

Moritz Wäschle

Ein Beitrag zur durchgängigen und kontinuierlichen Validierung für die Funktionale Sicherheit von Vehicle-to-Infrastructure als System of Systems durch MBSE-Unterstützung auf Basis des IPEK-XiL-Ansatzes

A contribution to the end-to-end and continuous validation for the Functional Safety of Vehicle-to-Infrastructure as a System of Systems through MBSE support based on the IPEK-XiL approach

Band 176

Systeme ■ Methoden ■ Prozesse

Univ.-Prof. Dr.-Ing. Dr. h.c. A. Albers
Univ.-Prof. Dr.-Ing. S. Matthiesen
(Hrsg.)

Moritz Wäschle

Ein Beitrag zur durchgängigen und kontinuierlichen Validierung für die Funktionale Sicherheit von Vehicle-to-Infrastructure als System of Systems durch MBSE-Unterstützung auf Basis des IPEK-XiL-Ansatzes

A contribution to the end-to-end and continuous validation for the Functional Safety of Vehicle-to-Infrastructure as a System of Systems through MBSE support based on the IPEK-XiL approach.

Band 176

Herausgeber Univ.-Prof. Dr.-Ing. Dr. h.c. A. Albers
Univ.-Prof. Dr.-Ing. S. Matthiesen

Copyright IPEK - Institut für Produktentwicklung, 2024
Karlsruher Institut für Technologie (KIT)
Die Forschungsuniversität in der Helmholtz-Gemeinschaft
Alle Rechte vorbehalten

Druck Designmedia Printhouse e.K., Stuttgart

ISSN 1615-8113

**Ein Beitrag zur durchgängigen und
kontinuierlichen Validierung für die Funktionale
Sicherheit von Vehicle-to-Infrastructure als
System of Systems durch MBSE-Unterstützung
auf Basis des IPEK-XiL-Ansatzes**

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Moritz Wäschele

Tag der mündlichen Prüfung: 31.05.2024
Hauptreferent: Univ.-Prof. Dr.-Ing. Dr. h.c. Albert Albers
Korreferent: Prof. Dr.-Ing. Lydia Kaiser

Vorwort der Herausgeber

Wissen ist einer der entscheidenden Faktoren in den Volkswirtschaften unserer Zeit. Der Unternehmenserfolg wird mehr denn je davon abhängen, wie schnell ein Unternehmen neues Wissen aufnehmen, zugänglich machen und verwerten kann. Die Aufgabe eines Universitätsinstitutes ist es, hier einen wesentlichen Beitrag zu leisten. In den Forschungsarbeiten wird ständig Wissen generiert. Dieses kann aber nur wirksam und für die Gemeinschaft nutzbar werden, wenn es in geeigneter Form kommuniziert wird. Diese Schriftenreihe dient seit mehr als 20 Jahren als eine Plattform zum Transfer und macht damit das Wissenspotenzial aus aktuellen Forschungsarbeiten am IPEK - Institut für Produktentwicklung Karlsruhe* am Karlsruher Institut für Technologie (KIT) verfügbar. Die Forschung des IPEK ist dabei strukturiert in die Kategorien Systeme, Methoden und Prozesse, um so der Komplexität heutiger Produktentwicklung ganzheitlich gerecht zu werden. Erst die Verknüpfung dieser drei Kategorien ermöglicht die Synthese innovativer Systeme durch Nutzung neuester Methoden und Prozesse. Gleichzeitig werden durch die Systemsynthese die erforschten neuen Methoden und Prozesse validiert und deren Mehrwert für die Praxis abgesichert. Dieses Forschungskonzept prägt nicht nur das IPEK-Leitbild, sondern auch den Charakter dieser Schriftenreihe, da immer alle drei Kategorien und deren Wechselwirkungen berücksichtigt werden. Jeder Band setzt hier individuelle Schwerpunkte und adressiert dabei folgende Forschungsgebiete des IPEK:

- das Entwicklungs- und Innovationsmanagement,
- die Entwicklungs- und Konstruktionsmethodik,
- der Leichtbau von der Ebene des ganzen Systems bis hinunter zur Optimierung des Bauteils,
- die Validierung technischer Systeme auch unter Berücksichtigung der NVH Aspekte (Noise, Vibration, Harshness) mit dem Fokus auf Schwingungen und Akustik an Komponenten und in den Gesamtsystemen sowie deren subjektiver Beurteilung durch den Menschen,
- die Antriebssystemtechnik mit den Schwerpunkten komplette Antriebslösungen für Fahrzeuge und Maschinen,
- das Design, die Tribologie und Erprobung von Kupplungen und Bremsen sowie
- die Gerätetechnik mit dem Schwerpunkt auf Power-Tools.

Die Forschungsberichte stellen Ergebnisse unserer Forschung sowohl anderen Wissenschaftlern als auch den Unternehmen zu Verfügung, um damit die Produktentwicklung in allen ihren Facetten mit innovativen Impulsen zu optimieren.

Albert Albers und Sven Matthiesen

* Eh.: Institut für Maschinenkonstruktionslehre und Kraftfahrzeugbau, Universität Karlsruhe (TH)

Vorwort zu Band 176

Mobilität ist ein Treiber der Entwicklung der Menschheit. Bereits in frühester Vorzeit wurde durch die Mobilität ein Austausch von Waren, Erfahrungen, Gedanken und Wissen möglich. Dieses Grundbedürfnis an Mobilität wurde über die Jahrhunderte durch unterschiedliche technische Lösungen immer weiter optimiert und verbessert. Über viele Jahrhunderte waren die vorwiegenden Mobilitätssysteme das Pferd, beziehungsweise das Pferdewerk, und das Schiff. Mit dem Beginn der Industrialisierung entstanden neue technische Lösungen, wie die Eisenbahn oder das Automobil. Insbesondere das Automobil hat zu einer radikalen Veränderung in der Gesellschaft geführt. Die hohe Verfügbarkeit individueller Mobilität hat völlig neue Möglichkeiten für das Individuum, aber auch für die Gesellschaft realisiert. Andererseits hat die starke Verbreitung dieser Mobilitätslösungen durch den Verbrauch an Landschaft für das Verkehrssystem, aber auch durch die Nutzung des Verbrennungsmotors und der fossilen Kraftstoffe zu erheblichen Umweltbeanspruchungen geführt. Ziel muss es daher sein, mit ganz neuartigen technischen Lösungen individuelle Mobilität komfortabel, sicher, nachhaltig, aber auch bezahlbar für die Menschen zur Verfügung zu stellen. Dabei ist zu berücksichtigen, dass heute erst im Bereich der sogenannten *Ersten Welt* Lösungen für die individuelle Mobilität breit verfügbar sind, der Bedarf aber auch in den anderen Ländern vorhanden ist und somit auch hier Lösungen geschaffen werden müssen. Um dies tun zu können, sind neue Lösungen für die Fahrzeuge notwendig, sogenannte CPS – Cyber-Physische-Systeme. Cyber-Physische-Systeme zeichnen sich dabei durch eine kontinuierliche Vernetzung der Systeme untereinander, aber auch mit der Infrastruktur, aus. Ein Beispiel ist die technischen Lösungen für hochautomatisierte Fahrzeuge, die mittlerweile eine intensive Kommunikation auch mit der Infrastruktur nutzen. Solche Systeme sind äußerst komplex. Die Herausforderung ist nicht nur ihre Synthese, sondern insbesondere auch die Validierung, das heißt, die Überprüfung, ob die gewünschten Nutzenpotenziale für den Kunden und den Anwender auch unter den Kriterien Komfort, Sicherheit, Funktion, Nutzbarkeit unter allen Randbedingungen gegeben sind. Diese Validierungsaufgabe ist eine der herausforderndsten der modernen Produktentwicklung. Hier einen Beitrag zu leisten, um zu einer kontinuierlichen Validierung für die funktionale Sicherheit von *Vehicle-to-Infrastructure* zu kommen, hat sich Herr Dr. Ing. Moritz Wäsche in seiner Forschung zum Ziel gesetzt. Er erarbeitet auf der Basis und als Teil der KaSPro – Karlsruher Schule für Produktentwicklung eine durchgängige, kontinuierliche Sprache und Methode zur Validierung der Funktionalen Sicherheit von *Vehicle-to-Infrastructure* (V2I) Systemen als SoS durch MBSE-Unterstützung unter Einsatz des IPEK-XiL-Ansatzes. Die Arbeit leistet einen wesentlichen Beitrag zum weiteren Ausbau des ASE -Advanced System

Engineering und ist gleichzeitig hochrelevant für die praktische Entwicklung von Elementen und Funktionen im Kontext des System of System „Mobilitätssystem“.

Juni, 2024

Albert Albers

Kurzfassung

Die Mobilität ist ein Grundbedürfnis des Menschen, wobei Sicherheit, Verkehrseffizienz und Komfort in der Mobilität zu gewährleisten sind. Diese Bedarfe können durch intelligente Transportsysteme erfüllt werden, die durch Vernetzung von Systemen wie Fahrzeug und Infrastruktur in einem System of Systems (SoS) realisiert werden. Bei der Entwicklung eines System of Systems stellt dessen Validierung aufgrund der Generierung von Wissen und der hohen Entwicklungskosten eine zentrale Herausforderung dar. Die vorliegende Arbeit zeigt auf, wie die Validierung des spezifischen SoS Vehicle-to-Infrastructure (V2I) durch Model-Based Systems Engineering (MBSE) unterstützt werden kann. Hierzu wird eine Vorgehensweise entwickelt, um das Zielsystem für die Validierung zu beschreiben und zentrale Anforderungen an die Validierung von V2I abzuleiten. Hierauf aufbauend werden Sprache und Methode für die Validierung von V2I entwickelt. Es zeigt sich, dass die modellbasierte Systementwicklung eine durchgängige, kontinuierliche Validierung unterstützt. Dies wird für den wesentlichen Bedarf der Funktionalen Sicherheit für den repräsentativen Anwendungsfall einer optimalen Geschwindigkeit zum Passieren einer Verkehrskreuzung gezeigt.

Die Arbeit zeigt drei wesentliche Ergebnisse auf. Erstens eine systematische Vorgehensweise zur verständlichen Zielsystembeschreibung von V2I für Produktentwickler. Zweitens eine Sprache zur Unterstützung der durchgängigen und kontinuierlichen Validierung, die drei Ausprägungsgrade von Modellen enthält. Drittens eine Methode zur Erfüllung des Bedarfs der Funktionalen Sicherheit für V2I. Diese Methode setzt die für die Validierung von V2I spezifische Sprache ein und bestätigt deren Eignung durch Anwendungsfälle in einer virtuell und virtuell-physisch ausgeprägten Validierungsumgebung.

Abstract

Mobility is a basic human need, and safety, traffic efficiency, and comfort in mobility must be ensured. These needs can be met by intelligent transportation systems, which are realized by networking systems such as vehicles and infrastructure in a System of Systems (SoS). During the development of a System of Systems, its validation is a key challenge due to the generation of knowledge and the high development costs. This thesis shows how the validation of the specific SoS Vehicle-to-Infrastructure (V2I) can be supported by Model-Based Systems Engineering (MBSE). For this purpose, an approach is developed to describe the system of objective for validation and to derive requirements for V2I validation. Based on this, a language and method for the validation of V2I are developed. It is shown that MBSE supports end-to-end, continuous validation. This is shown by the essential need of functional safety for the representative use case of an optimal speed for passing a traffic intersection.

The publication demonstrates three main results. First, a systematic approach to an understandable description of the system of objective of V2I for product developers. Second, a language to support end-to-end and continuous validation that includes three levels of model expression. Third, a method for meeting functional safety needs for V2I. This method leverages the language specific to V2I validation and its suitability is validated by use cases in a virtual and virtual-physical distinct validation environment.

Danksagung

Diese Arbeit ist über die letzten fünf Jahre entstanden und ich möchte mehreren Personen meinen Dank aussprechen.

Ein besonderer Dank gilt meinem Doktorvater Univ.-Prof. Dr.-Ing. Dr. h.c. Albert Albers, für die Betreuung der Arbeit und die zahlreichen Diskussionen zu den Themen rund um System of Systems.

Zudem bedanke ich mich bei Frau Univ.-Prof. Dr.-Ing. Lydia Kaiser für die Zusammenarbeit und die Zweitprüfung meiner Arbeit.

Meine Familie hat mich über die letzten Jahrzehnte mit Rat und Tat unterstützt. Ich bedanke mich für diese Unterstützung bei Silvia, Marielle und Michael Wäschle.

Als die wichtigste Wegbegleiterin, die von allen Steinen im Weg als erste erfuhr, möchte ich mich bei meiner Freundin Laura Louise Eberle für ihre Unterstützung in den letzten Jahren bedanken.

Bei Herrn Dr.-Ing. Simon Rapp bedanke ich mich für die Diskussionen und die Betreuung meiner Arbeit.

Bei Frau Dr. phil. Anne Petry möchte ich mich für den intensiven Austausch am Ende der Zeit bedanken.

Mit mehreren Kollegen am IPEK habe ich in den letzten Jahren intensiv diskutiert und in Projekten und Veröffentlichungen zusammengearbeitet. Insbesondere gilt mein Dank Constantin Mandel, Alex Martin und Stefan Schwarz.

„Im Großen müssen wir das Kleine kennenlernen“.

Georg Christoph Lichtenberg, 1742-1799

Inhaltsverzeichnis

Kurzfassung	i
Abstract.....	iii
Abbildungsverzeichnis	xiii
Tabellenverzeichnis	xxi
Abkürzungsverzeichnis	xxiii
Formelzeichen	xxix
1 Einleitung.....	1
2 Stand der Forschung und Technik	3
2.1 Systemtheorie und Systems Engineering.....	3
2.1.1 Systemtheorie	5
2.1.2 System of Systems	7
2.1.3 Model-Based Systems Engineering	14
2.1.4 Advanced Systems Engineering	20
2.1.5 Zwischenfazit	23
2.2 Validierung in der Produktentwicklung	24
2.2.1 Grundlegende Produktentwicklungsmodelle der KaSPro	24
2.2.2 Validierung in der Produktentwicklung	27
2.2.3 Zwischenfazit	33
2.3 Beschreibung und Validierung von Vehicle-to-Infrastructure.....	34
2.3.1 Systembeschreibung von V2I.....	36
2.3.2 Validierungsansätze für das hochautomatisierte Fahren	43
2.3.3 Zwischenfazit	51
2.4 Diskussion und Fazit	52
3 Bedarf und Zielsetzung.....	53
3.1 Forschungsbedarf	53
3.2 Grundannahmen und Forschungsziel	55
3.3 Grundthese und Forschungsfragen.....	57
3.3.1 Grundthese.....	57
3.3.2 Forschungsfragen.....	57
4 Methodisches Forschungsvorgehen	59
4.1 Vorgehen angelehnt an die Design Research Methodology von Blessing und Chakrabarti (2009)	59
4.2 Forschungsumgebung	63
4.3 Empirische Methoden	65
5 Beschreibung des generischen Zielsystems für die Validierung von V2I 67	

5.1	Empirische Studie zur Bedarfsermittlung.....	68
5.1.1	Systematisches Literaturreview.....	69
5.1.2	Befragungen in Live-Labs.....	73
5.1.3	Expertenbefragung.....	77
5.1.4	Zusammenfassung.....	80
5.2	Vorgehen zur Zielsystembeschreibung von V2I	80
5.2.1	Profilfindung für V2I als SoS.....	81
5.2.2	Ermittlung von SoS-Charakteristiken für V2I	85
5.2.3	Implikationen auf die Validierung von V2I	89
5.2.4	Zusammenfassung.....	91
5.3	Diskussion & Zwischenfazit	92
5.3.1	Limitierungen.....	92
5.3.2	Zwischenfazit zur Beantwortung der Forschungsfrage 1	93
5.3.3	Diskussion des Effektes Verständlichkeit	94

6 Modellierungssprache und -methode MEFS4SoS in MBSE zur Unterstützung der Validierung von V2I..... 97

6.1	Beiträge für die Modellierungssprache	99
6.1.1	Elementtypen im Problemraum des generischen Meta-Systemmodells	100
6.1.2	Elementtypen von Anforderungen des generischen Meta-Systemmodells	102
6.1.3	Elementtypen von Funktionen des generischen Meta-Systemmodells	106
6.1.4	Elementtypen von logischen Elementen des generischen Meta-Systemmodells	108
6.1.5	Referenzsystem im generischen Meta-Systemmodell	110
6.1.6	Meta-Systemmodell für das hochautomatisierte Fahren.....	112
6.1.7	Zusammenfassung.....	115
6.2	Konkretisierung der Modellierungssprache: Bereichsspezifisches Referenz-Systemmodell für das hochautomatisierte Fahren	116
6.2.1	Elemente des Problemraums (PS) im Referenz-Systemmodell.....	117
6.2.2	Elemente für Anforderungen im Referenz-Systemmodell	118
6.2.3	Elemente für Funktionen im Referenz-Systemmodell	118
6.2.4	Logische Elemente im Referenz-Systemmodell	120
6.2.5	Elemente des Referenzsystems des Referenz-Systemmodells.....	120
6.2.6	Zusammenfassung.....	123
6.3	MEFS4SoS – Methode zur Erreichung der Funktionalen Sicherheit für die Validierung von SoS.....	123
6.3.1	Systembeschreibung.....	124
6.3.2	Gefahrenidentifikation & Risikoevaluation	124
6.3.3	Funktionale Modifikation und Spezifikationen.....	126

6.3.4	Validierung.....	127
6.3.5	Zusammenfassung.....	130
6.4	Diskussion & Zwischenfazit.....	130
6.4.1	Limitierungen.....	130
6.4.2	Zwischenfazit zur Beantwortung der Forschungsfrage 2.....	131
6.4.3	Diskussion der Effekte und Erfolgsfaktoren.....	132
7	MBSE-unterstützte Validierung von V2I auf Basis des IPEK-XiL-Ansatzes	135
7.1	Systembeschreibung.....	138
7.2	Gefahrenidentifikation und Risikoevaluation.....	141
7.3	Funktionale Modifikation und Spezifikationen.....	145
7.4	Validierung – GLOSA virtuell.....	146
7.4.1	Beschreibung des Validierungssystems.....	147
7.4.2	Erstellung von konkreten Szenarien.....	150
7.4.3	Durchführung der Validierung.....	151
7.4.4	Zusammenfassung und Nutzung der Testergebnisse.....	157
7.5	Validierung – GLOSA virtuell-physisch.....	159
7.5.1	Beschreibung Validierungssystem.....	159
7.5.2	Erstellung von konkreten Szenarien.....	164
7.5.3	Durchführung der Validierung.....	165
7.5.4	Zusammenfassung und Nutzung der Testergebnisse.....	170
7.6	Diskussion & Zwischenfazit.....	170
7.6.1	Limitierungen.....	170
7.6.2	Zwischenfazit zur Beantwortung der Forschungsfrage 3.....	174
7.6.3	Diskussion der Effekte und Erfolgsfaktoren sowie der Anforderungen aus Kapitel 5.2.4.....	175
8	Zusammenfassung, Gesamtfazit und Ausblick	179
8.1	Zusammenfassung.....	179
8.2	Gesamtfazit.....	181
8.2.1	Erkenntnisse.....	182
8.2.2	Beitrag in der KaSPro – Karlsruher Schule für Produktentwicklung 184	
8.3	Ausblick.....	185
8.3.1	Zielsystembeschreibung von V2I.....	185
8.3.2	Sprache und Methode für V2I.....	186
8.3.3	Validierung von V2I.....	187
	Literaturverzeichnis.....	I
	Publikationsliste.....	XXXIX
	Glossar.....	XLIII

Anhang	XLV
A: Übersicht der Aktivitäten in ISO 26262 & ISO/PAS 21448	XLV
B: Übersicht der eingesetzten Koppelsysteme	XLVI
C: Fragebogen in den Live-Labs.....	XLVII
D: Erweiterung der Validierungsumgebung	XLIX

Abbildungsverzeichnis

Abbildung 1.1: Aufbau der Arbeit.....	2
Abbildung 2.1: Trend zu steigender Anzahl an Funktionen, Teilsystemen und Interaktionen (angepasst von Friedenthal et al., 2021) .	3
Abbildung 2.2: Drei Konzepte der Systemtheorie (Ropohl, 1975, 2009).	5
Abbildung 2.3: Abstraktionsgrade der Systemmodellierung (Albers, Matthiesen et al., 2015; Bursac, 2016; Muschik, 2011, S. 187).....	6
Abbildung 2.4: Die vier Typen von System of Systems (Dahmann & Baldwin, 2008; Dahmann, 2014; Lane & Epstein, 2013).	9
Abbildung 2.5: Vier Standards im Rahmen der Systementwicklung klassifiziert nach der Perspektive (angelehnt an BS ISO/IEC/IEEE 21841:2019).	10
Abbildung 2.6: Herausforderungen in der Entwicklung von SoS (Albers, Peglow et al., 2018).....	10
Abbildung 2.7: Herausforderungen für die Validierung von SoS (Albers, Peglow et al., 2018; Dahmann, 2014; Friedenthal et al., 2021; Nielsen et al., 2015).	11
Abbildung 2.8: Auszug aus der Ontologie für SoS (Holt & Perry, 2013a).....	13
Abbildung 2.9: Die drei Säulen des MBSE – Sprache, Methode und Tool (angelehnt an Delligatti, 2014).	14
Abbildung 2.10: Diagrammtypen in SysML (Object Management Group, 2019)...	15
Abbildung 2.11: Wesentliche Aktivitäten zur Systemmodellierung in der Methode SYSMOD dargestellt in einem Aktivitätendiagramm (Weilkiens, 2015).	17
Abbildung 2.12: Wesentliche Aktivitäten in der Methode FAS4M aufgeteilt nach Sichten (Moeser, 2016).....	18
Abbildung 2.13: Viewpoints und Layer im SPES Framework (Pohl et al., 2012)...	19
Abbildung 2.14: IPEK-MBSE-Framework (Mandel, Martin & Albers, 2022).	20
Abbildung 2.15: Die drei Handlungsfelder des ASE (Dumitrescu et al., 2021).	21
Abbildung 2.16: Venn-Diagramm mit den drei Systembegriffen SoS, IoT und CPS (übersetzt nach Henshaw, 2016).	22
Abbildung 2.17: Das erweiterte ZHO-Modell (Albers et al., 2011).....	25
Abbildung 2.18: iPeM – Integriertes Produktentstehungsmodell mit der Aktivität Validieren und Verifizieren und dem Layer Validierungssystem (Albers et al., 2010a; Albers, Reiß et al., 2016).....	26

Abbildung 2.19: Exemplarische Produktgenerationen und exemplarisches Referenzsystem für V2I im Modell der PGE – Produktgenerationsentwicklung (Struktur angelehnt an Albers, Bursac & Wintergerst, 2015).....	27
Abbildung 2.20: Design und Validierung im Produktentstehungsprozess nach dem ZHO-Modell (Albers, Behrendt et al., 2016).	28
Abbildung 2.21: V-Modell für die Entwicklung mechatronischer Systeme und Cyber-Physical Systems (VDI/VDE 2206).....	29
Abbildung 2.22: IPEK-X-in-the-Loop-Ansatz zur Validierung des Systems mit dessen Umwelt und dessen vernetzten Systemen (Albers & Düser, 2008; Albers, Behrendt et al., 2016).....	30
Abbildung 2.23: Elemente und Zusammenhänge des IPEK-X-in-the-Loop-(XiL)-Framework (Albers & Düser, 2008; Albers, Fischer, Behrendt & Lieske, 2014; Albers, Behrendt et al., 2016).....	31
Abbildung 2.24: IPEK-XiL-Ansatz und die Beziehung zum Produktentwicklungsprozess (übersetzt nach Mandel et al., 2020).	32
Abbildung 2.25: Pull- und Push-Prinzip der Validierung am Beispiel der Hybridantriebsentwicklung (Albers, Matros et al., 2015).....	33
Abbildung 2.26: Übersicht der vier V2X Kommunikationsarten V2V, V2I, V2P, V2N (Hsu, Wang, Zhou & Shawkat, 2016).	34
Abbildung 2.27: Übersicht der 23 wesentlichen V2I UseCases (vgl. CAR 2 CAR Communication Consortium, 2021; vgl. Connected Motorcycle Consortium, 2020).....	36
Abbildung 2.28: Stakeholder im Kontext von V2I geclustert nach ihren Kernaufgaben (vgl. ASAM e.V., 2022b; vgl. ETSI TS 103 301; vgl. BS ISO/IEC/IEEE 21841:2019; vgl. McKinley, 2021; vgl. SAE J3016_202104; vgl. Villarreal-Vasquez, Bhargava & Angin, 2017).	37
Abbildung 2.29: SAE Levels für automatisiertes Fahren (vgl. Bartels, Eberle & Knapp, 2015; vgl. convex-project.de, 2020; vgl. McKinsey, 2021).	38
Abbildung 2.30: links: montierte RSU (Siemens Industry, Inc., 2018) rechts: prinzipieller Aufbau einer RSU (Glimm & Zadek, 2021).	40
Abbildung 2.31: Kommunikationsschichten Modelle nach den drei Standards der Organisationen OSI, ETSI und IEEE (Abboud, Omar & Zhuang, 2016).....	42

Abbildung 2.32: Szenarienlayer mit sechs Schichten zur systematischen Beschreibung von Szenarien (vgl. Bagschik, Menzel, Reschka & Maurer, 2017; Bock et al., 2018).....	44
Abbildung 2.33: Die vier SOTIF Quadranten (links) (ISO/PAS 21448:2019(E)) und ein Regelkreis in STAMP (rechts) (Leveson & Thomas, 2018).	47
Abbildung 2.34: STPA Methode und Verbindung mit STAMP (vgl. Leveson & Thomas, 2018).....	48
Abbildung 2.35: Graphische Darstellung einer Szene in MATLAB ADT.	49
Abbildung 2.36: Graphische Darstellung einer Szene in CarMaker.	50
Abbildung 3.1: Aus der Forschungslücke abgeleiteter Forschungsbedarf und den hieraus resultierenden notwendigen Effekten, welche zu realisieren sind.....	55
Abbildung 4.1: Übersicht des Forschungsvorgehens angelehnt an die Design Research Methodology (Blessing & Chakrabarti, 2009).....	60
Abbildung 4.2: Forschungsfragen und deren Bezug zu drei Stadien der DRM.	61
Abbildung 4.3: Ableitung der Erfolgsfaktoren: Impact Model mit Ergebnissen der Kapitel, deren Effekte und Erfolgsfaktoren.....	62
Abbildung 4.4: Zeitlicher Einsatz der Validierungsumgebungen in der Forschungsumgebung.	64
Abbildung 4.5: Übersicht des Akustikrollenprüfstands und der eingesetzten (Teil-)Systeme (vgl. Aufbau bezüglich Sensorstimulation in Albers, Lutz, Behrendt, Düser & Hettel, 2017; zu allgemeinen Fahrzeugprüfständen in Düser et al., 2011).....	65
Abbildung 5.1: Übersicht der Erhebungsmethoden mit Datenquelle und Stichprobenanzahl (N).	68
Abbildung 5.2: Identifizierte Themengebiete mit der Suchphrase „System of Systems“ , sortiert nach der Anzahl an Veröffentlichungen. Die Größe der Boxen steht im Verhältnis zu der Anzahl an Veröffentlichungen.....	71
Abbildung 5.3: Anzahl an Publikationen nach Jahren bei den in der Legende genannten Quellen Web of Science Core Collection (WoSCC) und Tabelle 5.1 analysiert mit MAXQDA (MAXQDA) mit den darauffolgenden Suchphrasen, wobei SoS und MBSE ausgeschrieben und mit den Varianten mit Bindestrich und mit dem OR-Operator verbunden genutzt wurden (z.B. SoS OR System-of-Systems OR „System of Systems“).	72
Abbildung 5.4: Antworten zur initialen Befragung für fünf Live-Labs.....	75
Abbildung 5.5: Antworten zur finalen Befragung für sechs Befragungen in Live-Labs.	76

Abbildung 5.6: Übersicht über die Anzahl der durchgeführten Expertenbefragungen mit den jeweiligen Fachgebieten der befragten Experten aus den vier Quellen, wobei die 129 Interviews in Dumitrescu et al. 2021 rechts nach Fachgebieten aufgeschlüsselt sind.	77
Abbildung 5.7: Übersicht wesentlicher Aussagen von Experten unterschiedlicher Fachgebiete.	78
Abbildung 5.8: Vorgehen zur Zielsystembeschreibung und Ableitung von Anforderungen für ein SoS.....	80
Abbildung 5.9: Drei Perspektiven auf das SoS V2I.	81
Abbildung 5.10: V2I-Produktprofil (angelehnt an Albers, Heimicke, Walter et al., 2018).....	82
Abbildung 5.11: Venn-Diagramm verschiedener Systembegriffe (vgl. Henshaw, 2016).....	84
Abbildung 5.12: Charakteristiken des SoS V2I dargestellt als Spinnendiagramm (angelehnt an Albers, Mandel et al., 2018).....	87
Abbildung 5.13: Ausgeprägte Charakteristiken von V2I aus den Befragungen in IP21/22, IP22/23 und IP22/23 (Kontrollgruppe).....	88
Abbildung 5.14: Drei Perspektiven auf das SoS V2I und deren Charakteristiken. .	88
Abbildung 5.15: Anforderungen an die Validierung von V2I unter Berücksichtigung von MBSE.	90
Abbildung 6.1: Abstraktionsgrade und Fokus der Arbeit in Rot umrandete Kästen mit Einordnung der nachfolgenden Kapitel (Albers, Matthiesen et al., 2015; Bursac, 2016; vgl. Muschik, 2011, S. 187).	98
Abbildung 6.2: Ordnungsrahmen mit fünf Perspektiven des generischen Meta-Systemmodells aufgeteilt in zwei Layer des iPeM.....	100
Abbildung 6.3: Auszug der verknüpften Elementtypen des Problemraums im generischen Meta-Systemmodell.	101
Abbildung 6.4: Sprachliche Elementtypen von Problemraum und Anforderungen des generischen Meta-Systemmodells.	102
Abbildung 6.5: Ergänzung der sprachlichen Elementtypen von Anforderungen zu den Perspektiven Funktion und logische Elemente des generischen Meta-Systemmodells (angelehnt an Wäschle, Timucin et al. (2021)).	103
Abbildung 6.6: Sprachliche Elementtypen von Problemraum, Anforderungen und Funktionen des generischen Meta-Systemmodells.	107

Abbildung 6.7: Sprachliche Elementtypen von Problemraum, Anforderungen, Funktionen und Logische Elemente des generischen Meta-Systemmodells.	108
Abbildung 6.8: Wirkkette der Signalübertragung von GLOSA im Systemmodell und im C&C ² -Modell (Bildquelle BMWi3: EnBW Energie Baden-Württemberg AG, 2022;; Verteilerkasten: Shenzhen Fama Intelligent Equipment Co., Ltd., 2022).	109
Abbildung 6.9: Prinzipdarstellung von möglichen Auswirkungen einer in Rot gekennzeichneten Prinzipvariation in einem Systemtupel auf weitere Systemtupel (Wäschle, Timucin et al., 2021).	112
Abbildung 6.10: Elementtypen des bereichsspezifischen Meta-Systemmodells für das hochautomatisierte Fahren.	113
Abbildung 6.11: Referenz-Systemmodell mit Elementen deduziert aus dem Meta-Systemmodell (angelehnt an Wäschle und Wolter et al. (2022)).	117
Abbildung 6.12: Ausschnitt des Aktivitätendiagramms für den GLOSA-Algorithmus in einem ausführbaren SysML-Diagramm (Klein, 2022).	119
Abbildung 6.13: Spinnendiagramm mit Charakterisierung von vier Validierungsumgebungen für V2I.	122
Abbildung 6.14: MEFS4SoS - Methode zur Erreichung der Funktionalen Sicherheit für die Validierung im Bereich hochautomatisiertes Fahren für SoS in Anlehnung an ISO 26262, ISO/PAS21448, OpenScenario.	124
Abbildung 6.15: Auszug eines modellierten Regelkreises für das automatisierte Fahren mittels Lenkaktor (Wäschle, Wolter et al., 2020) angelehnt an (Leveson & Thomas, 2018).	126
Abbildung 6.16: Wirkkette der exemplarischen Schadensursache „Reibung“ dargestellt in einer Safety & Reliability Analysis Map.	127
Abbildung 6.17: Generelles Vorgehen und Erweiterung für MEFS4SoS zur Ableitung von konkreten Szenarien (angelehnt an Mandel, Wäschle et al., 2021; Wäschle, Wolter et al., 2021; Wäschle, Wang et al., 2022).	129
Abbildung 7.1: IPEK-XiL-Framework für V2I (in Anlehnung an Albers, Behrendt et al., 2016; Wäschle, Wang et al., 2022).	136
Abbildung 7.2: Konzeptdarstellung der Toolchain für die Verknüpfung von Systemmodell mit realem System (Wäschle et al. 2022 basierend auf Mandel et al. 2021).	137

Abbildung 7.3: Übersicht der Kapitel mit der Anwendung der vier Schritten von MEFS4SoS und den zwei Ausprägungen der Validierung in den Schritten 4a & 4b.	138
Abbildung 7.4: V2I-Systemkontext als White Box Ansatz mit Schnittstellen.	139
Abbildung 7.5: Auszug der Wirkkette von CS Fahrer zu Systemanforderungen mit unterschiedlichen Beziehungen.	139
Abbildung 7.6: Funktionsstruktur zur Realisierung des UseCase GLOSA.	141
Abbildung 7.7: Sequenzdiagramm zur Nachrichtenübertragung für GLOSA.	142
Abbildung 7.8: Auszug der Einflussfaktoren auf die Nachrichtenübertragung bei GLOSA (Yi, 2019).	143
Abbildung 7.9: GLOSA Algorithmen mit vier und sieben Systemzuständen (angelehnt an Glimm & Zadek, 2021).	146
Abbildung 7.10: Iteratives Prozessablaufdiagramm zur Testplanung und -auswertung für die Anwendung des IPEK-XiL-Ansatzes (Mandel, Wäschle et al., 2021).	147
Abbildung 7.11: Systemmodell des IPEK-XiL-Ansatzes für V2I.	149
Abbildung 7.12: Auszug der Modellbibliothek des Systemmodells zum Aufbau von realen Systemen in MATLAB-Simulink und ns-3 mit Referenz UseCases in der Kategorie Demos (Chen 2020).	150
Abbildung 7.13: Exemplarische Messergebnisse für GLOSA (zeitlich) (übersetzt nach Wäschle, Wang et al., 2022).	152
Abbildung 7.14: Scatter Plot für GLOSA mit Sicherheits- und Verkehrseffizienzgrenzwert (N=800) (übersetzt nach dem Vortrag der Publikation Wäschle, Wang et al., 2022).	153
Abbildung 7.15: Überprüfung der Erfüllung der drei Kriterien Sicherheit, Komfort und Verkehrseffizienz beim Passiermanöver für GLOSA (N=800).	154
Abbildung 7.16: Fahrzeughalteposition in Abhängigkeit der Latenz für zwei Verlustmodelle der Signalübertragung (RPLM und LDPM) mit Übertragungsfrequenzen (1,2 Hz) und Abständen (60,100 m) für GLOSA.	156
Abbildung 7.17: Traceability von Testergebnis zu Bedarfen realisiert mit Hilfe von unterschiedlichen Beziehungen.	159
Abbildung 7.18: Prinzipdarstellung der virtuell-physischen Validierungsumgebung mit drei Validierungskonfigurationen dargestellt in den alternativen Pfaden A1 bis A3 (übersetzt nach Wäschle, Wang et al., 2022).	161
Abbildung 7.19: Alternative 1 – Fahrroboter fährt auf dem Prüfstand.	162

Abbildung 7.20: Alternative 2 – Fahrer fährt auf dem Prüfstand.	163
Abbildung 7.21: Alternative 3 – Fahrer fährt auf der Straße.....	164
Abbildung 7.22: GUI zur Anzeige für den Fahrer.....	166
Abbildung 7.23: Testergebnisse zu erfolgreichen Passiermanövern ohne Halten mit Fahrer für GLOSA.....	167
Abbildung 7.24: Exemplarische Messergebnisse Passieren ohne Halten mit Fahrer (zeitlich) (übersetzt nach Wäschle, Jiang, Behrendt & Albers, 2020).	167
Abbildung 7.25: Testergebnisse für erfolgreiche Passiermanöver mit und ohne Halten mit Fahrroboter für GLOSA.....	168
Abbildung 7.26: Exemplarische Messergebnisse Passieren mit Halten mit Fahrroboter (zeitlich) (vgl. Wäschle, Behrendt et al., 2021). .	169
Abbildung 7.27: Einteilung der Untersuchung auf Plausibilisierung zu realen Messungen im Straßenverkehr für GLOSA (vgl. Yi (2019), Hintergrundbild unten: Hauptmann (2019)).....	172
Abbildung 8.1: Übersicht der wesentlichen Ergebnisse der vorliegenden Arbeit eingeteilt nach den drei Stadien DS1, PS und Initiale DS2. ..	183
Abbildung A.1: Übersicht der Methoden in ISO 26262 & ISO/PAS 21448.....	XLV

Tabellenverzeichnis

Tabelle 2.1:	Vergleich V2X-Technologien anhand fünf Kriterien.....	42
Tabelle 2.2:	Eingesetzte und weitere Simulationstools eingeteilt nach UseCases.....	49
Tabelle 5.1:	Überblick der Datenbasis sortiert nach Anzahl an Publikationen	69
Tabelle 5.2:	Anzahl der Treffer für ausgewählte Suchphrasen in den Datenbanken Web of Science CoreCollection (WoSCC) und Tabelle 5.1 analysiert mit MAXQDA	70
Tabelle 5.3:	Vorauswahl der Charakteristiken für die Validierung von V2I ..	86
Tabelle 5.4:	Priorisierte Herausforderungen für die Validierung von V2I abgeleitet aus acht Herausforderungen für die Validierung von SoS und den ausgeprägten V2I-Charakteristiken.....	89
Tabelle 6.1:	Auszug der verknüpften Elementtypen des Problemraums am Beispiel von GLOSA.....	102
Tabelle 6.2:	Beispiel der Anforderungsarten	105
Tabelle 6.3:	Viewpoints für das Meta-Systemmodell auf Produktebene....	106
Tabelle 6.4:	Beschreibungselemente der Schnittstelle und die angelegten Elemente im C&C ² -Ansatz.....	109
Tabelle 6.5:	Untersuchung der drei Variationsarten der SGE an den Elementen des Systemtupels	111
Tabelle 6.6:	Auszug aus der Szenarienbeschreibung mittels Verlinkung von Merkmalen mit der Allocate Beziehung zu UseCases	114
Tabelle 6.7:	Auszug der FMEA	120
Tabelle 6.8:	Auszug der Szenarienbeschreibung in SysML dargestellt in einer Design Structure Matrix	122
Tabelle 6.9:	Auszug einer HARA-Tabelle in SysML erstellt in dem Plugin „Cameo Safety & Reliability Analysis“	125
Tabelle 7.1:	Auszug der zu modellierenden Elemente für den UseCase GLOSA.....	140
Tabelle 7.2:	Übersicht – Elemente der Funktionalen Sicherheit für die beschriebene Gefahrensituation.....	143

Tabelle 7.3:	Risikotabelle (basierend auf ISO 26262-3:2018(E), S. 19; No Magic Product Documentation, 2022)	144
Tabelle 7.4:	Auszug der zu modellierenden veränderten Elemente für den UseCase GLOSA mit Variationsarten des Modells der PGE.	145
Tabelle 7.5:	Validierungsziele, -umgebung und Hypothesen für den UseCase GLOSA innerorts	148
Tabelle 7.6:	Test zum Systemverhalten für den UseCase GLOSA	150
Tabelle 7.7:	Übersicht - Test zum Systemverhalten für GLOSA	151
Tabelle 7.8:	Übersicht - Test zur Signalübertragung für GLOSA	155
Tabelle 7.9:	Analyse im Tool ModelCenter basierend auf Testergebnissen im Systemmodell	158
Tabelle 7.10:	Übersicht – Validierungsziele und Hypothesen für den UseCase GLOSA innerorts	160
Tabelle 7.11:	Übersicht - Tests zu Passiermanöver für den UseCase GLOSA innerorts.....	165
Tabelle 7.12:	Übersicht – Test zu Passiermanöver mit Fahrer auf dem Rollenprüfstand und auf der Straße für den UseCase GLOSA	166
Tabelle 7.13:	Übersicht - Rollenprüfstandstest zu Passiermanöver mit Fahrer für GLOSA.....	168
Tabelle B.1:	Übersicht der Funktionen einzelner Koppelsysteme	XLVI
Tabelle D.1:	Erweiterung der Validierungsumgebung.....	XLIX

Abkürzungsverzeichnis

ACC	Adaptive Cruise Control
ASAM	Association for Standardization of Automation and Measuring Systems
ASE	Advanced Systems Engineering
ASIL	Automotive Safety Integrity Level
ASPICE	Automotive Software Process Improvement and Capability Determination
AV	Ausprägungsvariation
AW	Auftretenswahrscheinlichkeit
BR	Business Requirement
BUS	Binary Unit System
C	Connector
CAM	Cooperative Awareness Message
CAD	Computer-Aided Design
CAN	Controlled Area Network
CONSENS	CONceptual design Specification technique for the ENgineering of mechatronic Systems
CPS	Cyber-Physical Systems
CS	Constituent Systems bzw. eigenständige Systeme
C&C ² -Ansatz	Contact & Channel – Ansatz
DeR	DetectionRate

DR	Domain Requirement
DRM	Design Research Methodology
DS	Deskriptive Studie
DSRC	Dedicated Short Range Communication
ECTS	European Credit Transfer and Accumulation System
ETSI	Europäische Institut für Telekommunikationsnormen
FAS(4M)(4X)	Functional Architectures of Systems (for Mechanical Engineers) (for Everything)
FF	Forschungsfrage
FMEA	Failure Mode Effect Analysis
G1	Gefahr Nummer 1
G _n	Systemgeneration n
GA	Grundannahme
GS	Gefahrensituation
GLOSA	Green Light Optimal Speed Advisory
GNSS	Global Navigation Satellite System
HARA	Hazard and Risk Analysis
HiL	Hardware-in-the-Loop
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council of Systems Engineering
ITS	Intelligente Transportsysteme
IoT	Internet of Things
IP	Integrierte Produktentwicklung
IPEK	Institut für Produktentwicklung am KIT

iPeM	integriertes Produktentstehungs-Modell
KaSPro	Karlsruher Schule für Produktentwicklung
KI	Künstliche Intelligenz
KIT	Karlsruher Institut für Technologie
LAN	Local Area Network
LDPLM	Log Distance Propagation Loss Model
LSA	Lichtsignalanlage
LSS	Leitstützstruktur
LTE	Long Term Evolution
MB(S)E	Model-Based (Systems) Engineering
MEFS4SoS	Methode zur Erfüllung der Funktionalen Sicherheit für die Validierung von System of Systems
ns3	Network Simulator 3
OBU	Onboard Unit
ODD	Operational Design Domain
OMG	Object Management Group
OSI	Open Systems Interconnection
PDM	Propagation Delay Model
PEP	Produktentstehungsprozesses
ProViL	Produktentwicklung im virtuellen Labor
PS	Problem Space bzw. Problemraum
PS1	Präskriptive Studie 1
PSS	Product-Service-Systems
PV	Prinzipvariation

R _n	Referenzsystem der Systemgeneration G _n
RC	Research Clarification
RFLP	Anforderungen, Funktionen, Logische-, Physische Elemente
RPZ	Risikoprioritätszahl
RSU	Roadside Unit
SAE	Society of Automotive Engineers
SE	Systems Engineering
SGE	Systemgenerationsentwicklung
SiD	System-in-Development
SoS	System of Systems
SOTIF	Safety of the intended function
SpaT	Signal Phase And Timing
SPES	Software Platform Embedded Systems
SS	Schadensstärke
STAMP	System-Theoretic Accident Model and Processes
STPA	Systems Theoretic Process Analysis
SuI	System-under-Investigation
SysML	Systems Modeling Language
SYSMOD	Systems Modeling Toolbox
SysR	Systemanforderung
UAF	Unified Architecture Framework
UDP	User Datagram Protocol
UML	Unified Modeling Language

V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Person
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VAMOS	Variant Modeling with SysML
VANET	Vehicular Ad Hoc Network
WF	Wirkfläche
WFP	Wirkflächenpaar
WoSCC	Web of Science Core Collection
XiL	X-in-the-Loop
XMI	XML Metadata Interchange
ZHO	Zielsystem, Handlungssystem, Objektsystem
ÜV	Übernahmevariation

Formelzeichen

N	Stichprobengröße
m	arithmetisches Mittel
σ	Standardabweichung der Stichprobe
L	Pfadverlust
L_0	Pfadverlust bei Referenzdistanz
n	Pfadverlust Distanz Exponent
d	Distanz
d_0	Referenzdistanz

1 Einleitung

Die Mobilität hat als eines der Grundbedürfnisse des Menschen einen hohen Stellenwert im täglichen Leben. Im Laufe der Zeit veränderte sich die Mobilität und die Bedarfe abhängig von technologischen Innovationen und einhergehender Auswirkungen. So führte die Erfindung der Pferdekutsche zu einem Platzbedarf für Pferde sowie zu Bedarfen nach Sicherheit aufgrund der steigenden Anzahl an Kutschen. In den letzten Jahren führen Faktoren wie die Industrialisierung und der steigende Energiebedarf der Weltbevölkerung zu einer globalen Erwärmung und zu dem Bedarf nach einer nachhaltigen Entwicklung. Dieser wird von den Vereinten Nationen in 17 Ziele adressiert (Vereinte Nationen, 2016).

In Zukunft kann die vernetzte Mobilität unterstützen, acht dieser Ziele zu erreichen. Die Vernetzung von Systemen wie Infrastruktur mit Fahrzeugen bewirkt beispielsweise eine Verringerung von Unfällen um ca. 80% (Nassir et al., 2020) und eine Reduzierung des Energiebedarfs um bis zu 8% auf Autobahnen (Rösener et al., 2019). Studien bestätigen den Trend zu vernetztem Fahren, der „zu einer deutlichen Zunahme von komplexen und voneinander abhängigen Systemen innerhalb und außerhalb des Fahrzeugs [führt]“ (Nationale Plattform Mobilität der Zukunft, 2020, S. 18).

In der vernetzten Mobilität können Systeme wie Fahrzeuge und Infrastruktur unabhängig entwickelt und betrieben werden. Trotz teilweiser eigener Zwecke werden sie zur Realisierung von Emergenzeffekten zu sogenannten Systems of Systems (SoS) vernetzt. Die Entwicklung dieser SoS führt zu mehreren Herausforderungen (Albers, Peglow, Powelske, Birk & Bursac, 2018; Dahmann, 2014), wobei die Validierung von SoS als eine der größten Herausforderungen angesehen wird (Dahmann, 2014). Daher wird in der vorliegenden Arbeit die Validierung eines SoS untersucht.

Die Größe eines SoS sowie Charakteristiken wie der heterogene Lebenszyklus der enthaltenen Systeme führen zu neuen Anwendungsfällen und der Notwendigkeit die Systeme nachvollziehbar zu beschreiben. Dies kann durch Systems Engineering erfolgen. Mit Hilfe von Modellen werden Ansätze des Model-Based Systems Engineerings auf ihren Nutzen für die Validierung untersucht.

