Faktor die zurzeit der Anfertigung der Ausarbeitung fehlende Integration der im Projekt entwickelten FZS in das Verzeichnis mit dem VTP. Ein dritter Einflussfaktor ist die Darstellung von äußeren Effekten auf die Sensormodule in der Simulationsumgebung. Ein Beispiel dafür ist die Beeinflussung der Sensorsichtbereiche durch Bodenwellen oder durch andere äußere Einflüsse, die zu einer wechselnden Erkennung von Objekten führen.

Jeder der genannten Aspekte hat Limitationen bei der Umsetzung und Durchführung von Testfällen zur Folge. Das Ziel ist daher die Beseitigung der Limitationen, sodass bei Tests in der Simulationsumgebung die gleichen wie im Konzept beschriebenen Fähigkeiten realisierbar sind.

8 Fazit und Ausblick

Die Einführung automatisierter Fahrzeuge mit Fahrerassistenzfunktionen und -systemen der Automatisierungsstufen 3, 4 und 5 erfordern neue Test- und Absicherungsverfahren für die Sicherstellung der funktionalen Sicherheit. Konventionelle Ansätze wie reale Testfahrten sind technisch und ökonomisch keine durchführbare Methode, um den erforderlichen Abdeckungsgrad zu erreichen. Ein im Projekt UNICAR*agil* verfolgtes Forschungsziel ist die Entwicklung von modularen und dienstorientierten automatisierten Fahrzeugen, deren modulare Architektur eine individuelle Erprobung und somit eine Abkehr von der Absicherung des Gesamtsystems ermöglicht. Die individuelle Erprobung einzelner Module erfordert die Berücksichtigung der Abhängigkeiten zu anderen Modulen und der Umgebung. Im Rahmen des Projekts UNICAR*agil* wird eine Methode zur Erbringung eines validen Sicherheitsnachweises erforscht und entwickelt.

Auf Basis dieser Motivation wurde in dieser Masterthesis ein Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung entwickelt. Dieses Verfahren hat zum Ziel, den Validierungs- und Verifikationsaufwand beim Sicherheitsnachweis für die Freigabe automatisierter Fahrfunktionen und -systeme zu reduzieren. Die identifizierten Abhängigkeiten zwischen den Modulen und zur Umgebung ermöglichen individuelle Modultests. Voraussetzung für die individuelle Erprobung ist die vollständige Aufdeckung der vorliegenden Abhängigkeiten. Mit individuellen Modultests wird der enorme Aufwand herkömmlicher System- und Integrationstests umgangen. Mit der Identifizierung der Abhängigkeiten eines Moduls ist es zudem möglich, dieses Modul in mehreren Fahrzeugmodellen zu verwenden, ohne dass die Notwendigkeit besteht, eine Gesamtabsicherung aufgrund der neuen Konstellation der Module durchzuführen. Ein weiterer Vorteil von identifizierten Abhängigkeiten eines Moduls besteht darin, bei Änderungen oder einem Update an diesem Modul lediglich dieses Modul abzusichern, da die anderen Module nicht betroffen sind. Diese Möglichkeiten der individuellen Erprobung aufgrund der identifizierten Abhängigkeiten der Module reduzieren infolgedessen den Test- und Absicherungsaufwand.

Neben diesem Verfahren zur Ermittlung und Darstellung von Abhängigkeiten wurde ein weiteres Verfahren entwickelt. Dieses zweite Verfahren zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten verfolgt das Ziel, die mit dem ersten Verfahren ermittelten Abhängigkeiten aufzuzeigen, obwohl diese vorher nicht offensichtlich waren, und weitere unbekannte Abhängigkeiten aufzudecken.

Zur Lösung der Aufgabenstellung wurden zu Beginn der Entwicklung beider Verfahren eine umfangreiche Literaturrecherche zum Stand der Technik durchgeführt und Anforderungen an die beiden Verfahren definiert. Der Lösungsraum möglicher Verfahren aus der Literatur wurde anhand der definierten Anforderungen bewertet.

Das Ergebnis der ersten Literaturrecherche zu Verfahren zur Ermittlung und Darstellung von Abhängigkeiten ist, dass mit der *Functional Resonance Analysis Method* und der *Infrastructure-Resilience-oriented Modelling Language* zwei vielversprechende Ansätze vorhanden sind. Diese stellen ein Modellierungswerkzeug dar, um Abhängigkeiten mit Fokus auf funktionale Abhängigkeiten zu ermitteln. Die *IRML*-Methode wurde für kritische Infrastrukturarchitekturen entwickelt und unterstützt mit

entsprechenden graphischen Elementen bei der Modellierung und Ermittlung von funktionalen Abhängigkeiten. Da die Systemarchitektur eines automatisierten Fahrzeugs einer Infrastrukturarchitektur ähnelt, beispielsweise bei Stromnetzen, ist daher hinsichtlich der Übertragbarkeit, Nachvollziehbarkeit und dem Aufwand die Entscheidung für die *IRML*-Methode gefallen. Diese Methode wurde anschließend an die in dieser Arbeit gewünschte Anwendung angepasst, sodass im Anschluss ein Verfahren zur Ermittlung und Darstellung von Abhängigkeiten hinsichtlich einer modularen Absicherung vorliegt.

Die zweite Literaturrecherche zu Verfahren zur Definition von Testfällen auf Grundlage ermittelter Abhängigkeiten hat ergeben, dass der szenariobasierte Ansatz zur Testfallgenerierung eine vielversprechende Lösung der Herausforderungen beim Sicherheitsnachweis von automatisierten Fahrfunktionen und -systemen ab der Automatisierungsstufe 3 darstellt. Für die Szenariogenerierung wurden der expertenbasierte und der anforderungsbasierte Ansatz ausgewählt, um auf Basis der identifizierten Abhängigkeiten Szenarien zu entwickeln und daraus Testfälle zu definieren. Mit der Entwicklung des zweiten Verfahrens ist der methodische Teil dieser Arbeit abgeschlossen.

Im ersten Teil des Hauptteils der Arbeit wurde anhand der Systemarchitektur eines UNICAR*agil*-Fahrzeugs das entwickelte Verfahren zur Ermittlung und Darstellung von Abhängigkeiten angewendet. Dabei sind mit Fokus auf die Fahrdynamikzustandsschätzung, den Verhaltens- und Trajektorienplaner, die Fahrschlauchabsicherung und die Fahrdynamik- und Trajektorienregelung die vorliegenden Abhängigkeiten identifiziert worden. Das erarbeitete Verfahren zur Ermittlung und Darstellung von Abhängigkeiten stellt einen systematischen Ansatz dar, die existierenden Abhängigkeiten zu ermitteln. Die Anwendung des Verfahrens hat gezeigt, dass ein tiefes und präzises Systemverständnis für jedes einzelne Modul Voraussetzung für eine Ermittlung der Abhängigkeiten ist. Dazu eignet sich in einem ersten Schritt eine intensive Einarbeitung in die Architektur und die einzelnen Module in Form von Literatur, von Beschreibungen oder von Simulationsmodellen. In einem zweiten Schritt sind Gespräche mit den Entwicklern der einzelnen Module hilfreich, um über die Literatur hinaus weitere Informationen über den Aufbau der Module und zu möglichen Auslösern von Zustands- und Verhaltensänderungen zu erhalten. Das Ergebnis der Anwendung des Verfahrens ist, dass eine Definition von Abhängigkeitskategorien und eine Einteilung der Abhängigkeiten notwendig ist, um eine Übersicht über die hohe Anzahl an Abhängigkeiten zu erhalten. Die Bewertung der ermittelten Abhängigkeiten und des Verfahrens hat aufgezeigt, dass der Fokus des Verfahrens auf den offensichtlichen Abhängigkeiten liegt und daher das Verfahren bei diesem Aspekt Verbesserungspotential aufweist. Ein Lösungsansatz ist die Durchführung einer weiteren Iteration mit dem alleinigen Fokus auf die datenbasierten Abhängigkeiten und die Identifizierung von unterschwelligen Informationen, die übertragen werden und ebenfalls Abhängigkeiten darstellen.

Unter Verwendung der mit dem ersten Verfahren in der Systemarchitektur von UNICAR*agil* ermittelten Abhängigkeiten wurde anschließend im zweiten Teil des Hauptteils der Arbeit das zweite Verfahren zur Definition von Testfällen angewendet. Mit dem experten- und anforderungsbasierten Ansatz zur Szenariogenerierung ließen sich mehrere Szenarien für die verschiedenen Abhängigkeitskategorien erzeugen. Die Testszenarien wurden anschließend in der Simulationsumgebung umgesetzt. Die durchgeführten Testfälle haben die identifizierten Abhängigkeiten bestätigt. Außerdem

ermöglichen die Definition und die Durchführung von Testfällen die Aufdeckung von weiteren Abhängigkeiten.

Insgesamt stellen die beiden entwickelten Verfahren einen erfolgversprechenden Ansatz dar, die in einer modularen Architektur vorliegenden Abhängigkeiten zu ermitteln, diese Abhängigkeiten mithilfe von Testfällen zu bestätigen und dabei noch weitere Abhängigkeiten aufzudecken.