Die vorliegende Arbeit untersucht das spezifische SoS Vehicle-to-Infrastructure (V2I). Anhand von konkreten Anwendungsfällen wird die modellbasierte Unterstützung der Validierung angewendet und geprüft.

Die vorliegende Arbeit untersucht in der Produktentwicklung die Validierung von V2I mittels des zentralen Ansatzes Model-Based Systems Engineering (siehe Kapitel 2). Die aus Kontext, zentralem Ansatz und Forschungsgegenstand aufgezeigte Forschungslücke wird in einem Forschungsbedarf bewertet und resultiert in der Zielstellung der vorliegenden Arbeit (siehe Kapitel 3). Diese lautet eine durchgängige, kontinuierliche Sprache und Methode zur Validierung der Funktionalen Sicherheit von Vehicle-to-Infrastructure als SoS durch MBSE-Unterstützung unter Einsatz des IPEK-XiL-Ansatzes zu entwickeln. Zur Erreichung des Ziels wird ein Forschungsvorgehen angelehnt an der Design Research Methodology angewendet (siehe Kapitel 4). Die aufgestellten drei Forschungsfragen werden in den drei Kapiteln 5 bis 7 beantwortet, woran Zusammenfassung, Gesamtfazit und Ausblick der vorliegenden Arbeit anschließen.

1. Einleitung	
2. Stand der Forschung und Technik	
2.1 Systemtheorie und Systems Engineering	2.2 Validierung in der Produktentwicklung
2.3 Beschreibung und Validierung von V2I	2.4 Diskussion und Fazit
3. Bedarf und Zielsetzung	
3.1 Forschungsbedarf	3.2 Grundannahmen und Forschungsziel
3.3 Grundthese und Forschungsfragen	
4. Methodisches Forschungsvorgehen	
4.1 Vorgehen angelehnt an die DRM ...	4.2 Forschungsumgebung
4.3 Empirische Methoden	
5. Beschreibung des generischen Zielsystems für die Validierung von V2I	
5.1 Empirische Studie zur Bedarfsermittlung	5.2 Vorgehen zur Zielsystembeschreibung von V2I
5.3 Diskussion und Zwischenfazit	
6. Modellierungssprache und -methode MEFS4SoS in MBSE zur Unterstützung der Validierung von V2I	
6.1 Beiträge für die Modellierungssprache	6.2 Konkretisierung der Modellierungssprache: ...
6.3 MEFS4SoS – Methode zur Erreichung der FuSi ...	6.4 Diskussion und Zwischenfazit
7. MBSE-unterstützte Validierung von V2I auf Basis des IPEK-XiL Ansatzes	
7.1 Systembeschreibung	7.2 Gefahrenidentifikation und Risikoevaluation
7.3 Funktionale Modifikation und Spezifikationen	7.4 Validierung – GLOSA virtuell
7.5 Validierung – GLOSA virtuell-physisch	7.6 Diskussion und Zwischenfazit
8. Zusammenfassung, Gesamtfazit und Ausblick	
8.1 Zusammenfassung	8.2 Gesamtfazit
8.3 Ausblick	

Abbildung 1.1: Aufbau der Arbeit.

2 Stand der Forschung und Technik

Grundlegend in der heutigen Produktentwicklung ist die bereits in den 1950er Jahren beschriebene Systembetrachtung im Sinne des Systems Engineering (siehe Kapitel 2.1). Systeme, wie beispielsweise in Ropohl (1975, 2009) beschrieben, ermöglichen das Aufstellen von Modellen der Produktentwicklung. Die vorliegende Arbeit fokussiert dabei auf die Validierung und nutzt Modelle der Karlsruher Schule für Produktentwicklung (KaSPro) (siehe Kapitel 2.2). Systems Engineering wird anschließend genutzt, um das System V2I im Kontext des hochautomatisierten Fahrens mit den momentanen Ansätzen der Validierung vorzustellen (siehe Kapitel 2.3).

2.1 Systemtheorie und Systems Engineering

Die Betrachtung eines Produkts als System ist notwendig, um die steigende Anzahl an Systemelementen in der Produktentwicklung zu berücksichtigen. Dies verdeutlicht Abbildung 2.1, in der die Zunahme von steigenden Systemelementen über die letzten 2.500 Jahre dargestellt ist. Dieser Trend führt zu steigender Komplexität aufgrund der Vernetzung der Systemelemente (siehe Ulrich & Eppinger, 2012).

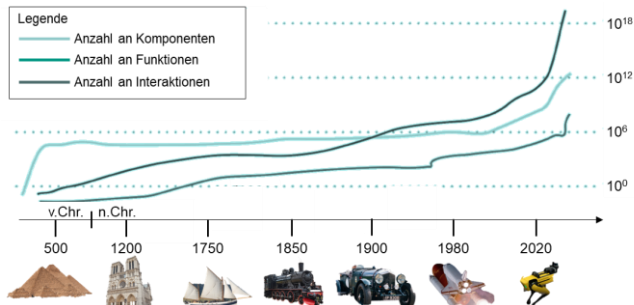


Abbildung 2.1: Trend zu steigender Anzahl an Funktionen, Teilsystemen und Interaktionen (angepasst von Friedenthal et al., 2021) ¹.

¹ Copyright: INCOSE

Daher kam bereits um 1950 im Bereich der materiellen Güter der Begriff der Systemtheorie durch die Arbeit „An Outline of General System Theory“ auf (Bertalanffy, 1950). Später beschrieb Ropohl (1975, 2009) Systeme mit Eigenschaften und Strukturen und erweiterte die Systemtheorie um immaterielle Güter.

Ebenfalls um 1950 wurde die systemische Entwicklung (engl. „Systems Engineering“ (SE)) beschrieben und angewendet. Die Forschungseinrichtung Bell Labs des Telekommunikationsunternehmens AT&T führte systematische Schritte ein, um die Entwicklung von Telekommunikationssystemen zu beschreiben (Fagen, 1978; Hall, 1962; Schlager, 1956). Hall (1962, S. 8–11) definierte auf dieser Basis Systems Engineering mit den fünf wesentlichen Phasen:

- (1) Systemstudien oder Programmplanung, mit der Untersuchung der Systemumwelt,
- (2) Sondierungsplanung, die die Problemdefinition, die Auswahl der Ziele, die Systemsynthese und -analyse mit der Auswahl des besten Systems umfasst,
- (3) Entwicklungsplanung, die Phase 2 in größerer Ausführlichkeit wiederholt,
- (4) Studien während der Entwicklung, die die Entwicklung von Teilen des Systems und die Integration und Prüfung dieser Teile umfassen,
- (5) laufendes Engineering, das stattfindet, während das System in Betrieb ist.

Kapitel 2.1.1 stellt die wesentlichen Inhalte der Systemtheorie vor, welche seit 1996 um den in der vorliegenden Arbeit genutzten Begriff des „System of Systems“ (SoS) (siehe Kapitel 2.1.2) ergänzt wurde. Als ein Zweig der Systemtheorie gilt die modellbasierte Entwicklung (engl. Model-Based Engineering (MBE)), welche um ein zentrales Systemmodell erweitert, zum „Model-Based Systems Engineering“ (MBSE) führt. MBSE wird als zentraler Ansatz für die vorliegende Arbeit in Kapitel 2.1.3 eingeführt. Neue Fortschritte wie die Digitalisierung und zunehmende Herausforderungen wie ein hoher Grad an Vernetzung und Interdisziplinarität der Systeme führen zu der Entwicklung des übergeordneten Begriffs „Advanced Systems Engineering“ (ASE), das MBSE umfasst. ASE beschreibt die neuartige bzw. zukünftige Systementwicklung und wird in Kapitel 2.1.4 vorgestellt.

2.1.1 Systemtheorie

Um die Validierung eines Systems zu untersuchen, wird zunächst das Verständnis zum System mit der Definition nach Ropohl (2009, S. 77) geschaffen:

„Ein **System** ist das Modell einer Ganzheit, die

- (a) Beziehungen zwischen Attributen (In-/Outputs, Zustände) aufweist
- (b) Aus miteinander verknüpften Teilen bzw. Subsystemen besteht
- (c) Von ihrer Umgebung bzw. einem Supersystem abgegrenzt wird.“

Ropohl stellt drei Konzepte von Systemen vor, das funktionale, strukturele und hierarchische Konzept (siehe Abbildung 2.2). Das funktionale Konzept beschreibt das System über die zu erzielende Funktion mit Input, Transfer und Output. Im strukturalen Konzept werden Relationen und Elemente innerhalb des Systems beschrieben. Das hierarchische Konzept konzentriert sich auf die Verschachtelung von Subsystemen zu System und weiter zu Supersystem.

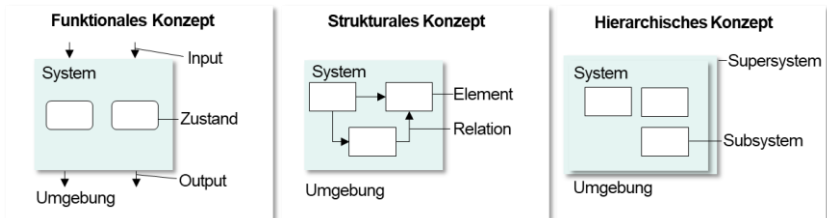


Abbildung 2.2: Drei Konzepte der Systemtheorie (Ropohl, 1975, 2009).

Um die Relationen im strukturalen Konzept zu modellieren, bietet sich der C&C²-Ansatz an. Der Ansatz verwendet die drei Schlüsselemente Wirkflächen (WF), Leitstützstrukturen (LSS) und Konnektoren (C) zur Beschreibung von Funktionalitäten (Albers & Wintergerst, 2014; Matthiesen, 2002). WF sind generalisierte Flächen wie Oberflächen von Körpern oder Grenzflächen von Flüssigkeiten, Gasen oder Feldern. Sie stehen in ständigem oder gelegentlichem Kontakt zu anderen WF, um ein Wirkflächenpaar (WFP) zu bilden und können Energie, Material und Informationen austauschen (Albers & Wintergerst, 2014; Wiedner, 2013). Leitstützstrukturen können die WF verbinden und Materie, Energie und Informationen transportieren. Sie existieren nur dann, wenn WFP verbunden sind und beschreiben den Wirkungsweg. Konnektoren integrieren für die Funktion relevanten Eigenschaften, die sich außerhalb des Gestaltungsraums befinden (Albers & Wintergerst, 2014). Ein Wirk-Netz ist die Summe aus WFP, C und LSS zur Ausführung einer Funktion in einem bestimmten Betriebszustand (Albers & Wintergerst, 2014). Eine Wirkfläche kann eine Bauteilfläche wie die Zahnflanke

eines Zahnrades sein. An den Zahnflanken sind die Räder in Kontakt und übertragen Energie. Daher überträgt das Rad ein Drehmoment auf ein anderes Rad.

Das System besteht aus ein oder mehreren Systemmodellen, welche wiederum ein oder mehrere Sichten enthalten, die mit Diagrammen visualisiert werden (Holt & Perry, 2018).

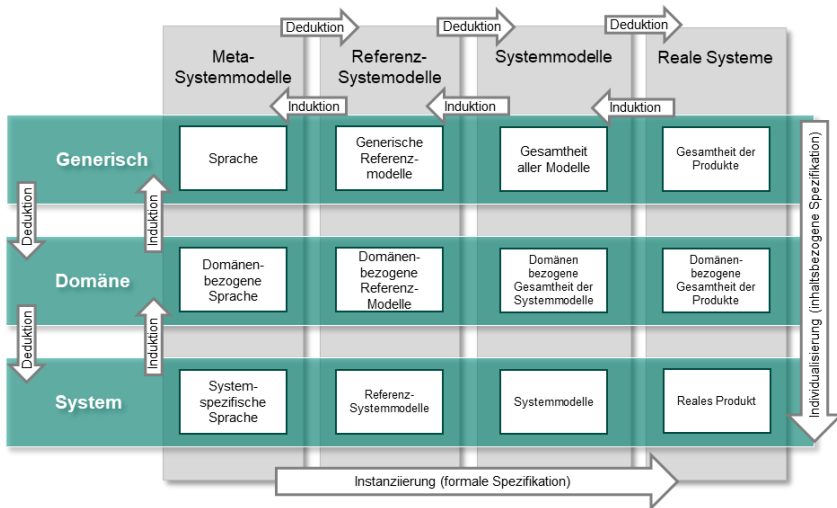


Abbildung 2.3: Abstraktionsgrade der Systemmodellierung (Albers, Matthiesen et al., 2015; Bursac, 2016; Muschik, 2011, S. 187).

Es werden die Abstraktionsgrade „Generisch“, das heißt über alle Domänen hinweg, „Domäne“ beispielsweise Maschinenbau und „System“ unterschieden. Weiterhin werden Modelle unterschieden nach Meta-, Referenz-Systemmodell, Systemmodell und realem System. Meta-Systemmodelle enthalten Regeln für die Beschreibung von Modellelementen und deren Beziehungen. Referenz-Systemmodelle unterstützen die Modellierung. Sie sind als eine Art „Baukasten“ zu verstehen, die Regeln aus dem Meta-Systemmodell spezifizieren. Die Spezifikation wird in Systemmodellen fortgesetzt. Systemmodelle enthalten bereits logische Elemente. Beispielsweise wird auf Systemebene ein Axialfluss Motor beschrieben, wobei das reale System Instanzen aufweist wie z.B. der Motor YASA 750 R der Firma YASA Limited. Dies führt dazu, dass reale Systeme ausgeführt werden können, beispielsweise in einer Simulation mit konkreten Parametersätzen.

Modelle können auch nach der Detaillierung des Systeminhalts in Black Box und White Box unterschieden werden (BCS SIGIST, 2001). Black Box Modelle enthalten

keine detaillierte Systembeschreibung und liegen dann vor, wenn ein System ohne genaue Kenntnisse seiner Struktur, Funktion und Hierarchie untersucht wird. Deutlich wird dies beispielsweise bei der Kooperation zwischen OEM und Lieferant, wenn Betriebsgeheimnisse geschützt werden, sodass in Bezug auf Input und Output des Partnerunternehmens nur ein Black Box Modell erstellt werden kann. Im Gegensatz zur Black Box Modellierung enthält die White Box Modellierung detaillierte Beschreibungen des Systems. Dies liegt üblicherweise bei der Inhouse-Entwicklung eines Systems vor.

Als eine Art des Systems wird das „soziotechnische System“ genannt. Dieses enthält technische Teilsysteme, die gesellschaftliche Sozio-Aspekte beeinflussen und mit ihnen wechselwirken. Diese Wechselwirkung veranlasst Hieber und Kammeyer (2014) zu der Aussage, dass es in Zukunft in der Verantwortung von Entwickelnden liegt, auch gesellschaftliche Auswirkungen zu betrachten. Nach Albers (2023) haben daher alle technischen Systeme eine Auswirkung auf die Gesellschaft und sind daher als soziotechnische Systeme zu bezeichnen.

Eine verwandte Art des Systems führt das nachfolgende Kapitel mit dem Begriff „System of Systems“ ein.

2.1.2 System of Systems

Einer der Treiber für diesen Systembegriff ist die Digitalisierung und die einhergehende Vernetzung von Systemen. Diese auch eigenständig betriebenen und verwalteten Systeme wurden 1996 von Maier „System aus Systemen“ (engl. „System of Systems“ (SoS)) bezeichnet. Er beschreibt fünf Charakteristiken von SoS (aus dem Englischen übersetzt nach M. Maier, 1996, S. 566):

- (1) Operative Unabhängigkeit,
- (2) Unabhängigkeit im Management,
- (3) Evolutionäres Verhalten,
- (4) Emergentes Verhalten,
- (5) Geographische Verteilung.

Zu den fünf von Maier genannten Charakteristiken beschreiben Nielsen, Larsen, Fitzgerald, Woodcock und Peleska (2015) abgeleitet aus einer Literaturrecherche:

- (6) Dynamische Rekonfiguration,
- (7) Wechselwirkungen,
- (8) Interoperabilität.

Jamshidi (2008) beschreibt das SoS in der von der INCOSE vertriebenen Zukunftsvision 2035 als eine Integration von eigenständigen Systemen (engl. Constituent Systems (CS)), welche unabhängig geführt und betrieben werden und

welche für einen bestimmten Zeitraum zusammen interagieren, um ein höheres Ziel zu erreichen (Friedenthal et al., 2021, S. 39). Dies und eine Vielzahl weiterer Veröffentlichungen veranlassten Maier zu der Aussage, dass es eine Vielzahl von Charakteristiken gibt, jedoch die ersten drei Charakteristiken: Operative Unabhängigkeit, Unabhängigkeit im Management und Emergentes Verhalten als Kerncharakteristiken bezeichnet werden können (M. Maier, 2021).

Albers, Mandel, Yan und Behrendt (2018) verwenden die Charakteristiken zur Beschreibung einer Validierungsumgebung. Angelehnt hieran wurde nach Untersuchungen in Kapitel 5.2 die Ausprägung der Charakteristiken auch für SoS unterschieden. Diese führt dazu, dass in der vorliegenden Arbeit die nachfolgende Definition, angelehnt an die Standards der ISO/IEC/IEEE, festgelegt wird:

Ein **System of Systems (SoS)** ist ein System, bestehend aus einer Menge interagierender Systeme, von denen jedes einzelne System in sich selbst als System betrachtet werden kann. Die wichtigsten Charakteristika von SoS sind:

- Jedes System kann unabhängig agieren und einen eigenen Zweck besitzen.
- Die individuellen Systeme der Menge werden unabhängig organisiert, um ihre Zwecke zu erfüllen.
- Die Systemkombination liefert Ergebnisse, die von einzelnen Systemen nicht erreicht werden können.

Die Charakteristika können unterschiedlich stark ausgeprägt sein. (vgl. Albers, Mandel et al. 2018, vgl. BS ISO/IEC/IEEE 21841:2019)

Die Beschreibung des SoS veranlasste M. Maier (1998) dazu, drei Typen von SoS einzuführen: (1) Zentral ausgerichtet (engl. Directed); (2) Kollaborativ; (3) Virtuell. Dahmann und Baldwin (2008) ergänzten die Typen um Anerkannt (engl. Acknowledged) und visualisierten die vier Typen in Abbildung 2.4. Die Typen werden im Folgenden beschrieben.

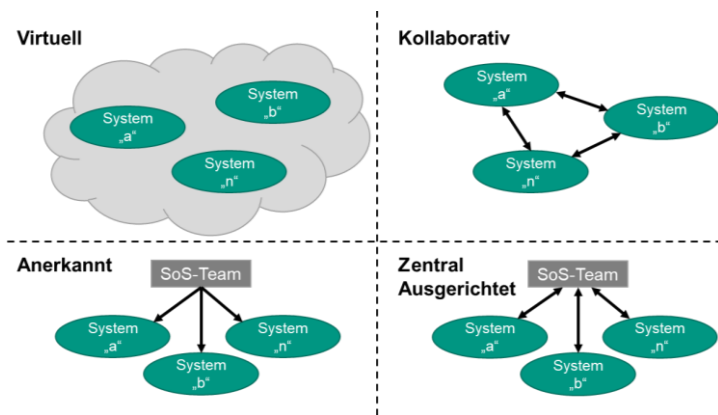


Abbildung 2.4: Die vier Typen von System of Systems (Dahmann & Baldwin, 2008; Dahmann, 2014; Lane & Epstein, 2013).

Das *virtuelle* SoS hat kein zentrales Management und keinen zentral beschlossenen gemeinsamen Zweck. Zudem sind nicht alle eigenständigen Systeme (engl. Constituent Systems (CS)) des SoS bekannt. Beispielsweise kann das Internet als virtuelles SoS bezeichnet werden. Das *kollaborative* SoS hat eine anreizbasierte Zusammenarbeit für einen gemeinsamen zentralen Zweck. Es enthält jedoch kein zentrales Management wie es beispielsweise in einem regionalen Krisensystem vorkommen kann. Im *anerkannten* (engl. Acknowledged) SoS gibt es ein Management mit anerkannten gemeinsamen Zielen sowie gemeinsame Ressourcen der CS. Die CS sind weiterhin unabhängig hinsichtlich ihrer Eigenverantwortung, Ziele, Finanzierung und Entwicklungsansätze. Ein CS ist nicht verpflichtet, Anforderungen des zentralen Managements nachzukommen. Das *direkte* SoS besitzt nun spezifische gemeinsame Ziele, ein zentrales Management mit Weisungsbefugnis zur Ressourcenallokation an die CS, beispielsweise in einer Inhouse-Entwicklung eines Batteriespeichersystems von Tesla, Inc.. Weiterhin können verbindliche Regelungen, wie sie im Gesundheitswesen vorkommen, zu einem direkten SoS führen. (Dahmann & Baldwin, 2008; M. Maier, 1998)

Die folgenden vier Standards der ISO/IEC/IEEE verdeutlichen einen wesentlichen Punkt bei der Betrachtung von SoS, die Perspektive auf das System (siehe Abbildung 2.5). Der Systembegriff wird im Standard ISO/IEC/IEEE 15288-2015 ohne die Berücksichtigung von SoS behandelt. Es kann jedoch auch eine Systementwicklung berücksichtigt werden, in der das zu entwickelnde System Teil eines SoS ist (ISO/IEC/IEEE 21839-2019). Weiterhin ist es möglich die Beziehungen zwischen Systemen innerhalb eines SoS zu betrachten (ISO/IEC/IEEE 21840-2019). Zuletzt kann das gesamte SoS untersucht werden (ISO/IEC/IEEE 21841-2019).

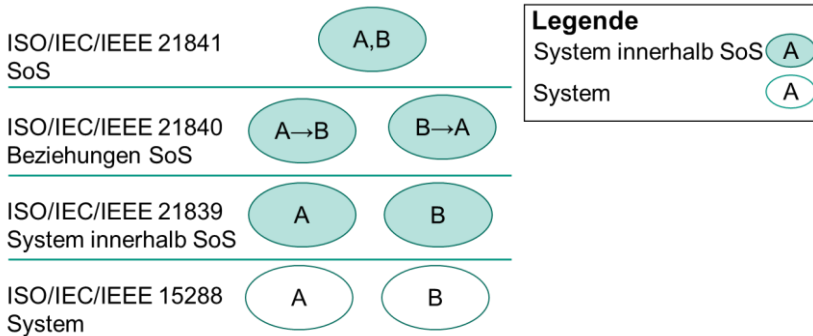


Abbildung 2.5: Vier Standards im Rahmen der Systementwicklung klassifiziert nach der Perspektive (angelehnt an BS ISO/IEC/IEEE 21841:2019).

Abgeleitet von den generellen Trends, den Charakteristika, Typen und Perspektiven von bzw. auf SoS wurden zahlreiche Herausforderungen identifiziert. Abbildung 2.6 verdeutlicht einige Herausforderungen in der Entwicklung von SoS.

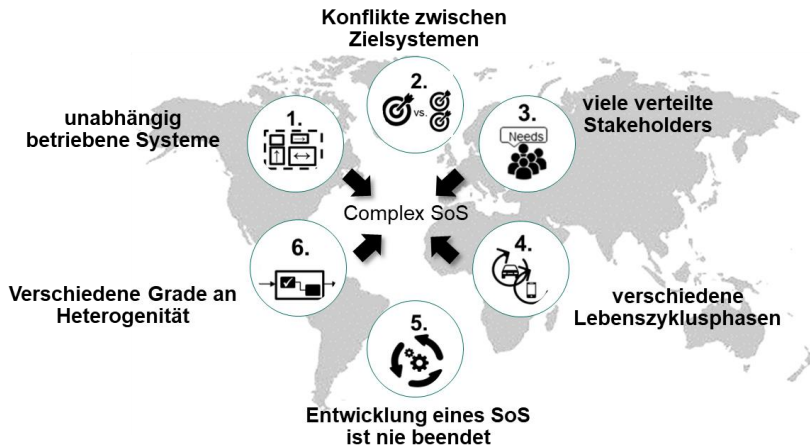


Abbildung 2.6: Herausforderungen in der Entwicklung von SoS (Albers, Peglow et al., 2018).

Eine wesentliche Herausforderung ist die Validierung von SoS (Dahmann, 2014; Friedenthal et al.). Aus diesem Grund wird im Folgenden auf diese Herausforderung detaillierter eingegangen. Die Validierung von SoS führt zu acht zentralen Herausforderungen, die sich durch Aggregation von mehreren Quellen ergeben (Albers, Peglow et al., 2018; Dahmann, 2014; Friedenthal et al., 2021; Gasser et al., 2015; BS ISO/IEC/IEEE 21841:2019; Nationale Plattform Mobilität der Zukunft, 2019; Nielsen et al., 2015) (siehe Abbildung 2.7):



Abbildung 2.7: Herausforderungen für die Validierung von SoS (Albers, Peglow et al., 2018; Dahmann, 2014; Friedenthal et al., 2021; Nielsen et al., 2015).

1. Die große Anzahl und Vielfalt an Stakeholder, Normen und Standards erschweren die Validierung von SoS. Dies sowie die Anzahl an Elementen und Beziehungen führen zu einer hohen Komplexität. Aus diesem Grund stellt sich die Frage, wie Unternehmen zukünftig zusammenarbeiten und wie die Systemkomplexität in der Validierung (z.B. in der Teststrategie) beherrscht werden kann.
2. Eine Vereinheitlichung von Modellierungen zur gemeinsamen Verifikation ist erforderlich. Beispielsweise existieren zustandsreiche, gleichzeitige, verteilte, autonome, zeitabhängige, mobile, hybride, diskrete und kontinuierliche und probabilistische Systeme. Eine Verknüpfung von Aktivitäten der Validierung und ein effektiver Einsatz ist erforderlich.
3. Eine dynamische Entwicklung von SoS erschwert die Testautomatisierung. Beispielsweise haben einzelne Systeme innerhalb des SoS häufig asynchrone Lebenszyklen. Beispielsweise hat die Infrastruktur eine Nutzungsdauer von meist mehr als zwanzig Jahre (vgl. Stadt Nürnberg, 2021) im Vergleich zu mechatronischen Systemen wie Fahrzeugen mit meist unter zwanzig Jahren (Krafftahrt-Bundesamt, 2021) und softwareintensiven Systemen wie Mobiltelefone mit einer mehrheitlichen Nutzungsdauer bis zu einem Jahr (Haas, 2023).

(Albers, Peglow et al., 2018; Albers, Mandel et al., 2018; Wäschle, Behrendt, Xing, Shi & Albers, 2021; Wäschle, Wang, Xinlu & Albers, 2022). Durch die dynamische Entwicklung können neue Systemeigenschaften entstehen, welche zu neuen Testzielen führen.

4. Es besteht der Bedarf an modellbasierten, datengetriebenen Methoden, Verfahren und Tests für inkrementelle SoS.
5. Eine Herausforderung ist die Integration der Validierung in die Entwicklungsprozesse und eine praxisnahe Skalierung der Methoden, Verfahren und Tests.
6. Zur Validierung sind die Ableitung und Pflege von SoS-Spezifikationen wesentlich, um unerwünschte Emergenzen zu identifizieren und zu vermeiden.
7. Es besteht der Bedarf an virtuellen Validierungsaktivitäten und Aktivitäten in einem Labor unter realen Bedingungen. Die Herausforderung ist einerseits die bedarfsgerechte Nutzung von Produkt-, Feld- und Simulationsdaten in Validierungsumgebungen. Andererseits, die sichtbare, verständliche Exploration des Verhaltens von hochautomatisierten, intelligenten Systemen.
8. Mit höherem Grad der Automatisierung scheinen bestehende Methoden, Verfahren und Tests an ihre Grenzen zu stoßen. Daher stellt sich die Frage, wie die Betriebssicherheit gewährleistet werden kann und welches beispielsweise die Einflussfaktoren im Straßenverkehr auf die Betriebssicherheit von hochautomatisierten Fahrzeugen sind.

Die nachfolgenden zwei wissenschaftlichen Beiträge nehmen sich der Herausforderungen von SoS an und bilden eine semantische Grundlage.

So findet sich in Coleman et al. (2012) eine Ontologie für SoS wieder, welche ausgehend vom System und dessen Kontext weitere zentrale Elemente wie Ziele, Anforderungen und Ressourcen beinhaltet. Holt und Perry (2013a) erweitern diese Ontologie um Prozesse und beschreiben kontextsensitiv die Anforderungen für SoS (siehe Abbildung 2.8). (Dandashi & Hause, 2015)

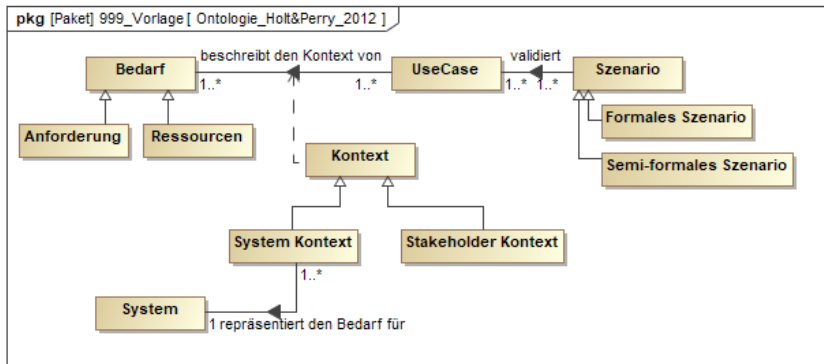


Abbildung 2.8: Auszug aus der Ontologie für SoS (Holt & Perry, 2013a).

Als gesamtheitliches Architektur Framework für SoS gibt in dem zweiten Beitrag das Unified Architecture Framework (UAF) Regeln und Methoden zur Modellierung von zehn Sichten auf ein SoS vor. Der Fokus liegt auf der Prozessmodellierung mit verknüpften Ressourcen. Die Sichten sind: strategische Fähigkeiten, operative Szenarien, Dienstleistungen, Ressourcen, Personal, Sicherheit, Projekte, Standards, Maßnahmen und Anforderungen. Für UAF existiert eine eigene Sprache, die UAF Modeling Language sowie die Implementierung in SysML mittels UAF-Profiles. Jede Domäne wird mit einem UAF Domain Metamodel beschrieben. Die große Anzahl an Sichten und Elementen, führt jedoch zu einem hohen Aufwand hinsichtlich der Prozessmodellierung ohne dabei die Validierung und Verifikation zu berücksichtigen. Die sehr geringe Anzahl an Suchergebnissen für die Begriffe „validation“ (vorkommen in UAFModelingLanguage ; UAFDomain Model: 0; 0), „verification“ (0; 1) und „testing“ (3; 3) führen zu dem Schluss, dass die Modellierung der Validierung nicht detailliert beschrieben ist. (Object Management Group, 2021; Object Management Group, 2022)

Die zwei Arbeiten zeigen die Modellierung als wesentliche Aktivität für SoS. Verschiedene Veröffentlichungen nutzen ein zentrales Systemmodell und beschreiben unterschiedliche Sichten auf dieses (vgl. Bilal, Daclin und Chapurlat (2014), Lollini, Mori, Babu und Bouchenak (2016)). Dies wird im Kontext des Systems Engineerings als Model-Based Systems Engineering bezeichnet und nachfolgend vorgestellt.

2.1.3 Model-Based Systems Engineering

MBSE ist ein Teil des Systems Engineering mit dem Einsatz eines zentralen Systemmodells. Das Modell fördert Konsistenzen und eine einheitliche Beschreibung. Mittels des föderativen oder zentralen Ansatzes ist das zentrale Systemmodell mit verschiedenen teilweise domänenspezifischen Expertentools verknüpft. Im zentralen Ansatz sind die Expertentools über eine Plattform eines Anbieters verknüpft (vgl. Cameo 3D Experience von Dassault Systèmes). Der föderative Ansatz beschreibt Expertentools aus verschiedenen Systemlandschaften, welche gekoppelt werden (vgl. Phoenix ModelCenter von Ansys).

MBSE setzt sich unabhängig von den zwei Ansätzen aus den in Abbildung 2.9 dargestellten drei zentralen Säulen Sprache, Methode und Tool zusammen (Delligatti, 2014). Der Fokus der vorliegenden Arbeit liegt auf Sprache und Methode, wobei das Tool Cameo Systems Modeler 19.0 von Dassault Systèmes hauptsächlich eingesetzt wird.

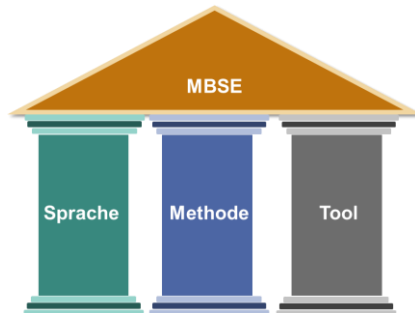


Abbildung 2.9: Die drei Säulen des MBSE – Sprache, Methode und Tool (angelehnt an Delligatti, 2014).

Unter **Sprache** werden „Wörter“ und „Grammatik“ für die Modellierung verstanden. Hierbei hat sich aus der in der Informatik verbreiteten „Unified Modeling Language“ (UML) die Sprache Systems Modeling Language (SysML) etabliert. Parallel hierzu gibt es weitere Bestrebungen eine Vereinfachung zu realisieren (vgl. iQuavis von TwoPillars) oder weitere Sprachen (vgl. Consens des Heinz-Nixdorf-Instituts) zu etablieren. In dieser Arbeit wird die Sprache SysML in der Version 1.6 eingesetzt. Die verschiedenen Sichten auf das Systemmodell sind in SysML in Diagrammen definiert (siehe Abbildung 2.10).

Beispielsweise weisen Strukturdiagramme, das nach Ropohl (2009) bekannte strukturele Konzept mittels unterschiedlicher Diagrammtypen auf. Das

Blockdefinitionsdiagramm definiert die wesentlichen Elemente des Systems als <Blöcke>. Mit der Bezeichnung „<>“ werden in der vorliegenden Arbeit Stereotypen bezeichnet. Jeder <Block> kann ein internes Blockdiagramm haben, was mit <Properties> die Struktur des <Blocks> beschreibt. Die generelle Organisationsstruktur aller Elemente wird ähnlich zu den Ordnersystemen im Betriebssystem Windows realisiert. Die Ordner heißen im SysML Sprachgebrauch <Packet> und werden im Packet-Diagramm modelliert.

Die Elemente vom Stereotyp <Block> lassen sich über die Verbindung <Generalisation> hierarchisch anordnen und setzen somit das von Ropohl (2009) definierte hierarchische Konzept ebenfalls um.

Ropohl (2009) beschreibt das Systemverhalten über ein funktionales Konzept. Dies wird generell und in Bezug zu <Stakeholder> durch Anwendungsfälle bzw. <UseCases> definiert und modelliert. <Stakeholder> sind alle Personen, welche mit dem System interagieren. <UseCases> lassen sich mittels <Aktivitäten> konkretisieren. Das Zustandsdiagramm beschreibt Systemzustände beispielsweise des Computers in „an“ und „aus“. Die Diagramme werden in Kapitel 5 bis 7 eingesetzt und dort beschrieben. Für eine detaillierte Beschreibung der einzelnen Diagramme wird auf Weilkens (2015) verwiesen. Eine Übersicht der relevanten SysML-Elemente und deren Beschreibung befindet sich im Glossar.

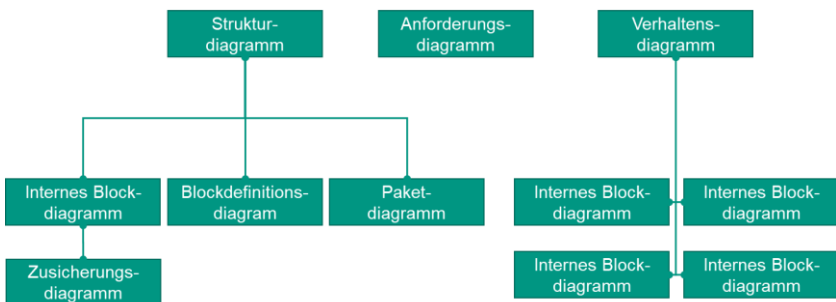


Abbildung 2.10: Diagrammtypen in SysML (Object Management Group, 2019).

Somit zeigt sich, dass UML und hiervon abgeleitet SysML die Analyse des Testbedarfs erzwingt, die Komplexität des Systems erfasst, eine plattformunabhängige Testbeschreibung ermöglicht, die Dokumentation erlaubt sowie die Wiederverwendbarkeit erhöht. (Gühmann, 2002)

Im Folgenden werden etablierte **Methoden** zur ganzheitlichen Modellierung wie CONSENS und SYSMOD eingeführt. Eine von der INCOSE beauftragte Übersicht

der MBSE-Methoden zeigt Estefan (2008). Die Methoden sind auf die Bedarfe der Unternehmen ausgelegt, daher wird auf die Beschreibung in der vorliegenden Arbeit verzichtet. Weiterhin gibt es Methoden zur Modellierung einzelner Aktivitäten oder Elementen wie FAS, VAMOS und OPM. Frameworks wie SPES oder das IPEK-MBSE-Framework führen die Struktur und Methode zusammen.

CONSENS („CONceptual design Specification technique for the ENgineering of mechatronic Systems“) ist eine Sprache und Methode zur Systemmodellierung (Gausemeier, Gaukster & Tschirner, 2013). SYSMOD – Systems Modeling Toolbox beschreibt ein Vorgehen beginnend mit der Modellierung von Systemidee und Basisarchitektur (siehe Abbildung 2.11). Ausgehend hiervon lassen sich Anforderungen, Anwendungsfälle und Systemkontext erstellen, um eine logische und physische Architektur abzuleiten. (Weilkiens, 2015)

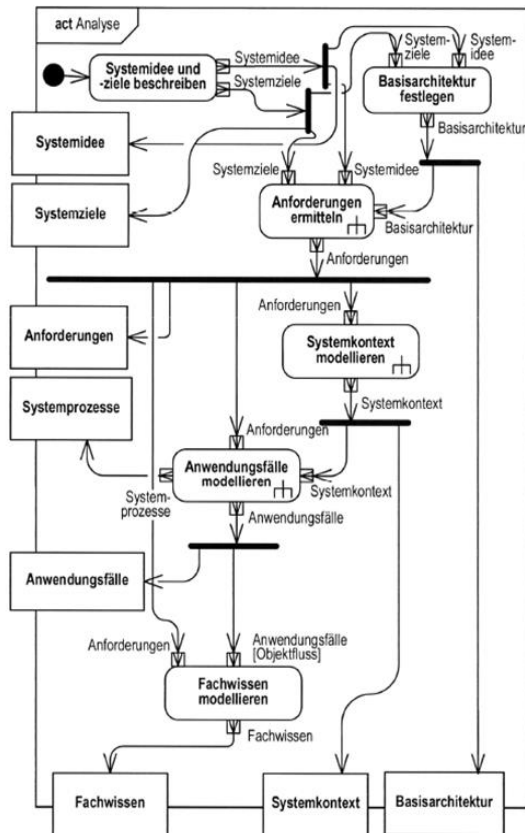


Abbildung 2.11: Wesentliche Aktivitäten zur Systemmodellierung in der Methode SYSMOD dargestellt in einem Aktivitätendiagramm (Weilkiens, 2015).

Im Unterschied zu SYSMOD fokussieren sich weitere Ansätze auf einzelne Aktivitäten oder Elemente der Modellierung. Einer dieser Ansätze ist die Methodik Functional Architectures of Systems (FAS). Sie beschreibt die Modellierung von Funktionen mittels einer Funktionalen Architektur (siehe Abbildung 2.12). Funktionale Blöcke und Gruppen ermöglichen eine systemgenerationsübergreifende Berücksichtigung der Funktionalen Architektur (Dänzer et al., 2014). FAS4M (FAS for Mechanical Engineers) betrachtet hierbei speziell die Schnittstelle für den Maschinenbau mit Hilfe von Elementen und Aktivitäten. Diese ermöglichen eine bidirektionale Verknüpfung von der Funktions-Sicht zu der Expertensicht in CAD (Computer-Aided Design) (Grundel et al., 2014). FAS4X erweitert dies für die Domänen Elektrotechnik und Informatik (Kleiner,

Husung, Mandel, Albers & Behrendt, 2017). Die Methode VAMOS – Variant Modeling with SysML ermöglicht über Stereotypen wie <variant> eine Modellierung mehrerer Produktvarianten in einem Systemmodell (Weilkiens, 2016). ISO/PAS 19459:2015 stellt die Object-Process Methodology (OPM) vor, mit der konzeptionelle Modelle unterschiedlicher Detaillierungsgrade erstellt werden können.

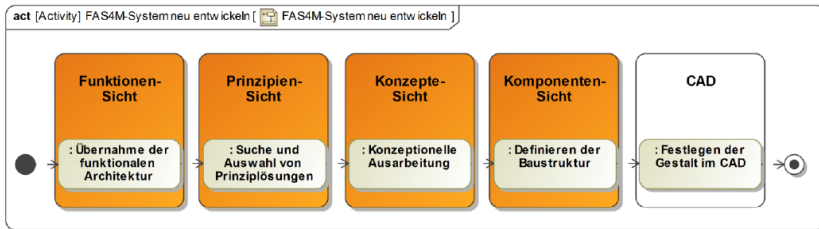


Abbildung 2.12: Wesentliche Aktivitäten in der Methode FAS4M aufgeteilt nach Sichten (Moeser, 2016).

Die SPES Methodologie (Software Platform Embedded Systems) beschreibt ein Framework, welches die Methoden einordnet und eine Gesamtstruktur des Systemmodells mit dessen Elementen und Beziehungen vorgibt (Pohl, Hönninger, Achatz & Broy, 2012). Wesentlich sind hierbei die in Abbildung 2.13 visualisierten Sichten auf das Systemmodell, welche in Anforderungs-, Funktionaler-, Logischer und Technischer Sicht unterschieden werden. Ein Framework für SoS wird vom Verteidigungsministerium der USA beschrieben (Baldwin, 2008). Es beinhaltet einen SoS-Guide mit sieben Aktivitäten: (1) übersetzen von Ressourcenziele, (2) verstehen von System und Beziehungen, (3) beurteilen von Leistung, (4) entwickeln einer SoS-Architektur, (5) überwachen und bewerten von Änderungen, (6) adressieren von Anforderungen und Ergebnismöglichkeiten, (7) ausführen von Upgrades. In dieser Arbeit werden aufgrund des Fokus auf der Validierung des SoS V2I die Aktivitäten (2), (4), (6), (7) betrachtet.

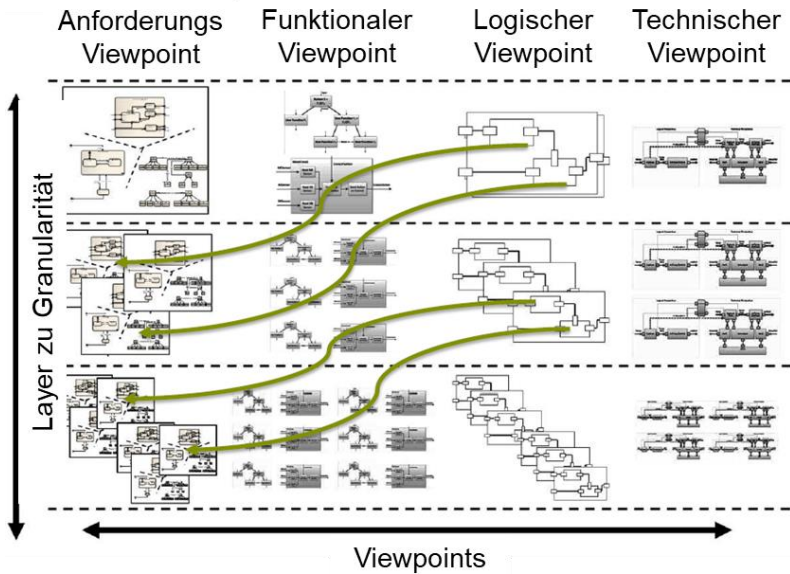


Abbildung 2.13: Viewpoints und Layer im SPES Framework (Pohl et al., 2012).

Basierend auf Arbeiten von Holt und Perry (2018) beschreiben Mandel, Martin und Albers (2022) ein IPEK-MBSE-Framework, mit dem Ziel eine hohe Akzeptanz bei den Nutzern zu realisieren. Kategorien des Framework sind direkt mit Aktivitäten verlinkt, welche von Nutzern als verständlich wahrgenommen werden. Aktivitäten der Modellerstellung lassen sich in Problemraum und Lösungsraum aufteilen. Der Problemraum enthält: Systemeingrenzung, Stakeholder-, Anforderungs- und UseCase-Bestimmung. Der Lösungsraum enthält neben zusätzlich abgeleiteten Systemanforderungen die Funktions-, Konzepte-, Teilsystem-Bestimmung. (Mandel, Martin & Albers, 2022)

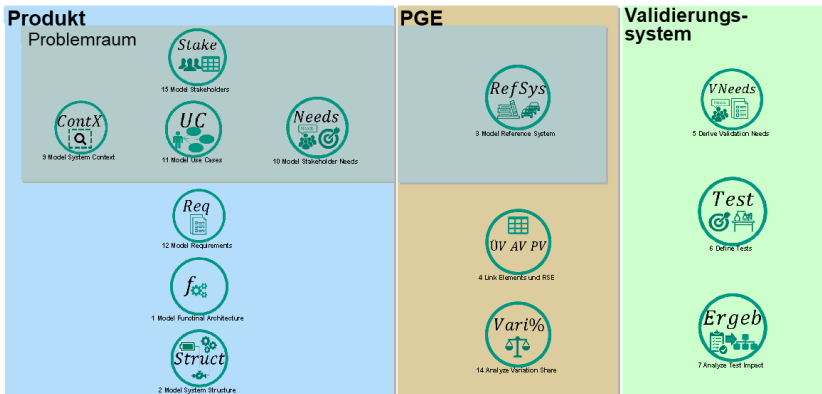


Abbildung 2.14: IPEK-MBSE-Framework (Mandel, Martin & Albers, 2022).

Die beschriebenen Methoden eignen sich jedoch nur teilweise für die modellbasierte Unterstützung der Validierung von SoS. Umfangreiche Methoden und Frameworks haben nicht den Schwerpunkt in SoS. Weitere Arbeiten untersuchen einzelne Sichten in MBSE speziell auf den Einsatz von SoS. Insbesondere für Schnittstellen in einem SoS existieren Ergebnisse zu optimierten Informationsflüssen (Luna, Lopes, Tao, Zapata & Pineda, 2013), speziellen Validierungsrichtlinien (Silingas & Butleris, 2009) und Schnittstellenverträgen (Bryans, Fitzgerald, Payne, Miyazawa & Kristensen, 2014; Faldik, Payne, Fitzgerald & Buhnova, 2017). Somit bestehen fragmentierte Methoden zu SoS oder Methoden ohne Bezug zu SoS und dessen Validierung. Somit erfolgt bislang keine über Systemebenen und Domänen durchgängige SoS Betrachtung. Dies sind Herausforderungen, die das Advanced Systems Engineering adressiert.

2.1.4 Advanced Systems Engineering

Neben dem zuvor eingeführten „Systems Engineering“ besteht das „Advanced Systems Engineering“ aus den zwei weiteren Begriffen „Advanced Systems“ und „Advanced Engineering“ und kann als eine Art „Zielbild“ für das zukünftige Systems Engineering verstanden werden. Die drei Begriffe „Advanced Systems“, „Systems Engineering“ und „Advanced Engineering“ werden als Handlungsfelder für das Zielbild verstanden und sind notwendig aufgrund von Megatrends wie Digitalisierung und eine nachhaltige Wertschöpfung von morgen (siehe Abbildung 2.15). (Albers & Lohmeyer, 2012; Albers, Dumitrescu et al., 2022; Albers, 2023; Dumitrescu, Albers, Riedel, Stark & Gausemeier, 2021).