In dieser Arbeit lag der Schwerpunkt auf der Entwicklung beider Verfahrenskonzepte und deren Anwendung mit Fokus auf die Fahrdynamikzustandsschätzung, den Verhaltens- und Trajektorienplaner, die Fahrschlauchabsicherung und die Fahrdynamik- und Trajektorienregelung. Erst mit der vollständigen Identifizierung der Abhängigkeiten zwischen den Modulen und zur Umgebung ist eine individuelle Erprobung möglich. Ein Anknüpfungspunkt an diese Arbeit, um mit reduziertem Aufwand die Abhängigkeiten zur Umgebung, die als ein weiteres Modul darstellbar ist, zu ermitteln, ist die Verknüpfung mit einem anderen Forschungsprojekt am Fachgebiet FZD. *Perception sensor Collaborative Effect and Cause Tree (PerCollECT)*[138] ist ein Ursache-Wirkungsbaum, in dem die Einflussfaktoren auf verschiedene Sensortypen eines Fahrzeugs identifiziert und in Graphenform zusammengestellt sind. Bei den Sensortypen handelt es sich um Umgebungswahrnehmungssensoren wie z.B. Radar, Lidar, Kamera und Ultraschall. In *PerCollECT* wird bei den Einflussfaktoren zwischen Systemdesign-Parametern, z.B. die Physik des Antennendesigns oder der Auflösung der Analog/Digital-Wandlung, und systemunabhängigen Ursachen, z.B. Wetterbedingungen, unterschieden. Diese beiden Arten von Einflussfaktoren lassen sich auf die in dieser Arbeit definierten Abhängigkeitskategorien übertragen. Die systemdesignbasierten Abhängigkeiten beschreiben ein ähnliches Verhältnis wie die Systemdesign-Parameter in *PerCollECT*. Die systemunabhängigen Ursachen lassen sich auf die systemunabhängigen Abhängigkeiten übertragen. Bei den ermittelten Ursache-Wirkungsketten handelt es sich im Prinzip um Abhängigkeiten. Daher sind die Einflussfaktoren bei der Umfelderfassung und infolgedessen die Abhängigkeiten zwischen Umgebung und Sensoren mit wenig Aufwand in das Abhängigkeitsnetz zu integrieren.

Eine Forschungsfrage für nachfolgende Arbeiten ist, wie die Unzugänglichkeiten hinsichtlich der Vollständigkeit bei den unterschwelligen Abhängigkeiten, den informationsbasierten Abhängigkeiten, zu beheben sind. Das Ziel dabei ist, weitere Abhängigkeiten aufzudecken und somit die Anforderungen einer individuellen Erprobung zu erfüllen und das Potential des Verfahrens auszuschöpfen.

Abschließend bleibt anzumerken, dass diese Arbeit einen Beitrag zu neuartigen Absicherungsmethoden für automatisierte Fahrzeuge liefert. Dennoch bleibt offen, ob nach der Anwendung der Verfahren auf das Gesamtsystem, also der Ermittlung der Abhängigkeiten in der Systemarchitektur im Projekt UNICAR*agil*, eine Durchführung von Modultests möglich ist oder ob noch weitere und bislang unbekannte Abhängigkeiten vorliegen und welche Optionen in diesem Fall bestehen, um das Verfahren gegebenenfalls noch weiter zu verfeinern und auszuarbeiten.

[138] Rosenberger, P.; Linnhoff, C.: PerCollECT.