Abbildung 2.15: Die drei Handlungsfelder des ASE (Dumitrescu et al., 2021).

Basierend auf der Systemtheorie und ihrer Weiterentwicklung sowie in Beispielen und Anwendungsfällen werden Advanced Systems beschrieben (Albers, 2023, S. 14–21). V2I gilt hierbei als Advanced System und wird in Kapitel 2.3 beschrieben. Advanced Systems weisen die Perspektiven Autonomie, Vernetzung, Interaktion und Smart Services auf (Dumitrescu et al., 2021, S. 10).

Advanced Systems beinhaltet somit mehrere Systemarten wie Internet of Things (IoT) und Cyber Physical Systems (CPS). Diese lassen sich teilweise nicht trennscharf formulieren. Demzufolge entwickelten Henshaw et al. ein Venn-Diagramm, welches die verschiedenen Systembegriffe ein- und abgrenzt. (Vgl. Henshaw, 2016)

Die verschiedenen Systembegriffe sind in Abbildung 2.16 in einem Venn-Diagramm dargestellt. Das aus der Mengenlehre bekannte Diagramm veranschaulicht die Mengen eines Begriffs und die Abhängigkeiten zwischen verschiedenen Mengen. So wird nach Henshaw (2016) IoT als Teil von SoS angesehen, sodass jedes IoT auch ein SoS ist.

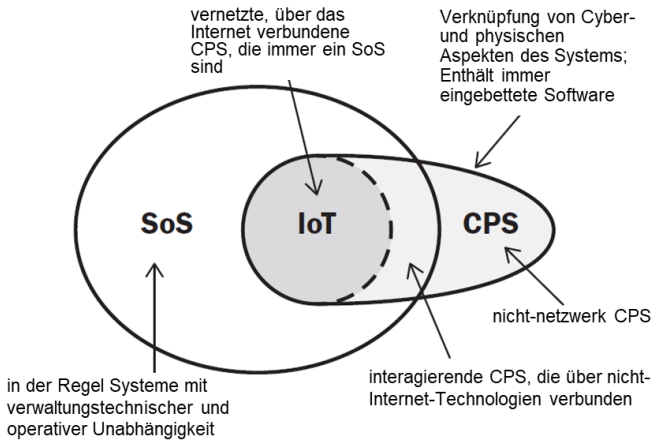


Abbildung 2.16: Venn-Diagramm mit den drei Systembegriffen SoS, IoT und CPS (übersetzt nach Henshaw, 2016).

Advanced Engineering beschreibt das Zielbild zukünftiger Methoden, Prozesse, Werkzeuge und Organisationen, welche sich aufgrund von Fortschritten und Herausforderungen anpassen. Die INCOSE sieht insbesondere die drei zentralen gesellschaftlichen Herausforderungen Klimawandel, Pandemische Antwort und weltweiter Zugang zum Gesundheitswesen als Treiber für Systems Engineering (Friedenthal et al., 2021). In der Studie zu Advanced Systems Engineering (Albers, Dumitrescu, Gausemeier, Riedel & Stark, 2018) wurden Experten zu den Fortschritten in digitalen Technologien und innovativen Methoden und deren Implikationen befragt. Digitale Technologien beinhalten im Rahmen der Leistungsstanderhebung die drei Themen:

- Digitale Durchgängigkeit und Produktlebenszyklusmanagement,
- Digitale Zwillinge und Betriebsdatennutzung im Engineering sowie
- Künstliche Intelligenz und Assistenzsysteme.

In der vorliegenden Arbeit wird das erste Thema der digitalen Durchgängigkeit und Produktlebenszyklusmanagement untersucht. Die Fortschritte in innovativen Methoden wurden in der Leistungsstanderhebung anhand der drei Themen betrachtet:

- Agilität im Engineering,
- Kreativität im Engineering,
- Systemgenerationsentwicklung im Engineering.

Eine detaillierte Beschreibung befindet sich in Albers (2023).

Mittels Advanced Engineering sollen somit bestehende Engineering-Ansätze kreative, agile und digitale Aspekte berücksichtigen. Aus diesem Grund dient Advanced Engineering der Zusammenfassung von Engineering-Ansätzen verschiedener Disziplinen für Aktivitäten und Aspekte der Marktleistung. (Dumitrescu et al., 2021)

Im Kontext der Digitalisierung im Engineering greift die Studie Advanced Systems Engineering das Thema Safety & Security by Design auf (Dumitrescu et al., 2021, S. 47). Wie insbesondere die Betriebssicherheit von Advanced Systems beim Design berücksichtigt werden kann, wird in der vorliegenden Arbeit untersucht. Daher wird nachfolgend das SoS Vehicle-to-Infrastructure als Advanced System eingeführt und bestehende Validierungsansätze insbesondere in Bezug zur Funktionalen Sicherheit berücksichtigt.

2.1.5 Zwischenfazit

Die etablierte Systemtheorie beschreibt Produkte als Systeme für eine vollständige Betrachtung und Analyse. In der Systemtheorie ermöglicht das Model-Based Systems Engineering die konsistente Beschreibung mit einem zentralen Systemmodell. Folgende Aussagen fassen das Kapitel 2.1 zusammen:

- Im **Systems Engineering** werden Produkte mit Hilfe der Systemtheorie betrachtet, wobei die neuen Herausforderungen wie die Validierung von komplexen Systemen zu der Einführung des **Advanced Systems Engineerings** führte.
- Aus dem anfänglichen Systembegriff haben sich spezifischere Begriffe wie System of Systems und cyberphysische Systeme entwickelt. **Die Implikationen der vielfältigen Begriffe gilt es zu verstehen und bedarfsgerecht zu nutzen.**
- System of Systems ist Teil der standardisierten Systemtheorie, wobei die **Validierung von SoS eine Herausforderung darstellt.**
- **Model-Based Systems Engineering** ermöglicht die Beschreibung der Systementwicklung in einem zentralen Systemmodell, jedoch fehlt eine ganzheitliche, durchgängige Betrachtung der **Validierung speziell für SoS.**

Die vorliegende Arbeit verwendet den Systembegriff und betrachtet Produkte als Systeme im Kontext der Validierung. Hierzu werden grundlegende Modelle und Ansätze in der Produktentwicklung genutzt, welche im Folgenden eingeführt werden.

2.2 Validierung in der Produktentwicklung

Die Validierung ist nach Albers et al. die zentrale Aktivität der Produktentwicklung (vgl. Albers, Braun & Muschik, 2010a). Demzufolge wird in Kapitel 2.2.1. die Produktentwicklung innerhalb des Produktentstehungsprozesses (PEP) vorgestellt, welche die Aktivität der Validierung einordnet und wesentliche Modelle der Produktentwicklung beinhaltet. In Kapitel 2.2.2 wird die Validierung mit wesentlichen Definitionen und Ansätzen eingeführt.

2.2.1 Grundlegende Produktentwicklungsmodelle der KaSPro

Die vorliegende Arbeit ist in der Karlsruher Schule für Produktentwicklung (KaSPro) eingeordnet und baut darauf auf. Daher werden einige grundlegende Modelle der KaSPro beschrieben.

Für die Betrachtung der Produktentwicklung existieren Modelle, die eine systematische Beschreibung des PEP ermöglichen. Als übergeordnetes Gedankenkonstrukt ist das ZHO-Modell zu verstehen, das die Produktentwicklung mit den drei Systemen Ziel-, Handlungs- und Objektsystem beschreibt (Albers, Lohmeyer & Ebel, 2011). Das Zielsystem enthält die für die Produktentwicklung relevanten Ziele, Anforderungen und Randbedingungen. Als Vorarbeit wird das Produktprofil zu Beginn der Produktentwicklung erstellt (Richter, Rapp, Kurtz, Romanov & Albers, 2019), welches den Produktzweck, Eigenschaften und eine mögliche Marktpositionierung enthält. Das Handlungssystem beinhaltet Aktivitäten, um die Zielsystemelemente zu erreichen. Hierzu ist es notwendig im Objektsystem Elemente wie Prototypen, Dokumente, Modelle und insbesondere das marktfähige Produkt zu erstellen. Die drei Systeme stehen im ständigen Austausch, was in Abbildung 2.17 dargestellt ist. Durch Analyse und Synthese werden die Systeme iterativ weiterentwickelt.

„Die **Analyse** beschreibt eine Handlung, die das Verstehen eines existierenden Systems bezweckt. Das Resultat einer Analyse ist Erkenntnis.“ (Lohmeyer, 2013)

„Die **Synthese** beschreibt eine Handlung, die das Erschaffen eines bis dato noch nicht existierenden Systems bezweckt. Das Resultat einer Synthese ist ein Ziel oder ein Objekt.“ (Lohmeyer, 2013)

Ausgehend vom Zielsystem werden mittels der Analyse von Zielsystemelementen Aktivitäten zur Beschreibung des Lösungsraums angestoßen. Auf Grund von Syntheseaktivitäten wie der Konstruktion werden anschließend

Objektsystemelemente wie z.B. ein Steuergerät erstellt. Wiederum die Analyse des Steuergerätes führt zu (neuen) Erkenntnissen, welche in einer Wissensbasis festgehalten werden. Die Erkenntnisse führen auch zur Prüfung der bisherigen Zielsystemelemente, sodass dies zu Anpassungen oder Bestätigungen der Elemente führen kann. Abbildung 2.17 verdeutlicht bei diesem iterativen Vorgehen die Richtung vom Ziel- zum Objektsystem, was als Kreation verstanden werden kann. Wiederum entgegengesetzt ist die Validierung, welche ausgehend von Objektsystemelementen die Erfüllung von Zielsystemelementen, wie beispielsweise Bedarfe, prüft.

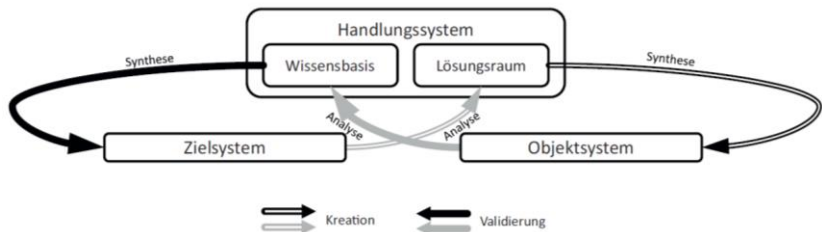


Abbildung 2.17: Das erweiterte ZHO-Modell (Albers et al., 2011).

Zur Beschreibung der Produktentstehung, die die Produktentwicklung enthält, erarbeiteten Albers et al. (2010a) das „Integrierte Produktentstehungsmodell“ (iPeM). Es beschreibt als Metamodell wesentliche Elemente und Beziehungen und basiert auf dem ZHO-Modell. Das iPeM enthält neben der Produktentstehung einer momentanen Generation G_n die Entwicklung der darauffolgenden Generationen G_{n+1} , G_{n+2} , ..., des Validierungssystems, des Produktionssystems und der Strategie für eine erfolgreiche Produktentstehung. Diese Elemente bilden die Layer im iPeM. Alle diese Layer verfügen über dasselbe für die Produktentstehung notwendige Zielsystem. Alle Layer haben ein individuelles Handlungssystem. Beispielsweise enthält das Validierungssystem die spezifische Aktivität „Validierungsziel objektivieren“ (Yan, 2020, S. 119). Das Handlungssystem wird in Abbildung 2.18 mit den Aktivitäten des jeweiligen Layers dargestellt, sowie Aktivitäten der Problemlösung, eines gemeinsamen Ressourcensystems zur Realisierung der Aktivitäten und eines Phasenmodells. Das Phasenmodell beschreibt die zeitliche Dimension der Aktivitäten in Phasen der Produktentstehung ähnlich zu einem Projektzeitplan. Es bildet die Aktivitäten auf der Zeitachse ab und beschreibt somit die Produktentstehung dynamisch (vgl. Albers, Reiß, Bursac & Richter, 2016, S. 101). Die Aktivitäten führen je Layer zu unterschiedlichen Objektsystemelementen, wie im Beispiel des Validierungssystems zu Prüfständen.

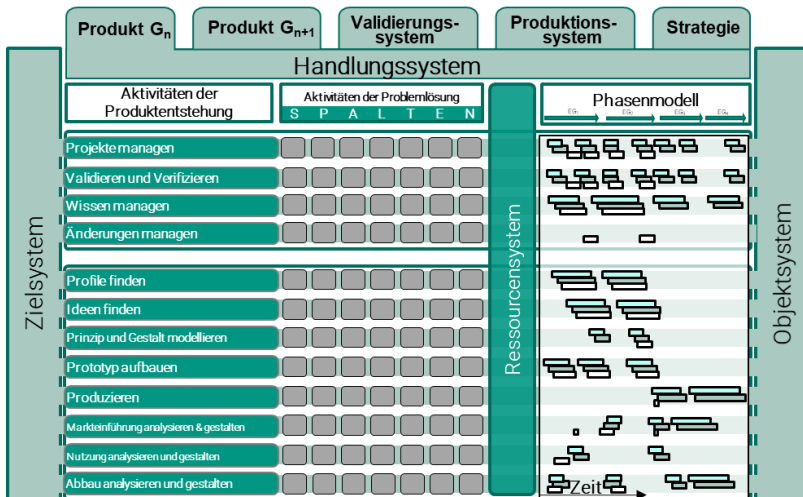


Abbildung 2.18: iPeM – Integriertes Produktentstehungsmodell mit der Aktivität Validieren und Verifizieren und dem Layer Validierungssystem (Albers et al., 2010a; Albers, Reiß et al., 2016).

Der in Abbildung 2.18 dargestellte Layer der momentan in Entwicklung befindlichen Systemgeneration (Produkt G_n) wird durch vorangegangene Generationen unterstützt. Die Entwicklung von Systemen in Generationen beschreibt das Modell der SGE - Systemgenerationsentwicklung (Albers & Wintergerst, 2014; Albers et al., 2019; Albers & Rapp, 2022; Albers, Kürten et al., 2022). Es beinhaltet zwei Hypothesen:

1. Jedes System wird auf der Grundlage eines Referenzsystems entwickelt. Elemente des Referenzsystems stammen aus bestehenden oder bereits geplanten soziotechnischen Systemen und der zugehörigen Dokumentation und dienen als Grundlage und Ausgangspunkt für die Entwicklung eines neuen Systems.
2. Die Elemente eines neuen Systems werden auf der Grundlage von Referenzsystemelementen mit Hilfe der Aktivitäten der Übernahmevariation (ÜV), der Ausprägungsvariation (AV) und der Prinzipvariation (PV) definiert.

Das Referenzsystem enthält Elemente wie beispielsweise die Generation der Vorgänger- oder Konkurrenzprodukte oder Teile davon basierend auf Referenzprodukten (siehe Abbildung 2.19). Die Idee der Systemgenerationen kann sogar auf eine Generation konzentriert werden. Dabei werden mehrere Entwicklungsgenerationen des nicht endgültigen Systems entwickelt, wie z.B.

mehrere Prototypen innerhalb einer Systemgeneration. Neben einer vollständigen Übertragung von Elementen in der ÜV beschreibt eine AV eine Übertragung der Hauptlösung durch Änderung einiger Attribute wie Parameterwerte. Bei einer PV verwendet die neue Generation ein neues Prinzip, wie beispielsweise ein neues Berechnungsprinzip für die Verkehrsplanung. Selbst den Sonderfall eines fehlenden Vorgängermodells beschreibt das Modell der PGE durch Referenzsystemelemente für die erste Produktgeneration (siehe Albers et al. (2020, S. 10)).

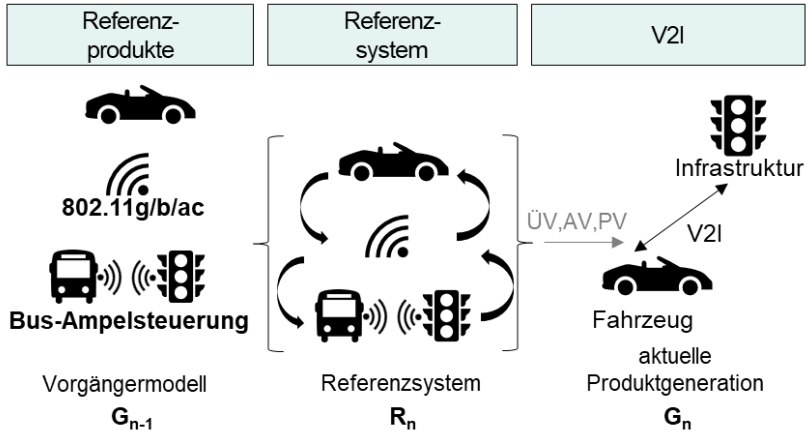


Abbildung 2.19: Exemplarische Produktgenerationen und exemplarisches Referenzsystem für V2I im Modell der PGE – Produktgenerationsentwicklung (Struktur angelehnt an Albers, Bursac & Wintergerst, 2015).

Neben den Layern unterschiedlicher Generationen (G_n und G_{n+1} in Abbildung 2.18) ist in der vorliegenden Arbeit der Layer des Validierungssystems relevant, der im nachfolgenden Kapitel vorgestellt wird.

2.2.2 Validierung in der Produktentwicklung

Analog zur übergreifenden Produktentwicklung wird die Validierung ebenfalls durch Elemente der KaSPro beschrieben und durch Analyse- und Synthese-Schritte umgesetzt. Auf Basis des ZHO-Modells wird die Validierung mit deren drei

Die **Validierung** beschreibt den Abgleich von Objektsystemelementen zu Bedarfen von Stakeholdern. Validierung besteht hierbei aus drei grundlegenden Aktivitäten: Bewertung, Objektivierung, Verifikation (Albert Albers, Behrendt, Klingler & Matros, 2016)

grundlegenden Aktivitäten Verifikation, Bewertung und Objektivierung definiert und in Abbildung 2.20 in das ZHO-Modell eingeordnet:

Objektsystemelemente wie Modelle und Prototypen werden analysiert. Dies beinhaltet die Bewertung durch Stakeholder. Beispielsweise werden Tests ausgeführt und die Testergebnisse bewertet. Ob die Ergebnisse, die im Zielsystem stehenden Anforderungen erfüllen, ist Teil der Verifikation. Nach der Bewertung erfolgt die Objektivierung der Ergebnisse mit Hilfe der Synthese. Die Testergebnisse werden interpretiert und es können Schlussfolgerungen in das Zielsystem aufgenommen werden. Beispielsweise wird eine Anforderung konkretisiert, aufgrund von neuen Erkenntnissen zu sicherheitskritischen Betriebsbereichen. (Albers, Behrendt, Klingler & Matros, 2016)

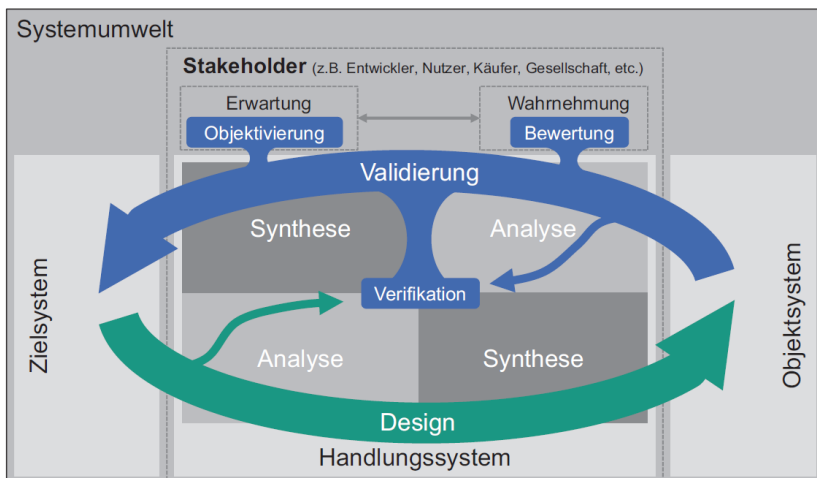


Abbildung 2.20: Design und Validierung im Produktentstehungsprozess nach dem ZHO-Modell (Albers, Behrendt et al., 2016).

Die Beschreibung der Validierung kann neben der Beschreibung durch das ZHO-Modell das V-Modell genutzt werden.

Das V-Modell stellt eine Vorgehensweise für den konkreten Einsatz der Validierung in der Produktentwicklung dar. Es zeigt durch den V-förmigen Aufbau inkrementelle Entwicklungsschritte in der zeitlichen Dimension von links nach rechts und im Detaillierungsgrad zunehmend von oben nach unten auf (siehe Abbildung 2.21). Aktivitäten auf der linken Seite werden durch entsprechende Aktivitäten der rechten Seite verifiziert und im letzten Schritt gegenüber den Kundenanforderungen validiert.

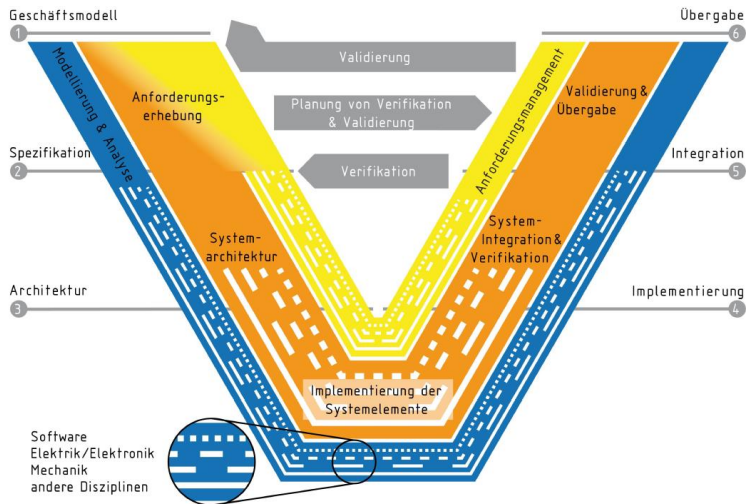


Abbildung 2.21: V-Modell für die Entwicklung mechatronischer Systeme und Cyber-Physical Systems (VDI/VDE 2206).

Das in der Fahrzeugentwicklung verbreitete V-Modell suggeriert eine zeitlich und im Detaillierungsgrad starre Produktentwicklung, die aufgrund von Änderungen häufig nicht einzuhalten ist (Christie, 2008). Im Gegensatz dazu ermöglicht das iPeM eine generische Beschreibung von Aktivitäten, wobei die detaillierte Anordnung anhand von Dimensionen wie der Zeit Unternehmensindividuell erfolgt.

Die Gesamtheit an Elementen in der Validierung beschreibt das Validierungssystem, welches folgendermaßen definiert ist (vgl. Albers, Reiß et al., 2016, S. 104):

Im **Validierungssystem** sind alle Elemente enthalten, welche die Erfüllung von Validierungsbedarfen des zu entwickelnden Produktes ermöglichen. Das Validierungssystem umfasst die gesamte Produktlebensdauer und alles zur Durchführung von Validierungsaktivitäten im Produktentstehungsprozess.

Die Elemente können über den IPEK-X-in-the-Loop-(IPEK-XiL)-Ansatz beschrieben werden. Der IPEK-XiL-Ansatz basiert auf den Ansätzen Modell-, Software-, Hardware-in-the-Loop und erweitert ihn auf mechatronische Systeme. Dieser Ansatz beschreibt das zu entwickelnde System als System-in-Development (SiD) oder System-under-Investigation (SuI). SiD beschreibt ein in der Entwicklung befindliches System, wobei die Entwickelnden Zugriff auf White Box-Modelle haben, wohingegen beim SuI das System nur als Black Box-Modell untersucht wird. Dabei

wird das "X" als das SiD oder Sul mit seinen angeschlossenen Systemen sowie die Umgebung betrachtet (siehe Abbildung 2.22). Das SiD oder Sul kann virtuell, physisch oder gemischt virtuell-physisch sein. Abhängig von diesem Reifegrad wird das Rest-Fahrzeug definiert. Es beinhaltet das notwendige Modell für eine Interaktion mit dem SiD. Das System Fahrzeug interagiert mit anderen Systemen, die in Abbildung 2.22 in Fahrer¹ und Umwelt dargestellt sind. Die Systeme interagieren für verschiedene Fahrmanöver und Testfälle. (Albers, Behrendt et al., 2016)



Abbildung 2.22: IPEK-X-in-the-Loop-Ansatz zur Validierung des Systems mit dessen Umwelt und dessen vernetzten Systemen (Albers & Düser, 2008; Albers, Behrendt et al., 2016).

Das SiD kann in unterschiedlichen Detaillierungsgraden dargestellt werden. Abbildung 2.23 zeigt das aus dem Ansatz hergeleitete Framework, welches das SiD in unterschiedlichen Ebenen enthalten kann. So können Systeme bis hin zu Wirkkontakten (WFP = Wirkflächen Paar) im IPEK-XiL-Ansatz untersucht werden. (Albers, Behrendt et al., 2016)

¹ In der vorliegenden Arbeit wird darauf verzichtet, alle Personenbezeichnungen z.B. in weiblicher und männlicher Form zu nennen. Die männliche Form gilt in allen Fällen, in denen dies nicht explizit ausgeschlossen wird, für alle Geschlechter.

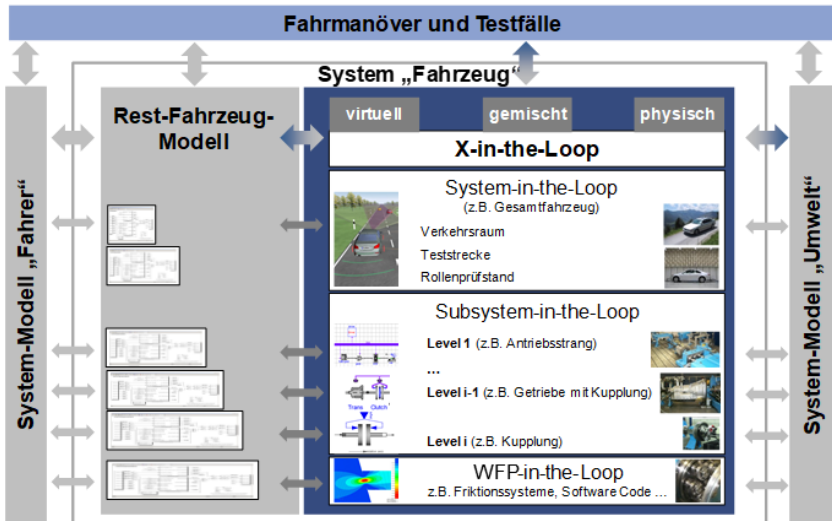


Abbildung 2.23: Elemente und Zusammenhänge des IPEK-X-in-the-Loop-(XiL)-Framework (Albers & Düser, 2008; Albers, Fischer, Behrendt & Lieske, 2014; Albers, Behrendt et al., 2016).

Eines der wesentlichen Elemente im Validierungssystem ist die Validierungsumgebung, welche wie folgt definiert wird:

„Die **Validierungsumgebung** bietet die Voraussetzungen für die Durchführung von Validierungsaktivitäten, bei denen ein (Teil-) System im Kontext des übergeordneten Systems und seiner Connected Systems validiert wird. Eine Validierungsumgebung ist somit eine konkrete Ausprägung eines Handlungssystems für die Validierung im Sinne von Methoden und des Ressourcensystems für mindestens eine Kombination aus Produkt und einem Validierungsziel zu einem bestimmten Zeitpunkt im Produktlebenszyklus“ (Albert Albers, Fischer, Klingler & Behrendt, 2014; Albert Albers, Mandel, Yan & Behrendt, 2018)

Abbildung 2.24 stellt dar, dass der Produkt-Layer des iPeM über einen Bedarf mit dem Validierungssystem-Layer verknüpft ist. Zudem verdeutlicht die Abbildung die wesentlichen Elemente der Validierung und ihre Beziehungen untereinander.

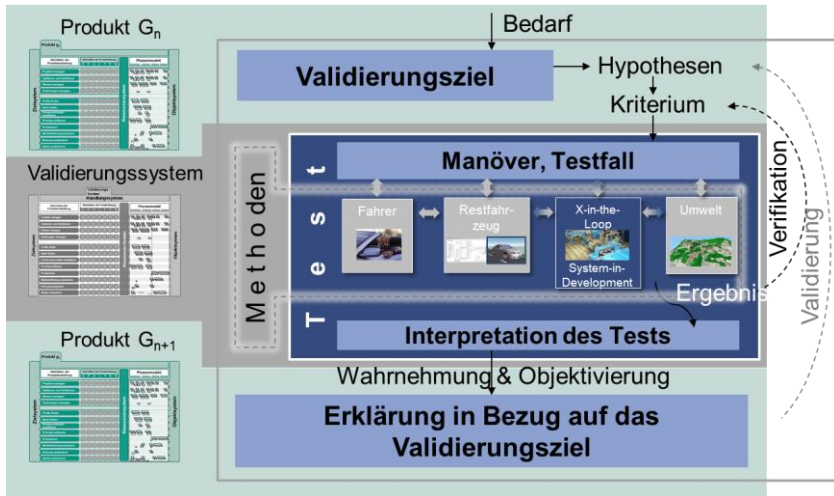


Abbildung 2.24: IPEK-XiL-Ansatz und die Beziehung zum Produktentwicklungsprozess (übersetzt nach Mandel et al., 2020).

Die Verknüpfung zwischen Produkt- und Validierungssystem-Layer auf Basis des iPeM kann nach dem Push- oder Pull-Prinzip erfolgen. Beim Push-Prinzip werden zu Beginn Aktivitäten der Produktentwicklung aus dem Handlungssystem ausgeführt. Diese führen dann zu Aktivitäten des Validierungssystems, welche „gepusht“ werden. Im Gegensatz hierzu werden beim Pull-Prinzip die sekundären Aktivitäten „gepullt“, um frühzeitig und kontinuierlich zusammen mit dem Produktsystem das Validierungssystem zu entwickeln. (Albers, Matros, Behrendt & Jetzinger, 2015)

Beispielsweise werden Untersuchungsumgebungen während der Produktentwicklung eingesetzt, um im Sinne des Pull-Prinzips frühzeitig Validierungsaktivitäten durchzuführen. Eine in dieser Arbeit verwendete Umgebung stellt das Live-Lab dar. Es ermöglicht die Erforschung und insbesondere Evaluierung von Methoden der Produktentwicklung in einem Entwicklungsprozess. Im Rahmen von Live-Labs führen Studierende des Maschinenbaus, Wirtschaftsingenieurwesens, der Mechatronik, Elektrotechnik und Informationstechnik, Technischen VWL und Informatik sechs Monate ein Entwicklungsprojekt mit Industriepartnern durch. Ziel ist in einer Umgebung durch gestaltbare Randbedingungen Methoden der Produktentwicklung für die Praxis in Industrieunternehmen zu evaluieren. (Walter, Albers, Haupt & Bursac, 2016)

Wesentliche Vorteile sind die Evaluierung durch Vergleichsgruppen und das realitätsnahe Verhalten der Probanden (Walter et al., 2016). Nachteile sind eine

mittlere externe Validität durch den teilweise unklaren Transfer der untersuchten Methode in die Industrie (Albers, Walter, Wilmsen & Bursac, 2018). Die Nutzung der Live-Labs in der vorliegenden Arbeit wird in Kapitel 4.2 beschrieben.

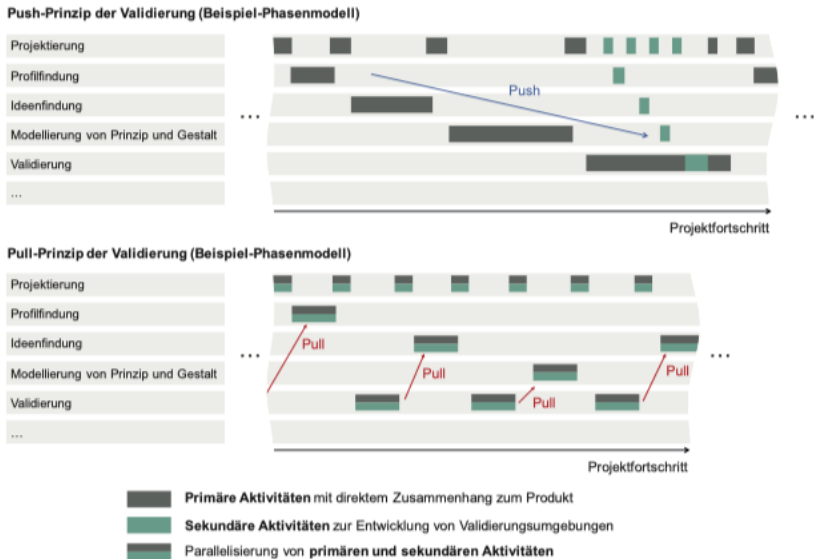


Abbildung 2.25: Pull- und Push-Prinzip der Validierung am Beispiel der Hybridantriebsentwicklung (Albers, Matros et al., 2015).

2.2.3 Zwischenfazit

Ausgehend von dem Produktentstehungsprozess beschreibt Kapitel 2.2 grundlegende Modelle, Aktivitäten und Elemente in der Produktentwicklung. Die vorliegende Arbeit fokussiert hierbei auf die Validierung als zentrale Aktivität der Produktentstehung (Albers, 2010). Folgende Aussagen fassen das Kapitel 2.2 zusammen:

- Die Validierung ist eine **zentrale Aktivität im Produktentstehungsprozess**. Sie kann mit Hilfe des ZHO-Modells beschrieben werden.
- Validierung wird als Aktivität im Produkt-Layer des iPeM eingesetzt und führt zu Aktivitäten und Objekten im Validierungssystem Layer.
- Der **IPEK-XiL-Ansatz** ermöglicht die Validierung eines SiD und Sul unter Berücksichtigung der Umwelt und des Systemkontextes.

Die im ZHO-Modell, iPeM und IPEK-XiL-Ansatz vorkommenden Systeme sind Teil der Systemtheorie. Nachfolgend wird untersucht, wie durch die Systemtheorie Ansätze und Modelle des Systems Engineering für die Validierung in der Produktentwicklung genutzt werden können.

2.3 Beschreibung und Validierung von Vehicle-to-Infrastructure

Bereits M. Maier (1996) beschrieb das Transportsystem als beispielhaftes SoS. Im Zuge vom Smart Cities und Smart Mobility wird häufig das Intelligent Transport System als Enabler genannt, welches jede Form von Transport wie Bahn und Fahrrad als System beschreibt. Dies verdeutlichen verschiedene Größen der Systeme, welche im Sinne einer Zwiebelschale Sub- oder Supersystem sein können. Intelligente Transportsysteme können mit Vehicle-to-Everything (V2X) realisiert werden, welches die in Abbildung 2.26 gezeigten Systeme beinhalten kann.

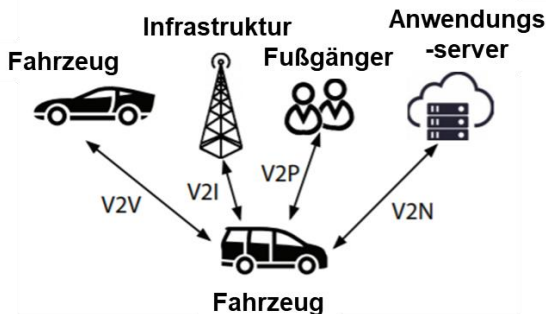


Abbildung 2.26: Übersicht der vier V2X Kommunikationsarten V2V, V2I, V2P, V2N (Hsu, Wang, Zhou & Shawkat, 2016).

In der vorliegenden Arbeit wird das spezielle SoS V2I untersucht, das sich aus der folgenden Definition ergibt.

Das Mobilitätssystem als Advanced System beinhaltet nun Systeme wie V2I bestehend aus Fahrzeug und Infrastruktur. Die vorliegende Arbeit fokussiert dabei auf V2I und nutzt hierbei das Verständnis der Commission Mobility and Transport (2016), des EUROPEAN PARLIAMENT und COUNCIL (2010, S. 14) und Albers (2023):

Das soziotechnische Advanced System **V2I** beinhaltet die Systeme Fahrzeug und Infrastruktur (z.B. mit Verkehrssignalanlage) sowie die kabellose Kommunikation zwischen den zwei Systemen. Hierbei kann V2I als SoS bezeichnet werden, da die Systeme Fahrzeug und Infrastruktur unabhängig voneinander verwaltet und betrieben werden können und Emergenzeffekte auftreten können.

V2I wird als SoS und Advanced System beschrieben. Im Sinne des Advanced Systems verfügt V2I über autonome, dynamisch vernetzte Systeme (vgl. Dumitrescu et al., 2021, S. 51). Neben den technischen Herausforderungen wird im V2I auch die Rolle des Menschen berücksichtigt, weshalb V2I als soziotechnisches System verstanden werden muss.

V2I bietet die Möglichkeit hochautomatisiertes Fahren zusammen mit anderen Systemen wie V2V und Radar zu realisieren (Bundesministerium für Verkehr und digitale Infrastruktur [BMVI], 2017). Zudem lassen sich Bedarfe nach Sicherheit, Nachhaltigkeit, Effizienz und Komfort erfüllen (vgl. Bezai, Medjdoub, Al-Habaibeh, Chalal & Fadli, 2021).

In Bezug zu V2X hat EURO NCAP eine Roadmap veröffentlicht, welche ab 2025 erste V2X Anwendungsfälle beinhaltet (Euro NCAP, 2017). Weitere Organisationen, wie IIHS (Cox, Mueller & Cicchino, 2023) und NHTSA (2022), forschen an der Einführung von V2X Lösungen. Erste Anwendungen, insbesondere zu V2V existieren bereits auf dem Markt, beispielsweise der VW Golf 8 (Thiemel, 2020; Volkswagen AG, 2020). Vorwiegend werden auf Testfeldern kommerzialisierte Lösungen untersucht (Bundesministerium für Digitales und Verkehr [BMDV], 2020). Mit Gesetzen im Bund und EU-Richtlinien wird nun auch der regulatorische Rahmen geschaffen (Bundestag, 2021; Europäische Kommission, 2020). In diesem Rahmen sind 23 <UseCases> für V2I realisierbar (siehe Abbildung 2.27).

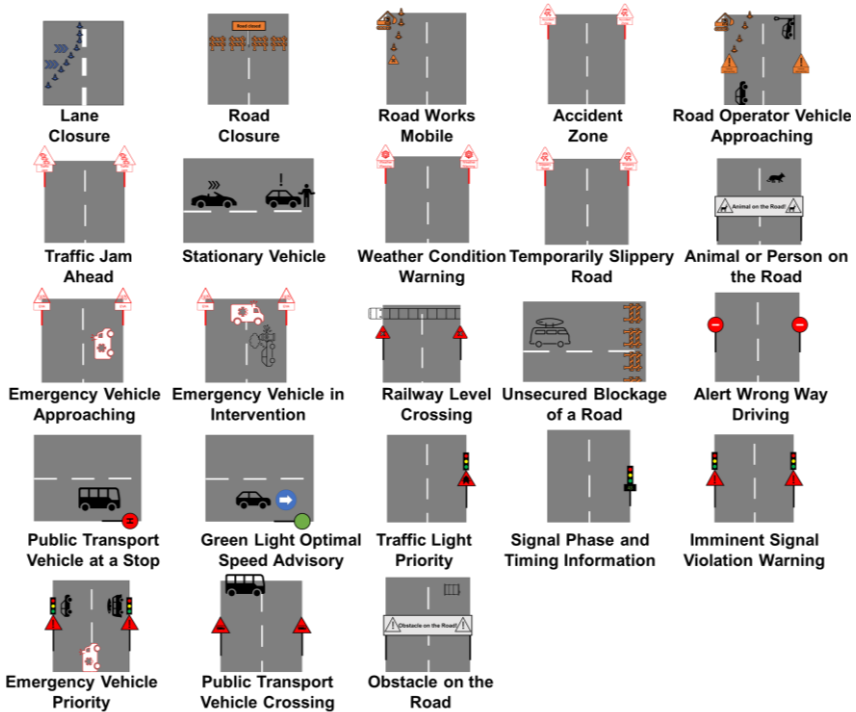


Abbildung 2.27: Übersicht der 23 wesentlichen V2I UseCases (vgl. CAR 2 CAR Communication Consortium, 2021; vgl. Connected Motorcycle Consortium, 2020).

V2I und dessen Validierung wird im Folgenden im Sinne des Systems Engineering vorgestellt.

2.3.1 Systembeschreibung von V2I

Abgeleitet aus den V2I UseCases werden zu Beginn die Stakeholder für V2I beschrieben und ausgehend von diesen das SoS V2I erläutert (siehe Abbildung 2.28). Dementsprechend werden im ersten Schritt die wesentlichen Standards und Organisationen in Bezug zu V2I eingeführt, sowie weitere Stakeholder wie Projektkonsortien mit deren wesentlichen Ergebnissen in Bezug zu V2I genannt. Die Stakeholder in dieser Arbeit sind unterteilt nach Rahmumgebung, Herstellung und Betrieb.

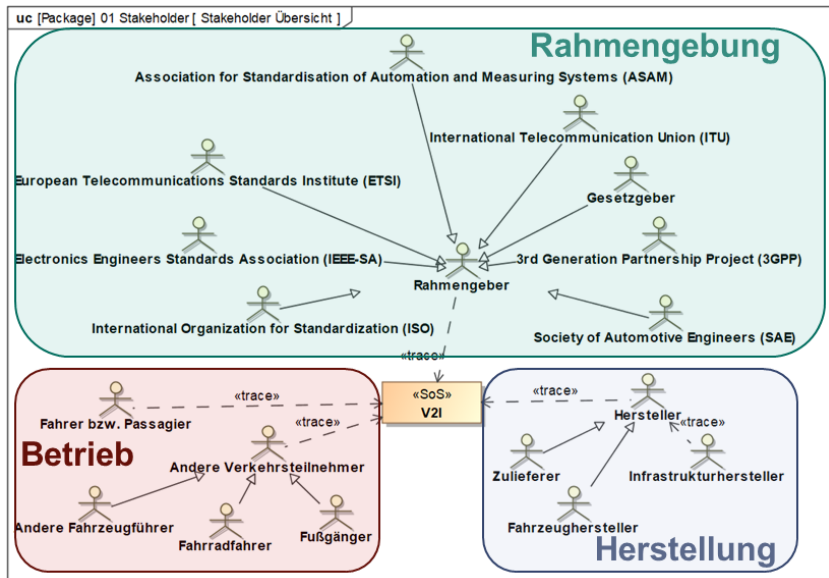


Abbildung 2.28: Stakeholder im Kontext von V2I geclustert nach ihren Kernaufgaben (vgl. ASAM e.V., 2022b; vgl. ETSI TS 103 301; vgl. BS ISO/IEC/IEEE 21841:2019; vgl. McKinley, 2021; vgl. SAE J3016_202104; vgl. Villarreal-Vasquez, Bhargava & Angin, 2017).

Rahmgebung

Die Forschung zu V2I ist abhängig von Gesetzen und Standards. Daher bezieht sich der folgende Abschnitt auf deren Einfluss auf die Validierung von V2I im Kontext des hochautomatisierten Fahrens.

Das Fahrzeug kann zusammen mit weiteren Systemen nach Automatisierungslevels auch „SAE-Levels“ (0 bis 5) eingestuft werden (siehe Abbildung 2.29). Den Einstufungen können beispielsweise Entwicklungskosten zugeordnet werden. Dabei kam eine Studie von McKinsey (2021) zu dem Ergebnis, dass die meisten Kosten für die Validierung bei höherem Level fällig werden.

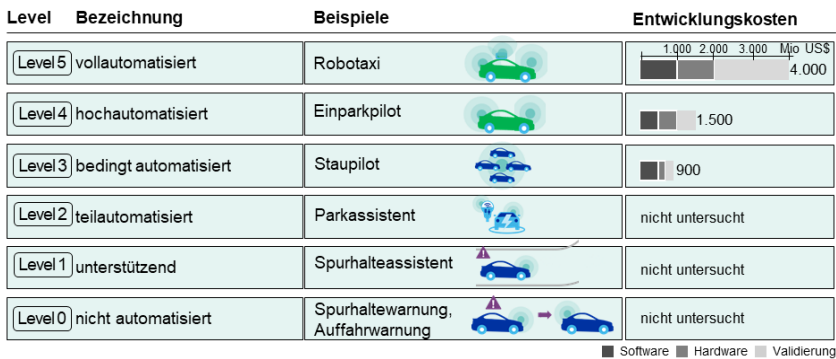


Abbildung 2.29: SAE Levels für automatisiertes Fahren (vgl. Bartels, Eberle & Knapp, 2015; vgl. convex-project.de, 2020; vgl. McKinsey, 2021).

Level 0 beschreibt eine vollständige Kontrolle des Fahrers, wobei Warnungen und aktive Sicherheitsfunktionen möglich sind. Beispielfhaft ließe sich eine Warnung vor einer roten Signalphase hier einordnen. Bei Level 1 übernimmt das Fahrzeug bereits die Fahrautomatisierung einer der Funktionen Lenken, Bremsen oder Beschleunigen, der Fahrer die anderen Funktionen. Beispiele sind Spurhalteassistent oder Fahrgeschwindigkeitsregelung (eng. Adaptive Cruise Control (ACC)). In Level 2 können mehrere Funktionen gleichzeitig übernommen werden, wobei der Fahrer weiterhin Funktionen übernimmt. Beispielsweise kann die Kombination von Spurhalteassistent und ACC in Level 2 erfolgen. In Level 3 sind sämtliche Fahrfunktionen temporär im Fahrzeug durchzuführen. Der Fahrer kann alle Fahraufgaben nach einem Übergabeprozess übernehmen. In Deutschland ist ab dem 21. Juni 2017 ein Gesetz in Kraft getreten, welches das Straßenverkehrsgesetz so ändert, dass Fahrer angepasste Pflichten und Rechte während der automatisierten Fahraufgabe haben, sodass Fahren bis Level 3 ermöglicht wird (Deutscher Bundestag, 2017). Bei Level 4 ist als Erweiterung zu Level 3 der Fahrer nicht verpflichtet eine Übernahmeanfrage des Fahrzeugs zu beantworten, siehe z.B. Einparkassistent. In Deutschland ist am 28. Juli 2021 ein Gesetz in Kraft getreten, das Level 4 in eingeschränkten Bereichen erlaubt (Bundestag, 2021). Bei einer vollständigen automatisierten Fahrt in SAE-Level 5 ist der Fahrer nicht für Fahraufgaben zuständig, was bislang nicht im öffentlichen Personennahverkehr eingesetzt wird. (SAE J3016_202104)

Für eine Realisierung des hochautomatisierten Fahrens wird bezogen auf V2I die Kommunikation in dem Standard IEEE 802.11p beschrieben. Dieser erweitert den weit verbreiteten W-Lan Standard 802.11 für Intelligente Transport Systeme (Ivanov et al., 2018). Der Standard IEEE 802.11p wird in Form einer Schichtenspezifikation für das 5GHz Frequenzband im European Telecommunications Standards Institute

(ETSI EN 302 663) detailliert. Für das korrekte Testen werden Testspezifikationen für die Kanalübertragung im 5.9GHz Frequenzband von der International Telecommunication Union beschrieben. Weitere Spezifikationen zu 802.11p, 5G und hybriden Architekturen beschreibt die 3rd Generation Partnership Project (McKinley, 2021).