Anhang

A.1 Abhängigkeitsnetz

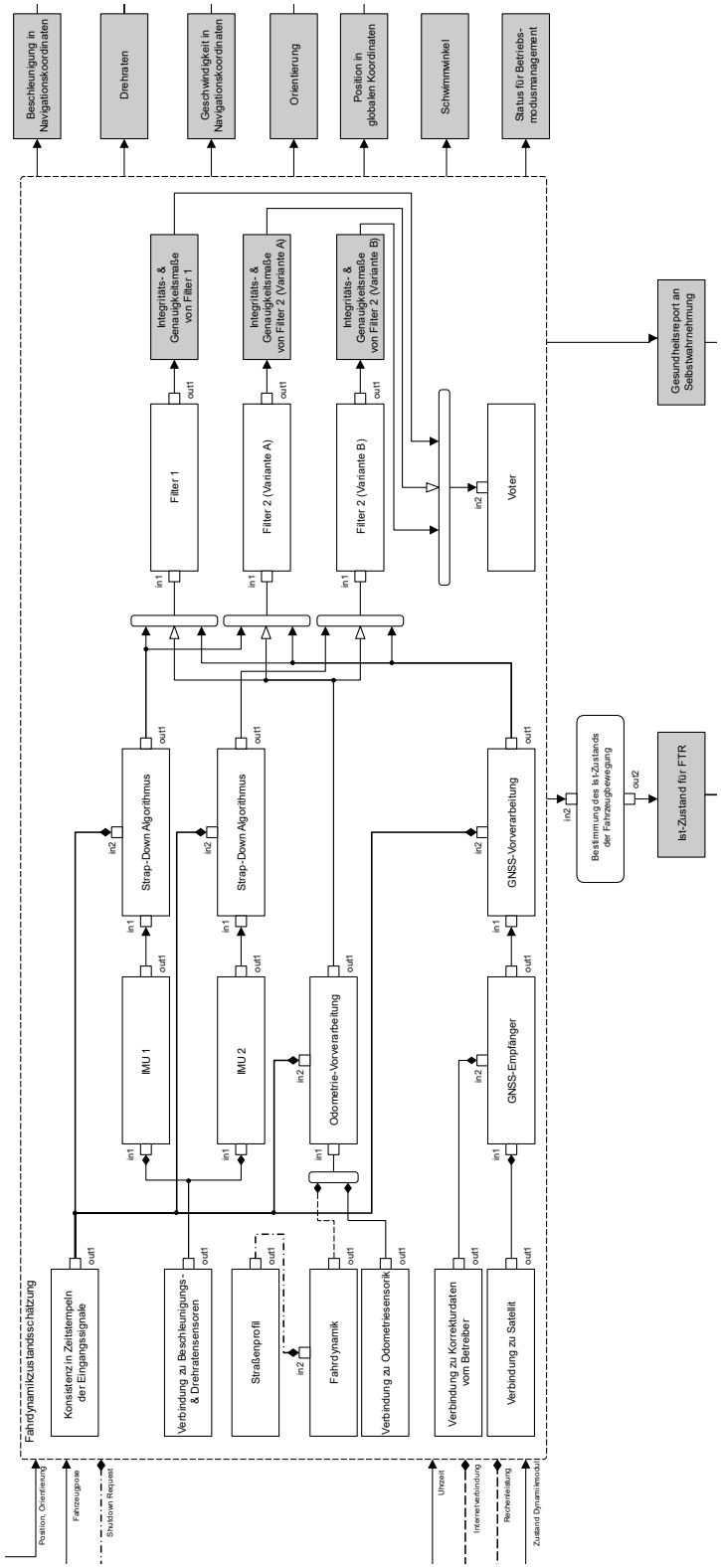

Abbildung A-1: Abhängigkeitsnetz – Ausschnitt Fahrdynamikzustandsschätzung

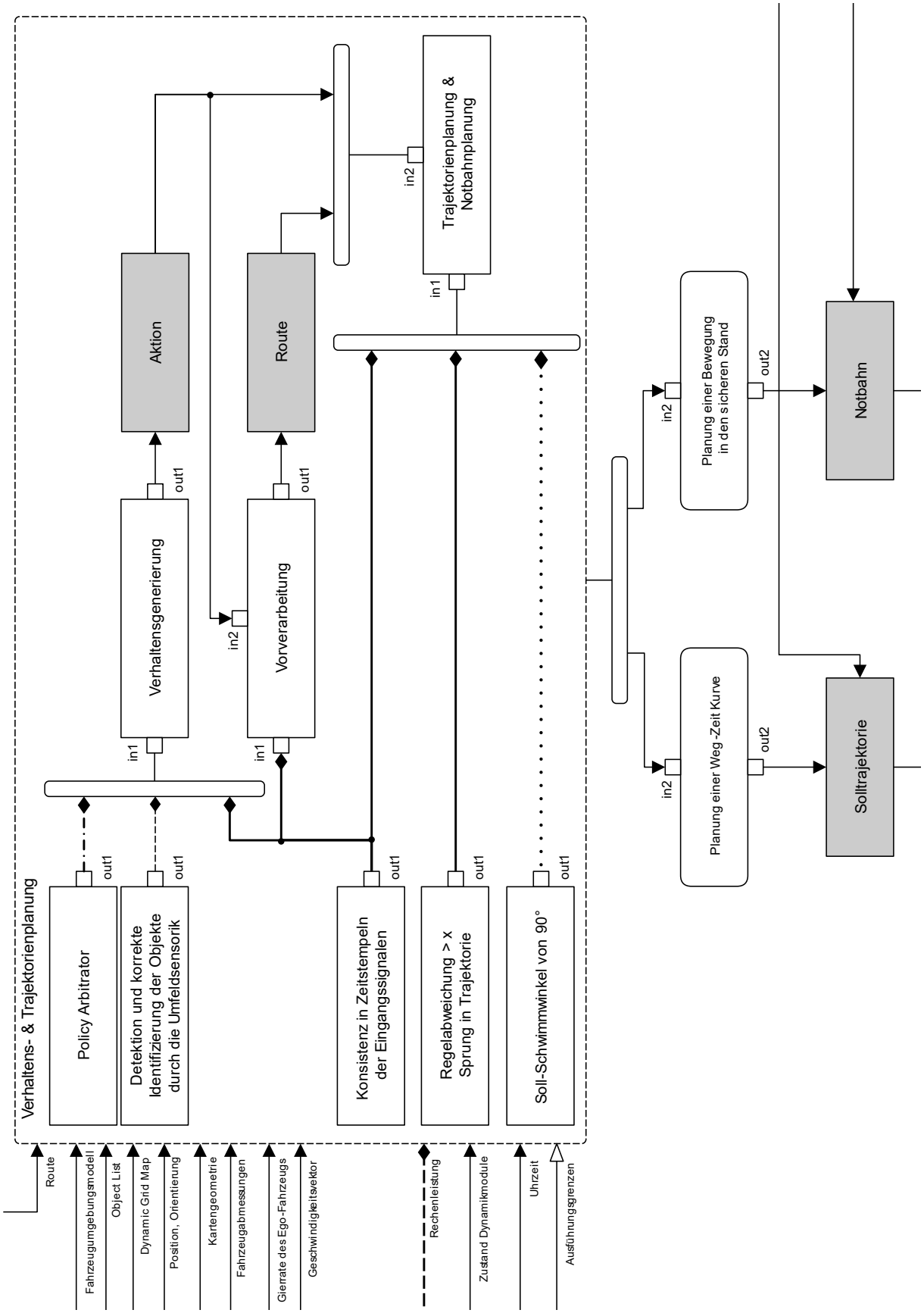

Abbildung A-2: Abhängigkeitsnetz – Ausschnitt Verhaltens- und Trajektorienplaner

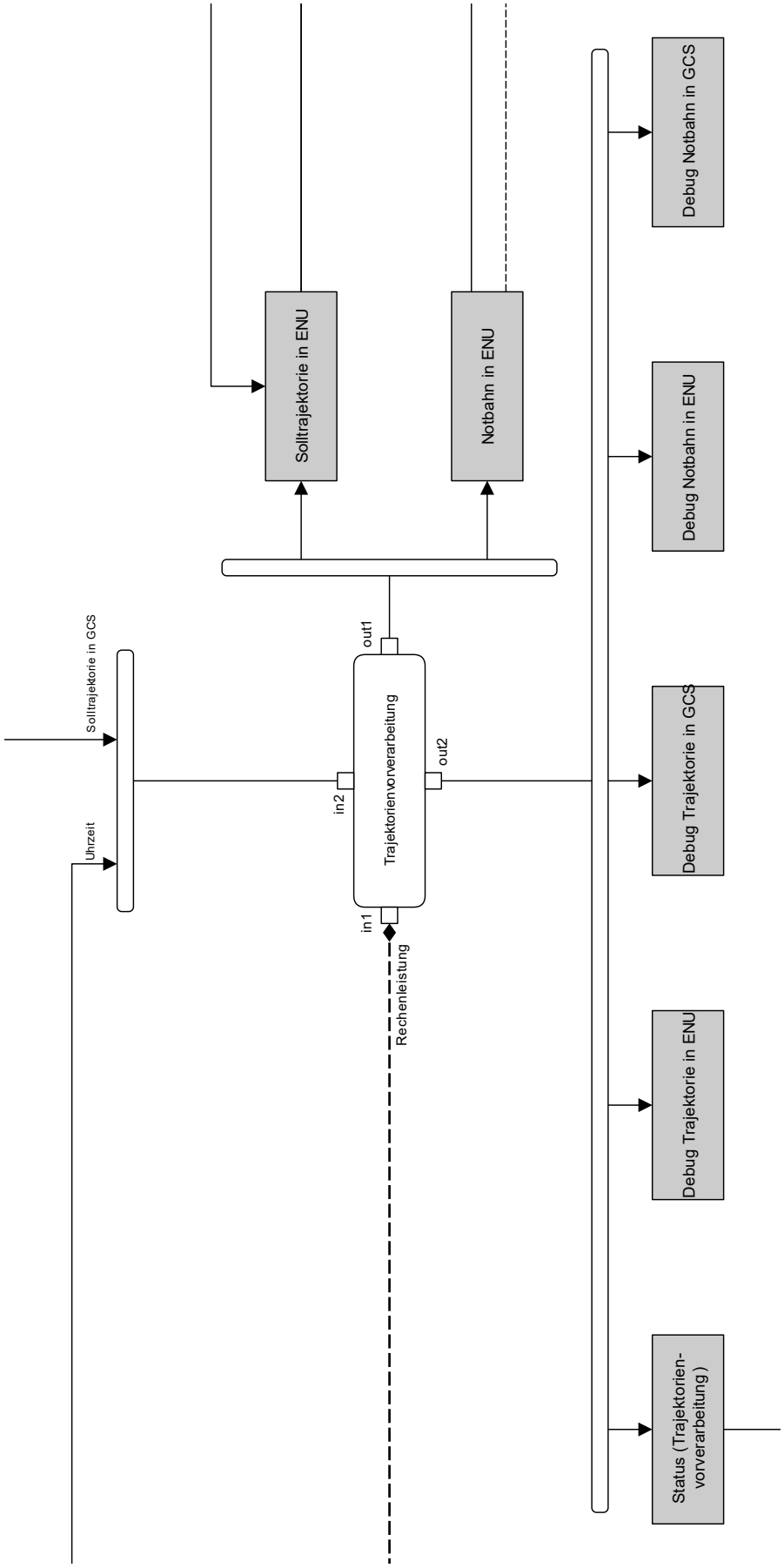

Abbildung A-3: Abhängigkeitsnetz – Ausschnitt Trajektorienvorverarbeitung

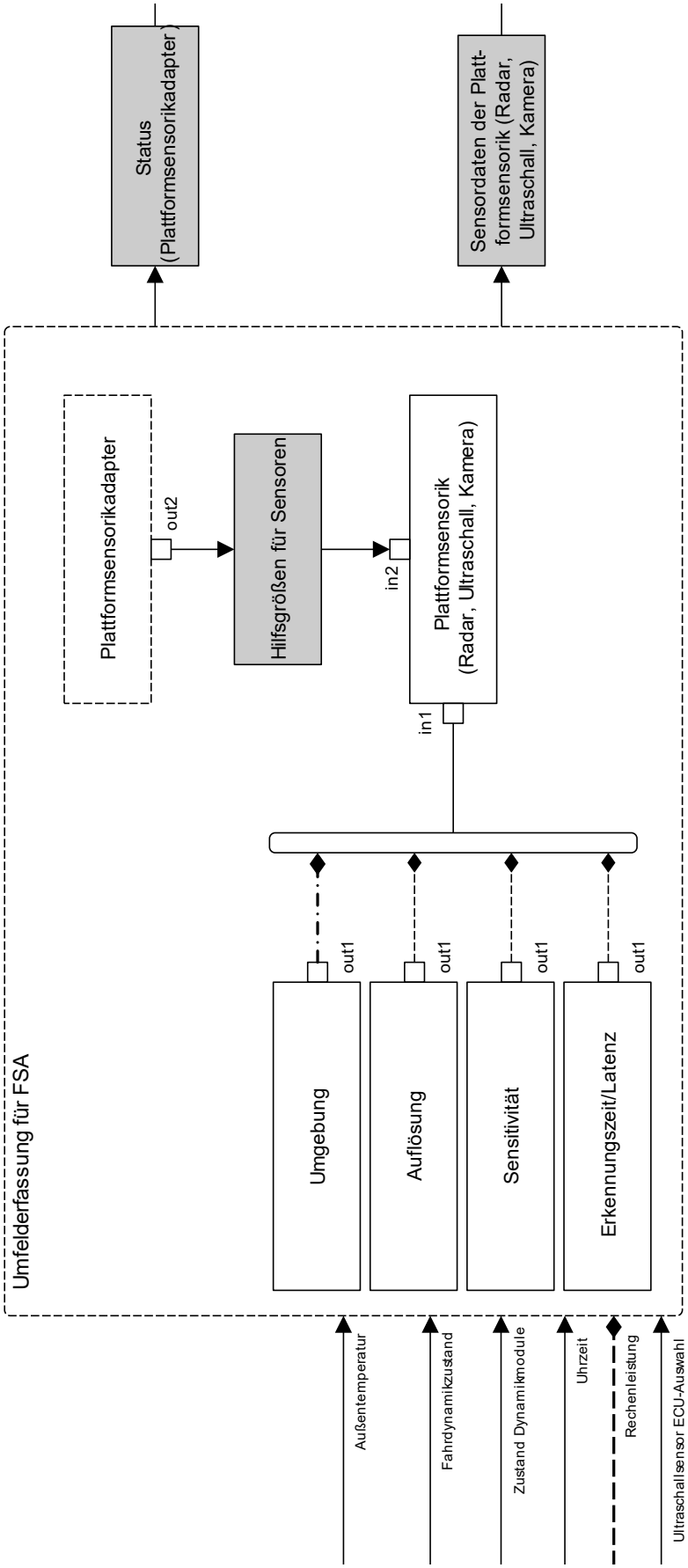

Abbildung A-4: Abhängigkeitsnetz – Ausschnitt Plattformsensorik

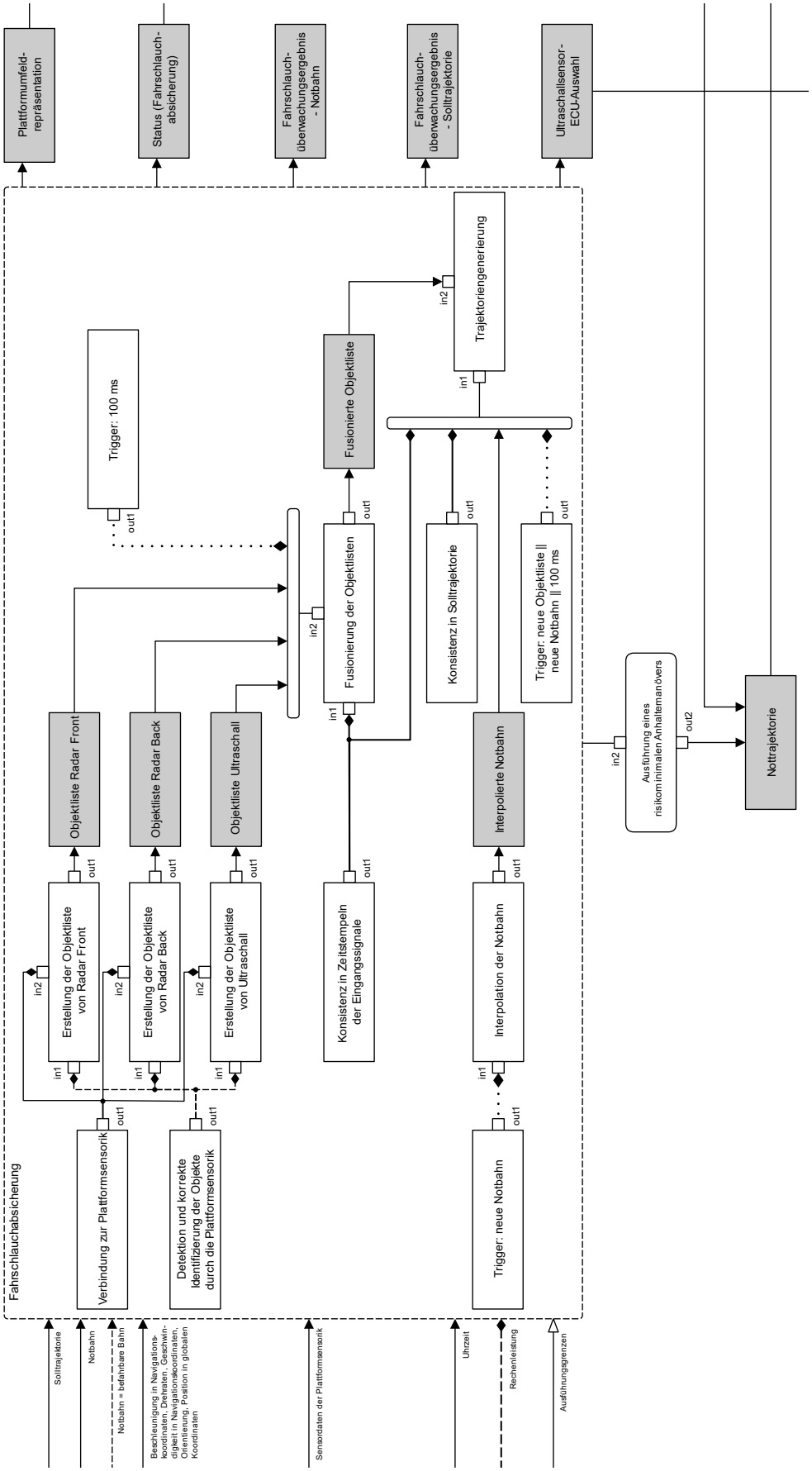

Abbildung A-5: Abhängigkeitsnetz – Ausschnitt Fahrschlauchabsicherung

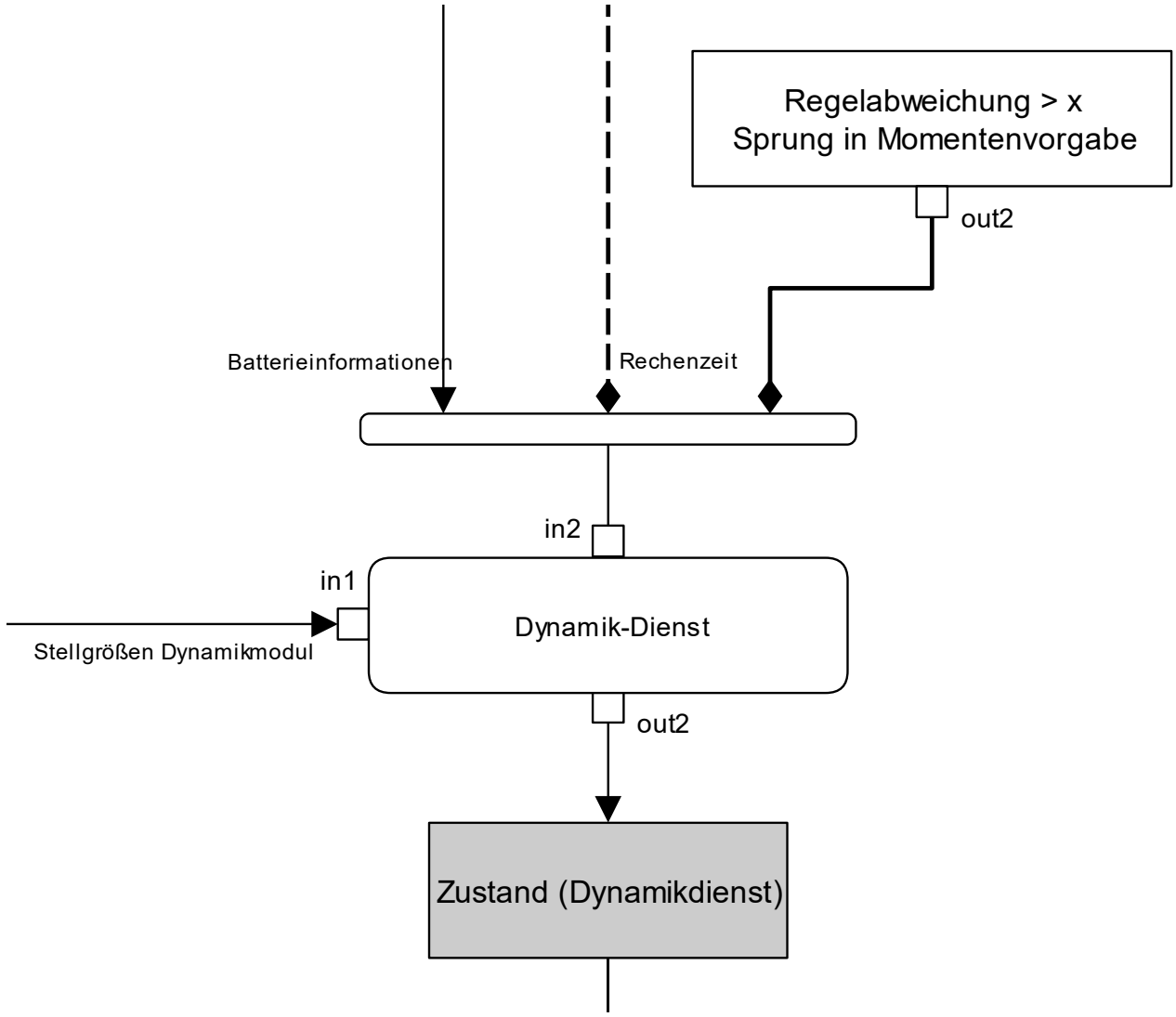

Abbildung A-6: Abhängigkeitsnetz – Ausschnitt Dynamikmodul

A.2 Ermittelte Abhängigkeiten – Fahrdynamikzustandsschätzung

Tabelle A-1: Ermittelte Abhängigkeiten – Fahrdynamikzustandsschätzung

ID	Abhängigkeit	Beteiligte Module	Gruppe	Beschreibung
DYM_FZS_001	Zustand Dynamikmodul	Dynamikmodul - FZS	Datenbasierte Abhängigkeit	
VLOF_FZS_002	Fahrzeugpose	Videobasierte Lokalisierung - FZS	Datenbasierte Abhängigkeit	
LOF_FZS_003	Position	Lokalisierungsfusion – FZS	Datenbasierte Abhängigkeit	
LOF_FZS_004	Orientierung	Lokalisierungsfusion – FZS	Datenbasierte Abhängigkeit	
FZS_FZS_005	Uhrzeit	FZS	Datenbasierte Abhängigkeit	
FZS_FZS_006	Daten der IMU 1 für Strap-Down Algorithmus (IMU 1)	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_007	Daten aus Strap-Down-Algorithmus (IMU 1) für Filter 1	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_008	Daten aus Strap-Down-Algorithmus (IMU 1) für Filter 2 (Variante A)	FZS - FZS	Datenbasierte Abhängigkeit	Filter 2 (Variante A) ist nach aktuellem Stand (Februar 2022) nicht vorhanden
FZS_FZS_009	Daten der IMU 2 für Strap-Down Algorithmus (IMU 2)	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_010	Daten aus Strap-Down-Algorithmus (IMU 2) für Filter 2 (Variante B)	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_011	Daten der Odometrie-Vorverarbeitung für Filter 1	FZS - FZS	Datenbasierte Abhängigkeit	Odometrie wird nach aktuellem Stand (Februar 2022) nicht genutzt
FZS_FZS_012	Daten der Odometrie-Vorverarbeitung für Filter 2 (Variante A)	FZS - FZS	Datenbasierte Abhängigkeit	Odometrie wird nach aktuellem Stand (Februar 2022) nicht genutzt
FZS_FZS_013	Daten der Odometrie-Vorverarbeitung für Filter 2 (Variante B)	FZS - FZS	Datenbasierte Abhängigkeit	Odometrie wird nach aktuellem Stand (Februar 2022) nicht genutzt
FZS_FZS_014	Daten des GNSS-Empfängers für GNSS-Vorverarbeitung	FZS - FZS	Datenbasierte Abhängigkeit	