Die generelle Beschreibung der Schnittstellen zu weiteren Systemen erfolgt in den Standards zur Systembeschreibung (siehe Kapitel 2.1.1). Bezüglich der Validierung werden in der vorliegenden Arbeit Standards zur Funktionalen Sicherheit (ISO 26262) berücksichtigt.

Aufgrund verschiedener beteiligter Systeme und einer Vielzahl involvierter Stakeholder in dem SoS V2I, beschreibt die Association for Standardisation of Automation and Measuring Systems (ASAM) eine unternehmensübergreifenden Nomenklatur zur Beschreibung von UseCases für SAE Levels (vgl. ASAM e.V., 2022b). Diese werden als OpenX Standards bezeichnet und in Kapitel 2.3.2 beschrieben sowie in Kapitel 6 und 7 berücksichtigt.

Herstellung

Basierend auf dem strukturalen Konzept der Systemtheorie (siehe Kapitel 2.1.1) werden nachfolgend die Strukturen mit den wesentlichen Elementen der Hersteller zur Realisierung der Kommunikation für die Systeme Fahrzeug, Infrastruktur sowie deren Schnittstellen betrachtet.

Innerhalb des Systems Fahrzeug bezieht sich eine bordeigene Kommunikation auf den Informationsaustausch zwischen verschiedenen Teilsystemen in einem Fahrzeug. Beispielsweise sind die Teilsysteme Onboard Unit (OBU), Sensoren und Anwendungseinheiten verbunden. Besondere Anforderungen an die Fahrzeugsteuerung, wie z. B. die Gewährleistung der Zustellung von Nachrichten oder konfliktfreie Nachrichten, erfordern die Verwendung spezieller Netzwerkprotokolle. Ein gängiges Hochgeschwindigkeitskommunikationsprotokoll für den Automobilsektor ist Controller Area Network (CAN). CAN ist ein serielles Feldbussystem, das 1983 von Robert Bosch GmbH entwickelt und 1993 als ISO 11898 standardisiert wurde. Die Teilsysteme innerhalb des Fahrzeugs sind mit einem Gateway verbunden. Dieses erlaubt die Kommunikation zwischen den CAN-Bussen. Beispielsweise wird das CAN-Signal der Fahrzeuggeschwindigkeit über den CAN-Bus zwischen mehreren Steuergeräten gesendet. (Lawrenz, 2011)

In der vorliegenden Arbeit wird die Geschwindigkeit des Fahrzeugs über das bordeigene CAN und die Geschwindigkeit der Rollen des Fahrzeugrollenprüfstands (siehe Kapitel 4.2) über CAN ermittelt.

Innerhalb des Systems Infrastruktur werden im Kontext von V2I die Objekte Lichtsignalanlage und Verkehrsschilder betrachtet. Die Infrastruktursysteme enthalten als funktionsgebendes Bauteil ein Lichtsignalsteuergerät (LSA-Controller). Die Roadside Unit (RSU) ist mit dem LSA-Controller verbunden, um die notwendigen Parameter der Verkehrslichtanlage zu versenden. Die RSU hat laut Herstellerangaben eine Sende-/Empfangsreichweite von ca. 300m bis 1,3km und kommuniziert im Frequenzband von 5,9 GHz (vgl. Herman elektronika). Der prinzipielle Aufbau ist in Abbildung 2.30 dargestellt und beinhaltet neben dem Mikrocontroller ein V2X-Modul zur Kommunikation und Nachrichtenmodulation. Zudem enthält es typischerweise weitere Kommunikationsmodule wie Long Term Evolution (LTE), Bluetooth, zur Standortbestimmung ein Global Navigation Satellite System-Modul (GNSS), sowie eine Stromversorgung. Weitere Teilsysteme sind Handbediengeräte, Signalgebermasten, Sensortaster, Induktionsschleifen, Infrarot-/Kameradetektoren (Landeshauptstadt Erfurt Stadtverwaltung, 2014).

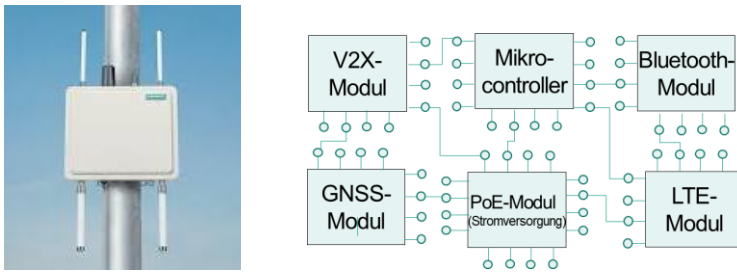


Abbildung 2.30: links: montierte RSU (Siemens Industry, Inc., 2018) rechts: prinzipieller Aufbau einer RSU (Glimm & Zadek, 2021).

Die Schnittstelle zwischen den Systemen Fahrzeug und Infrastruktur ist im Fokus der vorliegenden Arbeit, da sie zur Realisierung des SoS wichtig ist. Die Kommunikation zwischen den Systemen wird über „Board Units“ ermöglicht. Diese Steuergeräte mit Sender und Empfänger transformieren die Daten in das Sendeprotokoll, senden und transformieren sie wieder in eine vom Empfangssystem weiter verarbeitbare Form. Die Daten werden über Protokolle beschrieben, welche Regeln wie das Format, den Inhalt, die Bedeutung und Reihenfolge der Nachrichten definieren.

Für den spezifischen UseCase Green Light Optimal Speed Advisory (GLOSA) werden die drei Nachrichtenarten Cooperative Awareness Message (CAM), MAP topology (MAP) und Signal Phase And Timing (SpaT) eingesetzt (Botte, Pariota, D'Acerno & Bifulco, 2018).

CAM ist ein Awareness-Dienst der Anwendungsschicht, welcher einen periodischen Transfer von Status-Daten zu Single-hop Nachbarn realisiert. Beispielhafte Übertragungen sind Geschwindigkeit und geografische Position, welche nach dem folgenden Schema der Nachricht übertragen werden können.

CAM-Message: <Header> <Body>

- Header: Version, Nachrichten-ID, Erstellungszeit
- Body: Station-ID, (Mobile, Private, Physische) ITS Station, Referenz Position (Längen-/Breitengrad, Höhe, Ausrichtung)

MAP beinhaltet grundlegende kartenbasierte Daten z.B. zur Verkehrskreuzung mit RSU wie die ID, Geometrie und zugelassene Höchstgeschwindigkeit. SpaT enthält die aktuelle und folgende Signalphase und deren Dauer. (ETSI 103 301 1.3.1)

Der „Weg“ der Nachricht von Sender zu Empfänger kann über ein Schichtenmodell nachvollzogen werden (ETSI TS 103 301). Die Schichten, welche Funktionen für die Kommunikation informationsverarbeitender Teilsysteme beschreiben, werden hierbei von oben nach unten spezifischer. Jede Schicht erfüllt eine Aufgabe, die von den anderen Schichten abhängig ist. Je nach Art der Übertragung und Systeme müssen nicht alle Schichten implementiert werden. Abbildung 2.31 veranschaulicht die wesentlichen Schichtenmodelle OSI-Modell (Open Systems Interconnection Reference Model), sowie nach den Standards von ETSI und IEEE. Im Folgenden wird aufgrund der weiten Verbreitung im Bereich V2I das ETSI-Modell verwendet. Aufgrund des sehr ähnlichen Aufbaus sind die Überlegungen übertragbar. Die Nachricht wird in der Schicht Ressourcen erstellt. Anschließend wird sie an die Schicht Netzwerk & Transport zusammen mit Kommunikationsinformationen übergeben. (ETSI TS 103 301).

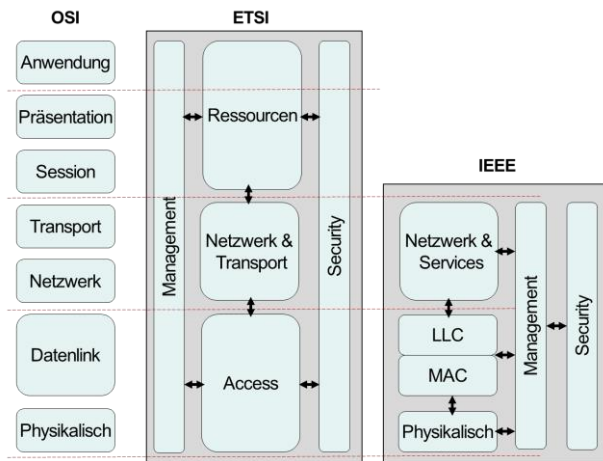


Abbildung 2.31: Kommunikationsschichten Modelle nach den drei Standards der Organisationen OSI, ETSI und IEEE (Abboud, Omar & Zhuang, 2016).

Neben des in der EU zugelassenen und zuvor beschriebenen Kommunikationsprotokolls 802.11p (auch DSRC genannt) werden C-V2X (Nutzung von LTE, 5G) und hybride Übertragungstechnologien unterschieden. Tabelle 2.1 veranschaulicht die wesentlichen Merkmale der Technologien. In der vorliegenden Arbeit wird auf den von der EU zugelassene Standard 802.11p fokussiert, zusätzlich werden Erweiterungen und erste Untersuchungen für 5G erforscht.

Tabelle 2.1: Vergleich V2X-Technologien anhand fünf Kriterien (Keysight Technologies, Inc., 26. Juli / 2018)

Kriterien	802.11p	C-V2X
Einsatzfähigkeit	In der EU zugelassen	Verschiedene Releases, keine vollständige Zulassung in der EU
Kosten	Benötigt RSUs	Parallel Nutzung von Telekommunikations-hardware.
Skalierbarkeit und Evolution	Bislang nichts geplant	5G und weitere Generationen
Latenz	<5ms	1ms bei 5G
Berücksichtigte SoS	Für V2I und V2V	Für V2V, V2I, V2N, V2P

Schnittstellen zwischen Fahrzeug und Infrastruktur werden bereits erfolgreich im Straßenverkehr eingesetzt (vgl. Gardner, D'Souza, Hounsell, Shrestha &

Bretherton, 2009; vgl. Volkswagen AG, 2020). Zwischen Bussen und Lichtsignalanlagen wird eine kürzere Wartezeit von Bussen realisiert. Dies wird im Kontext der in Kapitel 2.1 vorgestellten SGE – Systemgenerationsentwicklung als Referenzsystemelement verstanden. Die Systeme können über verschiedene Architekturlösungen realisiert werden. Beispielsweise über ein Infrarot-Signal (z.B. Buslinie 42 in Stuttgart) Informationen austauschen. Als übliche Technologie wird die Übertragung über GPS an die Verkehrsleitzentrale eingesetzt (z.B. in Southampton). Beispielsweise existiert der „grüne Welle UseCase“, bei dem Fahrzeugen Vorrang gegeben wird und eine lange Grünphase für das Vorrangsfahrzeug vorherrscht. Zudem existieren bereits lernende Systeme, welche sich adaptiv dem aktuellen Verkehrsverhalten anpassen können (z.B. in Stockholm). (Gardner et al., 2009)

Betrieb

Abbildung 2.28 stellt zudem Stakeholder im Bereich des Betriebes von V2I dar. Hersteller sowie Gesetzgeber und Standardisierungsorganisationen können ebenfalls einen Einfluss auf den Betrieb haben. Der Betrieb der Systeme durch einen Fahrer ist je nach SAE-Level in der Gesetzgebung festgelegt (vgl. VDI/VDE 2206). Zudem sind für das soziotechnische SoS V2I Erweiterungen möglich, sodass beispielsweise Fußgänger mittels Smartwatches Daten von Fahrzeug und Lichtsignalanlage erhalten. Es gibt jedoch Zweifel an der Betriebsfähigkeit des hochautomatisierten Fahrens. So herrscht in Ländern wie Deutschland und USA eine geringe Akzeptanz (McKinsey, 2018), die vornehmlich auf Vorbehalte gegenüber der Sicherheit von technischen Lösungen basiert (Schmitz & Hoppe, 2021, S. 12). Somit ist ein zentraler Stellhebel zur Erhöhung der Akzeptanz die Validierung der technischen Lösungen.

2.3.2 Validierungsansätze für das hochautomatisierte Fahren

In Bezug auf V2I existieren Forschungsergebnisse und Ansätze für die Validierung von hochautomatisierten Fahrzeugen. Für V2I sind in den vergangenen Jahren erste Anwendungen auf dem Markt erschienen (Herman elektronika, 2022; Siemens Industry, Inc., 2018), jedoch zögern Hersteller und Anwender bei der Umsetzung aufgrund von Unklarheiten bei den Frequenzen (vgl. Euro NCAP, 2017). Daher wird zu Beginn die Validierung des hochautomatisierten Fahrens beschrieben. Zudem werden einige Spezifika für die Validierung von V2I aufgezeigt. Die Anwendungen und Ansätze werden nach den in MBSE üblichen drei Säulen Sprache, Methode und Tool nachfolgend eingeordnet, wobei der Fokus der vorliegenden Arbeit auf Sprache und Methode liegt.

Sprache

Zur gemeinsamen Verständigung in einer einheitlichen Sprache werden Elemente in Standards definiert. Die UseCases bedürfen einer Verständigung und einer Validierung über Unternehmensgrenzen hinweg. Hierzu werden Elemente in Standards zur Beschreibung beginnend bei Fahrbahn (OpenCRG), Verkehrsnetz (OpenDrive) und Szenario (OpenScenario) definiert. Beispielsweise stellt Abbildung 2.32 die sechs Schichten zur systematischen Beschreibung von Szenarien dar. Nach der DIN SAE SPEC 91381:2019-06 ist ein Szenario eine Beschreibung einer zeitlichen Verkehrskonstellation. Nach dem Forschungsprojekt "PEGASUS" und weiterer Forschung umfasst ein Szenario Szenen über eine Zeitspanne (Pegasus, o.J.). Diese Szenen können als Schnappschüsse der Umgebung betrachtet werden (Pegasus, o.J.). Zu Beginn wird in Schicht eins die Fahrbahn definiert, anschließend weitere statische, dauerhafte Objekte an bzw. auf der Fahrbahn. Schicht drei erweitert dies um statische, temporäre Objekte und in der vierten Schicht werden bewegte Objekte beschrieben. Zum Schluss formuliert Schicht fünf die Umweltsituation wobei aufgrund der großen Thematik der Digitalen Informationen diese separat in Schicht sechs beschrieben wird.

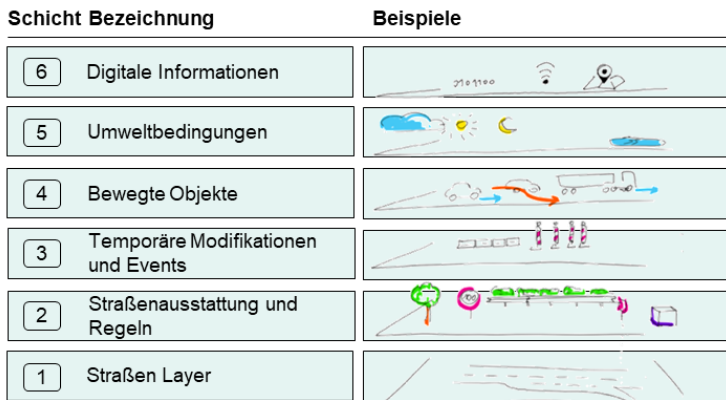


Abbildung 2.32: Szenarienlayer mit sechs Schichten zur systematischen Beschreibung von Szenarien (vgl. Bagschik, Menzel, Reschka & Maurer, 2017; Bock et al., 2018).

Die im Szenario beschriebenen Szenen zeigen das Verkehrsverhalten des Egofahrzeugs in einer Operational Design Domain (ODD). Nach der SAE J3016

zeigt die ODD Betriebsbedingungen, unter denen die Fahraufgabe oder Fahrfunktion spezifiziert ist und betrieben werden soll (SAE J3016_202104).

Methoden

Es lassen sich mehrere Methoden zur Validierung von V2I beschreiben (vgl. Adaptive Consortium (2017)). Diese folgen vornehmlich sechs wesentlichen Ansätzen, um insbesondere die Funktionale Sicherheit zu gewährleisten. Für weitere Validierungsansätze wie die Datensammlung und statistische Auswertung wird auf Dona und Ciuffo (2022) und ASAM e.V. (2022b) verwiesen. Mit ausschließlich realen Prüffahrten müssten ca. 120 Mio. km zurückgelegt werden. Da es beispielsweise auf Autobahnen je 12 Mio. km zu einem schweren Unfall kommt, ist etwa das zehnfache an Testkilometern nötig für valide statistische Aussagen. Zudem muss jede neue Funktionalität auch neu abgesichert werden. Daher bedarf es neuer Methoden, um die Validierung von hochautomatisierten Fahrzeugen effizient zu gestalten und Zeit und Kosten zu reduzieren. (Eberle und Jütten (2017, S. 15)) Nachfolgend werden die sechs Ansätze für neue Methoden vorgestellt:

- Virtualisierung (Dona & Ciuffo, 2022; Galbras, 2018),
- Szenariobasiertes Testen (Balic & Bulaja, 2021; Dona & Ciuffo, 2022),
- Black Box Testen (ISO 26262-4:2018(E)),
- Absicherung von Funktionen (ISO 26262-1:2018(E)),
- Sicherheitsabsicherung über Gefahren (ISO/PAS 21448:2019(E); Leveson & Thomas, 2018),
- Bedarfsgerechte Skalierung mittels X-in-the-Loop (Lutz, Behrendt & Albers, 2020; Stellet et al., 2015).

Eine effiziente Möglichkeit ist Testen mit **virtuellen Modellen**. Die Simulation in den Modellen ermöglicht es, unabhängig von zeitintensiven und schwierig zu reproduzierenden Feldtests, Erkenntnisse über das Systemverhalten zu gewinnen. In mehreren Tools können Simulationen ausgeführt werden und insbesondere Toolhersteller (vgl. Balic & Bulaja, 2021) beschreiben den Bedarf an Virtualisierung. Steimle, Weber und Maurer (2022) beschreiben im Rahmen des Projektes SET-Level eine Methode, um Testfällen Prüfstände zuzuordnen. Diese neuen Erkenntnisse verdeutlichen den aktuellen Stand der Forschung, der sich mit Selektion (Steimle et al., 2022), Realitätsnähe virtueller Tests (Ngo, Bauer & Resch, 2021) und deren Nutzung zur Absicherung beschäftigt. So ermöglichen zukünftig virtuelle Modelle einen Robustheitsnachweis von Einflüssen während einer Fahrt (Maschmeyer, Beidl, Düser & Schick, 2016).

Szenarien beschreiben UseCases, welche realisiert werden. Unter dem Begriff des „Szenarien basierten Testens“ werden von Szenarien Testfälle abgeleitet, um diese zu validieren. Basierend von Szenarien können nun von Inputdaten z.B. Karten, Manöversequenzen abgeleitet werden, welche zusammen mit einer Risikobewertung und einer Verteilungsfunktion für die Auftretenswahrscheinlichkeit von Szenarien führt (Bussler, 2022). Ausgehend von den funktionalen Szenarien kann nun über die Konkretisierung von logischen Szenarien, die maschinenlesbar sind und konkreten Szenarien mit einem Parameterset, ein Testfall abgeleitet werden (Bussler, 2022).

Für SoS sind **Black Box Tests** hauptsächlich dann notwendig, wenn nicht alle Systeminformationen vorliegen oder das vollständige Systemmodell zu aufwändig in der Erstellung ist, was in SoS häufig vorkommt. Die Funktionserfüllung innerhalb eines SoS kann nach Brade (2022) über die Betrachtung von Szenarien erfolgen. Demgegenüber liegen bei einem White Box Test alle Informationen bezüglich des zu untersuchenden Systems vollständig vor. So beschreiben Shalev-Shwartz, Shammah und Shashua (2017) ein interpretierbares, mathematisches Modell für Sicherheitszusagen bei Fahrmanövern. Meist werden White Box - und Black Box Tests separat voneinander betrachtet, wobei insbesondere Validierungsumgebungen für das gesamtheitliche Testen in verschiedenen Phasen der Produktentwicklung relevant und notwendig sind. Beispielhaft wird dies in Kapitel 7 angewendet, um die Sicherheitsdistanz zu bestimmen.

Der Ansatz weg von bauteilbezogenem Test hin zum **Testen von Funktionen** ist insbesondere im Bereich der Sicherheit relevant. Es existieren zahlreiche Metriken zur Bewertung der Sicherheit wie beispielsweise aus der Absicherung von Bauteilen mit der Failure Mode Effect Analysis (FMEA). Dabei wird die Risikopriorität als Produkt von Auftretenswahrscheinlichkeit, Schwere des Schadens und Entdeckungswahrscheinlichkeit des Bauteilversagens bestimmt. Hierzu sind Wirkketten im Sinne des MBSE sinnvoll, sodass die Durchgängigkeit und damit auch Nachvollziehbarkeit die Identifikation der Kriterien erleichtern. Jedoch ermöglichen die bislang eingesetzten Listen- und Tabellenansichten keine vollständige, durchgängige und konsistente Systembetrachtung.

Unter Berücksichtigung der drei **Sicherheitsdimensionen** in der Automobilindustrie: Funktionale Sicherheit, Sicherheit der beabsichtigten Funktionalität (SOTIF (ISO/PAS 21448:2019(E))) und Cybersicherheit (ISO/SAE 21434:2021) (Schnieder & Hosse, 2019), liegt der Fokus in der vorliegenden Arbeit auf der Integration von Funktionaler Sicherheit und SOTIF. Mit Hilfe der ISO 26262 lässt sich die Funktionale Sicherheit eines Systems beschreiben. Die ISO definiert sie als das „Fehlen von unangemessenen Risiken aufgrund von Gefahren, die durch Fehlverhalten von E/E Systemen auftreten (übersetzt nach ISO 26262-1:2018(E),

S. 14)⁴. Im Rahmen dieser Arbeit werden mechatronische Systeme betrachtet, sowie die Identifikation von Gefahren mit Methoden des Systems Engineering

Ein mechatronisches System ist **funktional sicher**, wenn keine unangemessenen Risiken aufgrund von Gefahren auftreten, welche durch Fehlverhalten des Systems oder durch Fehlverhalten von beabsichtigten Funktionen verursacht werden.

untersucht, daher wird eine Definition angelehnt an die ISO 26262-1 gewählt. SOTIF wird als Erweiterung der Sicherheit verstanden (ISO/PAS 21448:2019(E), S. 6). Aufgrund des Aspektes der Erweiterung wird in der vorliegenden Arbeit SOTIF als Teil der Funktionalen Sicherheit verstanden (vgl. Rau & Dojan, 2021, S. 11). Dies führt zu folgender Definition (angelehnt an ISO 26262-1:2018(E), S. 14; angelehnt an ISO/PAS 21448:2019(E), S. 6; angelehnt an Rau & Dojan, 2021, S. 11):

Die zwölf Standards zur Funktionalen Sicherheit verwenden hierzu das Automated Safety Level (ASIL), welches sich beispielsweise in Bezug zu Anforderungen in einer hierarchischen Systemstruktur aufgliedern lassen.

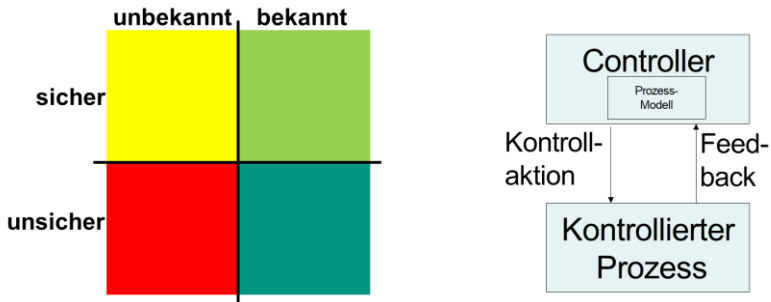


Abbildung 2.33: Die vier SOTIF Quadranten (links) (ISO/PAS 21448:2019(E)) und ein Regelkreis in STAMP (rechts) (Leveson & Thomas, 2018).

Der Standard zur Sicherheit einer beabsichtigten Funktion (engl. Safety of the Intended Function (SOTIF)) beschreibt die Einteilung nach Wissen und Gefahr (ISO/PAS 21448:2019(E)). Es werden Ansätze genannt, um die unbekannt, unsicheren UseCases zu ermitteln (siehe Abbildung 2.33). Anschließend können Maßnahmen getroffen werden, um die Sicherheit in den UseCases zu gewährleisten.

Insbesondere der Begriff der Gefahr (engl. Hazard) wird verwendet. Beispielsweise in den Methoden Hazard Analysis and Risk Assessment (HARA) und Systems Theoretic Process Analysis (STPA) werden gezielt Gefahren identifiziert, um

hieraus Gründe und Auswirkungen abzuleiten. In der vorliegenden Arbeit wird auf die STPA gezielt eingegangen, da sie einen vollständigen Ansatz beschreibt, um die Sicherheit von Systemen zu modellieren. Der Ansatz basiert auf STAMP (System-Theoretic Accident Model and Processes), das Regelkreise innerhalb eines Systems identifiziert. In Abbildung 2.33 ist der allgemeine Regelkreis zu sehen, in der sich die Kommunikation von Elementen zur Realisierung von Funktionen aufteilen lässt. Es werden somit Regelkreise aufgebaut, anhand derer nun mittels STPA Gefahren identifiziert werden können.

Die STPA-Methode ist in Abbildung 2.34 dargestellt. Zu Beginn wird der Zweck der Analyse bestimmt und das System mit seinen Systemgrenzen daraufhin modelliert. Im zweiten Schritt wird der Regelkreis basierend auf STAMP modelliert. Der dritte Schritt beinhaltet die Identifikation von unsicheren Kontrollaktionen innerhalb des Regelkreises. Zuletzt werden die Schadensszenarien identifiziert. (Leveson & Thomas, 2018)

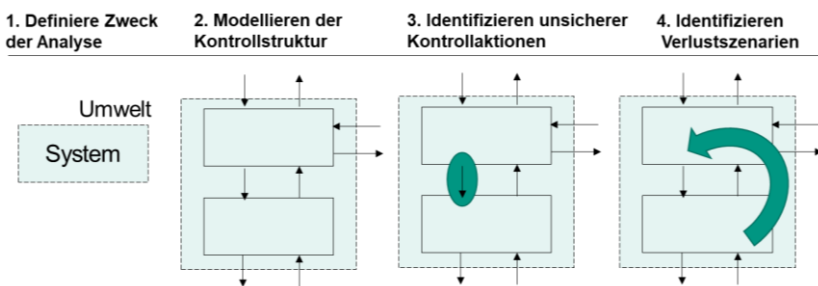


Abbildung 2.34: STPA Methode und Verbindung mit STAMP (vgl. Leveson & Thomas, 2018).

Die Analyse von Energie-, Informations- und Material-Kreisläufen wird insbesondere im Kontext von **X-in-the-Loop** eingesetzt. So formuliert Wood et al. (2019) den Bedarf an Driver-in-the-Loop Ansätzen. Im Kontext der Validierung von Fahrerassistenzsystemen wenden Lutz et al. (2020) den IPEK-XIL-Ansatz an (siehe in Kapitel 2.2.2 Abbildung 2.22).

Tools

Zahlreiche Tools ermöglichen die Validierung von hochautomatisierten Fahrzeugen und Infrastruktursystemen. Nachfolgend werden diese aufgeführt und in Bezug zu der vorliegenden Arbeit gesetzt. Die Werkzeuge lassen sich nach der Ausprägung unterscheiden in virtuelle Modelle (Simulationen), physische Modelle (Teilsystemprüfstände) und Gesamtsysteme (Testfelder, Gesamtprüfstand). Reine Softwarelösungen werden anwendungsfallspezifisch eingesetzt und sind für die

vorliegenden Arbeit in Tabelle 2.2 aufgeführt (vgl. Thorn, Kimmel Shawn & Chaka, 2018). Die einzelnen UseCases mit ihren Tools werden nachfolgend erläutert.

Tabelle 2.2: Eingesetzte und weitere Simulationstools eingeteilt nach UseCases (vgl. Thorn et al., 2018)

#	UseCase	Eingesetzte Tools in der Arbeit	Auswahl weiterer Tools
1	Emulation von Kommunikation zwischen Fahrzeug und Infrastruktur, um Parameter wie Latenz und Fehlerrate zu ermitteln	MATLAB 2022a, ns-3.30	Riverside Modeler, OMNET
2	Simulation des „Weltmodells“ mit der Umwelt	MATLAB 2022a	CARLA, Cognata, ADS
3	Fahrzeugverhalten	MATLAB 2022a, CarMaker 5.1.1	CARLA
4	Verkehrsverhalten	MATLAB 2022a	VISSIM, Aimsun, TransModeler
5	Testautomatisierung	ModelCenter 14.2	Model.CONNECT

#1: Für die Emulation von Kommunikationssignalen wird Network Simulator 3 (ns-3) lizenziert unter GNU General Public License v2 eingesetzt. Das Tool ist spezialisiert auf Netzwerkerstellung und -übertragungen. Es verwendet Knoten und Verbindungen zwischen Knoten zur Systembeschreibung und kann die Layer im OSI Netzwerkprotokoll abbilden. Zudem lässt es sich mit MATLAB-Simulink im Rahmen einer Co-Simulation koppeln.

#2: MATLAB von Mathworks kann zur Modellierung von Systemen verwendet werden und kommt ursprünglich aus der Rechnung mit Matrizen. Ein großes Einsatzgebiet ist die Steuerungs- und Regelungstechnik. Die Software bietet im Bereich des hochautomatisierten Fahrens Erweiterungen an, wie die „Automated Driving Toolbox“ (ADT). Die ADT beschreibt ein Szenario mit Umwelt

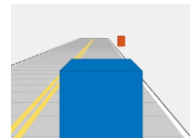


Abbildung 2.35: Graphische Darstellung einer Szene in MATLAB ADT.

(z.B. Einlesen von Kartendaten). Mit Hilfe des in ADT enthaltenen „Driving Scenario Designer“ wird die ODD beschrieben sowie eine virtuelle Repräsentanz (siehe Abbildung 2.27) dargestellt und Analysen ermöglicht.

#3: Das Fahrzeugverhalten kann ebenfalls mit MATLAB simuliert werden. Jedoch sind zahlreiche Modelle notwendig, um alle Funktionalitäten eines Gesamtfahrzeuges zu realisieren. Daher wird es oft für die Simulation von Teilaktivitäten eingesetzt. CarMaker von IPG Automotive GmbH enthält bereits eine Vielzahl von Modellen für die Gesamtfahrzeugsimulation. Zudem bietet sich eine realitätsnahe virtuelle Repräsentanz (vgl. Sensorstimulation mittels Bilder in Lutz et al. (2020)) sowie eine Einbettung in Prüfstandsarchitektur mittels eines Echtzeitsystems an.



Abbildung 2.36: Graphische Darstellung einer Szene in CarMaker.

#4: Für mikroskopische Verkehrssimulationen werden Tools wie VISSIM, AIMSUN und Paramics eingesetzt (Olstam, Lundgren, Adlers & Matstoms, 2008). Im Rahmen der Arbeit wird das Tool MATLAB genutzt.

#5: Testautomatisierungstools wie ModelCenter von Ansys ermöglichen die Erstellung von Testfällen und Ausführung von Tests. ModelCenter bietet zudem Kommunikationsschnittstellen zwischen einzelnen Expertentools wie Cameo Systems Modeler und MATLAB. (Vgl. Thorn et al., 2018)

Die Tools verdeutlichen das Potenzial zur virtuellen Validierung von V2I. Bislang gibt es zur Validierung von V2I vornehmlich Forschungsarbeiten mit Simulationen. Dies liegt daran, dass Zertifizierungen bislang lediglich über Straßenfahrten erfolgen. Außerdem sind Simulationsprogramme meistens mit einer Vereinfachung verbunden, welche quantifiziert werden muss. Weiterhin fokussiert sich die Fahrzeugindustrie stark auf Fahrerassistenzsysteme, da im Bereich von V2I kein weltweiter Standard vorherrscht (vgl. Euro NCAP, 2017)).

Trotz des Fokus auf Virtualisierung existieren erste hardwarebasierte Ansätze zur Validierung mit Prüfstandstests und Testfeldern, die nachfolgend vorgestellt werden.

Prüfstandstests werden von Ress und Wiecker (2016) im „Elektrischen Bremslabor“ (EBL) in Helmond durchgeführt. Beim EBL bekommt das Egofahrzeug vom Vorderfahrzeug ein Signal zur Bremsung vorgegeben. Hakim und Kreuer (2014) nutzen Hardwaresysteme basierend auf Funksignalemulatoren und

Testautomatisierung, welche die Nachrichtenübertragung simulieren können. Die von der Firma Qosmotec entwickelten Lösungen können bis zu 17 Funkgeräte simulieren. In Aachen werden von der Firma CANoe GPS Positionen simuliert und an Steuergeräte und Funkgeräte am Prüfstand weitergeleitet (Hakim & Kreuer, 2014). Hierbei kommt ein Luftschnittstellenemulator zum Einsatz (Hakim & Kreuer, 2014). Machbarkeitsversuche zur generellen Realisierung von V2X am Prüfstand erfolgten durch Kopplung des Verkehrssimulators Microsim mit einer Kommunikationseinheit und dem Fahr Simulator CarMaker (Albers, Düser, Sander, Roth & Henning, 2010).

Im Rahmen des Forschungsprojektes C-Roads werden zahlreiche **Testfelder** für hochautomatisiertes Fahren vorgestellt (Böhm, 2020). Meistens inkludieren diese auch V2I. Beispielsweise wurde an der Universität RWTH Aachen eine Forschungskreuzung aufgebaut. Diese beinhalteten Betonklötze zur Stimulation von Gebäudefassaden sowie Verkehrskreuzungen mit Verkehrslichtanlagen und RSUs. (Themann, Raudszus, Zlocki & Eckstein, 2016) Andere Forschungsprojekte wie das "Testfeld Sichere und Intelligente Mobilität Deutschland" oder das "Testfeld Autonomes Fahren Baden-Württemberg" integrieren die Validierungsumgebung in den Straßenverkehr.

2.3.3 Zwischenfazit

Die Validierung des in dieser Arbeit betrachteten System of Systems Vehicle-to-Infrastructure ist zwar Gegenstand der Forschung, jedoch ermöglichen Simulationen und Aktivitäten auf Testfeldern keine kontinuierliche Validierung des Gesamtsystems V2I, sondern nur eine Validierung von einzelnen Systemelementen wie Funktionen. Insbesondere in der wichtigen Bewertung der Funktionalen Sicherheit von V2I existieren nur einzelne Methoden wie SOTIF, welche abhängig des Reifegrads der Modelle eingesetzt werden. Folgende Aussagen fassen das Kapitel 2.3 zusammen:

- Die Validierung im Bereich des hochautomatisierten Fahrens soll mit Hilfe von **virtuellen, physischen und gemischt virtuell-physischen Modellen unterstützt werden.**
- Es fehlt an **durchgängigen, kontinuierlichen Ansätzen für eine Validierung des Gesamtsystems V2I**
- Die **Funktionale Sicherheit stellt ein zentrales Forschungsgebiet** in der Validierung des hochautomatisierten Fahrens und im Speziellen für V2I dar.

2.4 Diskussion und Fazit

Kapitel 2 verdeutlicht die Forschungslücke bezüglich einer ganzheitlichen, systemischen Betrachtung von V2I in der Validierung, wobei MBSE das Potenzial zur Unterstützung bietet. Die Forschungslücke insgesamt umfasst die in den Kapiteln identifizierten nachfolgenden Teile:

Bestehende Produktentwicklungsansätze beziehen sich nicht auf SoS

Es wird aufgezeigt, dass die Validierung als zentrale Aktivität im Produktentstehungsprozess für SoS bislang nicht erforscht ist. Wesentliche Modelle sind zwar generisch formuliert, jedoch fehlen Untersuchungen bezüglich SoS. (siehe Kapitel 2.2)

Keine Berücksichtigung der Systemeigenschaften von SoS

Die Betrachtung von Systemen und insbesondere System of Systems im Sinne des (Model Based) Systems Engineerings bieten Vorteile, die bislang nicht in der Validierung von V2I berücksichtigt werden. Inwieweit die Vorteile zu einem erhöhten Praxisnutzen führen ist bislang unklar. (siehe Kapitel 2.3)

Keine systemische, durchgängige Berücksichtigung der Funktionalen Sicherheit

Für hochautomatisiertes Fahren ist die Funktionale Sicherheit durchgängig, das heißt über Domänen und Systemebenen hinweg zu betrachten. Es existieren mehrere Standards und Methoden zur Funktionalen Sicherheit, jedoch keine Vereinheitlichung und keine Nutzung von zentralen, durchgängigen Systemmodellen im Sinne des MBSE. (siehe Kapitel 2.3)

Keine einheitliche, flexible Betrachtung von virtuellen und physischen Tests

Es wurde der Trend zur Virtualisierung in der Validierung aufgezeigt, der durch die Betrachtung von mehreren Tools unterstützt wird. Weiterhin existieren erste Hardwareanwendungen in Form von Prüfständen und Testfeldern. Eine Kombination von Simulation und Hardwareanwendungen für eine flexible Validierung von V2I unabhängig vom Zeitpunkt im Produktentwicklungsprozess fehlt bislang. (siehe Kapitel 2.3)

3 Bedarf und Zielsetzung

Ausgehend von der Forschungslücke von MBSE in der Validierung von SoS und insbesondere Vehicle-to-Infrastructure (V2I) werden nachfolgend der Forschungsbedarf untersucht sowie die Zielsetzung der vorliegenden Arbeit mit Annahmen, Ziel, Thesen und Forschungsfragen benannt.

3.1 Forschungsbedarf

Der allgemeine Trend führt dazu, dass immer häufiger Systeme auch Teile von größeren SoS sind (Friedenthal et al., 2021). Insbesondere Trends wie die Digitalisierung, Globalisierung und Automatisierung verstärken den Bedarf zusätzliche Funktionen durch das Zusammenspiel von mehreren Systemen zu realisieren. Die vorliegende Arbeit liefert einen Beitrag, wie ein System dabei unter Berücksichtigung des Einsatzes in einem SoS entwickelt werden sollte.

Verständlichkeit von V2I als SoS

Ein SoS muss im Rahmen dessen Entwicklung mit seinen Zielen verstanden werden, um es basierend auf dessen Charakteristiken geeignet zu entwickeln. Darüber hinaus sind SoS-Charakteristiken nach Maier (2021) nicht gleich wichtig, sodass dies in den Zielen berücksichtigt werden sollte. Somit bedarf es trotz bestehender Standards (siehe Abbildung 2.5) einer Beschreibung des Zielsystems (siehe Abbildung 2.2 nach dem Verständnis nach Albers et al. (2011)) für SoS, um die Verständlichkeit von SoS für Entwickelnde zu verbessern und aus SoS-Charakteristiken die richtigen Implikationen für Aktivitäten der Produktentwicklung zu ziehen.

Durchgängigkeit von Modellen, Methoden für die Validierung von V2I

Eine geeignete Berücksichtigung von SoS speziell in der Aktivität der Validierung ist bislang nur vereinzelt realisiert, da SoS durch ihre Größe und Vernetzung aufwändig zu betrachten sind (vgl. Kapitel 2.3.2). Zudem wird eine Nachvollziehbarkeit von Standards insbesondere zur Funktionalen Sicherheit (siehe ISO 26622, ASPICE) gefordert, um die Wirkung von Systemelementen wie Funktionen auf das Systemverhalten zu identifizieren. Dies soll insbesondere die Sicherheit im Betrieb der Systeme ermöglichen. Ansätze des Model-Based Systems Engineering können die Produktentwicklung und im Speziellen die Validierung von SoS unterstützen. Mittels Wirkketten wäre es für Entwickelnde möglich, eine durchgängige Verbindung

zwischen Ziel, Anforderung, UseCase, Funktion und Test zu realisieren. Dies kann eine effiziente Validierung ermöglichen, um beispielsweise schnell auf Systemänderungen reagieren zu können.

Kontinuität von Modellen und Methoden für die Validierung von V2I

SoS verändern sich kontinuierlich aufgrund von Änderungen einzelner Systeme mit unterschiedlichen Lebenszyklen. Modell und Methoden sollten demnach kontinuierlich über den Produktentwicklungsprozess anwendbar sein.

Vollständigkeit von Modellen und Methoden für die Validierung von V2I

Zudem formuliert INCOSE den Bedarf an modellbasierten Verfahren, um wesentliche Aktivitäten der Produktentwicklung wie die Validierung zu unterstützen (Friedenthal et al., 2021). Hierbei fehlt jedoch eine systematische, modellbasierte Betrachtung von Sprache und Methode für spezifische SoS. Nur hierdurch kann zusammen mit dem eingesetzten Tool MBSE mehrwertstiftend angewendet werden.

Flexible und erweiterbare Modelle und Methoden für die Validierung von V2I

Als prominentes Einsatzgebiet von SoS gilt die vernetzte Mobilität mit mehreren vernetzten Systemen. Die Kommunikation von Fahrzeug mit Infrastruktur (V2I) ist hierbei eines der ersten realisierten Systeme im Bereich der Flug-, Bus- und Bahntechnik. In der bisherigen Produktentwicklung von V2I werden jedoch die SoS-Eigenschaften und deren Implikationen selten berücksichtigt. Aufgrund von fehlenden Regularien sowie aufwändigen Validierungsaktivitäten werden bislang hauptsächlich Sensoren wie Kamera und Radar eingesetzt, wobei der mehrwertstiftende Einsatz beispielsweise für Sicherheit, Bequemlichkeit und Verkehrseffizienz von V2I in Studien gezeigt wurde (Nassir et al., 2020; Rösener et al., 2019). Demzufolge besteht der Bedarf die Validierung von V2I mittels nachvollziehbarer, gesamtsystemischer Betrachtung zu unterstützen.

Forschungslücke	Forschungsbedarf	Effekt
Bestehende Produktentwicklungsansätze beziehen sich meist nicht auf SoS	Ein einheitliches Verständnis zu V2I als SoS und dessen Implikationen für die Validierung ist notwendig.	Verständlichkeit
Bestehende Produktentwicklungsansätze beziehen sich meist nicht auf SoS	Häufige Veränderungen des SoS führen zu dem Bedarf an sich kontinuierlich ändernde Produktentwicklung	Kontinuität
Keine Berücksichtigung der Systemeigenschaften von SoS	Einzelne Publikationen zeigen Teilaspekte der Modellierung auf. Es Bedarf jedoch eine gesamtssystemische systematische Modellierung	Vollständigkeit
Keine einheitliche, flexible Betrachtung von virtuellen & physischen Tests	Häufige Veränderungen des SoS führen zu notwendigen, flexiblen Modellen in einer Validierungsumgebung	Flexibilität
Keine einheitliche, flexible Betrachtung von virtuellen & physischen Tests	Häufige Veränderungen des SoS führen zu dem Bedarf, dass sich die Entwicklung und in ihr die Validierungsumgebung erweitert, um weitere UseCases zu prüfen	Erweiterbarkeit
Keine systemische, durchgängige Berücksichtigung der Funktionalen Sicherheit	Durchgehende Betrachtung der Ziele im Gesamtsystem und seinen Teilsystemen.	Durchgängigkeit

Abbildung 3.1: Aus der Forschungslücke abgeleiteter Forschungsbedarf und den hieraus resultierenden notwendigen Effekten, welche zu realisieren sind.

Für den aufgezeigten Bedarf wird im Folgenden der Lösungsraum mit der Aufstellung zweier Grundannahmen und des Forschungsziels konkretisiert.

3.2 Grundannahmen und Forschungsziel

Der Bedarf an Unterstützung der Validierung von V2I soll in dieser Arbeit untersucht werden. Die aus der Produktentwicklung bekannten Ansätze Model-Based Systems Engineering und IPEK-X-in-the-Loop bieten erste Lösungskonzepte für einzelne Systeme. Sie führen zu zwei Grundannahmen (GA).

Grundannahmen

GA1: SoS lassen sich unter Einsatz von MBSE modellieren.

GA2: SoS lassen sich unter Einsatz des IPEK-xIL-Ansatzes validieren

Die erste Grundannahme beschreibt, dass mit MBSE ein SoS modelliert werden kann. Da ein SoS ein System ist, sollte MBSE dies ermöglichen. Zudem existieren

bereits Forschungsbeiträge, welche SoS im Verständnis von MBSE modellieren (siehe Kapitel 2.1.3).

Die zweite Grundannahme nennt die Eignung des IPEK-XiL-Ansatzes zur Validierung von SoS. Bislang wurde dies nicht betrachtet. Der generalisierte IPEK-XiL-Ansatz eignet sich jedoch für ein breites Einsatzgebiet mit einer Vielzahl von Systemen. Wesentlicher Fokus des Ansatzes ist die Betrachtung des zu entwickelnden Systems in der Vernetzung mit seiner Umwelt. Da Vernetzung und Systembetrachtung wesentlich für SoS sind, sollte der IPEK-XiL-Ansatz auch für SoS eingesetzt werden können.

Aus dem Bedarf der Validierungsunterstützung und den zwei Grundannahmen leitet sich das Forschungsziel der Arbeit ab.

Forschungsziel

Das Ziel ist eine durchgängige, kontinuierliche Sprache und Methode zur Validierung der Funktionalen Sicherheit von Vehicle-to-Infrastructure als SoS durch MBSE-Unterstützung unter Einsatz des IPEK-XiL-Ansatzes zu entwickeln.

Hierbei wird die Durchgängigkeit als über mehrere Bereiche¹ und Systemlayer² der Produktentwicklung und die Kontinuität als über mehrere Phasen der Produktentwicklung dauerhaft stattfindend verstanden. V2I wird als SoS in Kapitel 4 untersucht und die Bezeichnung von V2I als SoS bestätigt. V2I soll auf dessen Validierung im Hinblick auf seine SoS spezifischen Charaktereigenschaften untersucht werden, was zu nachstehender Grundthese und drei Forschungsfragen führt.

¹ Unter Bereich wird in der vorliegenden Arbeit eine Gruppierung von Systemen verstanden. Ein Bereich kann firmenspezifisch (vgl. Muschik (2011)), domänenspezifisch (vgl. Bursac (2016)) oder anwendungsfallspezifisch (z.B. hochautomatisiertes Fahren) sein. Die vorliegende Arbeit verwendet letzteres, um normenspezifische Sprachelemente einem Bereich zuzuordnen.

² im Sinne der hierarchischen Systemstruktur nach Ropohl (2009).

3.3 Grundthese und Forschungsfragen

Das Ziel der Forschung wird in diesem Kapitel konkretisiert. Zu Beginn wird eine Grundthese zur Orientierung der Forschungsfragen aufgestellt. Anschließend werden drei Forschungsfragen aus dem Forschungsbedarf hergeleitet, welche in der vorliegenden Arbeit beantwortet werden.

3.3.1 Grundthese

Die Grundthese benennt ein Beschreibungsmodell zur Validierung und gibt neben den drei Forschungsfragen die Orientierung für den Aufbau und den Inhalt der folgenden Kapitel vor.

Grundthese

Mittels eines Beschreibungsmodells lassen sich SoS-Charakteristiken identifizieren, die die Validierung von V2I-Anwendungsfällen mittels MBSE unterstützen.

3.3.2 Forschungsfragen

Abgeleitet aus dem Forschungsbedarf und der Grundthese stellt sich die Frage, wie sich V2I geeignet beschreiben und validieren lässt. Zudem gilt es, den Forschungsbedarf zu adressieren. Demzufolge werden die drei nachfolgenden Forschungsfragen aufgestellt.