FZS_FZS_015	Daten der GNSS-Vorverarbeitung für Filter 1	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_016	Daten der GNSS-Vorverarbeitung für Filter 2 (Variante A)	FZS - FZS	Datenbasierte Abhängigkeit	Filter 2 (Variante A) ist nach aktuellem Stand (Februar 2022) nicht vorhanden
FZS_FZS_017	Daten der GNSS-Vorverarbeitung für Filter 2 (Variante B)	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_018	Integritäts- & Genauigkeitsmaße von Filter 1 für den Voter	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_019	Integritäts- & Genauigkeitsmaße von Filter 2 (Variante A) für den Voter	FZS - FZS	Datenbasierte Abhängigkeit	Filter 2 (Variante A) ist nach aktuellem Stand (Februar 2022) nicht vorhanden
FZS_FZS_020	Integritäts- & Genauigkeitsmaße von Filter 2 (Variante B) für den Voter	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_009	Daten der FZS für die Bestimmung des Ist-Zustamds der Fahrzeugbewegung	FZS - FZS	Datenbasierte Abhängigkeit	
FZS_FZS_010	Konsistenz in Zeitstempeln der Eingangssignal auf Strap-Down Alorithmus (IMU 2)	FZS - FZS	Inter-Datenbasierte Abhängigkeit	
FZS_FZS_011	Konsistenz in Zeitstempeln der Eingangssignal auf Strap-Down Alorithmus (IMU 1)	FZS - FZS	Inter-Datenbasierte Abhängigkeit	
FZS_FZS_012	Konsistenz in Zeitstempeln der Eingangssignal für Odometrie-Vorverarbeitung	FZS - FZS	Inter-Datenbasierte Abhängigkeit	
FZS_FZS_013	Konsistenz in Zeitstempeln der Eingangssignal für GNSS-Vorverarbeitung	FZS - FZS	Inter-Datenbasierte Abhängigkeit	
FZS_FZS_014	Verbindung zu Beschleunigungs- & Drehratensensoren (IMU 1)	FZS - FZS	Datentransferbasierte Abhängigkeit	
FZS_FZS_015	Verbindung zu Beschleunigungs- & Drehratensensoren (IMU 2)	FZS - FZS	Datentransferbasierte Abhängigkeit	

FZS_FZS_016	Verbindung zwischen Korrekturdaten vom Betreiber und GNSS-Empfänger	FZS - FZS	Datentransferbasierte Abhängigkeit	
FZS_FZS_017	Verbindung zwischen Satellit und GNSS-Empfänger	FZS - FZS	Datentransferbasierte Abhängigkeit	
FZS_FZS_018	Verbindung zwischen Odometriesensorik und Odometrie-Vorverarbeitung	FZS - FZS	Datentransferbasierte Abhängigkeit	
FZS_FZS_019	Fahrdynamik auf Odometrie-Verarbeitung	FZS-FZS	Systemdesignbasierte Abhängigkeit	
STG_FZS_020	Rechenleistung	FZS Steuergerät - FZS	Ressourcenbasierte Abhängigkeit	
LTE_FZS_021	Internetverbindung	LTE-Modem - FZS	Ressourcenbasierte Abhängigkeit	
FZS_FZS_022	Shutdown Request	FZS	Systemunabhängige Abhängigkeit	
FZS_FZS_023	Straßenprofil auf Fahrdynamik	FZS-FZS	Systemunabhängige Abhängigkeit	

A.3 Ermittelte Abhängigkeiten – Verhaltens- und Trajektorienplanung

Tabelle A-2: Ermittelte Abhängigkeiten – Verhaltens- und Trajektorienplanung

ID	Abhängigkeit	Beteiligte Module	Gruppe	Beschreibung
CLO_VTP_001	Route	Cloud - VTP	Datenbasierte Abhängigkeit	
UGM_VTP_002	Fahrzeugumgebungs-modell	Fahrzeug-Umge-bungsmodell - VTP	Datenbasierte Abhängigkeit	
UGM_VTP_003	Object List	Fahrzeug-Umge-bungsmodell - VTP	Datenbasierte Abhängigkeit	
UGM_VTP_004	Dynamic Grid Map	Fahrzeug-Umge-bungsmodell - VTP	Datenbasierte Abhängigkeit	
LOF_VTP_005	Position	Lokalisierungs-fusion - VTP	Datenbasierte Abhängigkeit	
LOF_VTP_006	Orientierung	Lokalisierungs-fusion - VTP	Datenbasierte Abhängigkeit	
VTP_VTP_007	Kartengeometrie	Kartenbereitstel-lung - VTP	Datenbasierte Abhängigkeit	
VTP_VTP_008	Fahrzeugabmessungen	VTP	Datenbasierte Abhängigkeit	
FZS_VTP_009	Gierrate des Ego-Fahr-zeugs	FZS - VTP	Datenbasierte Abhängigkeit	
FZS_VTP_010	Geschwindigkeitsvek-tor	FZS - VTP	Datenbasierte Abhängigkeit	
DYM_VTP_011	Zustand Dynamikmo-dul	DYM - VTP	Datenbasierte Abhängigkeit	
FZS_VTP_012	Uhrzeit	VTP	Datenbasierte Abhängigkeit	
FTR_VTP_013	Ausführungsgrenzen	FTR - VTP	Datenbasierte Abhängigkeit	Im Konzept vorhan-den, aber Ausfüh-rungsgrenzen werden in der Umsetzung nicht verwendet
VTP_VTP_014	Aktion für die Trajekto-rien- & Notbahnpla-nung	VTP - VTP	Datenbasierte Abhängigkeit	
VTP_VTP_015	Aktion für die Vorver-arbeitung	VTP - VTP	Datenbasierte Abhängigkeit	
VTP_VTP_016	Route für die Trajekto-rien- & Notbahnpla-nung	VTP - VTP	Datenbasierte Abhängigkeit	
VTP_VTP_017	Daten des VTP für die Planung einer Weg-Zeit Kurve	VTP - VTP	Datenbasierte Abhängigkeit	

VTP_VTP_018	Daten des VTP für die Planung einer Bewegung in den sicheren Stand	VTP - VTP	Datenbasierte Abhängigkeit	
VTP_VTP_019	Konsistenz in Zeitstempeln der Eingangssignale für der Verhaltensgenerierung	VTP - VTP	Inter-Datenbasierte Abhängigkeit	
VTP_VTP_020	Konsistenz in Zeitstempeln der Eingangssignale für der Vorverarbeitung	VTP - VTP	Inter-Datenbasierte Abhängigkeit	
VTP_VTP_021	Konsistenz in Zeitstempeln der Eingangssignale für der Trajektorien- & Notbahnplanung	VTP - VTP	Inter-Datenbasierte Abhängigkeit	
VTP_VTP_022	Regelabweichung > x Sprung in Trajektorie	VTP - VTP	Inter-Datenbasierte Abhängigkeit	
VTP_VTP_023	Detektion und korrekte Identifizierung der Objekte durch Umfeldsensorik	VTP - VTP	Systemdesignbasierte Abhängigkeit	
STG_VTP_024	Rechenleistung	Großhirn Steuergerät - VTP	Ressourcenbasierte Abhängigkeit	
VTP_VTP_025	Sollschwimmwinkel von 90°	VTP - VTP	Codebasierte Abhängigkeit	
VTP_VTP_026	Policy Arbitrator	VTP - VTP	Systemunabhängige Abhängigkeit	

A.4 Ermittelte Abhängigkeiten – Trajektorienvorverarbeitung

Tabelle A-3: Ermittelte Abhängigkeiten – Trajektorienvorverarbeitung

ID	Abhängigkeit	Beteiligte Module	Gruppe	Beschreibung
VTP_TVV_001	Solltrajektorie in GCS vom VTP	VTP - TVV	Datenbasierte Abhängigkeit	
VTP_TVV_002	Notbahn in GCS vom VTP	VTP - TVV	Datenbasierte Abhängigkeit	
TSLTW_ TVV_003	Solltrajektorie in GCS von LTW	TSLTW - TVV	Datenbasierte Abhängigkeit	
TSLTW_TVV_004	Notbahn in GCS vom LTW	TSLTW - TVV	Datenbasierte Abhängigkeit	
TVV_TVV_005	Uhrzeit	TVV	Datenbasierte Abhängigkeit	
STG_TVV_006	Rechenleistung	Stammhirn Steu-ergerät - TVV	Ressourcenbasierte Abhängigkeit	

A.5 Ermittelte Abhängigkeiten – Plattformsensorik

Tabelle A-4: Ermittelte Abhängigkeiten – Plattformsensorik

ID	Abhängigkeit	Beteiligte Module	Gruppe	Beschreibung
PTS_PTS_001	Außentemperatur	PTS - PTS	Datenbasierte Abhängigkeit	
FZS_PTS_002	Fahrdynamikzustand	FZS - PTS	Datenbasierte Abhängigkeit	
DYM_PTS_003	Zustand Dynamikmodule	DYM - PTS	Datenbasierte Abhängigkeit	
PTS_PTS_004	Uhrzeit	PTS - PTS	Datenbasierte Abhängigkeit	
FSA_PTS_005	Ultraschallsensor ECU-Auswahl	FSA - PTS	Datenbasierte Abhängigkeit	
PTS_PTS_006	Hilfsgrößen für Sensoren vom Plattformsensorikadapter für Plattformsensorik	PTS - PTS	Datenbasierte Abhängigkeit	
PTS_PTS_007	Auflösung	PTS - PTS	Systemdesignbasierte Abhängigkeit	
PTS_PTS_008	Sensitivität	PTS - PTS	Systemdesignbasierte Abhängigkeit	
PTS_PTS_009	Erkennungszeit/Latenz	PTS - PTS	Systemdesignbasierte Abhängigkeit	
STG_PTS_010	Rechenleistung	Stammhirn Steuergerät - Plattformsensorik	Ressourcenbasierte Abhängigkeit	
PTS_PTS_011	Umgebung	PTS - PTS	Systemunabhängige Abhängigkeit	

A.6 Ermittelte Abhängigkeiten – Fahrschlauchabsicherung

Tabelle A-5: Ermittelte Abhängigkeiten – Fahrschlauchabsicherung

ID	Abhängigkeit	Beteiligte Module	Gruppe	Beschreibung
TVV_FSA_001	Solltrajektorie in ENU	TVV - FSA	Datenbasierte Abhängigkeit	
TVV_FSA_002	Notbahn in ENU	TVV - FSA	Datenbasierte Abhängigkeit	
FZS_FSA_003	Orientierung	FZS - FSA	Datenbasierte Abhängigkeit	
FZS_FSA_004	Position	FZS - FSA	Datenbasierte Abhängigkeit	
FZS_FSA_005	Geschwindigkeit	FZS - FSA	Datenbasierte Abhängigkeit	
FZS_FSA_006	Beschleunigung	FZS - FSA	Datenbasierte Abhängigkeit	
FZS_FSA_007	Drehraten	FZS - FSA	Datenbasierte Abhängigkeit	
PTS_FSA_008	Sensordaten Ultraschall (Plattformsensorik)	PTS - FSA	Datenbasierte Abhängigkeit	
PTS_FSA_009	Sensordaten Radar Back (Plattformsensorik)	PTS - FSA	Datenbasierte Abhängigkeit	
PTS_FSA_010	Sensordaten Radar Front (Plattformsensorik)	PTS - FSA	Datenbasierte Abhängigkeit	
FSA_FSA_011	Uhrzeit	FSA	Datenbasierte Abhängigkeit	
FTR_FSA_012	Ausführungsgrenzen	FTR - FSA	Datenbasierte Abhängigkeit	Im Konzept vorhanden, aber Ausführungsgrenzen werden in der Umsetzung nicht verwendet
FSA_FSA_013	Objektliste Ultraschall für Fusionierung der Objektlisten	FSA - FSA	Datenbasierte Abhängigkeit	
FSA_FSA_014	Objektliste Radar Back für Fusionierung der Objektlisten	FSA - FSA	Datenbasierte Abhängigkeit	
FSA_FSA_015	Objektliste Radar Front für Fusionierung der Objektlisten	FSA - FSA	Datenbasierte Abhängigkeit	
FSA_FSA_016	Interpolierte Notbahn für Trajektoriengenerierung	FSA - FSA	Datenbasierte Abhängigkeit	

FSA_FSA_017	Fusionierte Objektliste für Trajektoriengenerierung	FSA - FSA	Datenbasierte Abhängigkeit	
FSA_FSA_018	Daten der FSA für die Ausführung eines risikominimalen Anhaltemanövers	FSA - FSA	Datenbasierte Abhängigkeit	
VTP_FSA_019	Notbahn wird als befahrbare Bahn interpretiert	VTP - FSA	Informationsbasierte Abhängigkeit	Die grundsätzliche Befahrbarkeit der Notbahn wurde bei der Planung der abgesichert (z.B. Bordstein oder Graben wird von der Plattformsensorik nicht identifiziert)
FSA_FSA_020	Konsistenz in Zeitstempeln der Eingangssignale für Fusionierung der Objektliste	FSA - FSA	Inter-Datenbasierte Abhängigkeit	
FSA_FSA_021	Konsistenz in Zeitstempeln der Eingangssignale für Trajektoriengenerierung	FSA - FSA	Inter-Datenbasierte Abhängigkeit	
FSA_FSA_022	Konsistenz in Solltrajektorie	FSA - FSA	Inter-Datenbasierte Abhängigkeit	
FSA_FSA_023	Verbindung von der Plattformsensorik zur Sensor Schnittstelle Ultraschall	FSA - FSA	Datentransferbasierte Abhängigkeit	
FSA_FSA_024	Verbindung von der Plattformsensorik zur Sensor Schnittstelle Radar Back	FSA - FSA	Datentransferbasierte Abhängigkeit	
FSA_FSA_025	Verbindung von der Plattformsensorik zur Sensor Schnittstelle Radar Front	FSA - FSA	Datentransferbasierte Abhängigkeit	
FSA_FSA_026	Detektion und korrekte Identifizierung der Objekte durch die Plattformsensorik	FSA - FSA	Systemdesignbasierte Abhängigkeit	
STG_FSA_027	Rechenleistung	Stammhirn Steuergerät - FSA	Ressourcenbasierte Abhängigkeit	
FSA_FSA_028	Trigger alle 100ms für die Fusionierung der Objektlisten	FSA	Codebasierte Abhängigkeit	
FSA_FSA_029	Trigger bei neuer Objektliste, neuer Notbahn	FSA	Codebasierte Abhängigkeit	

	oder alle 100ms für die Trajektoriengenerierung			
FSA_FSA_030	Trigger bei neuer Notbahn für die Interpolation der Notbahn	FSA	Codebasierte Abhängigkeit	

A.7 Ermittelte Abhängigkeiten – Dynamikmodul

Tabelle A-6: Ermittelte Abhängigkeiten – Dynamikmodul

ID	Abhängigkeit	Beteiligte Module	Gruppe	Beschreibung
FTR_DYM_001	Stellgrößen von FTR für Dynamikmodule	FTR - DYM	Datenbasierte Abhängigkeit	
DLTWS_DYM_002	Stellgrößen von direkter Leitwartensteuerung für Dynamikmodule	DLTWS - DYM	Datenbasierte Abhängigkeit	
BMM_DYM_003	Stellgrößen von Betriebsmodusmanagement für Dynamikmodule	BMM - DYM	Datenbasierte Abhängigkeit	
BTM_DYM_004	Batterieinformationen	BTM - DYM	Datenbasierte Abhängigkeit	
DYM_DYM_005	Regelabweichung > x Sprung in Momentenvorgabe	DYM - DYM	Inter-Datenbasierte Abhängigkeit	
STG_DYM_006	Rechenzeit	Dynamikmodul Steuergerät - DYM	Ressourcenbasierte Abhängigkeit	

Literaturverzeichnis

Abbas, H. et al.: Requirement and risk-based scenarios (2017)
Abbas, Houssam; O'Kelly, Matthew E.; Rodionova, Alena; Mangharam, Rahul: A Driver's License Test for Driverless Vehicles, in: Mechanical Engineering (12), Jahrgang 139, S. S13-S16, 2017

Ackermann, S.; Winner, H.: Systemarchitektur beim Sicheren Anhalten (2020)
Ackermann, Stefan; Winner, Hermann: Systemarchitektur und Fahrmanöver zum Sicheren Anhalten modularer automatisierter Fahrzeuge, in: Eckstein, Lutz et al. (Hrsg.): 13. Workshop Fahrerassistenz und Automatisiertes Fahren, FAS 2020, Hotel Gut Moierhof, Walting im Altmühltal, Uni-DAS 13. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2020, Uni-DAS e.V, Darmstadt, 2020

Ackermann, S.; Winner, H.: UNICARagil - Sicheres Anhalten (2020)
Ackermann, Stefan; Winner, Hermann: UNICARagil - Digitales Halbzeitevent - Sicheres Anhalten; https://www.unicaragil.de/images/medien/news/NewsletterHZE_07-2020_DE.pdf, 2020, Zugriff 03.04.2022

Althoff, M.; Lutz, S.: Generation of Safety-Critical Test Scenarios (2018)
Althoff, Matthias; Lutz, Sebastian: Automatic Generation of Safety-Critical Test Scenarios for Collision Avoidance of Road Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, IEEE, 2018

Amersbach, C.; Winner, H.: Test Coverage for Automated Driving (2019)
Amersbach, Christian; Winner, Hermann: Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, IEEE, 2019

Arias, C. T. et al.: Dependency Analysis Solutions (2011)
Arias, Callo T. B.; van der Spek, Pieter; Avgeriou, Paris: A practice-driven systematic review of dependency analysis solutions, in: Empirical Software Engineering (5), Jahrgang 16, S. 544–586, 2011

Bach, J. et al.: Model based scenario specification (2016)
Bach, Johannes; Otten, Stefan; Sax, Eric: Model based scenario specification for development and test of automated driving functions, in: 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Schweden, IEEE, 2016

Bach, J.: Diss., Entwicklung und Test prädiktiver Fahrzeugregelungsfunktionen (2018)
Bach, Johannes: Methoden und Ansätze für die Entwicklung und den Test prädiktiver Fahrzeugregelungsfunktionen, Dissertation Karlsruher Institut für Technologie, Karlsruhe, 2018