Forschungsfragen

- FF1: Wie lässt sich das Zielsystem von V2I als SoS verständlich beschreiben und zu verteilten Systemen abgrenzen, um spezifische Herausforderungen in der Validierung von V2I zu identifizieren?
- FF2: Wie kann die durchgängige und kontinuierliche Validierung von V2I durch Modellierungssprache und Methode im Sinne von MBSE unterstützt werden?
- FF3: Wie sollte eine Validierungsumgebung zur durchgängigen und kontinuierlichen Validierung der Funktionalen Sicherheit von V2I realisiert sein?

Forschungsfrage 1 fördert das Systemverständnis und ermöglicht die Identifikation von Herausforderungen. Mittels der Antwort gilt es zu untersuchen, wie MBSE die Validierung von SoS unterstützt. Die vorliegende Arbeit fokussiert auf das System

V2I, um eine fachliche Tiefe zur Beantwortung der Forschungsfragen zu realisieren. Ein möglicher Transfer auf weitere Systeme wird in Kapitel 8.3 genannt.

Forschungsfrage 2 konzentriert sich auf die ersten zwei Säulen des MBSE: Sprache und Methode und setzt Grundannahme 1 voraus. Mit der Beantwortung der Frage sollen SoS-Charakteristiken und deren Implikationen aus der beantworteten Forschungsfrage 1 geeignet bei der Validierung von V2I berücksichtigt werden. Die Effekte der Durchgängigkeit und Kontinuität sind zur Beantwortung der Frage zu prüfen.

Forschungsfrage 3 fragt nach der Gestaltung einer Validierungsumgebung. Aufbauend auf Sprache und Methode gilt es diese im Rahmen der Validierung einzusetzen, um den Nutzen dieser zu prüfen. Die Validierungsumgebung bildet den Rahmen der Validierung und enthält alle notwendigen Elemente, um die Validierung des System-in-Development V2I durchzuführen. Die Validierung wird basierend auf Grundannahme 2 und dem Forschungsziel mithilfe des IPEK-XiL-Ansatzes durchgeführt. Mithilfe einer Validierungsumgebung mit spezifischen Validierungskonfigurationen und Messergebnissen soll die Forschungsfrage 3 beantwortet werden. Die Validierung mit ihren Analyse- und Syntheseschritten ist nach Albers (2010) die einzige Aktivität, bei der Wissen entsteht. Demzufolge fördert sie das Systemverständnis. Zudem nutzt die Validierung sprachliche Elemente und Methoden, sodass die Evaluation der Ergebnisse aus Forschungsfrage 2 möglich ist.

4 Methodisches Forschungsvorgehen

Die vorliegende Arbeit orientiert sich an der Design Research Methodology (DRM) von Blessing und Chakrabarti (2009). Das Vorgehen wird im Folgenden erläutert. Anschließend werden in Kapitel 5 eingesetzte, empirische Methoden eingeführt sowie die Forschungsumgebung beschrieben.

4.1 Vorgehen angelehnt an die Design Research Methodology von Blessing und Chakrabarti (2009)

Die DRM beschreibt das Vorgehen zur wissenschaftlichen Behandlung des Forschungsthemas der Validierung von V2I. Das Vorgehen besteht aus den vier Stadien Research Clarification (RC), Deskriptive Studie 1 (DS1), Präskriptive Studie (PS) und Deskriptive Studie 2 (DS2) (siehe Abbildung 4.1). Beginnend mit der Research Clarification werden Kriterien und die in Kapitel 2 aufgezeigten drei Themen „Systemtheorie und Systems Engineering“, „Validierung in der Produktentwicklung“ und „Beschreibung und Validierung von V2I“ untersucht und in Zusammenhang gesetzt. Sie führt zu einer Wissenslücke (siehe Kapitel 2.4), die den Forschungsbedarf beschreibt (siehe Kapitel 3.1). Anschließend behandelt die Deskriptive Studie 1 die Beschreibung des Forschungsvorgehens und die Prüfung des Forschungsbedarfs (siehe Kapitel 5). Darauf aufbauend wird in der Präskriptiven Studie der Lösungsvorschlag zur Beantwortung der Forschungsfrage erarbeitet (siehe Kapitel 6). In der initialen Deskriptiven Studie 2 wird die Lösung aus der Präskriptiven Studie auf deren Eignung geprüft (siehe Kapitel 7). Hierbei erfolgt eine initiale Evaluierung anhand des Einsatzes der Methode in der Prüfumgebung und Befragungen von Experten. Dies beinhaltet auch eine kritische Reflexion der Forschungsfragen.

Zusätzlich wurde innerhalb der Präskriptiven Studie ein explorativer Ansatz gewählt, sodass mehrere Iterationen möglich sind (siehe in Kapitel 4.2 Abbildung 4.4). Demnach sind die in Kapitel 6 erstellten Ergebnisse der Präskriptiven Studie durch mehrere iterative Schritte angelehnt an (vgl. L. Marxen & Albers, 2012, S. 1014; vgl. Leif Marxen, 2014) erstellt worden. Die Iterationen zwischen der zu entwickelnden Lösung und ersten Validierungsaktivitäten ermöglichen eine Wissensgenerierung und damit eine Synthese (vgl. L. Marxen & Albers, 2012, S. 1014; vgl. Leif Marxen, 2014).

Abbildung 4.1 zeigt neben dem Aufbau der vorliegenden Arbeit die Zielsetzung der einzelnen Kapitel. Es wird deutlich, dass die nachfolgenden drei Kapitel 5 bis 7 jeweils eine Forschungsfrage beantworten und in unterschiedliche Stadien der DRM einzuordnen sind. Der Zykluspeil bei der Präskriptiven Studie verdeutlicht die iterative Vorgehensweise angelehnt an Leif Marxen (2014).

DRM-Stadium	Kapitel	Forschungsinhalte	Zielsetzung
RC	1	Einleitung	Motivation
	2	Stand der Forschung und Technik	Forschungslücke
	3	Bedarf und Zielsetzung	Forschungsbedarf, -fragen
	4	Forschungsvorgehen	Forschungsdesign
DS I	5	Beschreibung des Zielsystems für die Validierung von V2I	Vertieftes Verständnis für die Bedarfe in der Validierung von SoS und Anforderungen an die sprachliche und methodische Unterstützung
PS	6	Modellierungssprache und Methode zur Unterstützung der Validierung von V2I	Unterstützung der Validierung von V2I im Sinne von MBSE
Initiale DS II	7	MBSE-unterstützte Validierung von V2I auf Basis des IPEK-XiL Ansatzes	Anwendung der Sprache und Methode zur Evaluation der Unterstützung für die Validierung von V2I
	8	Zusammenfassung und Ausblick	Aufzeigen von weiterführenden Forschungsarbeiten

Abbildung 4.1: Übersicht des Forschungsvorgehens angelehnt an die Design Research Methodology (Blessing & Chakrabarti, 2009).

Die Beantwortung einer Forschungsfrage gilt als Eingangsgröße für die darauffolgende Frage. Der Bedarf von Forschungsfrage 1 wird in der empirischen Studie in Kapitel 5.1 ermittelt. Ausgehend von den Ergebnissen zeigt Kapitel 5.2 ein Vorgehen für eine Systembeschreibung auf, das in einer empirischen Studie überprüft wird. Der Bedarf von Forschungsfrage 2 wird aus den Ergebnissen der ersten Forschungsfrage und der Literatur abgeleitet. Die Präskriptive Studie enthält die in Kapitel 5.1 bis 5.3 sprachlichen und methodischen Unterstützungen, deren Überprüfung mittels Forschungsfrage 3 durchgeführt wird. Forschungsfrage 3 dient zur Evaluation der Ergebnisse der Präskriptiven Studie. Hierzu wird eine Validierungsumgebung in Kapitel 7 aufgebaut sowie die Durchführung der

Validierung anhand dreier Validierungskonfigurationen untersucht. Die abschließende Bewertung der Messergebnisse überprüft die Validierungsumgebung und die Durchführung der Validierung. Somit führen die Ergebnisse der drei Forschungsfragen zu dem in Kapitel 3.2 formulierten Forschungsziel, die Validierung von V2I als SoS zu unterstützen. Abbildung 4.2 veranschaulicht die Einzahlung der drei Forschungsfragen auf dieses Ziel und die Abhängigkeiten der Forschungsfragen untereinander. Mittels Analyse- und Synthese-Aktivitäten fließen Ergebnisse einer Forschungsfrage in eine weitere Forschungsfrage ein.

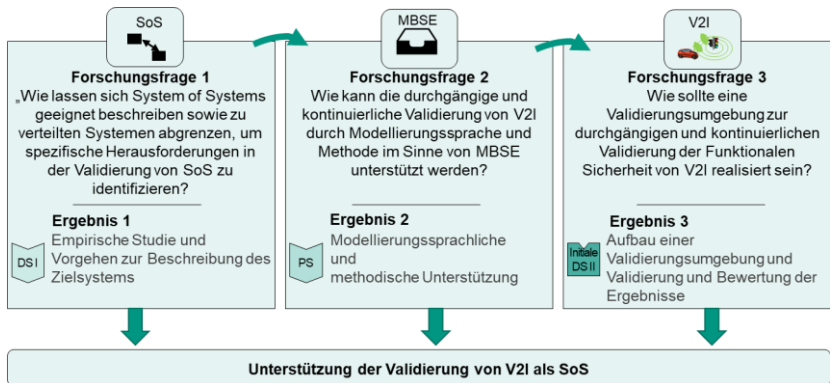


Abbildung 4.2: Forschungsfragen und deren Bezug zu drei Stadien der DRM.

Im Rahmen der Research Clarification werden zu Beginn Effekte und Erfolgsfaktoren identifiziert und in Abbildung 4.3 in einem Impact Model visualisiert. Somit werden die Wirkungen der Forschungsergebnisse auf direkt einflussbare, messbare Effekte geprüft. Diese Effekte führen zu Erfolgsfaktoren, die die Zielerreichung des Forschungsziels prüfen. Die Abhängigkeiten zwischen Effekten und Erfolgsfaktoren sind aus der Literatur nachgewiesen (siehe Quellennachweise an den Beziehungen in Abbildung 4.3). Effekte und Erfolgsfaktoren werden in den folgenden Teilkapiteln „Diskussion und Zwischenfazit“ der einzelnen Kapitel aufgegriffen, um die Beantwortung der jeweiligen Forschungsfrage zu prüfen.

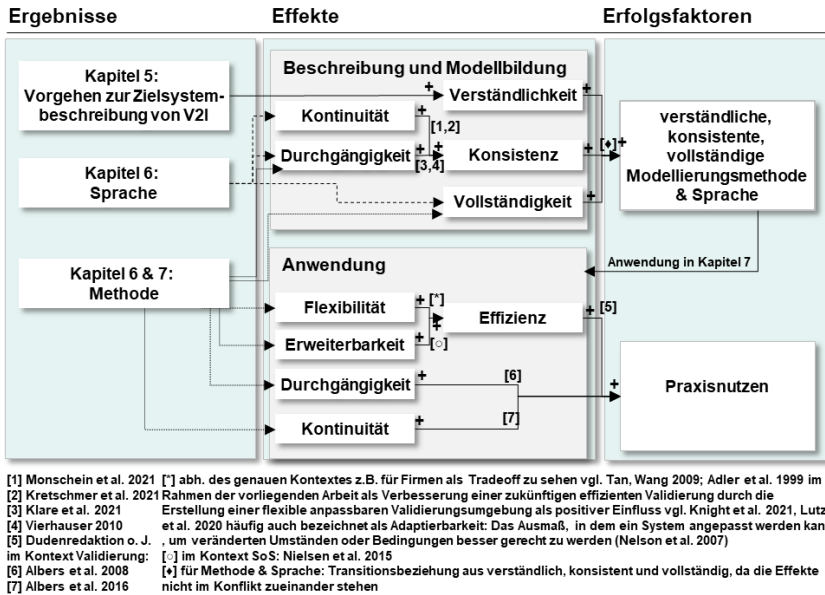


Abbildung 4.3: Ableitung der Erfolgsfaktoren: Impact Model mit Ergebnissen der Kapitel, deren Effekte und Erfolgsfaktoren.

So zielt die Beantwortung der Forschungsfrage 1 in Kapitel 5 darauf ab, die Verständlichkeit von SoS zu erhöhen. Dieser Effekt wird auch im Kontext der Zielsystembeschreibung als wichtig eingestuft (siehe Tabelle 1 in Richter, Witt, Gesk & Albers, 2019a) referenzierend auf (Albers, Klingler & Ebel, 2013; Götz, Stich, Backhaus & Reinhart, 2016; Messerle, 2016; Posner, Keller, Binz & Roth, 2012). Die Veröffentlichungen nennen zudem basierend auf Friedenthal, Griego und Sampson (2007) MBSE als ein Ansatz für die Zielsystembeschreibung, wobei MBSE neben der Zielsystembeschreibung auch in Forschungsfrage 2 zur Systemmodellierung genutzt werden kann.

Anschließend wird in Kapitel 6 Forschungsfrage 2 beantwortet, welche die zwei Kriterien der Durchgängigkeit und Vollständigkeit in der modellbasierten Beschreibung von SoS untersucht. Mit Durchgängigkeit im Problem- und Lösungsraum ist gemeint, sodass Elemente im Problemraum wie Stakeholderbedarfe mit Elementen im Lösungsraum wie Funktionen direkt oder indirekt verknüpft sind. Daher wird mit Durchgängigkeit eine nachvollziehbare Verknüpfung aller Modellelemente geprüft. Zur Vollständigkeit wird in Kapitel 5 die für den Modellzweck benötigte Modellierungstiefe oder auch vertikale Modellierung sowie die Modellierungsbreite oder horizontale Modellierung geprüft. Ergebnis ist

eine sprachliche und methodische Modellierung, deren Anwendung in Forschungsfrage 3 geprüft wird.

In Kapitel 7 wird die Forschungsfrage 3 beantwortet, wobei die drei Kriterien der Effizienz, Durchgängigkeit und Kontinuität zu erfüllen sind. Bezogen auf die Validierung von V2I soll eine Flexibilität und eine Erweiterbarkeit der Ansätze möglich sein, um ausgehend von voraussichtlichen Veränderungen in dem SoS V2I Anpassungen durchführen zu können und mit wenig zusätzlichem Aufwand die Validierung zu ermöglichen. Flexibilität und Erweiterbarkeit beeinflussen somit die Effizienz, da das zu erzielende Ergebnis einer erfolgreichen Validierung von SoS mit einem möglichst geringen Aufwand realisiert wird. Wie bereits in der Beschreibung und Modellbildung ermöglicht die Durchgängigkeit eine nachvollziehbare Verknüpfung der Modellelemente, wobei hier nun zusätzlich die eingesetzten Validierungssystemelemente wie Prüfstand und Validierungsmethode hinzugezogen werden. Die Kontinuität beschreibt die Validierung unabhängig vom Zeitpunkt im Produktentwicklungsprozess. Somit sind Modelle, beispielsweise für eine virtuelle oder physische Validierung, kontinuierlich verknüpft, sodass eine „nahtlose“ Validierung realisiert wird. Dies führt zu einer Verknüpfung von verschiedenen Reifegraden an Modellen und weiteren Validierungssystemelementen wie Prototypen.

Die Beantwortung der Forschungsfragen muss auf die Realisierung der Effekte abzielen, was in dem jeweiligen Teilkapitel „Diskussion & Zwischenfazit“ geprüft wird. Die Prüfung sowie die Erzielung der Ergebnisse erfolgt in einer Forschungsumgebung, die nachfolgend beschrieben wird.

4.2 Forschungsumgebung

Die Forschung wird am Karlsruher Institut für Technologie (KIT) am Institut für Produktentwicklung durchgeführt. Hierbei lassen sich Validierungsumgebungen in Live-Labs, virtuell, gemischte mit Rollenprüfstand und mit Straße unterscheiden (siehe Abbildung 4.4).

Für die Beantwortung der Forschungsfragen in den zwei deskriptiven Studien werden unter anderem Live-Labs genutzt. Im Rahmen von Live-Labs führen Studierende sechs Monate ein Entwicklungsprojekt mit Industriepartnern durch (siehe Kapitel 2.2.2). In den Lehrveranstaltungen IP – Integrierte Produktentwicklung und ProViL – Produktentwicklung im virtuellen Ideenlabor in 2020 bis 2022 wurden SoS-Workshops durchgeführt. ProViL kann hierbei als Variante von IP bezeichnet werden, mit den wesentlichen Unterschieden: Anzahl der Entwicklungsphasen (4 vs. 5), Teilnahme virtuell vs. Präsenz und Aufwand

eingestuft in ECTS (4 vs. 16). Aufgrund der Ähnlichkeit der Lehrveranstaltungen und der Befragungen innerhalb der Lehrveranstaltungen in derselben Entwicklungsphase ohne Einfluss der Betreuenden sind die nachfolgenden Ergebnisse vergleichbar.

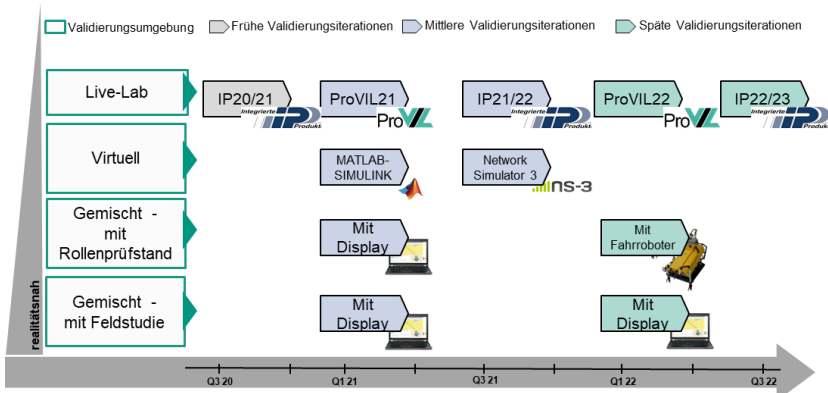


Abbildung 4.4: Zeitlicher Einsatz der Validierungsumgebungen in der Forschungsumgebung.

Weiterhin findet der Aufbau der Validierungsumgebung und die Validierung am Campus Ost des KIT am Fahrzeuggrollenprüfstand sowie dem Gelände des Campus Ost statt (siehe Abbildung 4.5).

Der genutzte Rollenprüfstand treibt das Fahrzeug zwei-achsig mit E-Maschinen über Scheitelrollen mit einem Durchmesser von 1,91m an. Die Rollen bringen eine Leistung von bis zu 220kW pro Achse auf. Das in dieser Arbeit eingesetzte Fahrzeug BMWi3 kann für Fahrmanöver mit geringen longitudinalen und lateralen Kräften über eine Zwei-Punkt Stangenfixierung befestigt werden. Für automatisierte Fahraufgaben wird der Fahreroter SAP2000 von Stähle GmbH eingesetzt. Dieser verfügt über drei Pedale (Fahrpedal, Bremse, Kupplung) und lässt sich ohne Veränderung am Fahrzeug auf dem Fahrersitz einbauen. Dabei können Messungen in Echtzeit mit Softwarelösungen wie AVL DRIVE TM aufgezeichnet werden. Wesentliche Messgrößen beziehen sich auf fahrzeugrelevante Größen und werden durch das CAN-Signal des Fahrzeug-BUS oder über Prüfstandsdaten gemessen. Der Prüfstand wird hierbei in der Straßensimulation betrieben. Der Straßensimulation liegt ein echtzeitfähiges Mehrkörpersimulationsmodell vom Fahrzeug zu Grunde. Dies berechnet basierend auf der Zugkraft des Fahrzeugs die Geschwindigkeit der Scheitelrollen. (Vgl. Düser et al., 2011)

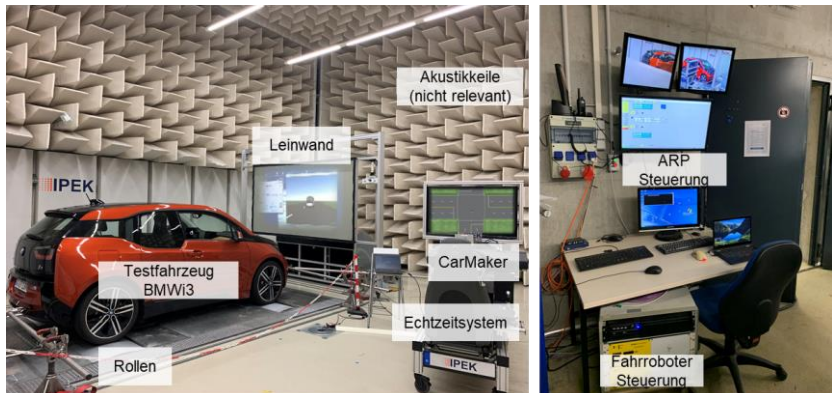


Abbildung 4.5: Übersicht des Akustikrollenprüfstands und der eingesetzten (Teil-)Systeme (vgl. Aufbau bezüglich Sensorstimulation in Albers, Lutz, Behrendt, Düser & Hettel, 2017; zu allgemeinen Fahrzeugprüfständen in Düser et al., 2011).

Im Rahmen von Live-Labs werden empirische Methoden eingesetzt, die nachfolgend vorgestellt werden.

4.3 Empirische Methoden

Die in Kapitel 5.1 genutzte empirische Studie beinhaltet drei Methoden, welche nachfolgend vorgestellt werden.

Die Durchführung der Methoden durchläuft dabei immer die drei Phasen: „Planung“, „Durchführung“ und „Dokumentation“ (Brereton, Kitchenham, Budgen, Turner & Khalil, 2007). Dieses Vorgehen für Literaturrecherchen wird auf empirische Methoden übertragen, wobei anstelle des Begriffs der „Dokumentation“ „Erkenntnisse“ verwendet werden, welche als Essenz der Dokumentation niedergeschrieben werden.

Eine **Umfrage** ist eine direkte Methode der Datenerhebung. In ihr wird die Perception von Menschen, Meinungen und Einstellung zu vordefinierten Fragen bestimmt. Aufgrund der vordefinierten, strukturierten Fragen sind diese wesentlich zur Auswertung der Daten, und können die Datenerhebung beeinflussen. (Leif Marxen, 2014).

In einem **Literaturreview** lassen sich empirisch erhobene Daten auswerten. Es existieren physische und virtuelle Datenquellen wie Protokolle, Prototypen, E-Mails.

Qualitativ erhobene Daten können mit der Methode der qualitativen Inhaltsanalyse aufbereitet werden. Quantitative Daten lassen sich mit Hilfe von Metriken aus den Datenquellen extrahieren. (Leif Marxen, 2014)

Das in dieser Arbeit angewendete **semi-strukturierte Interview** wird vor dem Interviewtermin vorbereitet und geplant. Zudem wurden den Interviewenden die Fragen zur Vorbereitung zugesendet. Die erhaltenen qualitativen Daten können meist nicht reproduziert werden, da die Interviewer nicht für weitere Interviews bereitstehen und möglicherweise andere Umweltbedingungen die Reproduzierbarkeit beeinflussen. (Heiser, 2018; Leif Marxen, 2014)

5 Beschreibung des generischen Zielsystems für die Validierung von V2I

Nach dem ZHO-Modell wird das Zielsystem bereits zu Beginn der Produktentwicklung initial bestimmt (vgl. Richter, Witt, Gesk & Albers, 2019b). Ebenfalls ist eine iterative Weiterentwicklung mittels Analyse und Synthese innerhalb der Produktentwicklung nötig, um Änderungen und Konkretisierungen des Zielsystems zu realisieren. Dies ermöglicht eine zielgerichtete Produktentwicklung, um den zuvor bestimmten Zweck zu erfüllen. Insbesondere in der Validierung müssen Elemente des Objektsystems mit Bedarfen und Anforderungen als Teil des Zielsystems abgeglichen werden. Nur dies ermöglicht die Bestätigung von beispielsweise Produktfunktionalitäten nach den Vorgaben der Anforderungen und nach den Bedarfen von Stakeholdern. Aus diesem Grund werden Bedarfe und Anforderungen als zwei wesentliche Elemente des Zielsystems bezeichnet (vgl. Ebel, 2015, S. 65) und in der vorliegenden Arbeit untersucht. Die Untersuchung erfolgt anhand des Forschungsgegenstands der Validierung von V2I, sodass kein spezifisches Zielsystem für ein spezifisches Produkt wie den Golf 8, sondern ein generisches Zielsystem untersucht wird.

Zur Identifikation von Bedarfen wird in Kapitel 5.1 eine empirische Studie durchgeführt. Dieser wird ein systematisches Literaturreview vorangestellt, um bereits bekannte Bedarfe zu ermitteln. Die empirische Studie enthält zur Datenerhebung Befragungen von angehenden Ingenieuren des Maschinenbaus und angrenzender Fachdisziplinen in Live-Labs sowie Expertenbefragungen. Die Analyse und die Erkenntnisse werden anschließend diskutiert.

Für die Spezifikation von Anforderungen wurde aus den Bedarfen der Wunsch nach Verständlichkeit bei der Beschreibung von SoS gefordert. Daher wird in Kapitel 5.2 ein Vorgehen erstellt, um ein SoS zu beschreiben und abzugrenzen. Basierend auf der Systembeschreibung werden anschließend Anforderungen an die Validierung von V2I formuliert.

Im Folgenden wird die Forschungsfrage 1 beantwortet: „Wie lässt sich das Zielsystem von V2I als SoS verständlich beschreiben und zu verteilten Systemen abgrenzen, um spezifische Herausforderungen in der Validierung von V2I zu identifizieren?“

5.1 Empirische Studie zur Bedarfsermittlung

Der empirischen Studie wird ein Literaturreview vorangestellt (siehe Abbildung 5.1). Dieses führt zur kritischen Betrachtung der Literatur basierend auf den zwei Datenbanken Web of Science und einer Datenbank mit ausgewählten Publikationen. Es werden diese zwei Datenbanken gewählt, da die CoreCollection von Web of Science (erste Datenbank) eine breite Abdeckung der Literatur ermöglicht, und somit Aussagen zur Gesamtbreite zulässt. Die gezielte Untersuchung in relevanten Veröffentlichungen mit der Software MAXQDA (zweite Datenbank) ermöglicht hingegen eine Literatursuche mit hoher Validität, da die Veröffentlichungen aufgrund ihres Ratings und ihrer Themenzugehörigkeit ausgewählt wurden.

Nach dem Literaturreview werden Befragungen in Live-Labs vorgestellt. Live-Labs stellen in der Produktentwicklung Umgebungen dar, welche die Untersuchung von wissenschaftlichen Methoden und Prozessen ermöglichen. Mehrere Studierende bearbeiten ein Entwicklungsprojekt, das in Kooperation mit einem Industriepartner durchgeführt wird. Hierbei soll durch den Methoden- und Prozesseinsatz ein Nutzen für die Studierenden entstehen. (Albers, Walter et al., 2018)

Zum Schluss werden die Ergebnisse der Befragungen von Experten vorgestellt.

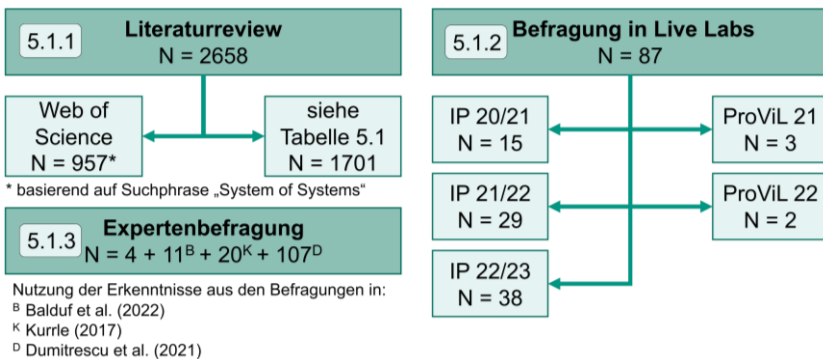


Abbildung 5.1: Übersicht der Erhebungsmethoden mit Datenquelle und Stichprobenanzahl (N).

5.1.1 Systematisches Literaturreview

Als erstes werden Rahmenbedingungen und der inhaltliche Fokus des Reviews erläutert. Anschließend erfolgen die systematische Suche und Auswertung. Nach der qualitativen Auswahl der Suchergebnisse, werden diese quantitativ ausgewertet. Daraufhin werden die Ergebnisse der Auswertung diskutiert und Erkenntnisse abgeleitet.

Planung – Rahmenbedingungen und inhaltlicher Fokus

Für das Literaturreview in Web of Science wurde die Core Collection gewählt ¹. Sie beinhaltet über 21.000 peer-reviewed Journals in 254 Themenkategorien (Clarivate, 2022). Die Suche wurde in allen Suchfeldern durchgeführt, sodass die Kategorien: Titel, Abstract, Volltext, Schlagworte in der Suche eingeschlossen sind.

Eine selektive Suche in Publikationen mit hoher Relevanz und Themenbezug ist basierend auf der Datenbasis ausgewählter Publikationen in Tabelle 5.1 durchgeführt.

Tabelle 5.1: Überblick der Datenbasis sortiert nach Anzahl an Publikationen

Typ	Publikation (für Journal: Impact Factor/SJR)	Zeitraum	Anzahl
Journal	Journal of Systems and Software (3,51/1,42)	2016 bis 2021	1187
	IEEE Intelligent Transportation Systems Magazine (5,3/0,99)	2016 bis 2021	411
	The Journal Systems Engineering (2,25/0,69)	2016 bis 2021	214
	International Journal of System of Systems Engineering (0,81/0,24)	2016 bis 2021	92
Konferenz	IEEE Systems Conference	2016 bis 2021	580
	IEEE International Conference on System of Systems Engineering	2016 bis 2021	422
	Tag des Systems Engineering	2016 bis 2021	175
Sonstiges	INCOSE International Symposium	2016 bis 2021	711
	Studien zu Advanced Systems Engineering ²	2018, 2021	2

¹ Bestimmte hierin enthaltene Daten gekennzeichnet mit WoSCC stammen aus Clarivate InCites. © Copyright Clarivate 202_. Alle Rechte vorbehalten.

² Albers, Dumitrescu, Marthaler et al. (2018); Dumitrescu, Albers, Riedel, Stark und Gausemeier (2021).

Durchführung – Systematische Suche und Auswertung

Tabelle 5.2 zeigt die Übersicht von Treffern zu Suchphrasen in den zwei Datenbanken.

Tabelle 5.2: Anzahl der Treffer für ausgewählte Suchphrasen in den Datenbanken Web of Science CoreCollection (WoSCC) und Tabelle 5.1 analysiert mit MAXQDA

#	Suchphrasen	Treffer WoSCC	Treffer MAXQDA
(1)	„System of Systems“ OR „System-of-Systems“	957	594
(2)	(1) AND Test*	108	55
(3)	(1) AND Validation	39	22
(4)	(1) AND (“Model-Based Systems Engineering” OR “Model Based Systems Engineering”)	22	5
(5)	(4) AND (Test* OR Validation)	3	3

Mehrere Einsatzgebiete verdeutlichen die Disziplinen³, in denen der Begriff SoS verwendet wird (siehe Abbildung 5.2). So wird neben „Ingenieurwesen E/E“, „Operations Research Management Wissenschaft“ und „Informatik Informationssysteme“ der Begriff in den Disziplinen „Ingenieurwesen Industrie“, „Ingenieurwesen Bauwesen“, „Umweltwissenschaft“ und „Automatisierungstechnik“ genannt. Dies verdeutlicht eine große Verbreitung des Begriffs in verschiedenen Disziplinen und den domänenübergreifenden Einsatz von SoS. Im Unterschied dazu kommen weitere Begriffe wie Cyber-Physical Systems (CPS) in WoSCC mit 5088 Publikationen sehr häufig vor, wobei sie in MAXQDA nur in 86 Dokumenten auftreten. CPS hat bis auf das Fehlen von Operations Research sehr ähnliche Disziplinen. So sind die häufigsten Disziplinen in WoSCC: E/E (1985), Informatik (1351), Telekommunikation (980) und Automatisierungstechnik (957). Dagegen tritt der Begriff Product-Service-Systems (PSS) nur in 737 Veröffentlichungen in WoSCC und acht Veröffentlichungen in MAXQDA auf. Hierbei zeigen die Disziplinen in WoSCC einen Trend zu Umweltwissenschaften: Umweltwissenschaft (276), Nachhaltigkeitstechnologie (272), Umweltingenieurwissenschaften (184), Fertigungstechnik (156), Management (126).

³ In Web of Science als „Feld“ bezeichnet, wobei diese Beschreibung im Folgenden durch die spezifischere Bezeichnung „Disziplin“ ersetzt wird.

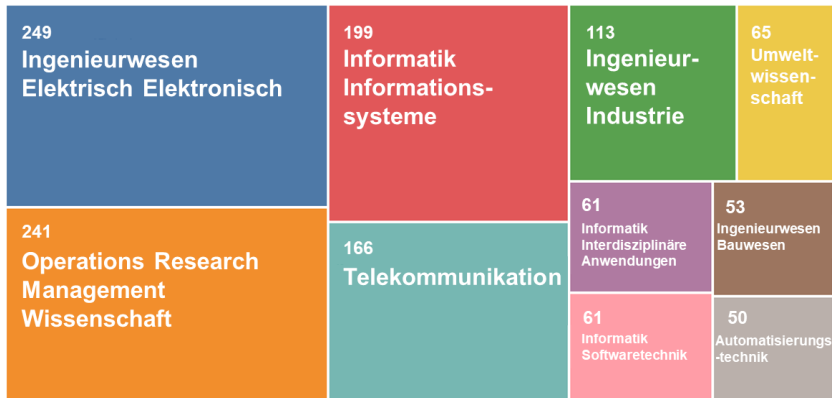


Abbildung 5.2: Identifizierte Themengebiete mit der Suchphrase „System of Systems“⁴, sortiert nach der Anzahl an Veröffentlichungen. Die Größe der Boxen steht im Verhältnis zu der Anzahl an Veröffentlichungen.

Für die Suche nach SoS und Validierung werden lediglich 36 Einträge gefunden. Dies verdeutlicht eine bislang geringe Forschung in der Validierung von SoS. Bei der Ergänzung um den im angloamerikanischen Raum und in Unternehmen bekannten Begriff „Test“⁴ (Trunkiert, um auch Unterbegriffe wie Testing zu finden) werden 108 Publikationen genannt. Dies deutet auf Forschung in Bezug auf das Testen von SoS hin. Abbildung 5.3 verdeutlicht, dass insbesondere in den Jahren ab 2018 ein stärkeres Forschungsinteresse an SoS aufkam.

⁴ Die Grafik des Zitationsberichts stammt von Clarivate Web of Science, Copyright Clarivate 202_.

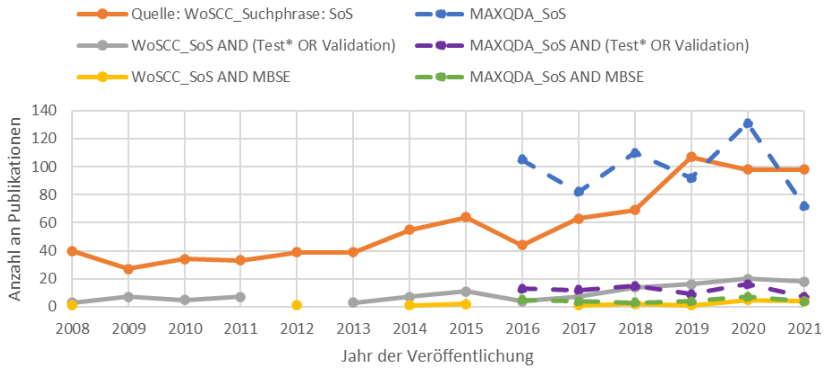


Abbildung 5.3: Anzahl an Publikationen nach Jahren bei den in der Legende genannten Quellen Web of Science Core Collection (WoSCC) und Tabelle 5.1 analysiert mit MAXQDA (MAXQDA) mit den darauffolgenden Suchphrasen, wobei SoS und MBSE ausgeschrieben und mit den Varianten mit Bindestrich und mit dem OR-Operator verbunden genutzt wurden (z.B. SoS OR System-of-Systems OR „System of Systems“).

In MAXQDA bleibt die Anzahl an Veröffentlichungen über die Jahre unverändert. Die Phrasenkombinationen mit „MBSE“ und „Test*“ und „Validierung“ ergeben in beiden Datenbanken eine geringe Anzahl über die Jahre. Insgesamt wurde in Kombination mit „Model-Based Systems Engineering“ der Begriff SoS lediglich 22-mal in WoSCC und fünfmal mit MAXQDA genannt.

Mit Fokus auf die Anwendung von Elektronik und Informationstechnik im Transportbereich wird häufig der Begriff „Intelligent Transport Systems“ (2183) verwendet, für Fahrzeuge im Speziellen auch „Vehicle-to-Everything“ (625), wohingegen die Begriffe „Car2X“ (13) und „Connected Cars“ (128) weniger geläufig sind. Daher wird in der vorliegenden Arbeit der Begriff Vehicle-to-Everything (V2X) mit dem Fokus auf Vehicle-to-Infrastructure (V2I) verwendet.

Die Möglichkeit ein System mit verschiedenen Begriffen zu beschreiben, birgt die Gefahr von Unklarheiten, welche zu Fehlinterpretationen und Missverständnissen führen können. Für Systeme gibt es die Begriffe „vernetzte Systeme“ (716), „verteilte Systeme“ (14569), „Cyber-Physical Systems“ (6687) und „Internet of Things“ (36159), welche in WoSCC häufig genannt werden (siehe Abbildung 5.11).

Die Untersuchung von postulierten Effekten und Erfolgsfaktoren in Kapitel 4.1 bezogen auf SoS mit der Suchphrase: „SoS AND Begriff“ zeigt, dass die Effekte Effizienz (effic* 62), Durchgängigkeit (end-to-end 49), Verständlichkeit (underst* 45)

und Kontinuität (continuit* 40) häufig genannt werden. Seltener kommen die Effekte Flexibilität (flexib* 20), Erweiterbarkeit (extend* 18) und Konsistenz (consist* 9) in Zusammenhang mit SoS vor. Synonyme wie Robustheit (robust* 19), Skalierbarkeit (scala* 14) können bezüglich der Begriffe Flexibilität respektive Erweiterbarkeit mit einbezogen werden.

Weiterhin wird deutlich, dass „Beispiele“ (60) häufig in den Veröffentlichungen in Bezug zu SoS eingesetzt werden. Diese beinhalten z.B. Roboterkommunikation (Smirnov, Kashevnik, Mikhailov, Mironov & Petrov, 2016) und Appartements (Kang & Park, 2016).

Erkenntnisse

Das Literaturreview verdeutlicht einen Trend in den letzten fünf Jahren zu mehr Veröffentlichungen im Themenfeld SoS (siehe Abbildung 5.3). Jedoch sind Kombinationen mit Test*, Validierung und MBSE nur einzeln veröffentlicht. Insbesondere die Kombination dieser drei Begriffe führt zu sehr wenigen Resultaten, wobei die Notwendigkeit die Validierung von SoS durch SE zu unterstützen in Dahmann (2014) explizit benannt wird (vgl. Dahmann & Roedler, 2016; vgl. Nielsen et al., 2015). Dies zeigt den Bedarf der vorliegenden Arbeit auf.

Die geringe Anzahl an Veröffentlichungen zu den Begriffen CPS und PSS in MAXQDA verdeutlicht den Fokus der selektierten Veröffentlichungen auf den Bereich SoS. Generell wird PSS nicht häufig verwendet, wobei in WoSCC sich das Auftreten des Begriffs von 2015 bis 2020 in etwa verdoppelt hat (von 41 zu 86). Der Begriff „CPS“ kommt in etwa fünfmal häufiger vor als „SoS“ (957 zu 5088). Dies zeigt eine breitere Anwendung des Begriffs „CPS“, wobei die Ähnlichkeit der Disziplinen zu einer Vernetzung der Begriffe führen kann.

In Veröffentlichungen zu SoS werden häufig Beispiele genutzt, um das System zu beschreiben. Dies verdeutlicht, dass die Konkretisierung für das Verständnis von SoS notwendig ist.

Im Rahmen einer Recherche im Automobil- und Produktentwicklungskontext konnten Anforderungen identifiziert werden, welche für SoS relevant sind. Um ein genaues Bild über das Verständnis von SoS-Experten und Experten der Produktentwicklung zu erhalten, wird nachfolgend eine empirische Studie in Live-Labs ausgeführt.

5.1.2 Befragungen in Live-Labs

Für eine Beschreibung der Live-Labs sei auf den Stand der Forschung und Technik in Kapitel 2.2.2 sowie auf die Forschungsumgebung der vorliegenden Arbeit in

Kapitel 4.2 verwiesen. Im Folgenden werden Planung, Durchführung und Erkenntnisse von Befragungen in Live-Labs vorgestellt (vgl. Brereton et al., 2007).

Planung

Die empirische Studie betrachtet die Forschungsfrage 1: „Wie lässt sich das Zielsystem von V2I als SoS verständlich beschreiben und zu verteilten Systemen abgrenzen, um spezifische Herausforderungen in der Validierung von V2I zu identifizieren?“ Zur Beantwortung der Forschungsfrage 1 werden zu Beginn zwei Hypothesen aufgestellt, welche Instrumente zum Verständnis von SoS sowie Methoden zur Adressierung von SoS im Kontext der Produktentwicklung ermöglichen:

1. „Bei der Beschreibung und dem Verständnis von SoS helfen Begriffsabgrenzung und -zuordnung, Typisierung, Beispiele“
2. „Methoden sind nützlich, um SoS geeignet adressieren zu können“

Zur systematischen Betrachtung wurde ein Studiendesign bestimmt, welches zwei Workshop-Vorträge, zwei Befragungen und eine Case Study beinhaltet. Zur Prüfung der Reliabilität und Reduzierung von zufälligen Fehlern wurde das Studiendesign mehrmals in Live-Labs angewendet. Die Befragung wurde innerhalb des Workshops mittels Fragebogen durchgeführt. Das mehrstufige Studiendesign dient dem Zweck zusätzliche Informationen den Workshopteilnehmenden zu präsentieren, um bei teilweise gleichbleibenden Fragen die Veränderung der Antworten zu identifizieren. Nachfolgend wird das Studiendesign vorgestellt:

1. *Workshop – Kurzeinführung:*
Basiert auf Grundlagen zu SoS wie den SoS-Typen (siehe Abbildung 2.4) und Anwendungsgebiete und Herausforderungen von SoS (siehe SEBoK, 2017).
2. *Initiale Befragung:*
Direkte und implizite Befragung in Papierform bzw. online über das Tool LimeSurvey (siehe Abbildung C.1 in Anhang C).
3. *Workshop – detaillierte Einführung:*
Mit weiterführenden Grundlagen wie mehreren Anwendungsbeispielen, einem Vergleich zu klassischen Systemen angelehnt an Abbildung 5.9 Spinnendiagramm mit Charakteristiken (siehe Abbildung 5.12, Abbildung 5.13)
4. *Case Study zum Industrieprojekt:*
Workshopteilnehmende untersuchen in Projektteams die erlernten Begriffe und Konzepte auf ihre Anwendbarkeit für ihre Projektideen.
5. *Finale Befragung:*
Direkte und implizite in Papierform bzw. online über das Tool LimeSurvey (LimeSurvey Project Team / Carsten Schmitz, 2012) (siehe Abbildung C.1 in Anhang C).

Als Zeitpunkt der Studie in der Lehrveranstaltung wird die Konzept- und Kreativitätsphase innerhalb der ersten Wochen gewählt. Hierbei ist eine lösungsoffene Ideenfindung möglich, sodass eine Systembetrachtung die Berücksichtigung wesentlicher Stakeholder und weitere Elemente sowie Schnittstellen ermöglicht.

Durchführung

Die Ergebnisse der initialen Befragung sind für die einzelnen Live-Labs in Abbildung 5.4 dargestellt. Die Live-Labs wurden zu den folgenden Themen durchgeführt:

- IP20/21: „Zukünftige Mobilität für Personen & Güter auf städtische Straßen“
- ProViL21: „XPAIRIENCE – Business opportunities around travel disruptions“ und „Steigerung des Fahrerlebnisses in Brennstoffzellenfahrzeugen durch effiziente Nutzung von Energie- und Stoffströmen“
- IP21/22: „Kundenorientierte, flexible Verpackungsprozesse (Maschinensysteme) für die Verpackung der Zukunft“
- ProViL22: „Baukörper der Zukunft“
- IP22/23: „Dachstuhl der Zukunft“

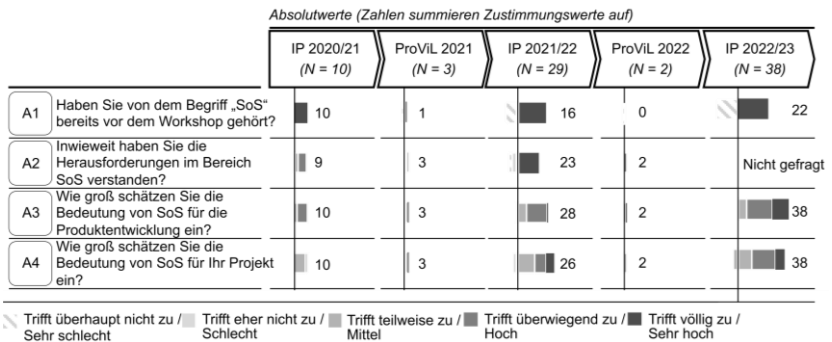


Abbildung 5.4: Antworten zur initialen Befragung für fünf Live-Labs.

Abbildung 5.5 veranschaulicht die Ergebnisse der finalen Befragung. Ein Spezialfall bildet IP 2022/23 mit einer komparativen Befragung mit Versuchsgruppe (VG) und Kontrollgruppe (KG). Hierbei erhielt die VG exklusiv Informationen zur Zielsystembeschreibung aus Kapitel 5.2. Eine Analyse dieser Auswirkungen wird in Kapitel 5.2 durchgeführt.

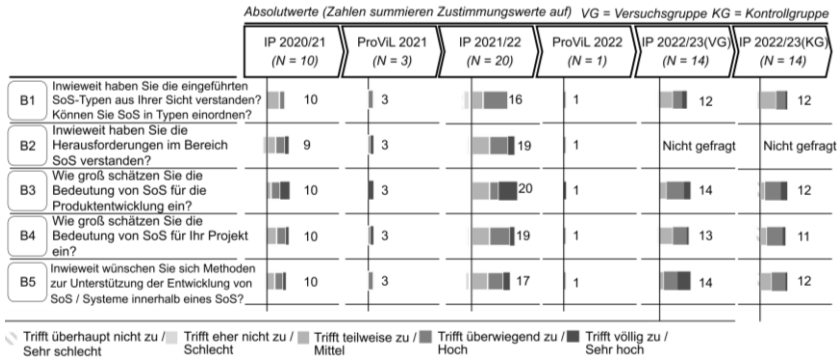


Abbildung 5.5: Antworten zur finalen Befragung für sechs Befragungen in Live-Labs.

Erkenntnisse

Aufgrund der niedrigen Teilnehmeranzahl ist die Aussagekraft gering bei den Befragungen innerhalb der Lehrveranstaltung ProViL. Dem wurde insofern Rechnung getragen, dass die nachfolgenden Erkenntnisse nicht aus den Befragungen der Lehrveranstaltung ProViL abgeleitet wurden, sondern diese dienen lediglich der Information.