Bagschik, G. et al.: Szenarien für automatisierte Fahrzeuge (2017)
Bagschik, Gerrit; Menzel, Till; Reschka, Andreas; Maurer, Markus: Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen, in: Bengler, Klaus et al. (Hrsg.): 11. Workshop Fahrerassistenz und Automatisiertes Fahren, FAS 2017, Hotel Gut Moierhof, Walting im Altmühltal, Uni-DAS 11. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2017, Uni-DAS e.V, Darmstadt, 2017

Bagschik, G. et al.: Architecture Framework for Automated Vehicles (2018)
Bagschik, Gerrit; Nolte, Marcus; Ernst, Susanne; Maurer, Markus: A System's Perspective Towards an Architecture Framework for Safe Automated Vehicles, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018

Bagschik, G. et al.: Ontology based Scene Creation (2018)
Bagschik, Gerrit; Menzel, Till; Maurer, Markus: Ontology based Scene Creation for the Development of Automated Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, IEEE, 2018

Bagschik, G. et al.: Wissensbasierte Szenariengenerierung für Betriebsszenarien (2018)
Bagschik, Gerrit; Menzel, Till; Körner, Christian; Maurer, Markus: Wissensbasierte Szenariengenerierung für Betriebsszenarien auf deutschen Autobahnen, in: Winner, Hermann (Hrsg.): 12. Workshop Fahrerassistenz und Automatisiertes Fahren, FAS 2018, Hotel Gut Moierhof, Walting im Altmühltal, Uni-DAS 12. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2018, Uni-DAS e.V, Darmstadt, 2018

BMWI: PEGASUS Method (2019)
BMWI: PEGASUS Method; https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf, 2019, Zugriff 05.04.2022

Buchholz, M. et al.: Automation of the UNICARagil vehicles (2020)
Buchholz, Michael; Gies, Fabian; Danzer, Andreas; Henning, Matti; Hermann, Charlotte; Herzog, Manuel; Horn, Markus; Schön, Markus; Rexin, Nils; Dietmayer, Klaus; Fernandez, Carlos; Janosovits, Johannes; Kamran, Danial; Kinzig, Christian; Lauer, Martin; Molinos, Eduardo; Stiller, Christoph; Ackermann, Stefan; Homolla, Tobias; Winner, Hermann; Gottschalg, Grischa; Leinen, Stefan; Becker, Matthias; Feiler, Johannes; Hoffmann, Simon; Diermeyer, Frank; Lampe, Bastian; Beemelmanns, Till; van Kempen, Raphael; Woopen, Timo; Eckstein, Lutz; Voget, Nicolai; Moormann, Dieter; Jatzkowski, Inga; Stolte, Torben; Maurer, Markus; Graf, Jürgen; Hinüber, Edgar L. von; Siepenkötter, Norbert: Automation of the UNICARagil vehicles, in: 29. Aachen Colloquium Sustainable Mobility, Institute for Automotive Engineering RWTH Aachen University; Institute for Combustion Engines RWTH Aachen University, Aachen, October 2020

Carvalho, E. A. de **et al.: FRAM and Complex Systems (2021)**
Carvalho, Elaine A. de; Gomes, José O.; Jatobá, Alessandro; Da Silva, Mônica F.; Carvalho, Paulo V. R. de: Employing resilience engineering in eliciting software requirements for complex systems: experiments with the functional resonance analysis method (FRAM), in: Cognition, Technology & Work (1), Jahrgang 23, S. 65–83, 2021

Dahlstedt, Å. G.; Persson, A.: Requirements Interdependencies and Interdependency Types (2005)
Dahlstedt, Åsa G.; Persson, Anne: Requirements Interdependencies: State of the Art and Future Challenges, in: Aurum, Aybüke; Wohlin, Claes (Hrsg.): Engineering and Managing Software Requirements, Springer-Verlag, Berlin/Heidelberg, 2005

Dajsuren, Y. et al.: Correspondence Rules for Automotive Architecture (2014)
Dajsuren, Yanja; Gerpheide, Christine M.; Serebrenik, Alexander; Wijs, Anton; Vasilescu, Bogdan; van den Brand, Mark G.: Formalizing correspondence rules for automotive architecture views, in: Seinturier, Lionel; Bures, Tomas; McGregor, John D. (Hrsg.): Proceedings of the 10th international ACM Sigsoft conference on Quality of software architectures - QoSA '14, Marcq-en-Bareul, France, ACM Press, New York, New York, USA, 2014

Eckstein, L. et al.: 13. Workshop Fahrerassistenz und Automatisiertes Fahren, FAS 2020 (2020)
Eckstein, Lutz; Bengler, Klaus; Dietmayer, Klais, et al. (Hrsg.) 13. Workshop Fahrerassistenz und Automatisiertes Fahren, FAS 2020, Uni-DAS 13. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2020, Uni-DAS e.V, Darmstadt, 2020

Ferreira, P. N.; Cañas, J. J.: FRAM and Impacts of Automation (2019)
Ferreira, Pedro N. P.; Cañas, José J.: Assessing operational impacts of automation using functional resonance analysis method, in: Cognition, Technology & Work (3), Jahrgang 21, S. 535–552, 2019

Filippini, R.; Silva, A.: Modeling Framework for Networked Systems-of-Systems (2014)
Filippini, Roberto; Silva, Andrés: A modeling framework for the resilience analysis of networked systems-of-systems based on functional dependencies, in: Reliability Engineering & System Safety, Jahrgang 125, S. 82–91, 2014

Gottschalg, G. et al.: Concept for Sensor Fusion Algorithms (2020)
Gottschalg, Grischa; Becker, Matthias; Leinen, Stefan: Integrity Concept for Sensor Fusion Algorithms used in a Prototype Vehicle for Automated Driving, in: 2020 European Navigation Conference (ENC), 2020

Gottschalg, G. et al.: Inkonsistente Lokalisierungsdaten in modularen Systemen (2020)
Gottschalg, Grischa; Homolla, Tobias; Hermann Winner: Verfahren zur Korrektur von inkonsistenten Lokalisierungsdaten in modularen technischen Systemen, in: Eckstein, Lutz et al. (Hrsg.): 13. Workshop Fahrerassistenz und Automatisiertes Fahren, FAS 2020, Hotel Gut Moierhof, Walting im Altmühltal, Uni-DAS 13. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2020, Uni-DAS e.V, Darmstadt, 2020

Gottschalg, G. et al.: Data Fusion of Redundant Fusion Filters (2021)
Gottschalg, Grischa; Becker, Matthias; Leinen, Stefan: Integrity Based Data Fusion of Redundant Fusion Filters for Vehicle Dynamic State Estimation in Automated Driving, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, IEEE, 2021

Grabbe, N.: FRAM and Automated Driving (2019)
Grabbe, Niklas: Risk Assessment of Automated Driving – A Systemic Approach, in: Uni-DAS Doktorandenworkshop, Garching, 29.03.2019

Grabbe, N. et al.: Safety of automated driving: FRAM (2020)
Grabbe, Niklas; Kellnberger, Anna; Aydin, Beyza; Bengler, Klaus: Safety of automated driving: The need for a systems approach and application of the Functional Resonance Analysis Method, in: Safety Science, Jahrgang 126, S. 104665, 2020

Höhn, R.; Höppner, S.: Das V-Modell XT (2008)
Höhn, Reinhard; Höppner, Stephan: Das V-Modell XT, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008

Hollnagel, E. et al.: The Functional Resonance Analysis Method (2014)
Hollnagel, Erik; Hounsgaard, Jeanette; Colligan, Lacey: FRAM - the Functional Resonance Analysis Method, 1. Auflage, Centre for Quality, Region of Southern Denmark, 2014

Hosseini, S.; Ivanov, D.: Bayesian networks for risk analysis (2020)
Hosseini, Seyedmohsen; Ivanov, Dmitry: Bayesian networks for supply chain risk, resilience and ripple effect analysis: A literature review, in: Expert Systems with Applications, Jahrgang 161, S. 113649, 2020

Hu, J. et al.: Dynamic Resilience of LNG System (2021)
Hu, Jinqiu; Khan, Faisal; Zhang, Laibin: Dynamic resilience assessment of the Marine LNG offloading system, in: Reliability Engineering & System Safety, Jahrgang 208, S. 107368, 2021

Institute of Electrical and Electronics Engineers: IEEE 829: Standard for Tests (2008)
Institute of Electrical and Electronics Engineers: IEEE 829: Standard for Software and System Test, IEEE, 2008

International Electrotechnical Commission: IEC 61508 (2010)
International Electrotechnical Commission: Functional Safety of Electrical/Electronic/ Programmable Electronic Safety-Related Systems (E/E/ PES), 2010

International Organization for Standardization: ISO 26262 (2018)
International Organization for Standardization: Road vehicles – Functional safety, 2018

IPG Automotive GmbH: CarMaker
IPG Automotive GmbH: CarMaker; https://ipg-automotive.com/de/produkte-loesungen/software/carmaker/, Zugriff 26.04.2022

Kammouh, O. et al.: Use of Bayesian networks (2020)
Kammouh, Omar; Gardoni, Paolo; Cimellaro, Gian P.: Probabilistic framework to evaluate the resilience of engineering systems using Bayesian and dynamic Bayesian networks, in: Reliability Engineering & System Safety, Jahrgang 198, S. 106813, 2020

Kamran, D.; Wang, L.: UNICARagil - Verhaltens- und Trajektorienplanung (2020)
Kamran, Danial; Wang, Lingguang: UNICARagil - Digitales Halbzeitevent - Verhaltens- und Trajektorienplanung; https://www.unicaragil.de/images/medien/news/NewsletterHZE_07-2020_DE.pdf, 2020, Zugriff 03.04.2022