Aus der *initialen Befragung* sind folgende Erkenntnisse ableitbar:

- Das Vorwissen zu SoS ist verteilt. Es haben im Schnitt über 50% bereits Vorwissen zu SoS, welches anhand der Rückmeldung in den Kommentarfeldern vornehmlich durch Lehrveranstaltungen am Institut für Produktentwicklung (IPEK) vermittelt wurde.
- Die Herausforderungen von SoS wurden als verstanden beurteilt.
- Die Bedeutung für Produktentwicklung und Projekt wurde überwiegend mit mittel bis hoch beurteilt.

Aus der *finalen Befragung* sind folgende Erkenntnisse ableitbar:

- Das Verständnis zu SoS nimmt von initialer zu finaler Befragung leicht ab (initial 3,4, final 3,1 von max. 5,0 als Mittelwert der Befragungen in IP). Gründe hierfür können ein nicht geeignetes Workshopkonzept sein oder zusätzliche Informationen, die zu einer neuen Bewertung der Thematik führen.
- Es bestehen Wünsche nach bedarfsgerechten Methoden (3,7), Leitfragen (3,9), Begriffsabgrenzungen (3,2) für ein besseres Verständnis von SoS.
- Die generelle Bedeutung von SoS in der Produktentwicklung wird erkannt (3,7).

Basierend hierauf lassen sich die Hypothesen evaluieren. Unter der Annahme, dass die Antworten den tatsächlichen Wunsch ausdrücken, wird Hypothese 1 bewertet. SoS-Typen, Leitfragen und Begriffszuordnungen mittels Venn-Diagramm werden als nützlich beurteilt. Ebenso kann Hypothese 2 verifiziert werden, da Methoden für ein besseres Verständnis von SoS als nützlich angesehen werden.

5.1.3 Expertenbefragung

Neben den Workshops mit Studierenden wurden Experten unterschiedlicher Themenfelder befragt.

Planung

Mit Hilfe von semi-strukturierten Interviews, Ergebnissen aus dem Literaturreview, sowie eines online Fragebogens wurden Experten befragt. In Abbildung 5.7 sind die Fachgebiete der Experten aufgeschlüsselt nach den vier Quellen dargestellt. Hierbei werden verschiedene Fachgebiete abgedeckt, wobei der Fokus auf Automotive liegt. Dieser Fokus entspricht dem, der vorliegenden Arbeit. Ganzheitliche Aussagen über Fachbereichsgrenzen hinweg lassen sich ausgehend von diesem Fokus nicht treffen.

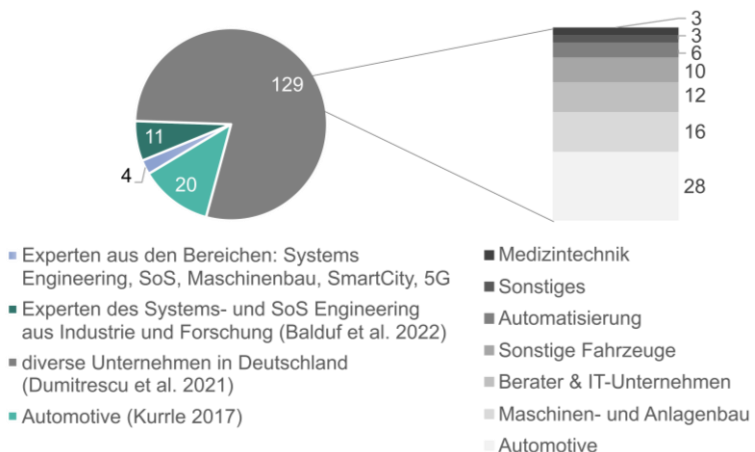


Abbildung 5.6: Übersicht über die Anzahl der durchgeführten Expertenbefragungen mit den jeweiligen Fachgebieten der befragten Experten aus den vier Quellen, wobei die 129 Interviews in Dumitrescu et al. 2021 rechts nach Fachgebieten aufgeschlüsselt sind.

In Abbildung 5.7 sind wesentliche Aussagen mit dem zugehörigen Fachgebiet des Experten illustriert, deren Durchführung und Erkenntnisse im Folgenden dargelegt werden.

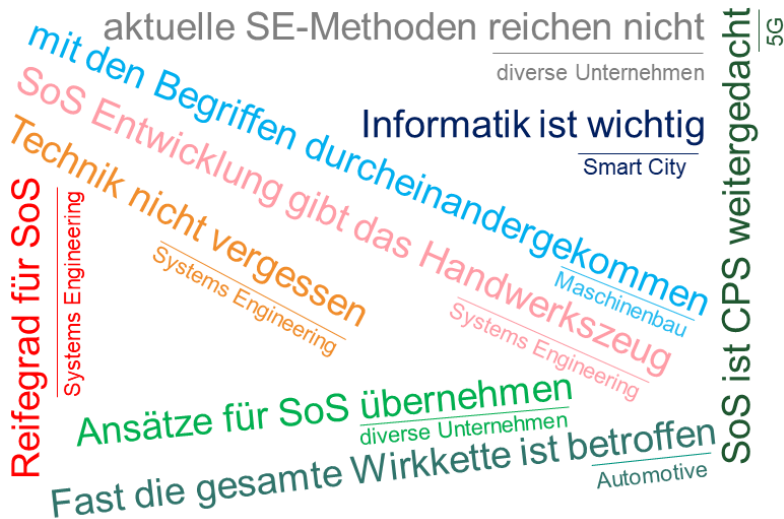


Abbildung 5.7: Übersicht wesentlicher Aussagen von Experten unterschiedlicher Fachgebiete.

Durchführung und Erkenntnisse

Es besteht teilweise das Verständnis über verschiedene Systembegriffe, allerdings ist die Unterscheidung nicht vollständig bekannt: „SoS ist das Konzept cyber-physical System ein bisschen weitergedacht.“ (Domänenexperten zu 5G aus Unternehmen) und „Da bin ich mit den Begrifflichkeiten SoS, System und Supersystem etwas durcheinandergekommen.“ (Domänenexperte im Maschinenbau aus Unternehmen⁵)

⁵ Ermittelt mit semi-strukturiertem Interview

Die Domäne der Informatik wird als wichtig angesehen: „*Das Gebiet der Informatik scheint wichtig zu sein.*“ (Paraphrasiert und übersetzt aus dem Englischen nach Domänenexperte zu Smart City an einem Forschungsinstitut⁵)

SoS sowie die Unterstützung der Validierung wird als sehr wichtig angesehen. Methoden, Leitfragen und Beispiele können hierbei viel helfen (Domänenexperte zu SoS an einem Forschungsinstitut⁶).

Expertenbefragungen von Mitarbeitenden bei der Dr. Ing. h.c. F. Porsche AG in Kurrle (2017) meinten zur Validierung für das Connected Car, dass Testen und die Verknüpfung mit Anforderungen große Herausforderungen sind. Zudem wird die Suche nach Fehlern als sehr schwierig beschrieben und die große Anzahl Stakeholder benannt: „*Fast die gesamte Wirkkette der Automobilunternehmen ist betroffen und fast alle Bereiche müssen ihre Prozesse anpassen.*“ (Kurrle, 2017)

Weiterhin wurden im Rahmen von Experteninterviews in der Studie zu Advanced Systems Engineering (Dumitrescu et al., 2021) Aussagen zu SoS getroffen. Es zeigt sich, dass der Begriff SoS teilweise bekannt ist und als relevant angesehen wird. Jedoch muss die technische Perspektive innerhalb eines SoS mitbetrachtet werden. „*System of Systems [... ist zurecht ein Schlagwort, das] uns im Markt, im Maschinenbau und im Fahrzeugbau auch heute schon beschäftigt und weiter beschäftigt. Aber [ich bin] der Meinung, wir sollten über diese ganzen schönen Buzzwörter hinaus, was da dahinter steht nicht vergessen, dass es auch noch die elementare Technik weiter geben muss und [sie] gebraucht wird*“ (Interviews im Rahmen von Dumitrescu et al., 2021).

Zudem wird die Produktentwicklung im Kontext SoS angesehen, um den Entwickelnden Handwerkszeuge an die Hand zu geben. „*Systems Engineering sehen wir mehr als diese Herangehensweise. Ich persönlich würde sagen, eher so ein bisschen dieses Mindset der Domain übergreifenden Produktentwicklung und im Kontext System of Systems Entwicklung und die Advanced Engineering gibt uns das notwendige Handwerkszeug dazu*“ (Interviews im Rahmen von Dumitrescu et al., 2021). Zudem wird die Notwendigkeit der Quantifizierung des Reifegrades des SoS-Levels genannt. „*Right now, we are finding out for example that we need to quantify the level of the readiness of the overall operation, the systems level, the system of systems level, [...]*“ (Interviews im Rahmen von Dumitrescu et al., 2021).

Weiterhin führten Balduf et al. elf Expertenbefragungen durch und leiteten Statements zu SoS ab. So lauteten sieben Statements, dass aktuelle Systems

⁶ Ermittelt mit online Fragebogen

Engineering Methoden für SoS nicht ausreichen. Zudem benennen zwölf Statements den Bedarf Ansätze zu übernehmen. (Balduf, Heinrich, Becker & Adler, 2022)

5.1.4 Zusammenfassung

Im Rahmen der empirischen Studie wurde festgestellt, dass das Thema SoS als abstrakt wahrgenommen wird. Mit Hilfe von Beispielen und Konkretisierungen sehen die Befragten einen Mehrwert für die Produktentwicklung.

- Es gibt **kein einheitliches Begriffsverständnis**, sondern Unsicherheit bzgl. der Begriffe.
- Es existieren **Bedarfe für Methoden, Leitfragen und Beispiele** für ein besseres Verständnis von SoS zur Unterstützung der Validierung von SoS.

Das Literaturreview und die empirische Studie zeigen den Bedarf an einem besseren Verständnis zu SoS auf. Es ist daher notwendig diesen Bedarf zu adressieren, um darauf aufbauend Zielsystemelemente wie Anforderungen an die Validierung von V2I abzuleiten.

5.2 Vorgehen zur Zielsystembeschreibung von V2I

Die in Kapitel 5.1 ermittelten Bedarfe zeigen den Wunsch nach einer verständlichen Beschreibung des SoS auf. Nach Aussage der Befragten können Beispiele und Begriffsabgrenzungen die Systembeschreibung unterstützen. Hieraus abgeleitet wurde ein Vorgehen zur Zielsystembeschreibung für V2I entwickelt, das zu drei Ergebnissen im Objektsystem führt. Abbildung 5.8 zeigt die drei Schritte mit deren jeweiliger Kapitelnummer an. Nachfolgend werden die drei Schritte vorgestellt.

Kapitel	Schritt	Ergebnis
5.2.1	Profilfindung	Produktprofil im Kontext SoS
5.2.2	Ermittlung von SoS-Charakteristiken	Spinnendiagramm der Ausprägungen von SoS-Charakteristiken
5.2.3	Implikationen für Aktivitäten der Produktentwicklung	Anforderungskatalog

Abbildung 5.8: Vorgehen zur Zielsystembeschreibung und Ableitung von Anforderungen für ein SoS.

In Schritt 1 wird das Produktprofil für SoS bestimmt (siehe Kapitel 5.2.1). Hierbei wird auf dem bestehenden Produktprofil aufgebaut (vgl. Albers, Heimicke, Walter et

al., 2018) und dieses für SoS erweitert. Beispielsweise sind SoS-Typen und Beschreibungen der CS für ein SoS wesentlich.

In Schritt 2 werden Ausprägungen der aus dem Produktprofil abgeleiteten wesentlichen SoS-Charakteristiken abgefragt und in einem Spinnendiagramm abgebildet (siehe in Kapitel 5.2.2).

Schritt 3 leitet aus den SoS-Charakteristiken und bisherigen Zielsystemelementen Anforderungen für die Validierung von V2I ab. (siehe Kapitel 5.2.3).

5.2.1 Profilfindung für V2I als SoS

Das Profil ist je nach Perspektive auf das jeweilige System zu bestimmen. Angelehnt an BS ISO/IEC/IEEE 21841:2019 können drei wesentliche Perspektiven auf das SoS V2I betrachtet werden (siehe Abbildung 5.9):

1. Das CS Fahrzeug als eigenständiges System,
2. Das CS Fahrzeug als System innerhalb von V2I und
3. V2I als SoS.

Weitere Perspektiven wie die Kommunikation einer Untermenge an CS in einem SoS oder das Mobilitätssystem als Übermenge von V2I sind nicht im Fokus der vorliegenden Arbeit.

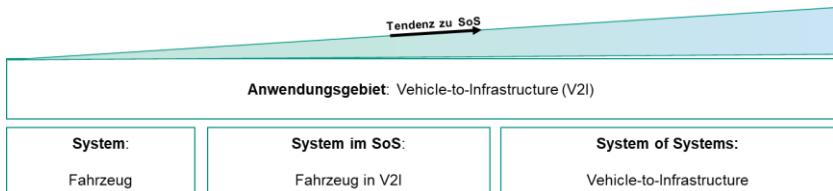


Abbildung 5.9: Drei Perspektiven auf das SoS V2I.

Nach M. Maier (1996) und Dahmann (2014) können SoS typisiert werden. Der Typ unterstützt in der zielgerichteten Betrachtung und im Vergleich verschiedener SoS. Hierbei ist nach Dahmann das Management entscheidend bei der Einteilung in einen der vier SoS-Typen (vgl. Kapitel 2.1.2). (Dahmann, 2014)

Nach Richter und Witt et al. (2019b) ist der Zweck einer Produkt- bzw. Systementwicklung frühzeitig zu identifizieren. Hierzu wird das Produktprofil entwickelt, um das SoS mit wesentlichen UseCases, Nutzen und Referenzsystemelementen zu. Dies veranschaulicht Abbildung 5.10 mit dreizehn Feldern ausgefüllt für das Produktprofil für V2I.

<h3>Produktprofil für V2I</h3>		
<p>Produktclaim V2I <i>Wir brauchen ein Produkt, dass das hochautomatisiertes Fahren durch Senden von Informationen über Frequenzen durch die Luft ermöglicht, um ein verkehrseffizientes, sicheres und bequemes Fahren zu ermöglichen</i></p>		
<p>Initiale SoS-Beschreibung</p> <ul style="list-style-type: none"> ■ SoS-Typ: Anerkannt ■ SoS-Eingrenzung: CPS // SoS-Abgrenzung: PSS, IoT ■ Fahrzeug Infrastruktur Kommunikation führt zu verbesserten Verkehrsentscheidungen 		
<p>Initiale CS Beschreibungen</p> <ul style="list-style-type: none"> ■ Fahrzeug, das eine OBU hat, welche Daten sendet und empfängt und an ein Steuergerät für das Fahrverhalten Daten versendet ■ Infrastruktur, die z.B. in Form einer Lichtsignalanlage (LSA) Informationen versendet 		
<p>Referenzsystemelemente</p> <ul style="list-style-type: none"> ■ Bus-LSA Kommunikation ■ Fahrerassistenz z.B. mit Monokamera 	<p>Use Cases CS in SoS</p> <ul style="list-style-type: none"> ■ Fahrzeug empfängt von der LSA Informationen zur Ampelphase und zum Phasenwechsel über WLAN ■ Das Fahrzeug kann die Trajektorie neu berechnen und die Beschleunigung anpassen 	
<p>Anbiaternutzen SoS</p> <ul style="list-style-type: none"> ■ Integration von Features in bekanntes Konzept ■ Daten aus Nutzungsphase sammeln ■ Stauvermeidung zur Wirtschaftlichkeit 	<p>Kundennutzen SoS</p> <ul style="list-style-type: none"> ■ Sichere, bequemere, verkehrseffizientere Fahrt 	<p>Anwieldernutzen SoS</p> <ul style="list-style-type: none"> ■ Sichere, bequemere, verkehrseffizientere Fahrt
<p>Wettbewerbssituation</p> <ul style="list-style-type: none"> ■ Mehrere Protokolle sind in unterschiedlichen Ländern zugelassen (EU: 802.11p) 	<p>Nachfrage SoS</p> <ul style="list-style-type: none"> ■ Passagiere und Fahrende, die in kurzer Zeit, bequem und sicher zwischen zwei Orten mit einem Fahrzeug transportiert werden möchten 	
<p>Validierung des ... durch</p> <ul style="list-style-type: none"> ■ Validierungsumgebung aufbauen ■ Verschiedene Tests in Validierungsumgebung zu Systemverhalten, Signalübertragung 		
<p>Rahmenbedingungen / Einschränkungen</p> <ul style="list-style-type: none"> ■ Gesetzliche Rahmenbedingungen (z.B. StVO) ■ Standards (z.B. 802.11p) 		

Abbildung 5.10: V2I-Produktprofil (angelehnt an Albers, Heimicke, Walter et al., 2018).

SoS-spezifische Felder sind:

- *Initiale CS Beschreibungen*: Das CS wird nach dem in Kapitel 2.1.2 eingeführten Standard zu SoS (BS ISO/IEC/IEEE 21839:2019) als System innerhalb eines SoS beschrieben. Demzufolge sind die Schnittstellen und Ressourcen zu wählen. In den initialen Workshops hat sich herausgestellt, dass insbesondere die Schnittstellen zu einem Verständniserfolg führen. Demzufolge sind die zu erzielenden Funktionen durch das CS mit den Schnittstellen zu weiteren CS zu beschreiben.

Zusätzlich erfolgen SoS-spezifische Ergänzungen zu dem allgemeinen Produktprofil (vgl. Albers, Heimicke, Walter et al., 2018) in den Feldern:

- *UseCases*: Hierbei sind Emergenzeffekte zu beschreiben, welche zu UseCases führen, die nicht von CS realisiert werden können. Dies verdeutlicht den Zweck des SoS und ist mit dem Feld *SoS-Claim* verknüpft.
- *Rahmenbedingungen*: Zur Realisierung eines SoS sind teilweise Managementaktivitäten notwendig sowie eine Zuordnung von systemeigenen Ressourcen zum SoS.
- *Referenzsystemelemente*: Wesentliche Referenzsystemelemente hängen von Domäne, SoS-Typ und Systembegrifflichkeiten ab. Demzufolge kann nachstehend die Begriffseinordnung im Feld *Initiale SoS Beschreibung* erfolgen. Das Referenzsystem sollte SoS-Beispiele zur Orientierung einer Einordnung und zum Verständnis enthalten. Hierbei lässt sich auf bestehende Listen wie auf SEBoK (2017) aufbauen. Das Referenzsystem enthält zudem weitere Systembegrifflichkeiten wie „Cyber-Physisches System“ und „Internet-of-Things“.
- *Anbiaternutzen*: Hierunter sind Stakeholder der CS zu verstehen, die gemeinsam das SoS für Kunden anbieten. Dies kann je nach Typ von SoS unbeabsichtigt (z.B. virtuelles SoS) oder kooperativ (z.B. anerkanntes SoS) erfolgen. Im Falle von V2I handelt es sich um ein anerkanntes SoS mit den Anbietern Fahrzeughersteller/-anbieter, Verkehrsleitzentrale und Staatlicher Instanz (siehe Kapitel 2.3.1)

Eine Begriffsabgrenzung und -einordnung hilft beim Verständnis und bei der Suche nach weiteren Referenzsystemelementen. Abbildung 5.11 verdeutlicht dies in einem Venn-Diagramm mit wesentlichen System-Begrifflichkeiten angelehnt an Henshaw (2016). Die Abbildung zeigt, dass jedes System Teil einer Umgebung ist, die nicht mit anderen Systemen interagiert. Der Umgebungsbegriff ist allgemein gefasst und nicht aus Sicht eines bestimmten Systems. Dies beinhaltet die Annahme, dass es kein System gibt, welches mit jedem System interagiert. In der Legende werden wesentliche Systemeigenschaften und Ableitungen aufgegriffen. So kann die Art der Systemgrenze unterschieden werden. Eine hohe Wahrscheinlichkeit an Veränderungen der Systemgrenze während des Systemlebenszyklus wird hier als weiche Systemgrenze bezeichnet. Insbesondere SoS, CPS und PSS haben

aufgrund der Vernetzung eine hohe Wahrscheinlichkeit sich während des Lebenszyklus zu verändern. Albers und Peglow et al. (2018) bezeichnen die Entwicklung von SoS als nie beendet aufgrund von kontinuierlicher Veränderung. Dies verdeutlichen Veränderungen in den Systemgrenzen aufgrund von Modifikationen sowie neu hinzukommende und wegfallende Systeme in einem SoS. Es kann von einer zeitvarianten, weichen Systemgrenze gesprochen werden. Dem gegenüber ist eine harte Systemgrenze bei klassischen, eigenständigen Systemen fest vorgegeben, da sie keine Vernetzung mit weiteren Systemen haben und keine Digitalisierungsanteile, sodass sie sich in ihrem Lebenszyklus nicht verändern.

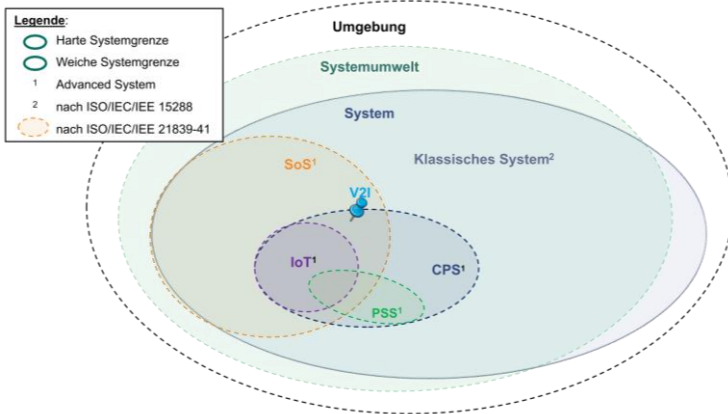


Abbildung 5.11: Venn-Diagramm verschiedener Systembegriffe (vgl. Henshaw, 2016).

- Die meisten Systeme sind nach Ropohl (2009) offene Systeme, welche eine Interaktion mit anderen Systemen aufweisen und daher Teil der **Systemumwelt** sind. Der Umweltbegriff ist immer aus Sicht eines spezifischen Systems benannt.
- Das **SoS** ist ein System, das in der Menge von Systemen enthalten ist. SoS umfasst mehrere vernetzte Teilsysteme, aber diese können auch eingebettete Künstliche Intelligenz enthalten. Der Schwerpunkt liegt demnach auf der Frage, wie man die Vernetzung der einzelnen Systeme realisiert, damit sie angemessen zusammenarbeiten. (Henshaw, 2016)
- **IoT** ist immer auch ein SoS, da es vernetzte eigenständige Systeme enthält. Zudem ist IoT immer ein CPS. (Henshaw, 2016)
- **CPS** kann ein SoS sein und ein SoS kann ein CPS sein. CPS ist die Verbindung von „cyber“ und physischen Aspekten von Systemen (Lee, 2015). CPS, welche nicht als SoS anzusehen sind, bilden vernetzte Systeme, welche nicht eigenständig sind.

- **PSS** verhält sich zu SoS gleich wie CPS zu SoS. Meistens beinhaltet PSS vernetzte eigenständige Systeme, sodass es SoS sind. Es können jedoch auch Service Systeme sein, die nur für das spezielle Produkt eingesetzt werden und nicht operativ eigenständig sind. PSS sind immer CPS, da der „cyber“ Anteil als Service und der Produkt Anteil als physisch in CPS vorkommt.
- **Advanced System** ist als Sammelbegriff für alle CPS und SoS anzusehen (vgl. Kapitel 2.1.4 und (Albers, Dumitrescu, Gausemeier et al., 2018)).
- **Verteiltes System** ist als Abkürzung des Begriffs standortverteiltes System zu verstehen. Dementsprechend wird die räumliche Distanz über eine Vernetzung realisiert.
- **Vernetztes System** meint eine Vernetzung von Systemen. Innerhalb des IPEK-XiL Ansatzes ist darunter die Vernetzung zwischen SiD/Sul (siehe Kapitel 2.2.2) und weiteren Systemen beispielsweise der Umwelt zu verstehen.

Es zeigt sich, dass verschiedene Systembegriffe existieren, die in ein Venn-Diagramm eingeordnet werden können (siehe Abbildung 5.11). In den Befragungen in Live-Labs wurde die Beschreibung mittels Venn-Diagramm als das Verständnis überdurchschnittlich verbessernd bewertet (IP21/22: 2,32; IP22/23: 2,43; Kontrollgruppe 2,36 Skala 1 bis 3). Weiterhin hilft das in den Lehrveranstaltungen IP und ProViL etablierte Produktprofil, den Zweck sowie Referenzsystemelemente und weitere Systeme frühzeitig zu betrachten. Hierzu wurde das Produktprofil um SoS spezifische Themen ergänzt.

5.2.2 Ermittlung von SoS-Charakteristiken für V2I

Abgeleitet aus dem Produktprofil sind die Charakteristiken des SoS zu bestimmen, um das spezifische SoS in der Produktentwicklung zu berücksichtigen. Bereits in der Ursprungsveröffentlichung zu SoS sind zentrale Charakteristiken von M. Maier (1996) benannt. Im Laufe der Zeit sind weitere Charakteristiken identifiziert worden, welche spezifischen SoS zuzuordnen sind und dem SoS gewisse charakterliche Eigenschaften zuweisen (vgl. Dahmann, 2014; Nielsen et al., 2015). Tabelle 5.3 listet Charakteristiken auf und bewertet sie angelehnt an Nielsen et al. (2015), um Doppeldeutigkeiten zu vermeiden und um den Fokus auf relevante Charakteristiken zu legen.

Tabelle 5.3: Vorauswahl der Charakteristiken für die Validierung von V21

#	Kriterium Charakteristik [Quelle] (Suchtreffer in MAXQDA)	Eindeutigkeit		Relevanz
		Score	Referenz	Bei eindeutigen Charakteristiken ▶ Suchtreffer > 5
1	Unabhängiges Management [1-3] (10)	✓		✓
2	Operative Unabhängigkeit [1-3] (14)	(✓)	#5	✓
3	Geographische Verteilung [1] (9)	✓		✓
4	Emergenzen [1-4] (109)	✓		✓
5	Autonomie [3, 4] (213)	-	#1,2	
6	Zugehörigkeit [4] (16)	-	#13,14	
7	Konnektivität [4] (72)	✓		✓
8	Heterogenität (99)	(✓)	#9	✓
9	Diversität [4] (79)	-	#8	
11	Evolutionär [3] (183)	(✓)	#12	✓
12	Dynamische Rekonfiguration [3] (12)	-	#11	
13	Interdependenz [3, 5] (24)	-	#6	
14	Interoperabilität [3] (61)	-	#6	
15	Gemeinsamer Zweck [5] (0)	-	#4	
16	Gemeinsame Architektur [5] (2)	✓		
17	Ausmaß, Größe & Black Box [5] (212)	✓		✓
18	Verschiedene Lebenszyklen [5] (7)	✓		✓

Für die Zuordnung von Charakteristiken zu einem SoS sind zuvor deren Ausprägungen zu bestimmen. Ausprägungen sind notwendig, da Charakteristiken unterschiedlich stark auftreten. So kann beispielsweise eine räumliche Entfernung zu einer anderen Prüfhalle, einem Bundesland oder einem Kontinent variieren. Dies hat Auswirkungen auf Implikationen der Charakteristiken wie einer echtzeitfähigen Vernetzung und sollte daher erfasst werden. Zudem zeigen M. Maier (1996) und Dahmann (2014) die Notwendigkeit einer Ausprägung anhand des unabhängigen Managements, die zu unterschiedlichen SoS-Typen führt. Die Ausprägung zeigt demnach, ob das Management nicht vorhanden (Typ: virtuell und kollaborativ), vorhanden (Typ: anerkannt) oder sogar weisungsbefugt (Typ: zentral ausgerichtet) ist.

Ein Spinnendiagramm (siehe Abbildung 5.12) beschreibt die Ausprägungen der Charakteristiken (vgl. Yan, Mandel, Behrendt & Albers, 2019). Der eigentliche Zweck des Spinnendiagramms ist dreigeteilt: die Beschreibung des SoS, der Vergleich unterschiedlicher SoS bezüglich ihrer Charakteristiken und die Implikationen auf Aktivitäten der Produktentwicklung. Es ist zu beachten, dass ein volles Spinnennetz nicht als besser oder schlechter einzuordnen ist. Somit ist ein Vergleich des Flächeninhalts zwischen SoS zwar möglich, jedoch ohne Aussagekraft, da sich die Charakteristiken und Ausprägungen unterscheiden. Die Ausprägungen sind in der Legende in Abbildung 5.12 genannt. Hierbei ist eine eindeutige, abgrenzbare Wahl der diskreten Ausprägungsschritte zu wählen.

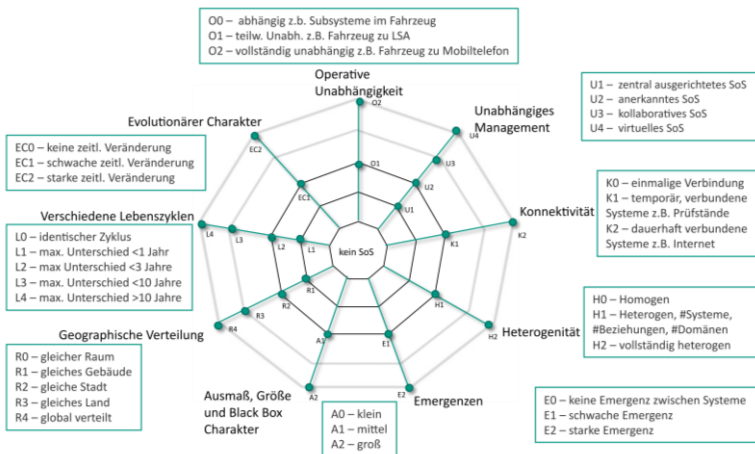


Abbildung 5.12: Charakteristiken des SoS V2I dargestellt als Spinnendiagramm (angelehnt an Albers, Mandel et al., 2018).

In der Befragung in IP21/22 (N=20) wurden die Ausprägungen der Charakteristiken für V2I durch Mittelwertbildung abgeleitet (siehe grüne Fläche in Abbildung 5.11). Die Mittelwertbildung ist notwendig, da die Befragten die Ausprägungen der Charakteristiken von V2I unterschiedlich eingeordnet haben. Die Standardabweichung verdeutlicht die Bandbreite der quantifizierten Antworten, sodass bei geringer Standardabweichung ein kausaler Zusammenhang zu einem einheitlichen Verständnis vorliegt.

In IP22/23 (N=14) wurde zuvor in der Versuchsgruppe die Vorgehensweise zur Zielsystembeschreibung inklusive Produktprofil hergeleitet, wobei eine Kontrollgruppe (N=14) dies nicht zur Verfügung hatte. Dies führte zu in Abbildung 5.11 in Blau mit Orange gestricheltem Rahmen (Versuchsgruppe) und Blau mit Rot gestricheltem Rahmen (Kontrollgruppe) dargestellter Fläche und Extremwerten.

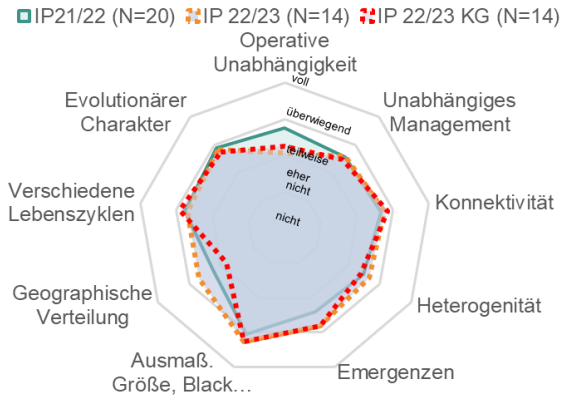


Abbildung 5.13: Ausgeprägte Charakteristiken von V2I aus den Befragungen in IP21/22, IP22/23 und IP22/23 (Kontrollgruppe).

Neben der Betrachtung von V2I lassen sich auch Systeme innerhalb des SoS betrachten. Hierzu können die zwei weiteren Perspektiven auf Systeme: System und System im SoS beschrieben werden. Abbildung 5.14 visualisiert die drei Perspektiven für V2I am Beispiel des Einzelsystems Fahrzeug.

	System: Fahrzeug	System im SoS: Fahrzeug in V2I	System of Systems: V2I
Operative Unabhängigkeit	Fahrzeug eigenständig betreibbar	Fahrzeug eigenständig betreibbar, Schnittstellen zu weiteren Systemen	Fahrzeug unabh. von LSA betreibbar
Unabhängiges Management	OEM	OEM mit Interaktion zu Stakeholder	Fahrzeug und Infrastruktur unabh. verwaltet z.B. OEM, Verkehrsplaner
Konnektivität	Bestimmt im Fahrzeugentwicklungsprozess	z.B. Anpassung von Android Schnittstelle	Schnittstellen zwischen Fahrzeuge, LSA
Heterogenität	Homogen	Homogen, Diversität an Schnittstellen und Backend	Heterogen, verschiedene Funktionen z.B. Transport, Stadtplanung
Emergenzen	Bestimmt im Fahrzeugentwicklungsprozess	Bestimmt im Fahrzeugentwicklungsprozess, Änderung über Zeit	Volle Funktionalität kommt erst durch das Operieren im SoS Kontext
Ausmaß, Größe, Black Box	OEM hat WhiteBox Sicht	WhiteBox Sicht + Schnittstellen nach Standards	Black Box Sichten auf andere Systeme
Geographische Verteilung	Fahrzeugteilsysteme an einem Ort	an einem Ort, Schnittstelle zu z.B. Fahrzeuge an anderen Orten	Fahrzeuge und Infrastruktur sind örtlich verteilt
Versch. Lebenszyklen	Zyklus ca. 18 Jahre	Ein Produktlebenszyklus, ändernde Systeme an Schnittstellen, Upgrades	mehrere Zyklen, <14Jahre bis >30 Jahre
Evolutionärer Charakter	mit Over-the-Air Updates	Updates/Upgrades von Umgebung, ggfs. Fahrzeug	Permanente Wechsel/Updates/Upgrades

Abbildung 5.14: Drei Perspektiven auf das SoS V2I und deren Charakteristiken.

Für jede Perspektive lässt sich demnach ein eigenständiges Spinnendiagramm erstellen, da auch eigenständige Systeme Ausprägungen der SoS-Charakteristiken aufweisen können.

Basierend auf den Charakteristiken lassen sich im folgenden Kapitel Aussagen zur Validierung von V2I treffen.

5.2.3 Implikationen auf die Validierung von V2I

Aus der Literatur wurden acht zentrale Herausforderungen für die Validierung von SoS identifiziert und in Abbildung 2.8 dargestellt (Albers, Peglow et al., 2018; Dahmann, 2014; Friedenthal et al., 2021; Nielsen et al., 2015). In der vorliegenden Arbeit wird auf das spezifische SoS V2I fokussiert. Demzufolge sind die Herausforderungen zu spezifizieren, um von diesen Implikationen in Form von Anforderungen an die Validierung von V2I abzuleiten.

Die Anpassung wird in Tabelle 5.4 aufgezeigt, in der der Einfluss von ausgeprägten Charakteristiken von V2I auf die acht zentralen Herausforderungen der Validierung von SoS dargestellt ist. Dies ermöglicht eine systematische und wiederverwendbare Priorisierung von Herausforderungen, die an die jeweilige Aktivität der Produktentwicklung (in diesem Fall Validierung) anpassbar ist.

Tabelle 5.4: Priorisierte Herausforderungen für die Validierung von V2I abgeleitet aus acht Herausforderungen für die Validierung von SoS und den ausgeprägten V2I-Charakteristiken

Herausforderungen für die Validierung von SoS	V2I-Charakteristiken							
	1. Vielfalt	2. Einheit / Modellierung	3. Testautomatisierung	4. Methoden, Verfahren, Tests	5. Integration in PEP	6. Ableitung & Pflege	7. Virtuell & reale Bedingungen	8. Betriebsicherheit
Operative Unabhängigkeit 69%								
Unabhängiges Management 64%								
Geographische Verteilung 56%								
Emergenz 60%								
Evolutionäres Verhalten 72%								
Autonomie 61%								
Zugehörigkeit 66%								
Vernetzung 67%								
Vielfältigkeit / Diversität 62%								
Untersch. Lebenszyklen 69%								
Summe	2,67	4,69	6,16	6,46	5,825	4,955	4,235	4,93

Legende

- = Bewertung 1
- = Bewertung 0,5
- = Bewertung 0
- = Stärke der Ausprägung

Tabelle 5.4 beinhaltet auf der linken Seite die ausgeprägten SoS-Charakteristiken als arithmetisches Mittel aus der Befragung in IP21. Der Einfluss der Charakteristiken wurde auf die Herausforderungen für die Validierung von SoS mit

{0;0,5,1} bewertet. Mittels Summenbildung der Produkte von Stärke der Ausprägung und Bewertung lassen sich die Gesamtbewertungen der Herausforderung bilden. Beispielsweise wird für die erste Herausforderung der Vielfalt gerechnet $64\%*1+72\%*1+62\%*1+69\%*1=2,67$.⁷

Aus den priorisierten spezifischen Herausforderungen für die Validierung von V2I werden nun Anforderungen abgeleitet (siehe Herausforderung zwei bis acht in Abbildung 5.15). Zu der ersten Herausforderung wurde aufgrund des geringen Wertes von 2,7 keine Anforderung erstellt.

#	Anforderung an die Validierung von V2I	Herausforderungen für die Validierung von SoS
A1	Bei der Validierung von V2I sollten Schnittstellen mit Hilfe von Systemmodellen analysiert und angepasst werden.	2. Einheitliche Modellierung 3. Testautomatisierung 4. Methoden, Verfahren, Tests
A2	Bei der Validierung von V2I sollte eine flexible, erweiterbare Validierungsumgebung eingesetzt werden, um die SoS-Charakteristiken geeignet zu modellieren darzustellen.	4. Methoden, Verfahren, Tests 7. Virtuelle und reale Bedingungen
A3	Bei der Validierung von V2I sollten Stakeholder, Elemente, CS und Umgebung kontinuierlich einbezogen werden, um die Charakteristiken des SoS geeignet zu berücksichtigen.	4. Methoden, Verfahren, Tests 5. Integration in PEP 6. Ableitung & Pflege
A4	Bei der Validierung von Funktionen in V2I muss die Funktionale Sicherheit berücksichtigt werden, um Standards einzuhalten und die Sicherheit zu zeigen.	8. Betriebssicherheit

Abbildung 5.15: Anforderungen an die Validierung von V2I unter Berücksichtigung von MBSE.

Die in Abbildung 5.15 beschriebenen Anforderungen fokussieren auf die Validierung von V2I unter Berücksichtigung des in der vorliegenden Arbeit relevanten Themenbereiches MBSE. Sie stellen jedoch nur ein Ausschnitt der Anforderungen

⁷ Die Multiplikationen mit dem Wert 0 wurden aufgrund der Übersichtlichkeit weggelassen.

dar, sodass beispielsweise Dahmann, Lane, Rebovich und Lowry (2010) weitere Anforderungen benennen (vgl. zu Funktionaler Sicherheit Pelliccione et al., 2020, S. 16). Dahmann et al. (2010) leiten hierbei aus den Charakteristiken unter Berücksichtigung des SoS Anforderungen an die Validierung des SoS ab. Dies verdeutlicht, dass die in der vorliegenden Arbeit untersuchten Anforderungen einen wesentlichen Teil des Zielsystems ausmachen. Jedoch liegt keine Vollständigkeit des Zielsystems vor, wie sie in Albers, Heimicke, Hirschter et al. (2018) durch die Anzahl an definierten Zielen zu der Gesamtanzahl an Zielen am Ende eines Projektes definiert ist (vgl. Costello & Liu, 1995).

5.2.4 Zusammenfassung

Die Bedarfe für eine verständliche Beschreibung und Einordnung von SoS sind bereits im Zielsystem zu adressieren. Hierzu wird ein Vorgehen mit den Schritten Profilfindung, Charakteristiken von SoS und Implikationen auf die Validierung von V2I angewendet. Dieses Vorgehen führt zu Anforderungen an die Validierung von V2I. Das Vorgehen ist iterativ auszuführen, da sich Anforderungen ändern und insbesondere aufgrund der zeitlich hohen Variabilität eines SoS Änderungen auftreten. Eine Übertragung des systematischen Vorgehens auf weitere SoS und weitere Aktivitäten der Produktentwicklung ist zukünftig zu prüfen und wird im Ausblick adressiert.

- Das **Vorgehen zur Zielsystembeschreibung zur Validierung von V2I** besteht aus den drei Schritten: Profilfindung, Ermittlung von SoS-Charakteristiken und Implikationen für die Validierung von V2I.
- Die **Profilfindung** erfolgt durch das Produktprofil für eine spezifische Perspektive auf das eigenständige System oder SoS. Hierbei werden SoS-Typen bestimmt und eine Begriffsabgrenzung vorgenommen.
- Die in einer Literaturanalyse akkumulierten **SoS-Charakteristiken** werden für den Themenbereich V2I konkretisiert und in einem Spinnendiagramm mit deren Ausprägungen dargestellt.
- **Implikationen** der ausgeprägten Charakteristiken und der acht zentralen Herausforderungen auf die Validierung von V2I werden mit Hilfe von Tabelle 5.4 bestimmt. Es lassen sich vier Anforderungen ableiten, welche die wesentlichen Charakteristiken und Herausforderungen adressieren und in den nachfolgenden Kapiteln untersucht werden.

5.3 Diskussion & Zwischenfazit

Zu Beginn wird in Kapitel 5.3 die Aussagekraft der Erkenntnisse aus der empirischen Studie mittels der Kriterien Validität, Reliabilität und Objektivität überprüft. Anschließend wird ein Fazit aus den Ergebnissen der Kapitel 5.1 und 5.2 gezogen. Zum Schluss wird der wesentliche postulierte Effekt der Verständlichkeit überprüft.

5.3.1 Limitierungen

Im Rahmen der empirischen Studie wurde die Erfüllung der Messgütekriterien Validität, Reliabilität und Objektivität angelehnt an Moosbrugger und Kelava (2012) geprüft, um Limitierungen der empirischen Studie zu prüfen.

Live-Labs kombinieren die Vorteile von Laborstudien mit einer mittleren bis hohen **internen Validität** und im Vergleich zu Feldstudien mit einer mittleren bis hohen **externen Validität**. Die Befragungen im Rahmen der Live-Labs wurden in einer frühen Phase der Produktentwicklung und somit zu einem ähnlichen Zeitpunkt im Produktentwicklungsprozess durchgeführt. Aufgrund der Ausweitung der Studie auf Experteninterviews wurden mehrere Personengruppen befragt, sodass es zu keiner Voreingenommenheit bei der Auswahl der Befragten kam. Die Gefahr von Mehrdeutigkeit von Aussagen wurde berücksichtigt und durch klare Aussagen, Erklärungen und die Möglichkeit von Nachfragen beim Workshopleiter sichergestellt. (Albers, Walter et al., 2018)

Die **Reliabilität** der Studie ist durch die Struktur des Fragebogens und durch Wiederholungen der Fragen im gleichen Live-Lab und in weiteren Live-Labs realisiert. Die Reproduzierbarkeit von Live-Labs kann generell als mittel eingestuft werden (Albers, Walter et al., 2018).

Die **Objektivität** wird in die drei Teilkriterien Durchführungs-, Auswertungs- und Interpretationsobjektivität unterteilt. Die Durchführungsobjektivität ist aufgrund des einheitlichen Fragebogens und des einheitlichen Workshopkonzeptes hoch. Es wird lediglich die Präsenz der Teilnehmenden und somit das Format des Fragebogens online und in Papierform unterschieden⁸. Die Auswertungsobjektivität ist durch den Einsatz von Multiple-Choice Fragen gegeben (Moosbrugger & Kelava, 2012, 9-10).

⁸ - Präsenztermin mit Fragebogen in Papierform (IP20/21)
- Onlinetermin mit online Fragebogen (ProViL 21, 22)
- Präsenztermin mit online Fragebogen (IP 21/22, 22/23)

Ausnahme bilden die Freitextfragen, welche aufgrund der klaren Aussagen gut ausgewertet werden konnten. Die Interpretationsobjektivität ist gegeben, da basierend auf der ordinalen Likert-Skala (Likert, 1932) und aufgrund der klaren Zweiteilung in der Visualisierung der Ergebnisse in den Abbildung 5.4 und Abbildung 5.5 eine objektive Interpretation ermöglicht wird.

5.3.2 Zwischenfazit zur Beantwortung der Forschungsfrage 1

Im Folgenden wird die Beantwortung der Forschungsfrage 1 zusammengefasst:

FF1: Wie lässt sich das Zielsystem von V2I als SoS verständlich beschreiben und zu verteilten Systemen abgrenzen, um spezifische Herausforderungen in der Validierung von V2I zu identifizieren?

Kapitel 5 stellt die Bedarfe der Validierung von SoS und im Speziellen V2I vor. Diese werden aus einem Literaturreview sowie einer empirischen Studie identifiziert. Die Bedarfe zeigen die Notwendigkeit einer Zielsystembeschreibung von V2I, die mit Hilfe von Profilfindung, Charakteristiken und Implikationen durchgeführt wird.

Zu Beginn wurden Bedarfe in einer empirischen Studie identifiziert und beschrieben. Hieraus wurde ein Vorgehen entwickelt, um das Zielsystem verständlich zu beschreiben.

In der Profilfindung wird das Produktprofil für SoS eingesetzt, um den Zweck, Nutzen und weitere Elemente frühzeitig zu benennen. Innerhalb der Profilfindung gilt es auch den Systembegriff und die Abgrenzung zu Begriffen wie verteilte Systeme durchzuführen. Laut Befragungen kann das Venn-Diagramm die Unterscheidungen der Systembegriffe gut darstellen.

Die Charakteristiken ermöglichen im zweiten Schritt des Vorgehens die Einordnung des SoS Begriffs und Abgrenzung zu weiteren Begriffen. Es wird jedoch deutlich, dass SoS Ausprägungen aufweisen. Mit Hilfe der Charakteristiken lassen sich auch weitere Systeme beschreiben. So hätte das verteilte System bei der geographischen Verteilung eine hohe Ausprägung. Über die Ausprägungen der anderen Charakteristiken muss der Einzelfall betrachtet werden. So bezieht sich Yan (2020) auf verteilte Prüfstände und es zeigen Albers und Mandel et al. (2018) die Beschreibung von Validierungsumgebungen anhand SoS-Charakteristiken.

Basierend auf den Bedarfen nach konkreten Handlungsempfehlungen für die Validierung von V2I wurde Schritt 3 des Vorgehens spezifiziert, um Herausforderungen für die Validierung von V2I und davon abgeleitet Anforderungen zu benennen. Es zeigt sich, dass Herausforderungen in der Literatur vorliegen, wobei diese für den Einzelfall von V2I priorisiert und konkretisiert werden. Die

benannten vier Anforderungen für die Validierung von V2I werden in den nachfolgenden Kapiteln 6 und 7 untersucht. Die als wesentlicher Effekt postulierte Verständlichkeit wird im Folgenden überprüft.

5.3.3 Diskussion des Effektes Verständlichkeit

In den Kapiteln 3 und 4 wird der Effekt „Verständlichkeit“ beschrieben (siehe Abbildung 4.3).

Der Effekt „**Verständlichkeit**“ **wird erreicht** durch eine Zielsystembeschreibung mit Hilfe des Produktprofils, welches SoS-spezifische Erweiterungen enthält. Mehr als 290 Produktprofile wurden in zwölf Projekten erstellt, was für eine verständliche Beschreibung spricht (Albers, Heimicke, Walter et al., 2018). Weiterhin werden die zehn SoS-Charakteristiken in einem Spinnendiagramm veranschaulicht, wobei die Fläche des Diagramms keine eigenständige Bewertung zulässt. Weiterhin wurde die Veranschaulichung im Diagramm als gut und verständlich bewertet (siehe Abbildung 5.12 und Abbildung 5.13). Zusätzlich bietet die Betrachtung unterschiedlicher Perspektiven des Systems eine Übersicht und eine verständlichere Beschreibung des SoS (siehe Abbildung 5.14). Die Ausprägungen der Charakteristiken variieren teilweise stark, sodass eine genaue Beschreibung der Charakteristiken mit Beispielen notwendig ist. Die Implikationen für die Validierung von V2I werden aus den ausgeprägten Charakteristiken von V2I und den zentralen Herausforderungen der Validierung von SoS hergeleitet. Das Vorgehen wurde als verständlich angesehen (siehe Abbildung 5.4 und Abbildung 5.5) und kann reproduzierbar für unterschiedliche Aktivitäten der Produktentwicklung ausgeführt werden. In der vorliegenden Arbeit werden Anforderungen für die wesentlichen Herausforderungen formuliert. Die Ableitung von konkreten Anforderungen wird als nützlich angesehen (siehe Abbildung 5.5). Eine Identifizierung und insbesondere Konkretisierung der Anforderungen ist jedoch nur über ein tieferes Systemverständnis möglich. Hierzu wird in Kapitel 6.1.2 der Begriff der Anforderung als Teil der Modellierung integriert und konkretisiert.

Der Effekt „**Verständlichkeit**“ **wird gemessen** durch eine direkte Befragung von Nutzergruppen im Rahmen einer empirischen Studie. Die Datenermittlung wurde durch Workshops mit Interview-Befragungen durchgeführt. Im Rahmen des Workshops in IP22/23 wurde mit Hilfe einer Kontrollgruppe die Verständlichkeit überprüft. Es zeigte sich, dass die Verständlichkeit durch die Anwendung der Vorgehensweise verbessert werden konnte (siehe Abbildung 5.5). So konnte das Verständnis zu SoS-Typen verbessert werden auf einer Skala von 1 (sehr schlecht) bis 5 (sehr gut) (Versuchsgruppe: $m=3,4$, $\sigma=1,0$, Kontrollgruppe: $m=3,3$, $\sigma=0,8$ Kurz : [3,4/1,0//3,3/0,8]). Weiterhin wird der Nutzen von Methoden zur Unterstützung der Entwicklung von SoS durch das Vorgehen höher bewertet [4,3/0,7//3,3/1,1].

Einzelne Hilfsmittel werden als nützlich eingestuft. Dies gilt insbesondere für Beispiele auf einer Skale 1 bis 3 mit den Werten [2,9/0,3//2,6/0,5], wobei neben den hohen arithmetischen Mittel die niedrigen Standardabweichungen auf eine breite Zustimmung deuten. Weitere Hilfsmittel wie Leitfragen [2,2/0,6//2,4/0,6] und Venn-Diagramm [2,4/0,7//2,4/0,6], werden ebenfalls als nützlich, mit einer breiten Zustimmung, eingestuft. Eine retrospektive Befragung des IP-Jahrgangs 22/23 zu der Nützlichkeit des SoS-Workshops in einer späteren Phase in der Produktentwicklung hat ein gutes Ergebnis mit einer hohen Standardabweichung ergeben [2,5/1,0]. Mittelwert und Standardabweichung sind hierbei plausibel, da der Mittelwert stark vom Projektinhalt beeinflusst werden kann, der in diesem Jahrgang bereits sehr fokussiert war. Die hohe Standardabweichung ist durch unterschiedliche Rollen der Entwickelnden z.B. Validierungs- und Konstruktionsingenieure zu erklären.

6 Modellierungssprache und -methode MEFS4SoS in MBSE zur Unterstützung der Validierung von V2I

Zur Umsetzung von MBSE gilt es die drei Säulen Sprache, Methode und Tool zusammen zu betrachten. Die vorliegende Arbeit beantwortet die Forschungsfrage 2: „Wie kann die durchgängige und kontinuierliche Validierung von V2I durch Modellierungssprache und Methode im Sinne von MBSE unterstützt werden?“. Somit liegt der Fokus auf Sprache und Methode, sodass das Tool „Cameo Systems Modeler“ als Enabler gilt, um die sprachliche und methodische Unterstützung zu visualisieren und dokumentieren.

Ausgehend von der Modellierungssprache SysML werden qualitative und quantitative Modelle erstellt, um die Validierung des SoS V2I zu unterstützen. Diese lassen sich in Abbildung 6.1 in die drei Abstraktionsebenen: „Generisch“, „Bereich“¹ und „System“ einordnen (Albers, Matthiesen et al., 2015). Modelle werden hierbei in die vier Abstraktionsgrade: „Meta-“, „Referenz-“, „Systemmodelle“ und „Reale Systeme“ klassifiziert (Albers, Matthiesen et al., 2015). In der vorliegenden Arbeit werden beginnend mit generischen SoS-Eigenschaften im Meta-Systemmodell, über domänenbezogene Modelle, Modelle für die Validierung von V2I erstellt und untersucht. In Abbildung 6.1 werden die betrachteten Modelle und deren Deduktion in den rot umrandeten Boxen dargestellt. Neben der Deduktion können mittels Induktion Systemmodelle erstellt werden. Dies ist beispielsweise bei einer detaillierten Betrachtung eines realen Systems als Element des Objektsystems möglich. Die Aktivitäten der Analyse und Synthese führen zu neuem Wissen, welches zu einer Anpassung oder Erstellung eines Systemmodells auf einem anderen Abstraktionsgrad führen kann. Beispielsweise führte die Analyse von realen Szenarien zu dem Standard OpenScenario, welcher im Bereich des hochautomatisierten Fahrens im Meta-Systemmodell aufgenommen ist. In der

¹ Unter Bereich wird in der vorliegenden Arbeit eine Gruppierung von Systemen verstanden. Ein Bereich kann firmenspezifisch (vgl. Muschik (2011)), domänenspezifisch (vgl. Bursac (2016)) oder anwendungsfallspezifisch (z.B. Hochautomatisiertes Fahren) sein. Die vorliegende Arbeit verwendet letzteres, um normenspezifische Sprachelemente einem Bereich zuzuordnen.

vorliegenden Arbeit erfolgt, angelehnt in der Kapitelstruktur, eine sequentielle Einführung der Systemmodelle von hohem Abstraktionsgrad zu niedrigem.

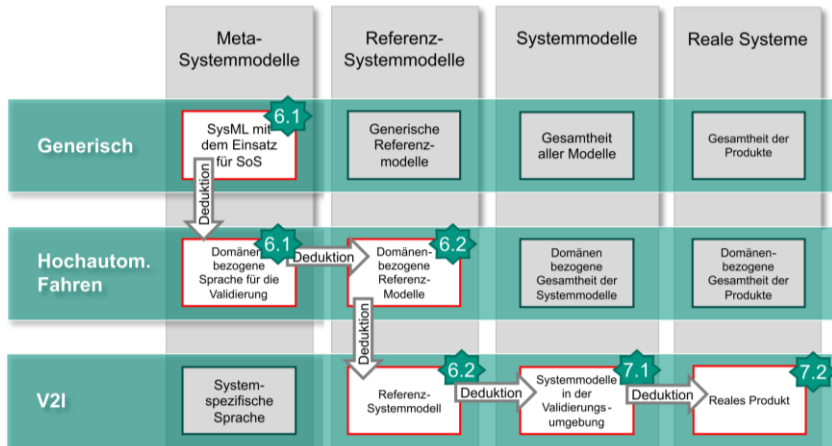


Abbildung 6.1: Abstraktionsgrade und Fokus der Arbeit in Rot umrandete Kästen mit Einordnung der nachfolgenden Kapitel (Albers, Matthiesen et al., 2015; Bursac, 2016; vgl. Muschik, 2011, S. 187).

Beginnend mit dem Meta-Systemmodell zeigen die nachfolgenden Beiträge, unterteilt in die drei Abstraktionsgrade, wie die Validierung von „generischen SoS“, „hochautomatisiertem Fahren“ und „V2I“ mittels Elemente einer Modellierungssprache und methodisch unterstützt werden kann. Basierend auf dem Meta-Systemmodell wird ein Referenz-Systemmodell für den Bereich des hochautomatisierten Fahrens in Kapitel 6.2 vorgestellt, welches Elemente mit ihren Beziehungen für deren Anwendung für die Funktionale Sicherheit beinhaltet. Hierbei werden Beispiele für V2I benannt. Der Bedarf für das Themenfeld Funktionale Sicherheit wird in Kapitel 3.1 beschrieben, in Kapitel 5 geprüft und in Anforderung 4 in Abbildung 5.15 festgehalten. Funktionale Sicherheit ist zudem wesentlich in der Validierung in dem Bereich des hochautomatisierten Fahrens (vgl. ISO 26262-1:2018(E); ISO/PAS 21448:2019(E)). Das Referenz-Systemmodell ermöglicht nun die Erstellung von Systemmodellen und realen Systemen, welche für V2I in Kapitel 7 beschrieben und für Validierungsaktivitäten genutzt werden. In Kapitel 6.3 werden die Elemente des Referenz-Systemmodells verwendet und die methodische Unterstützung beschrieben anhand der Methode „MEFS4SoS – Methode zur Erreichung der Funktionalen Sicherheit für die Validierung von SoS“.

Zur Erreichung des Ziels der Arbeit werden die in Abbildung 6.1 rot markierten Modelle untersucht. Weitere Modelle sind je nach Anwendungsfall nützlich. Beispielsweise ermöglichen generische (Referenz-)Systemmodelle und generische reale Produkte unternehmensübergreifende Modelle, die einen späteren Wissen- und Informationsaustausch erleichtern. Dies wird im Forschungsprojekt MoSyS untersucht (siehe Mandel, Guenther et al., 2022). Die Betrachtung der Modelle wird im Ausblick beschrieben. Disziplinspezifische Systemmodelle und reale Produkte sind nicht Fokus der vorliegenden Arbeit, die das spezifische SoS V2I untersucht. In Bezug zu V2I existieren kaum spezifische Sprachelemente, da diese übergreifend auf V2X-Ebene in der Disziplin hochautomatisiertes Fahren gebündelt werden. Gründe hierfür sind derselbe Standard der Nachrichtenübertragung 802.11p z.B. für V2V, V2N und die Anforderung der vorliegenden Arbeit die Validierung zu erweitern, sodass gemeinsame Sprachelemente im Fokus liegen.

6.1 Beiträge für die Modellierungssprache

Das generische Meta-Systemmodell ist die Basis für die Sprache und enthält Elementtypen der Modellierungssprache SysML für die Beschreibung von Systemen und demzufolge auch SoS (Holt & Perry, 2013b). Deduzierte bereichsspezifische Modelle beinhalten Erweiterungen beispielsweise in Form von Profilen, umgesetzt in Plugins für SysML (vgl. Baldwin (2008)). So ermöglicht das Plugin „Cameo Safety & Reliability Analyzer“ spezifische Sichten und Elementtypen für die Sicherheits- und Zuverlässigkeitsbetrachtung von Systemen. Enthalten sind hierbei Standards und Normen, welche meist bereichsspezifisch festgelegt werden. Beispielsweise listen Guidelines der UNECE Sicherheitsanforderungen für das hochautomatisierte Fahren auf (United Nations Economic and Social Council, 2022). Nachfolgend werden bezogen auf Funktionale Sicherheit Sprachelemente für ein Meta-Systemmodell für SoS beschrieben sowie wesentliche Schnittstellen genannt.

Abbildung 6.2 veranschaulicht den in dieser Arbeit betrachteten Ordnungsrahmen für die Elementtypen und deren Beziehungen für das generische Meta-Systemmodell, aufgeteilt nach den iPeM-Layern „Systemgeneration G_n “ und „Validierungssystem für die Systemgeneration G_n “. Der Layer weiterer Systemgenerationen ist Teil des Referenzsystems und wird in Kapitel 6.1.5 beschrieben. In der vorliegenden Arbeit werden angelehnt an das Produktprofil der Problemraum (engl. Problem Space (PS)) sowie die aus dem RFLP-Ansatz im Kontext von MBSE bekannten Perspektiven beschrieben: Anforderungen, Funktionen, Logische Elemente und Physische Elemente (vgl. Mandel, Guenther et al., 2022; Pohl et al., 2012). Nachfolgend werden die Perspektiven von Anforderungen bis logische Elemente für das generische Meta-Systemmodell

beschrieben. Die physischen Elemente werden im bereichsspezifischen Meta-Systemmodell vorgestellt, das für das hochautomatisierte Fahren den Fokus auf die Funktionale Sicherheit legt.

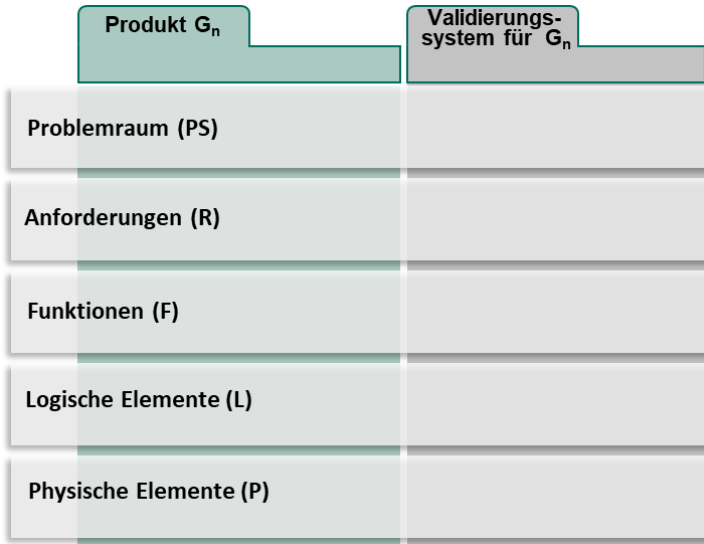


Abbildung 6.2: Ordnungsrahmen mit fünf Perspektiven des generischen Meta-Systemmodells aufgeteilt in zwei Layer des iPeM.

6.1.1 Elementtypen im Problemraum des generischen Meta-Systemmodells

In der Systemmodellierung wesentlich ist die Bestimmung des Systemkontexts (siehe Abbildung 6.3). Dieser beinhaltet Stakeholder wie Hersteller und Standards wie ISO 26262 und ist sowohl für das einzelne System als auch für das SoS mit mehreren Systemen anzulegen. Der <Systemkontext> ist mit einem oder mehreren Stakeholderbedarfen verknüpft. Beispielsweise ist ein <Bedarf> die sichere Autofahrt an ein zuvor festgelegtes Ziel. Ein Stakeholder ist hierbei nur dann zu modellieren, wenn ein <Bedarf> abzuleiten ist. Die <Bedarfe> können mit <UseCases> und diese wiederum mit einem <Validierungsbedarf> im Layer Validierungssystem verknüpft werden. Je nach Art der Beschreibung sind <Bedarfe> meist spezifisch für <UseCases> beschrieben (Kubin, Wagenmann, Reichert, Mandel & Albers, 2023). Die <Bedarfe> sind für jeden UseCase

notwendig, sodass ein UseCase für die Modellierung auch einen Bedarf befriedigt. Dies wird durch die Notation der Beziehung 1 zu 1..* benannt, welche zeigt, dass die Assoziation von einem Systemkontext auf ein bis beliebig viele Bedarfe führt und durch die Notation² 1 zu * ein Bedarf auf kein bis beliebig viele UseCases. Der <Validierungsbedarf> beschreibt eine Wissenslücke, welche durch die Aktivität der Validierung geschlossen werden soll. Der <Validierungsbedarf> kann daher auch von anderen Elementtypen als vom UseCase herkommen, wobei in der vorliegenden Arbeit der UseCase wesentlich zur Herleitung der Validierung ist und für weitere Verknüpfungen auf Kubin et al. (2023) verwiesen sein. Der <Validierungsbedarf> stellt somit einen Übergang zu weiteren Elementtypen im Validierungssystem dar (vgl. Mandel, Böning, Behrendt & Albers, 2021).

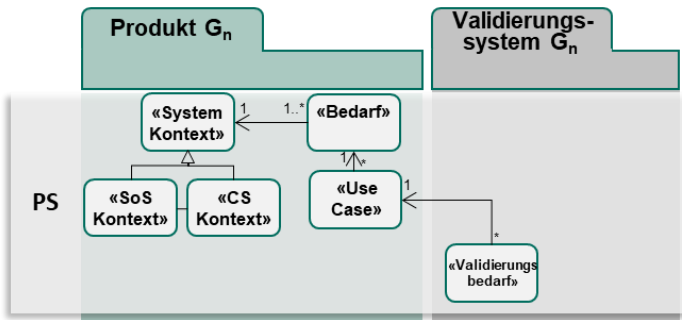


Abbildung 6.3: Auszug der verknüpften Elementtypen des Problemraums im generischen Meta-Systemmodell.

Tabelle 6.1 zeigt die Abhängigkeiten in SysML in einer generischen Tabelle als eine beispielhafte Sicht auf verwendete Elementtypen des Meta-Systemmodells. Hierbei sind die <Bedarfe> der jeweiligen Stakeholder, mit den zugehörigen <UseCases> zum <Validierungsbedarf> verknüpft. Beispielsweise wünscht sich der Fahrer einen automatischen Bremsvorgang, der in dem UseCase automatisches Bremsen bei einem Notfall formuliert ist. Dies führt zu dem <Validierungsbedarf>, den automatischen Bremsvorgang bei aktivierter ACC-Funktion zu evaluieren.

² $* \in \mathbb{N}_0$, $(1..*) \in \mathbb{N}$; $(2..*) \in (\mathbb{N} \geq 2)$; $2^* \in (2 * \mathbb{N})$

Tabelle 6.1: Auszug der verknüpften Elementtypen des Problemraums am Beispiel von GLOSA

Bedarf	Stakeholder	UseCase	Validierungsbedarf
Automatische Bremsung	Fahrer des betrachteten Fahrzeugs Fahrer anderer Fahrzeuge	Automatisches Bremsen bei einem Notfall	144 Automatisches Bremsen während einer Fahrt mit aktiviertem ACC
GLOSA für den Verkehr	Fahrer anderer Fahrzeuge Fahrer des betrachteten Fahrzeugs	Fahren mit aktiviertem GLOSA-Algorithmus außerorts Fahren innerorts	138 GLOSA Verkehrsteilnahme 135 GLOSA Effizienz
GLOSA für Sicherheit	Fahrer des betrachteten Fahrzeugs Verkehrsteilnehmer	Fahren mit aktiviertem GLOSA-Algorithmus außerorts Fahren innerorts	134 GLOSA Sicherheit

6.1.2 Elementtypen von Anforderungen des generischen Meta-Systemmodells

Abbildung 6.4 zeigt die im Rahmen der Arbeit wesentlichen Elementtypen der Perspektive Anforderung mit ihren Beziehungen zu dem Problemraum. Neben der im bereichsspezifischen Meta- und Referenz-Systemmodell angewendeten <Validierungsmethode> (siehe Kapitel 6.1.6, 6.3 und 7), werden nachfolgend die drei Elementtypen: <Anforderung>, <Vertrag> und <Validierungsziel> beschrieben.

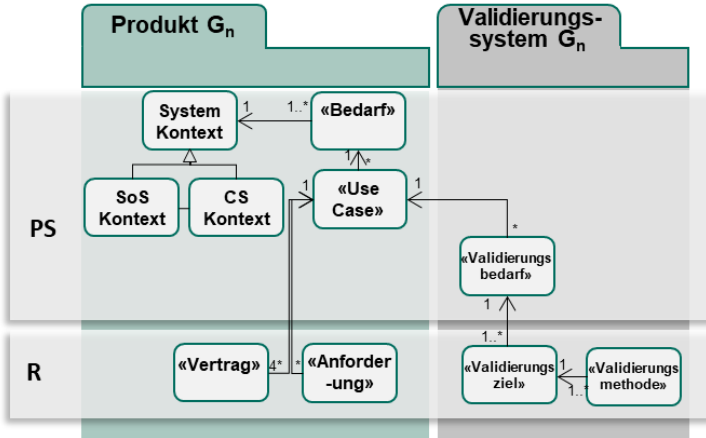


Abbildung 6.4: Sprachliche Elementtypen von Problemraum und Anforderungen des generischen Meta-Systemmodells.

Abbildung 6.5 verdeutlicht die Beschreibung von <Anforderungen>, welche meist individuell bzw. unternehmensspezifisch sind. Diese sind nicht SoS spezifisch, wobei verschiedene Anforderungsarten und -listen organisatorisch

zusammengefügt werden müssen. ASPICE beschreibt, dass Systemanforderungen mit ihrem Bezug zu Elementen betrachtet werden müssen und eine bidirektionale Nachvollziehbarkeit (engl. Traceability) realisiert werden muss (VDA QMC Working Group 13 / Automotive SIG, 2017, S. 41). Es gibt je nach Art der <Anforderung> eine Vernetzung auf unterschiedlichen Perspektiven zu den Elementen <Funktion> und <logisches Teilsystem>.

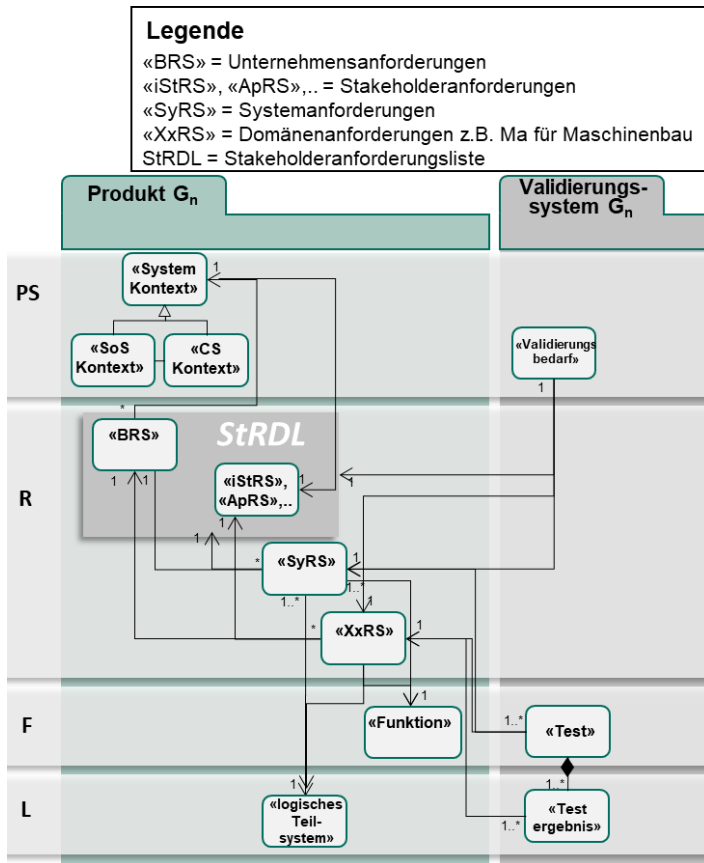


Abbildung 6.5: Ergänzung der sprachlichen Elementtypen von Anforderungen zu den Perspektiven Funktion und logische Elemente des generischen Meta-Systemmodells (angelehnt an Wäschle, Timucin et al. (2021)).

Für die Identifizierung von Unternehmens- und Stakeholderanforderungen wird der <System Kontext> benötigt, welcher die relevanten Systeme in der Umgebung beschreibt. Kubin et al. (2023) bestimmen sogar die Unternehmensanforderungen

im Problemraum, da diese frühzeitig genannt werden und eng mit dem Systemkontext vernetzt sind. Da Unternehmensanforderungen im Problem- oder Lösungsraum eingeordnet werden können, werden in der vorliegenden Arbeit die Unternehmensanforderungen in der Perspektive Anforderung als Teil des Lösungsraum dargestellt. Um die Auswirkungen von Anforderungen auf weitere Perspektiven wie die Funktion betrachten zu können, muss die Traceability innerhalb der weiteren Elemente gewährleistet sein. Die hieraus folgende Verknüpfung wird mithilfe einer Klassifizierung von Anforderungen realisiert. Je nach Anforderungsart müssen diese zu weiteren Elementen unterschiedlich verknüpft werden. Im vorliegenden Beispiel wird die Anforderungsklassifizierung für mechatronische Systeme nach Krusche angewendet (Krusche, 2000):

- Funktionale Anforderungen richten sich an eine zu erfüllende <Funktion> und sind mit dieser verknüpft. Die <Funktion> kann wiederum mit einem logischen Teilsystem umgesetzt werden. Die funktionale Anforderung wird anschließend mit den realisierenden Wirkprinzipien und dem <Testfall> verknüpft.
- Gestaltungsanforderungen und (Lean) Constraints haben in der Regel keinen direkten Einfluss auf eine <Funktion> des Systems. Beispielsweise ist die Farbe eines Bauteils eine Gestaltanforderung, die meistens unabhängig von einer Funktion realisiert werden kann. Beispielsweise haben die Farben eines Anzeigeelements im Fahrzeug zur Anzeige der LSA keinen direkten funktionalen Einfluss. Daher werden Gestaltungsanforderungen mit dem logischen Element wie das betroffene <logische Teilsystem> verknüpft. Gestaltungsanforderungen müssen mit einem verknüpften <Testfall> verifiziert werden. Im Gegensatz dazu sind Constraints Randbedingungen, welche in der Regel keinen eigenen <Testfall> haben. So kann es beispielsweise vorgegeben sein, dass bei Warnmeldungen eine gelbe Anzeige auftritt.

Abgeleitete <Anforderungen> machen einen großen Teil der Systemanforderungen aus und werden beispielsweise in der ISO 26262 unter kaskadierende Fehler beschrieben, welche sich zwischen den Hierarchieebenen der (Teil-)Systeme vererben lassen. Zum einen kann eine <Anforderung> aus einer anderen Systemanforderung direkt abgeleitet werden. Beispielsweise ist in Tabelle 6.2 die Systemanforderung 165.1 nach der minimalen Geschwindigkeit für die Überquerung einer Verkehrskreuzung von der Richtgeschwindigkeit abgeleitet. Neben einer direkten Ableitung kann eine <Anforderung> durch die Konkretisierung des Systemmodells festgelegt werden. Beispielsweise ist die Domänenanforderung nach der Übertragungsreichweite der OBU erst durch die Konkretisierung des Systems erstellt worden. Die Verknüpfung von <Anforderungen> zu weiteren Elementen verdeutlicht Tabelle 6.2. In der Tabelle sind verschiedene Arten von

<Anforderungen> mit <Funktionen>, logischen Elementen, <Validierungsbedarfe> und Stakeholder vernetzt. Beispielsweise ist die Europäische Union als Zielmarkt für eine Anforderung vom Unternehmen vorgegeben. Dies führt zu Auswirkungen auf alle Modellelemente und zu dem Validierungsbedarf der Einhaltung der EU-Richtlinien.

Tabelle 6.2: Beispiel der Anforderungsarten Unternehmensanforderung (BR), Stakeholderanforderung (SR), Systemanforderung (SysR), mit Verlinkung zu Validierungsbedarf, Funktion und logischem Element

Anforderung	Validierungsbedarf	Funktion	Log. Element
207 BR_Zielmarkt_EU	220 Einhaltung_der_EU_Richtlinien	Model	Unternehmen
205 BR_Zielapplikation_GLOSA	221 Validierung_des_UseCase_GLOSA	GLOSA	Unternehmen
SR_automatischesFahrst 239 em_dynamsicheFahraufga be_inODD_auszuführen		GLOSA	UNECE
165 SysR_GLOSA_v_pass_max	134 GLOSA Sicherheit	GLOSA Braking Infrastruktur/ Roadside Unit Onboard Unit	Gesetzgebung_ Deutschland
165.1 SysR_GLOSA_v_pass_min	135 GLOSA Effizienz	GLOSA Braking Infrastruktur/ Roadside Unit Acceleration Onboard Unit	165 SysR_GLOSA_v _pass_max

Für die wesentlichen Schnittstellen zur Verbindung von CS innerhalb eines SoS sind <Anforderungen> nicht bindend genug. Hierzu gibt es den <Vertrag>, welcher Schnittstellen bereits in der frühen Phase der Entstehung eines SoS spezifiziert.

Um die Notwendigkeit des <Vertrags> zu zeigen, werden vier Sichten darauf definiert. Angelehnt an Faldik et al. (2017) werden die vier Viewpoints³ auf Schnittstellenverträge definiert, die in Tabelle 6.3 mit Beispielen aufgelistet sind. Der Fahrer als eigenständiges System wird durch einen Fahrroboter realisiert. Der Fahrroboter wandelt die Umgebungs- und Fahrzeuginformationen in festgelegte Pedalpositionen um. In der Vertragsprotokoll Sicht geben die Vertragsregeln die Einschränkungen jeder Funktion an, die mit dem <Vertrag> der betreffenden Schnittstelle zusammenhängen.

³ Viewpoints spezifizieren Regeln zur Realisierung eines Views (deutsch: Sicht) (siehe Kapitel 2.1.3)

Tabelle 6.3: Viewpoints für das Meta-Systemmodell auf Produktebene (Faldik et al., 2017; Wäschle, Behrendt et al., 2021))

Name	Zweck des Viewpoints
Contractual SoS Definition Viewpoint	Vertrag zwischen CS Fahrzeug und CS Fahrer
Contract Conformance Viewpoint	Dieser Vertrag regelt die Schnittstelle zwischen CS Fahrzeug und CS Fahrer. Der Vertrag ist nur erfüllt, wenn beide Schnittstellenfunktionen erfüllt sind.
Contract Connections Viewpoint	Die Schnittstellenfunktion 1 wird durch die OBU des Fahrers und die OBU des Fahrzeugs erreicht; die Schnittstellenfunktion 2 wird durch den Fahrroboter und das Pedalsystem erreicht.
Contract Definition Viewpoint	Geschwindigkeitskontrolle: Der CAN BUS sendet die aktuelle Fahrzeuggeschwindigkeit mit einer festen Frequenz als Inputgröße an den Fahrroboter, der bei Bedarf eine neue Geschwindigkeit einstellt. Zustandsvariablen: Frequenz für Senden und Empfangen Invariante des Zustands: Geschwindigkeit in jedem Zyklus

<Anforderung> und <Vertrag> beschreiben somit die Forderungen an das System bzw. SoS. Die Forderungen an das Validierungssystem können aus dem <Validierungsbedarf> in einem <Validierungsziel> abgeleitet werden. Beispielsweise ist der Bedarf an die Validierung für V2I eine Überprüfung der Funktionalen Sicherheit. Dies ist nun mit dem <Validierungsziel> verknüpft, welches benennt, wie die Funktionale Sicherheit am Rollenprüfstand quantitativ zu belegen ist. Das <Validierungsziel> ist somit konkreter als der <Validierungsbedarf>, der sich im Problemraum befindet. Die Einordnung des <Validierungsziels> erfolgt nun mit der Verknüpfung zu weiteren Perspektiven, welche die nachfolgenden Kapitel beginnend mit der Funktion beschreiben.

6.1.3 Elementtypen von Funktionen des generischen Meta-Systemmodells

<Funktionen> lassen sich mit Hilfe von Schnittstellen beschreiben, angelehnt an das dritte Axiom von Newton zusammengefasst in der Aussage „actio gleich reactio“ (Vgl. Albers & Matthiesen, 2002). Für SoS können durch Emergenzeffekte neben den in einzelnen CS realisierten <Funktionen> weitere <Funktionen> realisiert werden, beispielsweise die Warnung des Fahrzeugs durch Informationen des Verkehrssystems. Somit lassen sich einige <Funktionen> für SoS auf mehrere Systeme aufteilen. Dies führt zu dem Problem, das mindestens nach der

Funktionsbeschreibung CS spezifische eigene Modelle eingesetzt werden können, wobei Stakeholder keinen freien Wissensaustausch von unternehmensgeschützten Informationen in dem zentralen SoS Modell ermöglichen. Daher sind für das SoS Black Box Modelle notwendig, die auf die entsprechenden unternehmenseigenen Modelle verweisen und die Funktionalität beschreiben. Die Black Box Modelle können mit Hilfe von <Variablen> realisiert werden, welche Input und Output einer Funktion sein können (siehe Abbildung 6.6). Angelehnt an Modellierungskonzepte in der Informatik (vgl. Object Management Group, 2022)) lassen sich <Variablen> eines CS in „benötigt“ und „bereitgestellt“ kategorisieren (Wäschle, Imamovic, Rapp & Albers, 2022). Somit kann ein CS eine <Variable> bereitstellen und ein weiteres CS innerhalb des SoS diese anfordern und verwenden. Diese Unterscheidung ermöglicht es <Variablen> frühzeitig zu modellieren und somit die Schnittstelle zu spezifizieren.

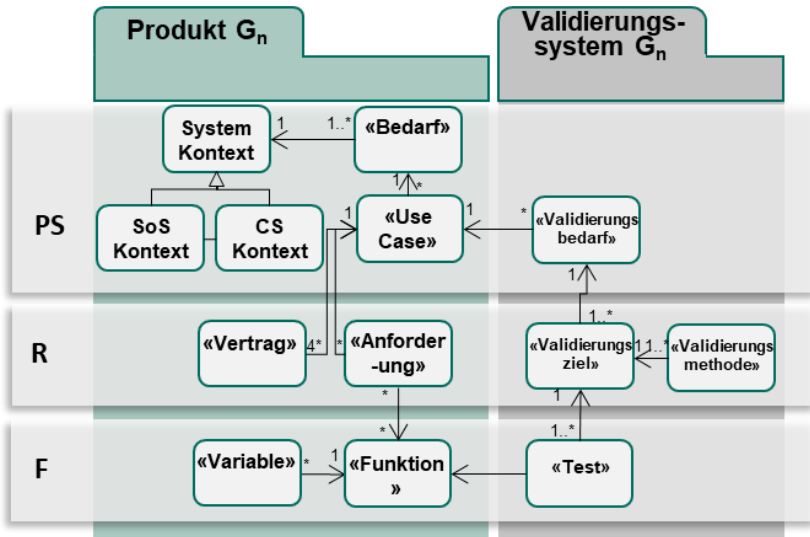


Abbildung 6.6: Sprachliche Elementtypen von Problemraum, Anforderungen und Funktionen des generischen Meta-Systemmodells.

Weiterhin lassen sich im Validierungssystem <Tests> beschreiben (siehe Abbildung 6.6). Diese sind als Sammelbegriff für mehrere <Testfälle> zu verwenden und mit den logischen Elementen <Testfall> und <Testumgebung> verknüpft. Diese werden nachfolgend in der Perspektive der logischen Elemente beschrieben.

6.1.4 Elementtypen von logischen Elementen des generischen Meta-Systemmodells

Schnittstellen zwischen CS sind in der Modellierung von SoS wesentlich und können in der logischen Architektur mit Hilfe des <SoSInterfaceBlock> dargestellt werden (siehe Abbildung 6.7). Zudem werden auf der Perspektive der logischen Elemente basierend auf dem hierarchischen Konzept von Ropohl (2009) <logische Teilsysteme> eines Systems beschrieben.

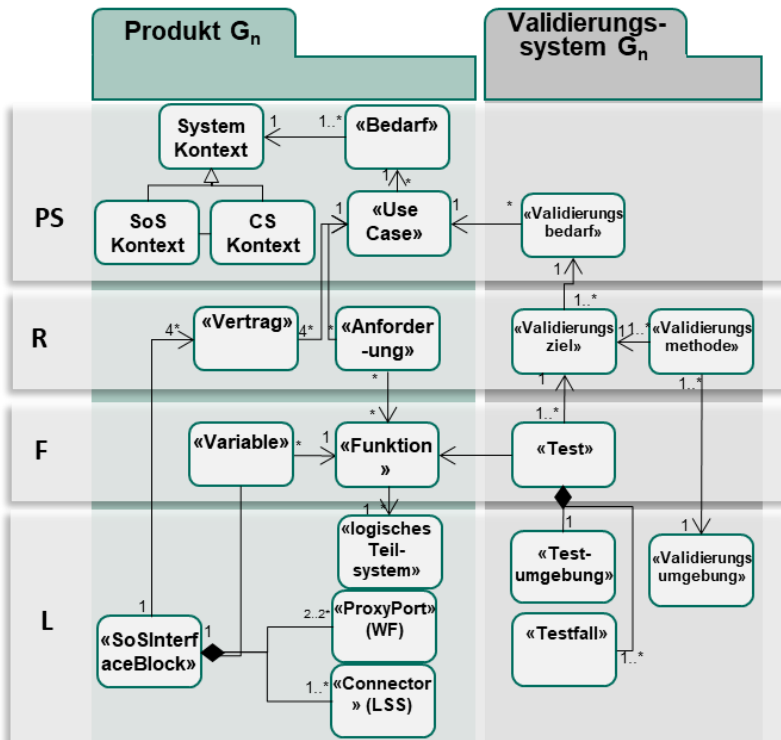


Abbildung 6.7: Sprachliche Elementtypen von Problemraum, Anforderungen, Funktionen und Logische Elemente des generischen Meta-Systemmodells.

Innerhalb eines <SoSInterfaceBlock> können Informationen, Energie oder Materie zwischen Teilsystemen beschrieben werden, die über die modellierten Schnittstellen übertragen werden. Die Übertragung kann angelehnt an den C&C²-Ansatz beschrieben werden, wobei Tabelle 6.4 die wesentlichen Elemente darstellt und in Bezug zum C&C²-Ansatz setzt.

Tabelle 6.4: Beschreibungselemente der Schnittstelle und die angelegten Elemente im C&C²-Ansatz

Systemelement in SysML	Beschreibung im C&C ² -Ansatz
<ProxyPort>	Connector
<ProxyPort>	Wirkfläche
Zwei <ProxyPorts> verbunden mit einem <Connector>	Wirkflächenpaar
<Block> bzw. <PartProperty>	Leitstützstruktur

So wird die Schnittstelle zwischen zwei Elementen mit einem Wirkflächenpaar (WFP) realisiert. In SysML werden analog zum WFP zwei <ProxyPorts> mit einem <Connector> verknüpft. Der <Connector> ermöglicht einen richtungsabhängigen ItemFlow. Zusammengenommen bilden zwei <ProxyPorts> und ein <Connector> einen <InterfaceBlock>. Beispielsweise sind in Abbildung 6.8 die wesentlichen Schnittstellen für die Datenübertragung des UseCases GLOSA als Wirkkette dargestellt.

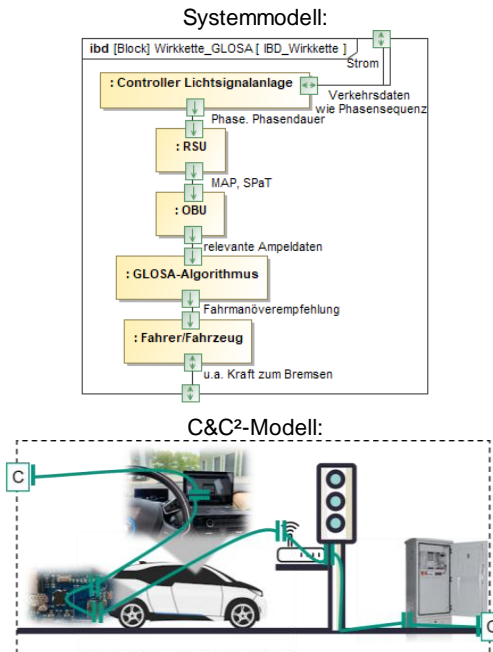


Abbildung 6.8: Wirkkette der Signalübertragung von GLOSA im Systemmodell und im C&C²-Modell (Bildquelle BMWi3: EnBW Energie Baden-Württemberg AG, 2022;; Verteilerkasten: Shenzhen Fama Intelligent Equipment Co., Ltd., 2022).

Der <Test> im Validierungssystem enthält eine <Testumgebung> sowie einen <Testfall>. Ein Testfall ist eine konkrete Beschreibung von einem durchzuführenden Test. Dieser wird in einer <Testumgebung> ausgeführt, welche die notwendigen Ressourcen beinhaltet. Für SoS kann diese <Testumgebung> Ressourcen unterschiedlicher CS enthalten. Dies führt zu der Notwendigkeit, <Testumgebungen> speziell für SoS zu erstellen. Beispielsweise können standortverteilte Prüfstände gekoppelt werden, sodass Interaktionen hervorgerufen durch Emergenzeffekte frühzeitig getestet werden können (vgl. und siehe Beispiele in Wäschle, Wolter et al., 2021; Wäschle, Wolter, Bause, Behrendt & Albers, 2022)

Innerhalb der Produktentwicklung werden die einzelnen Elementtypen abhängig von Referenzprodukten bestimmt. Demzufolge ist das Referenzsystem ein essenzieller Bestandteil der Produktentwicklung und somit auch der Modellierung. Dies wird im nachfolgenden Kapitel bei der Beschreibung von Elementtypen für das Meta-Systemmodell berücksichtigt.

6.1.5 Referenzsystem im generischen Meta-Systemmodell

Im Sinne des Modells der SGE – Systemgenerationsentwicklung wird das Referenzsystem R_n mit der Systemgeneration G_n und Validierungssystem für G_n verknüpft. Um die Auswirkungen und Abhängigkeiten zwischen den Systemen zu beschreiben wird ein Systemtupel vorgestellt, welches wesentliche Elemente der Validierung verknüpft. Somit lassen sich die Variationsarten für verschiedene Elementtypen beschreiben und Potenziale zur Risikoanalyse aufzeigen.

Ein System- n -tupel beschreibt eine Anzahl $n \in \mathbb{N}$ geordneter Elemente. Ein System-3-tupel (nachfolgend Systemtupel) zeigt die für die Validierung der repräsentativen Elemente <Anforderung>, <logisches Element> und <Testfall> und ihre Verknüpfungen für ein (Teil-)system. Weitere Elemente wie die <Funktion> sind für eine vollständige Betrachtung der Validierung zu berücksichtigen.

Erkenntnisse über das Systemtupel können helfen, die Auswirkungen von Änderungen eines Elements auf weitere Elemente eines Systems zu identifizieren und analysieren und somit das Ausmaß der Änderung zu bestimmen. Jedes System, ungeachtet des Abstraktionsgrades, kann demnach mit dem Systemtupel beschrieben werden. Die Elemente des Systemtupels haben im Sinne der SGE wiederum korrespondierende Elemente in einem Referenzsystem. Das Referenzsystem kann Elemente aus unterschiedlichen Referenzprodukten enthalten. Wenn beispielsweise ein Sensor von der Vorgängergeneration übernommen wird, aber der zugehörige <Testfall> aus einem Testfallkatalog eines anderen Systems stammt, bilden diese beiden unterschiedlichen Referenzprodukte

für das gemeinsame Referenzsystem. Somit kann auch die Analyse der Variationsanteile auf die Elemente des Systemtupels erweitert werden. Für diesen Schritt ist es notwendig, <Anforderungen> und <Testfälle> genauso wie <logisches Teilsystem> über die Variationsarten kategorisch einzuordnen. Tabelle 6.5 zeigt Beispiele, wie das Modell der SGE auf <Anforderungen> und <Testfälle> übertragen werden kann.

Tabelle 6.5: Untersuchung der drei Variationsarten der SGE an den Elementen des Systemtupels (vgl. Wäschle, Timucin et al., 2021)

Elementtypen	ÜV	AV	PV
<logisches Teilsystem>	Übernahme Übertragungsprotokolle	R_n : 4 Zustände im Algo. ⁴ G_n : 7 Zustände im Algo. ⁴	R_n : Technologie 5G G_n : Technologie 802.11p
<Anforderung>	Übernahme Spezifikation	R_n : Geschw. max. 50km/h G_n : Geschw. max. 50km/h	R_n : keine Laufzeitanf. G_n : Laufzeitanf.
<Testfall>	Übernahme Testfall	R_n : Lastfall Strecke 100m G_n : Lastfall Strecke 150m	R_n : bisheriger Prüfstand G_n : neuer Prüfstand

So sind unveränderte <logisches Teilsysteme>, <Anforderungen> und <Testfälle> als Übernahmevariation (ÜV) zu bezeichnen. AV kann durch Änderungen von Parametern der Elemente hervorgerufen werden. Eine grundsätzliche Veränderung z.B. der Übertragungstechnologie sowie neue Elemente wie neue Prüfstände führen zu Prinzipvariationen (PV) bezogen auf das Referenzsystem. Unter der Annahme, dass die Variationsarten unterschiedlich große Risiken mit sich führen (vgl. Albers, Reiß, Bursac, Schwarz & Lüdcke, 2013; vgl. Albers, Dumitrescu, Marthaler et al., 2018) kann eine Abschätzung zu Entwicklungsrisiko und -aufwand erfolgen. Die Variationen eines Elements haben somit direkte Auswirkungen auf benachbarte Elemente des Systemtupels. Weiterhin können auch Auswirkungen auf verknüpfte Systemtupel beispielsweise anderer CS erfolgen. Abbildung 6.9 veranschaulicht die Auswirkungen auf unterschiedliche Systemtupel für SoS und CS.

⁴ Algorithmus innerhalb des Fahrzeugs zur Bestimmung der Fahraufgabe basierend auf den Informationen des Systems Infrastruktur

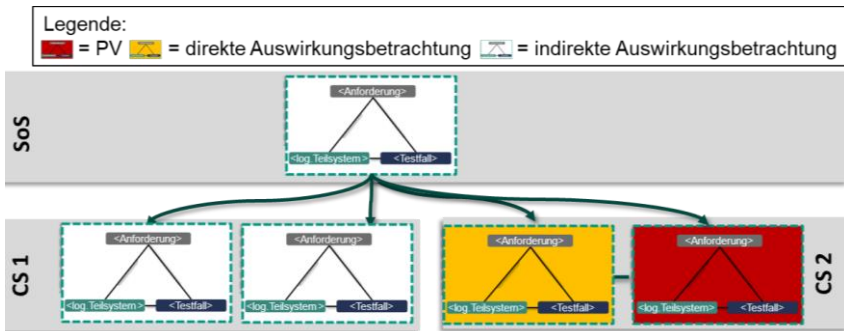


Abbildung 6.9:Prinzipdarstellung von möglichen Auswirkungen einer in Rot gekennzeichneten Prinzipvariation in einem Systemtupel auf weitere Systemtupel (Wäschle, Timucin et al., 2021).

Neben den direkten Beziehungen innerhalb und zwischen Systemtupeln ist es notwendig weitere Beziehungen zum Referenzsystem und weiteren Elementen zu erstellen. Beispielsweise können im Bereich der Validierung weitere Elemente wie <Validierungsziel> mitberücksichtigt werden.