Klamann, B.; Lippert, M.: UNICARagil - Modulare Absicherung (2020)
Klamann, Björn; Lippert, Moritz: UNICARagil - Digitales Halbzeitevent - Modulare Absicherung; https://www.unicaragil.de/images/medien/news/NewsletterHZE_07-2020_DE.pdf, 2020, Zugriff 03.04.2022

Lippert, M. et al.: Bestehens- & Versagenskriterien für Tests (2020)
Lippert, Moritz; Klamann, Björn; Amersbach, Christian; Winner, Hermann: Definition von Bestehens-/Versagenskriterien für das partikuläre Testen von automatisierten Fahrfunktionen, in: Automatisiertes Fahren, 2020

Menzel, T. et al.: Scenarios for Test and Validation (2018)
Menzel, Till; Bagschik, Gerrit; Maurer, Markus: Scenarios for Development, Test and Validation of Automated Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, IEEE, 2018

Pfeffer, R.: Diss., Funktionale Absicherung hochautomatisierter Fahrfunktionen (2020)
Pfeffer, Raphael: Szenariobasierte simulationsgestützte funktionale Absicherung hochautomatisierter Fahrfunktionen durch Nutzung von Realdaten, Dissertation Karlsruher Institut für Technologie, Karlsruher Institut für Technologie (KIT), 2020

Pohl, K.: Process-centered requirements engineering (1996)
Pohl, Klaus: Process-centered requirements engineering, Advanced software development series, Jahrgang 5, Research Studies Press, Taunton, Somerset, England, 1996

Rosenberger, P.; Linnhoff, C.: PerCollECT
Rosenberger, Philipp; Linnhoff, Clemens: PerCollECT; https://github.com/PerCollECT, Zugriff 16.04.2022

RWTH Aachen: UNICARagil (2018)
RWTH Aachen: UNICARagil; https://www.unicaragil.de/de/, 2018, Zugriff 09.04.2022

SAE International: SAE J3016: Taxonomy and Definitions (2021)
SAE International: SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles; https://www.sae.org/standards/content/j3016_202104/, 2021, Zugriff 06.04.2022

Salay, R. et al.: ISO 26262 and Automotive Software (2018)
Salay, Rick; Queiroz, Rodrigo; Czarnecki, Krzysztof: An Analysis of ISO 26262: Machine Learning and Safety in Automotive Software, in: SAE Technical Paper Series, SAE Technical Paper Series, SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2018

Salehi, V. et al.: Modeling Complex Socio-Technical Systems (2021)
Salehi, Vahid; Veitch, Brian; Smith, Douglas: Modeling complex socio-technical systems using the FRAM: A literature review, in: Human Factors and Ergonomics in Manufacturing & Service Industries (1), Jahrgang 31, S. 118–142, 2021

Schuldt, F. et al.: Effiziente systematische szenarienbasierte Testgenerierung (2013)
Schuldt, Fabian; Saust, Falko; Lichte, Bernd; Maurer, Markus; Scholz, Stephan: Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen; https://publikationsserver.tu-braunschweig.de/receive/dbbs_mods_00052570, 2013, Zugriff 14.04.2022

Shuttleworth, J.: SAE J3016 (2019)
Shuttleworth, Jennifer: SAE Standards News: J3016 automated-driving graphic update; https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic, 2019, Zugriff 06.04.2022

Silva, A.; Filippini, R.: Infrastructure Resilience-oriented Modelling Language (2011)
Silva, Andres; Filippini, Roberto: Infrastructure (Resilience-oriented) Modelling Language (I®ML), EUR (Luxembourg), Jahrgang 24727, Publications Office, Luxembourg, 2011

Steimle, M. et al.: Terminologie für den szenarienbasierten Testansatz (2018)
Steimle, Markus; Bagschik, Gerrit; Menzel, Till; Wendler, Jan T.; Maurer, Markus: Ein Beitrag zur

Terminologie für den szenarienbasierten Testansatz automatisierter Fahrfunktionen, in: ITS mobility e.V (Hrsg.): AAET Automatisiertes und Vernetztes Fahren, ITS mobility e.V, 2018

Strode, D. E.; Huff, S. L.: Taxonomy of Dependencies (2012)
Strode, Diane E.; Huff, Sid L.: A Taxonomy of Dependencies in Agile Software Development, in: ACIS 2012 Proceedings, 2012

Ulbrich, S. et al.: Begriffe für das automatisierte Fahren (2015)
Ulbrich, Simon; Menzel, Till; Reschka, Andreas; Schuldt, Fabian; Maurer, Markus: Definition der Begriffe Szene, Situation und Szenario für das automatisierte Fahren, in: Stiller, Christoph (Hrsg.): 10. Workshop Fahrerassistenzsysteme, FAS 2015, Hotel Gut Moierhof, Walting im Altmühltal, Uni-DAS 10. Workshop Fahrerassistenz und automatisiertes Fahren. FAS 2015, Uni-DAS e.V, Darmstadt, 2015

UNICARagil: UNICARagil news - 1. Ausgabe (2019)
UNICARagil: UNICARagil news - 1. Ausgabe; https://www.unicaragil.de/images/medien/news/Newsletter1_09-2019_DE.pdf, 2019, Zugriff 06.04.2022

UNICARagil: UNICARagil news - Halbzeitevent (2020)
UNICARagil: UNICARagil news - Halbzeitevent; https://www.unicaragil.de/images/medien/news/NewsletterHZE_07-2020_DE.pdf, 2020, Zugriff 04.04.2022

UNICARagil: UNICARagil news - 4. Ausgabe (2021)
UNICARagil: UNICARagil news - 4. Ausgabe; https://www.unicaragil.de/images/medien/news/Newsletter4_02-2021_DE.pdf, 2021, Zugriff 06.04.2022

UNICARagil Architektur Tool UNICARagil Architektur Tool; https://architektur.unicaragil.embedded.rwth-aachen.de/index.php?action=show_view&view_id=1, Zugriff 17.04.2022

Wachenfeld, W. et al.: Objective Identification of Scenarios (2016)
Wachenfeld, Walther; Junietz, Philipp; Winner, Hermann; Themann Philipp; Pütz, Andreas: Safety Assurance Based on an Objective Identification of Scenarios.; https://www.pegasusprojekt.de/files/tmpl/pdf/Automation%20Vehicle%20Symposium%202016%20BreakoutSession%20Folien.pdf, 2016, Zugriff 14.04.2022

Wachenfeld, W.; Winner, H.: Die Freigabe des autonomen Fahrens (2015)
Wachenfeld, Walther; Winner, Hermann: Die Freigabe des autonomen Fahrens, in: Maurer, Markus et al. (Hrsg.): Autonomes Fahren, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015

Wilhelm, U. et al.: Funktionale Sicherheit und ISO 26262 (2015)
Wilhelm, Ulf; Ebel, Susanne; Weitzel, Alexander: Funktionale Sicherheit und ISO 26262, in: Winner, Hermann et al. (Hrsg.): Handbuch Fahrerassistenzsysteme, ATZ/MTZ-Fachbuch, 3. Auflage, Springer Fachmedien Wiesbaden, Wiesbaden, 2015

Winner, H. et al.: Handbuch Fahrerassistenzsysteme (2015)
Winner, Hermann; Hakuli, Stephan; Lotz, Felix; Singer, Christina (Hrsg.) Handbuch Fahrerassistenzsysteme, ATZ/MTZ-Fachbuch, 3. Auflage, Springer Fachmedien Wiesbaden, Wiesbaden, 2015

Winner, H.: Quo vadis, FAS? (2015)
Winner, Hermann: Quo vadis, FAS?, in: Winner, Hermann et al. (Hrsg.): Handbuch Fahrerassistenzsysteme, ATZ/MTZ-Fachbuch, 3. Auflage, Springer Fachmedien Wiesbaden, Wiesbaden, 2015

Woopen, T. et al.: UNICARagil - Disruptive modulare Architektur (2018)
Woopen, Timo; Eckstein, Lutz; Kowalewski, Stefan; Moormann, Dieter; Maurer, Markus; Ernst, Rolf; Winner, Hermann; Katzenbeisser, Stefan; Becker, Matthias; Stiller, Christoph; Furmans, Kai; Bengler, Klaus; Lienkamp, Markus; Reuss, Hans-Christian; Dietmayer, Klaus; Lategahn, Henning; Siepenkötter, Norbert; Elbs, Martin; v. Hinüber, Edgar; Dupuis, Marius; Hecker, Christian: UNICARagil - Disruptive modulare Architektur für agile, automatisierte Fahrzeugkonzepte, in: 27. Aachen Colloquium Automobile and Engine Technology, 1. Auflage, Institute for Automotive Engineering RWTH Aachen; Institute for Combustion Engines RWTH Aachen, Aachen, October 2018

yWorks GmbH: yEd
yWorks GmbH: yEd; https://www.yworks.com/products/yed, Zugriff 26.04.2022

Zhang, H. et al.: Dependencies in Software Requirements (2014)
Zhang, He; Li, Juan; Zhu, Liming; Jeffery, Ross; Liu, Yan; Wang, Qing; Li, Mingshu: Investigating dependencies in software requirements for change propagation analysis, in: Information and Software Technology (1), Jahrgang 56, S. 40–53, 2014