Für die Auswirkungen im Systemtupel bietet die Schnittstellenbeschreibung des vorherigen Kapitels das Potenzial, Systeme mit dessen Variationsanteilen für die einzelnen Schnittstellen zu modellieren. Ein <logisches Teilsystem> auf einer Ebene ist eine LSS in einer hierarchisch höheren Ebene. Erhöht man die Granularität um eine Ebene, hat das Teilsystem wiederum Verbindungen WFP und LSS. Die Ebene des Modells zeigt also die spezifischen Elemente dieser Ebene. Ein Beispiel wäre ein polykristallines Metall, das elektrische Energie und Kräfte an den Schnittstellen überträgt. Eine Betrachtungsebene tiefer werden nun Korngrenzen des Materials sichtbar. Jede Korngrenze ist ein WFP zur Übertragung dieser Energie. Im Sinne des Systemtupels sind nun Anforderungen und Testfälle zum Beispiel an das Material mit zu betrachten. (Wäschle, Martin, Radimersky, Behrendt & Albers, 2020)

6.1.6 Meta-Systemmodell für das hochautomatisierte Fahren

Speziell in dem Bereich des hochautomatisierten Fahrens sind Elementtypen im Kontext des Szenarios und der damit verbundenen Szenarien basierten Validierung wesentlich. Weiterhin ist für das hochautomatisierte Fahren die Perspektive der physischen Elemente relevant, da die Untersuchung der Funktionalen Sicherheit an dem spezifischen Teilsystem erfolgt. Die bereichsspezifischen Elementtypen werden in Abbildung 6.10 in grau visualisiert und zu den generischen Meta-Systemmodellelementtypen verknüpft. Angelehnt an den ASAM Standard OpenScenario (ASAM e. V., 2022a) lassen sich logische und konkrete Szenarien

und die Operational Design Domain (ODD) beschreiben. Die ODD enthält den Kontext, in dem das System betrieben und die Funktionale Sicherheit entsprechend validiert werden muss.

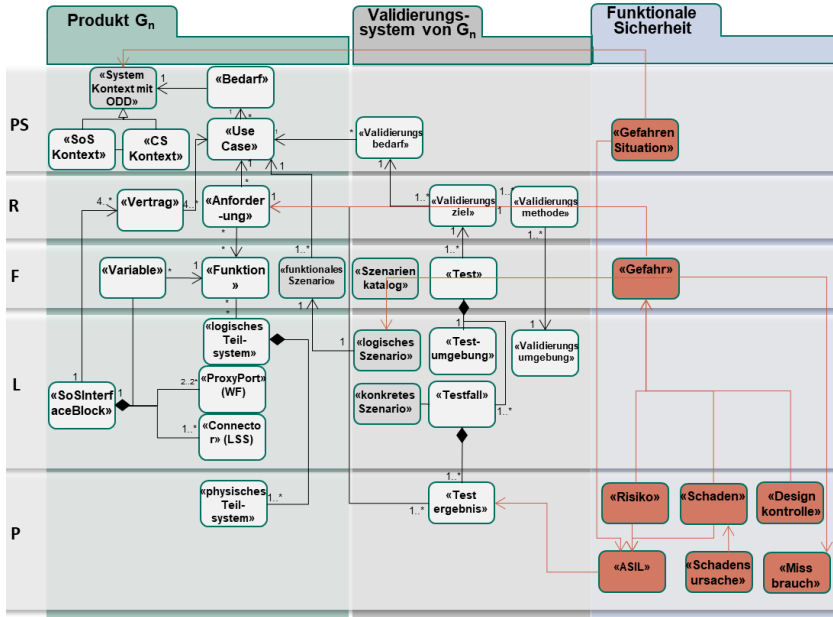


Abbildung 6.10: Elementtypen des bereichsspezifischen Meta-Systemmodells für das hochautomatisierte Fahren.

Richtlinien für die korrekte Beschreibung von Szenarien erfüllt der Standard OpenScenario mit einem Bereichsmodell (ASAM e. V., 2022a). In dem umfangreichen Modell sind beispielsweise Distanz, Nick-Winkel und Trajektorie vorgegeben. Die Szenarienbeschreibung lässt sich auch in SysML darstellen. In einer Dependency Matrix in Tabelle 6.6 lassen sich die Elemente zur Szenarienbeschreibung wie Distanz darstellen und Szenarien zuweisen, um deren Nachvollziehbarkeit zu erhöhen. Beispielsweise können somit alle Szenarien mit einer Haltebucht neben der Straße schnell gefunden werden.

Tabelle 6.6: Auszug aus der Szenarienbeschreibung mittels Verlinkung von Merkmalen mit der Allocate Beziehung zu UseCases

	Ausweichmanöver	Bergabfahrt	Fahrzeug Be- & Entladen	Haltestellenanfahrt
Szenarienlayer	8	7		8
04_Bewegliche Objekte	1	1		1
Fahrzeug auf entgegenkommender Spur	3	↗	↗	
Fahrzeug auf Nachbarspur links	2			↗
Fahrzeug auf Nachbarspur rechts	1			
01_Fahrbahneigenschaften	1	2		2
Haltebucht neben Straße	1			↗

Weiterhin kann explizit für das Thema der Funktionalen Sicherheit ein MBSE-Layer beschrieben werden. Dieser MBSE-Layer ist im Speziellen für die Systemmodellierung notwendig. Somit kann der in Abbildung 6.2 dargestellte Ordnungsrahmen um wesentliche Themenfelder wie die Funktionale Sicherheit erweitert werden, deren Einfluss auf das System durchgängig über mehrere Perspektiven erfolgt und die Elementtypen erweitert. Daher werden in der vorliegenden Arbeit die Layer im Kontext der Systemmodellierung als MBSE-Layer bezeichnet. Die Notwendigkeit für eine Rückverfolgbarkeit der Funktionalen Sicherheit zu anderen Layern ist in der Norm ISO 26262 und in ASPICE vorgegeben. Der Layer Funktionale Sicherheit ist in Abbildung 6.10 in einer blauen Box und Elementtypen in roten Boxen dargestellt. Hierbei werden die wesentlichen Normen berücksichtigt, um einen Transfer zu Unternehmen und einheitlicher Nomenklatur zu ermöglichen. Wesentlich ist die Kombination der in Normen genannten Elementtypen (angelehnt an VDA QMC Working Group 13 / Automotive SIG, 2017), sodass eine ausführliche Betrachtung der Funktionalen Sicherheit ermöglicht wird. Die SysML-Profile „Safety & Reliability Analysis“, „ISO-26262“, „FMEA“ und „MedicalRisk“ bieten bereits einige Elementtypen zur bereichsspezifischen Systembeschreibung (No Magic Product Documentation, 2022). Trotz des fehlenden Domänenbezugs benennt das MedicalRisk-Profil wesentliche Elemente der Funktionalen Sicherheit. Prinzipiell ist die Beschreibung eines SoS analog zu einem eigenständigen System vorzunehmen, allerdings kann es zu Emergenzeffekten bei der Kombination verschiedener Ressourcen kommen. Dies führt zu neuen Gefahren und muss methodisch berücksichtigt werden. (vgl. Wäschle und Wolter et al. (2022)).

Innerhalb des <System-Kontextes mit ODD> lassen sich <Gefahrensituationen> beschreiben. Diese werden über verschiedene Methoden der Funktionalen Sicherheit identifiziert. Die vorliegende Arbeit stellt die Methode MEFS4SoS in Kapitel 6.3 vor.

Für die Modellierung der <Gefahrensituation> wird in den Methoden STPA und SOTIF der <System-Kontext mit ODD> modelliert. Aus ihr lässt sich die für <ASIL> notwendige Auftretenswahrscheinlichkeit einer <Gefahr> ableiten. Die <Gefahr>⁵ ist ein wesentlicher Elementtyp in den Normen ISO 26262 und SOTIF. Im Systemmodell wird die <Gefahr>⁶ in einem logischen Szenario beschrieben. Aus der <Gefahr> folgt <Risiko> und <Schaden>, welche zur Berechnung von <ASIL> notwendig sind. Zudem ist die <Designkontrolle>⁷ in der FMEA enthalten, um die Überwachung der <Gefahr> zu ermöglichen. Der <Missbrauch> wird in SOTIF genannt und führt zu einer Betrachtung von nicht intendiertem Systemverhalten, welches ebenfalls Bestandteil der Funktionalen Sicherheit ist. Die <Schadensursache>⁸ wird in der FMEA eingesetzt und beschreibt den Grund des Schadens bzw. der Gefahr. Weitere Stereotypen sind je nach Anwendungsfall notwendig und werden daher in der Systemebene eines Meta-Systemmodells oder weiterer Modelle beschrieben, die nicht im Fokus der vorliegenden Arbeit sind. Die in der Funktionalen Sicherheit verbreiteten „Sicherheitsanforderungen“ beschreiben <Anforderungen> an System und Domäne, um die Sicherheit zu gewährleisten. Sie sind in der vorliegenden Arbeit Teil der Systemanforderungen. Daher ist die Gefahr mit den <Anforderungen> und explizit mit den System- und Domänenanforderungen verbunden.

6.1.7 Zusammenfassung

Kapitel 6.1 führt ein generisches und bereichsspezifisches Meta-Systemmodell für die Validierung von SoS ein. Hierzu werden die Layer Systemgeneration (G_n) und Validierungssystem für G_n sowie das Referenzsystem verwendet. Anhand der Perspektiven (PS-R-F-L-P) werden wichtige Elementtypen zur Validierung von SoS eingeführt und mit ihren Beziehungen in das Meta-Systemmodell integriert. Es zeigt

⁵ In der FMEA kann der Effekt eines Fehlers auf <Gefahr> gemappt werden (Chen, Jiao und Zhao (2020))

⁶ In der Arbeit wird die in der STPA unsichere Kontrollaktion und in der FMEA Schadensmöglichkeiten als Teil der Gefahr angesehen. Die sichere Kontrollaktion in der STPA wird als Funktion angesehen (Chen et al. (2020))

⁷ In der STPA wird Sicherheits Randbedingung verwendet (Chen et al. (2020))

⁸ In der STPA können die kausale Faktoren auf <Fehlerursache> gemappt werden (Chen et al. (2020))

sich, dass themenspezifische Elementtypen wie zur Funktionalen Sicherheit in einem eigenen MBSE-Layer dargestellt werden können, um einen bedarfsgerechten Einsatz zu ermöglichen. So können Systemarchitekten und andere Rollen, welche das Modell verwenden, den Layer nach Bedarf ein- oder ausblenden.

6.2 Konkretisierung der Modellierungssprache: Bereichsspezifisches Referenz-Systemmodell für das hochautomatisierte Fahren

Nachfolgend wird ein bereichsspezifisches Referenz-Systemmodell beschrieben, welches die sprachlichen Elementtypen des Meta-Systemmodells konkretisiert und wesentliche Referenzen einführt. Hierbei wird ein Fokus auf die Funktionale Sicherheit von SoS gelegt, deren Bedarf in Studien aufgezeigt wurde (vgl. Sikora, Tenbergen und Pohl (2012)). Um die Beziehungen zwischen den Elementtypen im Meta-Systemmodell zu beschreiben und konkretisieren, visualisiert Abbildung 6.11 ein bereichsspezifisches Referenz-Systemmodell. Das Referenz-Systemmodell wurde im Rahmen des Forschungsprojekts „SmartLoad“⁹ für die Validierung von hochautomatisierten Fahrzeugen erstellt. Abbildung 6.11 verdeutlicht nun, dass die Elemente des Referenz-Systemmodells in die in Kapitel 6.1 eingeführte Struktur zur Modellbeschreibung (siehe Abbildung 6.2) integriert werden können. So können Elemente nach den vier Layern und den fünf Perspektiven unterteilt werden. Angelehnt an die Norm SOTIF (ISO/PAS 21448:2019(E)) wird das Referenz-Systemmodell als bekannter Bereich deklariert und es können mögliche unbekannte Elemente für die Funktionale Sicherheit berücksichtigt werden. Das Referenz-Systemmodell visualisiert daher die beiden Hauptbereiche von SOTIF, bekannt und unbekannt, mit dem Ziel, den unbekannteren Bereich unter Berücksichtigung der Funktionalen Sicherheit zu reduzieren. Hierzu wurden bereits in Kapitel 6.1.6 im bereichsspezifischen Meta-Systemmodell wesentliche Elementtypen der Funktionalen Sicherheit vorgestellt. Die Deduktion der Elementtypen des Meta-Systemmodells in Elemente des Referenz-Systemmodells erfolgt in diesem Kapitel.

⁹ Innerhalb des Forschungsprojektes SmartLoad „Neue Methoden zur Zuverlässigkeitssteigerung von hoch automatisierten elektrischen Fahrzeugen“ wurden Methoden auf deren Anwendbarkeit untersucht und ein Referenz-Systemmodell abgeleitet. (siehe Wäschle, Wolter et al. (2021); Wäschle, Wolter, Bause, Behrendt und Albers (2022))

Hierbei werden aus dem Bereich des hochautomatisierten Fahrens Referenzen durch Beispiele am Ende der Teilkapitel expliziert.

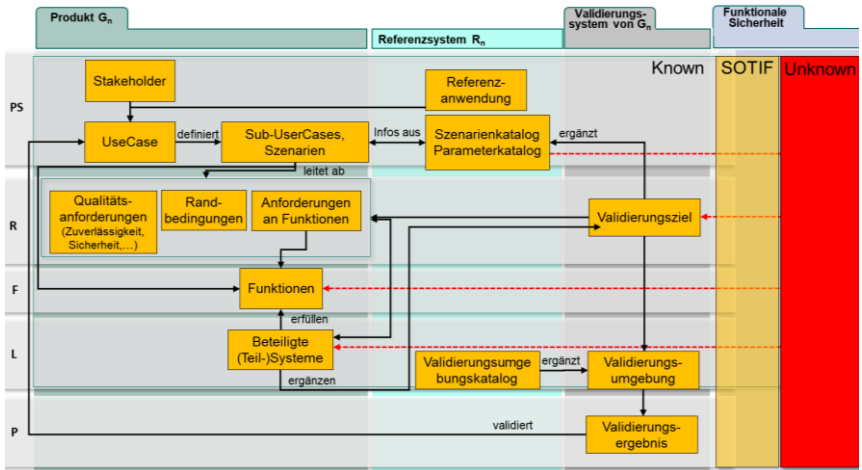


Abbildung 6.11: Referenz-Systemmodell mit Elementen deduziert aus dem Meta-Systemmodell (angelehnt an Wäschle und Wolter et al. (2022)).

6.2.1 Elemente des Problemraums (PS) im Referenz-Systemmodell

Wie bereits im Meta-Systemmodell wird der Systemkontext mit dem konkreten Elementtyp Stakeholder bestimmt und mit UseCases verknüpft. Stakeholder werden nach Weilkiens (2015) in einem Anwendungsfalldiagramm modelliert. Die Bedarfe der Stakeholder werden erfasst, um das Produkt zu validieren. Die Stakeholder werden daher über eine Assoziationsverbindung mit jeweils weiteren Stakeholdern und Anwendungsfällen, in denen sie interagieren, verknüpft. Spezifisch für den Bereich des hochautomatisierten Fahrens sind zahlreichen Normen und Gesetze, welche Einfluss auf das System nehmen. Bei einer Änderung von diesen muss das System dementsprechend angepasst werden, was eine Flexibilität in der Entwicklung fordert und durch eine nachvollziehbare Verknüpfung der Elemente in Modellen unterstützt wird. Zudem führt die häufig hohe Anzahl Stakeholder in einem SoS zu konflikthafter Zielen und somit der Notwendigkeit dies abzubilden und flexibel auf Änderungen reagieren zu können.

Es müssen Fahrgäste sowie Personen in der Umgebung wie Fußgänger berücksichtigt werden. Außerdem müssen öffentliche Einrichtungen wie Kommunen für die Verkehrsplanung sowie privatwirtschaftliche Einrichtungen wie

Versicherungsgesellschaften, Hersteller, Dienstleister und Stakeholder von interagierenden Systemen, modelliert werden.

Je nach Stakeholder lassen sich zu Beginn wesentliche „Haupt-UseCases“ bestimmen. Beispielsweise kann nach Transportgut unterschieden werden in den Transport von Personen, Gütern und Gütern mit Anhänger. Für diese Haupt-UseCases können spezifischere Sub-UseCases gebildet werden. Diese sind mit einer include-Beziehung zu logischen Szenarien und von logischen zu konkreten Szenarien verknüpft (Pegasus, o.J.; vgl. Wäschle, Wolter et al., 2022). Beispielsweise ist der Haupt-UseCase der Personentransport. Nun können Sub-UseCases wie Green Light Optimal Speed Advisory (GLOSA) bestimmt werden. Bei GLOSA wird eine Information von Lichtsignalanlage an Fahrzeug zur Vorausplanung der Fahrgeschwindigkeit übermittelt (siehe Kapitel 7.2). Dabei können weitere Subsub-UseCases für GLOSA definiert werden, wie die Fahrt außerorts und innerorts. Am Schluss enthält das konkrete Szenario Ereignisse wie das Beschleunigen mit konkreten Parametersätzen.

6.2.2 Elemente für Anforderungen im Referenz-Systemmodell

Für das hochautomatisierte Fahren sind Anforderungen wesentlich, deren Beschreibung in Normen vorgegeben werden. Der Zusammenhang zwischen Anforderungen und weiteren Elementtypen wird bereits im Meta-Systemmodell vorgegeben (siehe Kapitel 6.1.2). Nach der Norm ISO 26262 müssen die Anforderungen des exemplarischen Anwendungsfalls die Betriebsart, die Fehlertoleranz, das Zeitintervall, sichere Zustände, Notbetriebszeitintervall und funktionale Redundanzen enthalten (ISO 26262-9:2018(E)).

Zum Beispiel wird die Anforderung zum automatisierten Fahren über die Verkehrskreuzung für das gefährliche Ereignis eines Unfalls während des Überfahrens der Verkehrskreuzung mit der ASIL D bestimmt. ASIL werden berechnet aus der Beherrschbarkeit, der Eintrittswahrscheinlichkeit und der Schwere des Ereignisses (siehe Kapitel 7.2). Anforderungen bezogen auf die Sicherheit werden für jede Gefahr basierend auf dieser Gefahren- und Risikobewertung bestimmt. (ISO 26262-1:2018(E), S. 10)

6.2.3 Elemente für Funktionen im Referenz-Systemmodell

Insbesondere für SoS sind White Box Modelle für CS unterschiedlicher Unternehmen schwierig zu realisieren. Der Wissensaustausch kann daher über Black Box Modelle realisiert werden. Hierbei können wesentliche Modellelemente bestimmt werden, die kein zu schützendes Unternehmenswissen preisgeben. Diese Modellelemente können nach dem funktionalen Modell von Ropohl (2009)

Funktionen durch die Modellierung von Input und Output abstrahiert beschreiben. Dies ist insbesondere in der Schnittstellenmodellierung wesentlich, bei der ein systemübergreifender Austausch notwendig wird.

Für eine konkrete Funktionsbeschreibung werden White Box Modelle eingesetzt. So beschreibt Kapitel 6.1.4 die Modellierung von Schnittstellen angelehnt an den C&C²-Ansatz (siehe Abbildung 6.8). Abbildung 6.12 zeigt ein Aktivitätsdiagramm zur detaillierten Beschreibung einer Funktion. Die Norm ISO 26262 konzentriert sich auf sicherheitsrelevante Funktionen, die das Potenzial haben, Teil einer Gefahr oder einer Anforderung zu sein (ISO 26262-1:2018(E), S. 23).

Abbildung 6.12 zeigt den wesentlichen ausführbaren Ablauf des GLOSA-Algorithmus. Die Funktion automatisiertes Beschleunigen kann innerhalb des Sub-UseCase GLOSA realisiert werden. Für die Bewertung von Sicherheitsaspekten ist die Aggregation aller Funktionen zur Erreichung der automatisierten Beschleunigung zu betrachten.

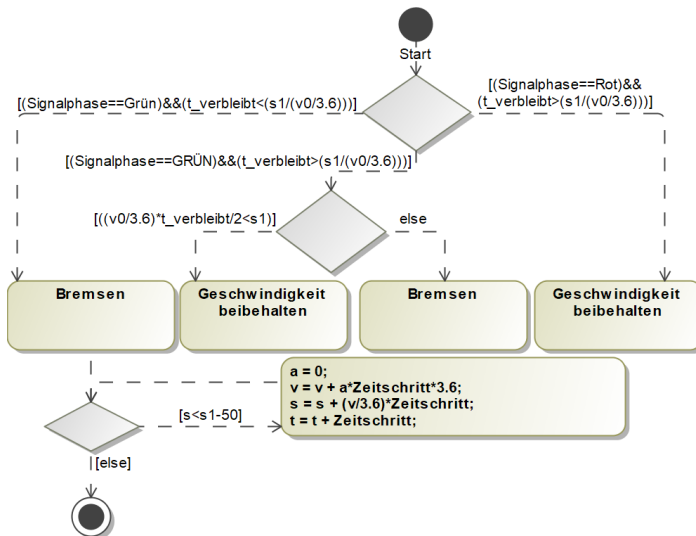


Abbildung 6.12: Ausschnitt des Aktivitätendiagramms für den GLOSA-Algorithmus in einem ausführbaren SysML-Diagramm (Klein, 2022)¹⁰.

¹⁰ Abschlussarbeit (unveröffentlicht)

6.2.4 Logische Elemente im Referenz-Systemmodell

Teilsysteme lassen sich in der logischen Perspektive in dem SysML-Element <Block> beschreiben. Beispielsweise kann die Funktion „Signalphase erkennen“ über das Teilsystem Kamera oder OBU mit RSU realisiert werden. Mit Hilfe von Teilsystemen und weiteren Elementen kann die Validierungsumgebung modelliert werden. Die Modellierung enthält eine Abhängigkeitsmatrix und ein Blockdefinitionsdiagramm, um einen Überblick über die Prüfstände mit der Systemaufbaustruktur in einer logischen und physischen Perspektive zu geben. Auf diese Weise kann eine flexible Validierung in Abhängigkeit des SiD erreicht werden. Um eine Verknüpfung und Kommunikation zwischen den verteilten Prüfständen zu erreichen, werden Koppelsysteme eingesetzt (siehe Albers, Pinner, Yan, Hettel & Behrendt, 2016). Sie dienen zur Überwindung von Inkompatibilitäten zwischen Modellen beispielsweise durch ein Kabel zur Kopplung bei einer räumlichen Trennung.

Tabelle 6.7 nutzt wesentliche Elemente der FMEA. Die Tabelle enthält die verknüpften Elemente: Teilsysteme, Fehler, Fehlerursache, Design Kontrolle sowie die drei Kennwerte Schadensstärke, Auftretenswahrscheinlichkeit und Detektionsrate zur Berechnung der Risikoprioritätszahl. Die FMEA unterstützt die systematische Identifizierung von Fehlern und Protokollierung von Fehlern für den zukünftigen Bedarf. Die Tabelle führt die FMEA aus für den Einsatz von Routern zur Kommunikation zwischen Prüfständen für das Testen von UseCases hochautomatisierter Fahrzeuge.

Tabelle 6.7: Auszug der FMEA mit den Kennwerten Schadensstärke (SS), Auftretenswahrscheinlichkeit (AW), Detektionsrate (DeR), Risikoprioritätszahl (RPZ) modelliert in SysML

Komponente	Fehler	Fehlerursache	SS	AW	Design Kontrolle	DeR	RPZ
RouterMikrotik...	KeinSignalvonRouter	RouterHardwareSchaden	3	2	ResetHardware	2	12.0
RouterMikrotik...	KeinSignalzuRouter	RouterFalschKonfiguriert	3	1	Signalüberwachung	1	3.0

6.2.5 Elemente des Referenzsystems des Referenz-Systemmodells

Als Referenzsystemelemente werden nachfolgend zwei Kataloge für Validierungsumgebungen und Szenarien vorgestellt.

Für die Validierungsumgebung kann ein Katalog Teil des Referenzsystems sein, um eine modulare Wiederverwendung von Elementen zu ermöglichen (vgl. Yan et al., 2019). Der Katalog enthält Bewertungen und beispielhafte Konfigurationen von Validierungsumgebungen. Basierend auf neun Kriterien lassen sich Validierungsumgebungen im Katalog bewerten und vergleichen (Albers, Mandel et al., 2018). Auf diese Weise kann eine Wissensdatenbank geschaffen werden, deren Einträge als Referenzsystemelement für die Empfehlung von Maßnahmen und die Entwicklung neuer Generationen von Validierungsumgebungen dient. Die neun Kriterien wurden von den Charakteristiken des SoS abgeleitet und weiter unterteilt Qi (2019)¹¹. Beispielsweise wird die Heterogenität unterteilt in Quelle und Ziel, um die verschiedenen Typen und Frequenzen von Empfangsdaten zu beschreiben und zu bewerten.

In Abbildung 6.13 werden vier Validierungsumgebungen im Kontext V2I in einem Spinnendiagramm verglichen. Die Validierung eines BUS (vgl. Kapitel 2.3.1) , Hardware Antriebsstrangprüfstand (vgl. Yan et al., 2019 Bezeichnung HiL in Abbildung 6.13), V2I-Prüfstand (vgl. Abbildung 7.18) und Testfeld (vgl. 2.3.2) werden in einem Spinnendiagramm verglichen. Es wird ersichtlich, dass verschiedene Validierungsumgebungen unterschiedlich starke Ausprägungen der Bewertungsgrößen und somit Stärken aufweisen. Daraus folgt, dass Referenzsystemelemente nicht nur für Anwendungsfälle geeignet sind, sondern auch für Elemente des Validierungssystems wie die Validierungsumgebung zu betrachten sind.

¹¹ Abschlussarbeit (unveröffentlicht)

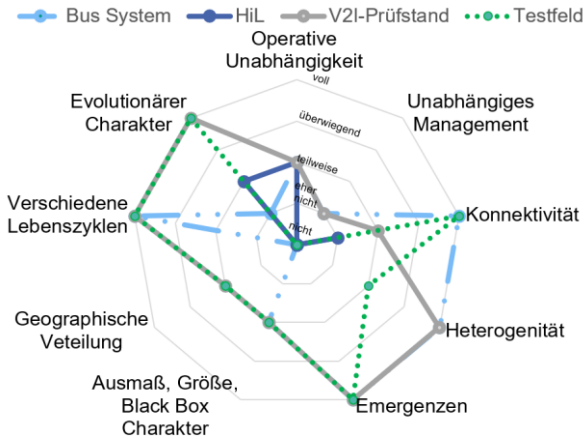


Abbildung 6.13: Spinnendiagramm mit Charakterisierung von vier Validierungsumgebungen für V2I.

Im Szenarien katalog werden vordefinierte Szenarien aus Referenzsystemelementen, wie z.B. aus der Norm EURO-NCAP (Euro NCAP, 2019) betrachtet. Zudem lassen sich Referenzprodukte wie Flugzeuge heranziehen, die in Bezug auf die Funktionale Sicherheit in hochautomatisierten Systemen ähnliche Themen in den Normen DO-254 (U.S. Department of Transportation - federal aviation administration, 2017) und UL4600 (Underwriters' Laboratories, 2020) adressieren. Weiterhin können Szenarien mit Hilfe von Methoden entwickelt werden. In Kapitel 6.3 wird eine solche Methode eingeführt.

Der Szenarien katalog basiert auf dem PEGASUS-Szenarien katalog, dessen Beschreibungsschichten im ASAM Standard OpenScenario standardisiert wurden (Pegasus, o.J.). In Tabelle 6.8 werden verschiedene Szenarien tabellarisch in dem Szenarien katalog in SysML beschrieben. Hierzu wird die „betriebliche Situation Tabelle“ eingesetzt, die den Szenarien bereits eine Häufigkeit zuordnet.

Tabelle 6.8: Auszug der Szenarienbeschreibung in SysML dargestellt in einer Design Structure Matrix

Name	Häufigkeit	Begründung der Häufigkeit	Fahrzeugeinsatz	Umwelt und Infrastruktur
Autobahnfahrt	E4	Tägliches Fahren auf Autobahnen	Fahrt mit hoher Geschwindigkeit	Verkehrsfreie Fahrt
Kurvenfahrt	E4	Fahren auf leicht kurvigen Straßen ist Teil des normalen Fahrens	Fahrt mit hoher Geschwindigkeit	Verkehrsfreie Fahrt

6.2.6 Zusammenfassung

Kapitel 6.2 beschreibt ein bereichsspezifisches Referenz-Systemmodell für das hochautomatisierte Fahren. Hierbei werden wesentliche Elemente zusammenhängend zur Funktionalen Sicherheit beschrieben und in die Struktur des Meta-Systemmodells eingeordnet. Das Modell kann in SysML implementiert werden und beinhaltet Sichten und Diagramme zu Elementen und deren Beziehungen. Die Sichten werden anhand von V2I Beispielen in SysML beschrieben.

6.3 MEFS4SoS – Methode zur Erreichung der Funktionalen Sicherheit für die Validierung von SoS

Angelehnt an die Normen der ISO 26262 und SOTIF wurde die in Abbildung 6.14 dargestellte Methode bestehend aus vier Schritten mit zugeordneten Aktivitäten für die Validierung von SoS entwickelt (siehe Übersicht der untersuchten Aktivitäten und die berücksichtigten Schritte in MEFS4SoS Abbildung A.1 in Anhang A). Hierbei sind die Schritte nicht notwendigerweise sequentiell auszuführen. Beispielsweise kann eine Gefahrenidentifikation zu einer neuen notwendigen Systembeschreibung führen. Weiterhin kann die Methode wiederholt werden. So führen Ergebnisse der Validierung zu Syntheseaktivitäten, die das Modell verändern und in die Systembeschreibung aufgenommen werden können.

Die Zwischenergebnisse der einzelnen Schritte können als Entwicklungsgenerationen angesehen werden. So führen Modifikationen im zweiten Schritt zu Variationen basierend auf dem Modell der SGE. Aus diesem Grund resultiert ein Durchlauf der Schritte eins bis vier in einer neuen Systemgeneration. Die Erkenntnisse der Validierung werden anschließend mittels Analyse und Synthese in das Systemmodell (mit dessen Beschreibung in Schritt 1) zurückgeführt und führen zu einer neuen Systemgeneration.

MEFS4SoS – Methode zur Erreichung der Funktionalen Sicherheit		
1	Systembeschreibung	Erstellen eines Modells mit PSRF Elementen und Schnittstellen ^{3,4,5,6}
2	Gefahrenidentifikation & Risikoevaluation	Identifizieren von (nicht-sicheren, unbekanntem) Gefahren ^{1,3}
3	Funktionale Modifikation und Spezifikationen	Adaption von PSRFLP-Elementen zur Reduzierung des Risikos ^{1,3}
4	Validierung	Beschreibung des Validierungssystems und erstellen konkreter Szenarien ²
		Evaluieren des Risikos & bestimmen von ASIL, Ursache, Designkontrolle ^{3,9}
		Analyse & Adaption der Auswirkungen mit abhängigen Fehlern im Modell ⁹
		Durchführung der Validierung, Zusammenfassung, Nutzung der Testergebnisse ^{1,5,6,9}

¹ ISO/PAS21448 ³ ISO26262-3 ⁵ ISO26262-5 ⁹ ISO26262-9
² ASAM OpenSCENARIO® 2.0.0 ⁴ ISO26262-4 ⁶ ISO26262-6

Abbildung 6.14: MEFS4SoS - Methode zur Erreichung der Funktionalen Sicherheit für die Validierung im Bereich hochautomatisiertes Fahren für SoS in Anlehnung an ISO 26262, ISO/PAS21448, OpenScenario.

6.3.1 Systembeschreibung

Meta-Systemmodell und Referenz-Systemmodell beinhalten bereits eine Systembeschreibung mit dem System-, SoS-, und CS-Kontext sowie UseCase Beschreibungen. Dies verdeutlicht den Mehrwert einer Systemmodellierung, da wesentliche Aktivitäten der Modellbildung und eine nachvollziehbare Vernetzung bereits in SysML erfolgt. Basierend auf dem SoS-Kontext können in den Folgeaktivitäten potenzielle Gefahren identifiziert und bewertet werden.

6.3.2 Gefahrenidentifikation & Risikoevaluation

Die Aktivitäten in Stufe 2 sind mit bestehenden Methoden der Funktionalen Sicherheit realisierbar. HARA und STPA identifizieren Gefahren. Zudem ermöglicht die FMEA die Identifikation von Fehlern, Ursachen, Designkontrolle und der Risikoprioritätszahl (siehe Tabelle 6.7). Der Nutzen der FMEA für SoS wurde im Rahmen einer Analyse mit der Bewertung von verschiedenen Methoden untersucht (Wäschle, Behrendt et al., 2021). Eine Kombination von Methoden der Funktionalen Sicherheit wie HARA, STPA und FMEA wurde bereits in Veröffentlichungen dargestellt und als mehrwertstiftend bezeichnet (Chen, Jiao & Zhao, 2020; Sulaman, Beer, Felderer & Höst, 2019). Die drei Methoden ermöglichen drei wesentliche Aktivitäten (vgl. Weikiens et al., 2015, S. 3):

- (1) die Identifikation fehlender Elementtypen und Verknüpfungen und somit der Vergrößerung des SOTIF-Bereiches „Known“;
- (2) die Verifikation der bestehenden Elementtypen und Verknüpfungen;
- (3) die Validierung gemäß Normen z.B. zur Funktionalen Sicherheit.

Tabelle 6.9 zeigt die wesentlichen Elemente zur Beschreibung und Beurteilung einer Gefahr (vgl. HARA). Hierbei werden Risiken identifiziert und kategorisiert anhand

der drei Faktoren: „Schweregrad“ des potenziellen Schadens, „Aufretenswahrscheinlichkeit“ der Betriebssituation und „Kontrollierbarkeit“ des gefährlichen Ereignisses. Der Schweregrad kann durch den potenziellen oder gemessenen Schaden beurteilt werden. Die Auftretenswahrscheinlichkeit wird anhand der Gefahrensituation und der Schadensursache beurteilt. Ein ASIL wird einer Gefahr zugewiesen, basierend auf der Analyse von Fahrsituationen in Bezug auf die drei Faktoren. Nach der Zuweisung wird eine Systemanforderung¹² formuliert, um ein unangemessenes Risiko zu vermeiden, und es werden weitere Anforderungen zur Risikominderung abgeleitet. (Vgl. ISO 26262-2:2018(E))

Tabelle 6.9: Auszug einer HARA-Tabelle in SysML erstellt in dem Plugin „Cameo Safety & Reliability Analysis“

Schadensursache	Gefahr	Schaden
(R) Reibung	(H2) H Getriebeschaltung funktioniert nicht	(H) Zahnradschaden
(R) Transistorschaden	(H2) K Inkorrekte Signale von Motor	(H) Überhitzung Motorschaltung
(R) Kapazitätsschaden	(H2) E Inkorrekte Signale von HV-Batterie	(H) Kurzschluss

Durch den Einsatz von Elementen der Funktionalen Sicherheit können die Gefahren identifiziert und mit Elementen des Referenz-Systemmodells verknüpft werden (siehe rote Pfeile in Abbildung 6.11). Es wird das SoS V2I untersucht. Innerhalb des SoS ergibt sich ein emergentes Verhalten mit neuen Ursachen und abhängigen Fehlern. Die Gefährdungen haben ihre Ursache in spezifischen Systemkonfigurationen, welche sich nicht immer klar einem CS zuordnen lassen, sondern aufgrund der Emergenz teilweise nur innerhalb des SoS auftreten.

Die in Kapitel 2.3.2 eingeführte Norm SOTIF weist Aktivitäten zur Identifikation von unsicheren, noch nicht bekannten Gefahren auf. Aufgrund der Modellierung des System- bzw. SoS-Kontextes können Gefahren abgeleitet werden. Hierzu kann die Methode STPA eingesetzt werden, in der unsichere Kontrollaktionen identifiziert werden. Abbildung 6.15 zeigt einen Regelkreis, der mit UseCases verknüpft werden kann. Teilsysteme wie der Lenkaktor (1), Sensoren (2) oder die Autobox (Steuergerät) (3) tauschen Daten und Materie aus. Dadurch entsteht ein Regelkreis, der mögliche Fehler aufzeigt. In diesem Beispiel sendet die Autobox einen Satz von Lenkwinkeln an den Lenkaktor. Sensoren ermitteln das Lenkmoment und den Winkel der Lenkwelle und übermitteln sie zurück an die Regelungseinheit Autobox. Der Regelkreis kann auf verschiedenen Perspektiven verwendet und modelliert

¹² In der Norm ISO 26262-2 (ISO 26262-2:2018(E)) wird der Begriff Sicherheitsziel anstelle von Systemanforderung verwendet. Im Sinne des Metamodells mit der in Kapitel 6.1.2 eingeführten verschiedenen Arten von Anforderungen wird das Sicherheitsziel als eine Systemanforderung klassifiziert.

werden. In der Autobox regelt zum Beispiel ein Controller die Einstellung des Soll-Lenkswinkels. Wie in Abbildung 6.15 dargestellt, müssen mehrere Elemente modelliert und miteinander verbunden werden. Die Integration der STPA in das Referenz-Systemmodell hilft, UseCase-spezifische Gefährdungen frühzeitig zu erkennen. Darüber hinaus ermöglicht diese Integration im MBSE die Verknüpfung und damit die Identifizierung von Redundanzen in Funktionen und Teilsystemen. (Wäschle, Wolter, Bause & Behrendt, 2020)

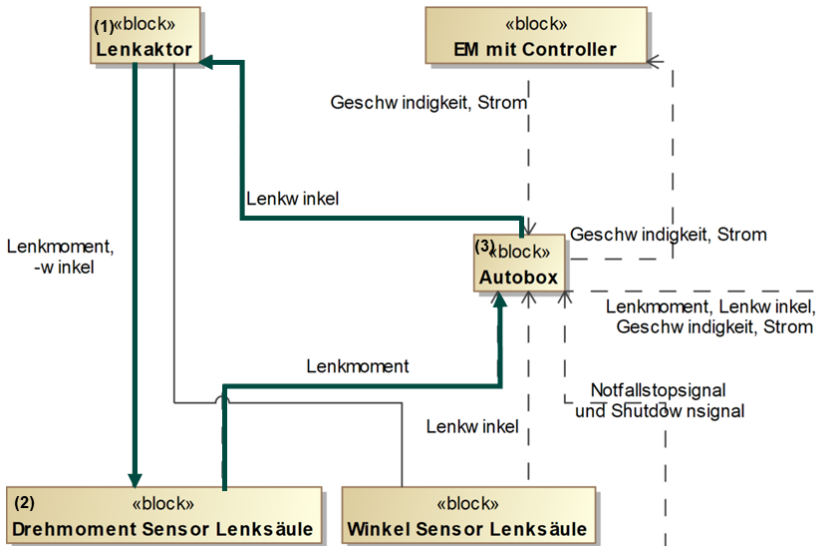


Abbildung 6.15: Auszug eines modellierten Regelkreises für das automatisierte Fahren mittels Lenkaktor (Wäschle, Wolter et al., 2020) angelehnt an (Leveson & Thomas, 2018).

6.3.3 Funktionale Modifikation und Spezifikationen

Ausgehend von der Einordnung von Elementen nach ASIL können Anpassungen an Elementen vorgenommen werden. Die Auswirkungen, welche beispielsweise kaskadierende Fehler (siehe Kapitel 6.1.6) enthalten, können mit einer Abhängigkeitsmodellierung untersucht werden. Ein mögliches Ergebnis der Abhängigkeitsmodellierung ist in Abbildung 6.16 dargestellt. Die Abbildung visualisiert als Beispiel aufgrund von Reibung abisolierte Kabel als sicherheitsrelevantes Element und deren Verknüpfung mit Schäden. Somit kann die systemübergreifende Interdependenz in SoS modelliert werden. Die Funktionen

sind mit Anwendungsfällen verknüpft und unterstützen die Identifizierung von Ereignissen.

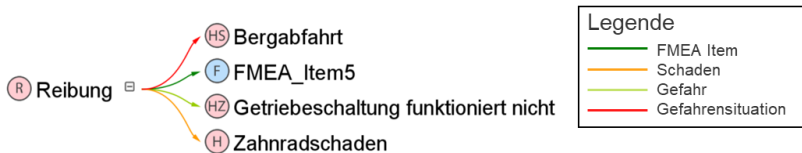


Abbildung 6.16: Wirkkette der exemplarischen Schadensursache „Reibung“ dargestellt in einer Safety & Reliability Analysis Map.

6.3.4 Validierung

Zur Durchführung der Validierung wird in Vorbereitung die Validierungsumgebung aufgebaut. Zudem gilt es, basierend auf den vorgestellten sprachlichen Elementen konkrete Szenarien abzuleiten, die anschließend angewendet und deren Ergebnisse evaluiert werden können.

Aufbau der Validierungsumgebung: Bereits während der Produktentwicklung erfolgen nach dem Pull-Prinzip Aktivitäten der Validierung. Hierbei ist die Validierungsumgebung als wesentlicher Teil des Validierungssystems parallel zum Produkt aufzubauen. Basierend auf dem IPEK-XiL-Ansatz enthält die Validierungsumgebung Systeme, welche mit dem SiD bzw. Sul interagieren. Aufgrund der Tatsache, dass spezifische Validierungsumgebungen mehrere Systeme enthalten, die sowohl eigenständig betrieben und verwaltet werden sowie standortverteilt vorliegen können, können diese Validierungsumgebungen als SoS kategorisiert werden. Demzufolge lässt sich für deren Validierung ebenfalls die Methode MEFS4SoS anwenden.

Ableitung von konkreten Szenarien: Abbildung 6.17 veranschaulicht ein Vorgehen in sechs Schritten, um ausgehend von Stakeholderbedarfen konkrete Szenarien abzuleiten. Das generelle Vorgehen zur Validierung dient dabei als Ausgangsbasis und ist in Abbildung 6.17 auf der linken Seite dargestellt. Es ist in verschiedenen Forschungsarbeiten beschrieben und ist hier aus Gründen der Übersicht als sequenzielles Vorgehen dargestellt (vgl. Albers, Behrendt et al., 2016; vgl. Kubin et al., 2023; vgl. Mandel, Wäsche, Lutz & Behrendt, 2021; vgl. Yan, 2020). Wie bereits in Kapitel 1.1 beschrieben ist die Validierung ein häufig durchzuführende Aktivität nach dem Pull-Prinzip (siehe Abbildung 2.25).

Zu Beginn des generellen Vorgehens wird basierend auf Bedarfen und den hieraus formulierten Anforderungen ein Validierungsziel formuliert und eine Hypothese zur

Verifikation aufgestellt. Szenarien sind für den Bereich des hochautomatisierten Fahrens direkt mit dem Testfall verknüpft. Beispielsweise wünscht sich ein Fahrer sicher durch einen Baustellenbereichen zu Fahren und dementsprechend die Geschwindigkeit einzuhalten. Eine Anforderung beschreibt nun das automatisierte Setzen eines Geschwindigkeitsgrenzbereichs für den Baustellenbereich. Diese Anforderung kann mit dem V2I-UseCase „RoadWorksWarning“ erfolgen. Das Validierungsziel ist die Untersuchung, ob die Informationen an Fahrzeuge zuverlässig für den Fahrzeuggeschwindigkeitsbereich bis z.B. 150km/h funktioniert. Eine Hypothese kann nun eine Abhängigkeit zwischen Geschwindigkeit und der Übertragungreichweite zwischen Baustellensender und -empfänger im Fahrzeug aufstellen, die es zu prüfen gilt.

Abbildung 6.17 stellt auf der rechten Seite dar, wie ausgehend von dem Stakeholderbedarf der UseCase ① (Nr. in Abbildung 6.17) ausgewählt wird. Dieser lässt sich in Szenarien aufteilen und basierend auf einem Szenarienkatalog ② ein logisches Szenario ③ auswählen. Der Szenarienkatalog ② enthält parametrisierte Szenarien und ist Teil des Referenzsystems¹³. Die Auswahl des logischen Szenarios ③ basiert auf einer Priorisierung und Zielsetzung mit den Methoden FMEA und HARA. Das logische Szenario ④ beinhaltet: Parametergrenzwerte, stochastische Verteilungen, betroffene Systeme und Teilsystem, Umweltbedingungen und Erfüllungskriterien. Beispielsweise kann die Datenübertragung des Geschwindigkeitsgrenzwerts der RSU untersucht werden mit der Signalübertragungreichweite von max. 200m, normalverteilten Testfällen, Regen, den betroffenen Systemen RSU & OBU und dem Erfüllungskriterium, dass die Geschwindigkeit bis 200km/h zu einem rechtzeitigen Abbremsvorgang führt. Mit Hilfe einer x -fachen Generierung¹⁴ ⑤ (z.B. Monte-Carlo Analyse, siehe Kapitel 7.4) können ausgehend von einem logischen Szenario konkrete Szenarien ⑥ abgeleitet werden. (Wäschle, Wolter et al., 2021)

¹³ Es ist anzunehmen, dass Szenarien vorliegen. Dies ist im Automotive Bereich durch zahlreiche Szenarienkataloge häufig der Fall. Alternativ ist ein Szenario zu erstellen (siehe beispielsweise Carvalho Pinheiro et al. (2021)).

¹⁴ Der Faktor $x \in \mathbb{N}$ ist abhängig der Hypothese zu wählen. Beispielsweise beeinflussen die Verteilungsfunktionen von Parametern die Anzahl der Testfälle.

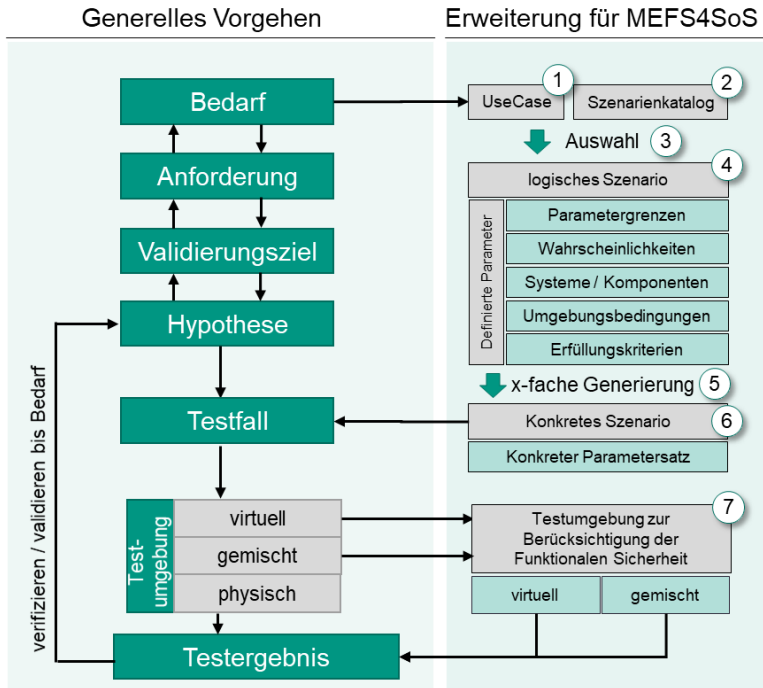


Abbildung 6.17: Generelles Vorgehen und Erweiterung für MEFS4SoS zur Ableitung von konkreten Szenarien (angelehnt an Mandel, Wäschle et al., 2021; Wäschle, Wolter et al., 2021; Wäschle, Wang et al., 2022).

Das konkrete Szenario kann in einer virtuellen, physischen oder gemischt physisch-virtuellen Testumgebung ausgeführt werden. In MEFS4SoS wird in ⑦ auf den Aufbau von virtuell und gemischten Testumgebungen fokussiert. Dies ist je nach Hypothese und Reifegrad des Systems zu wählen. Das Testergebnis kann zur Verifikation oder Falsifikation der Hypothese genutzt werden. Die in Abbildung 6.17 nach oben zeigende Pfeile validieren die Testergebnisse mit einer Verifikation von Hypothese und Anforderung und einer Validierung zu den Bedarfen (siehe Kapitel 7, vgl. Düser, 2010; vgl. Kubin et al., 2023; vgl. Yan, 2020)

Durchführung der Validierung: Die ISO 26262-4 gibt vier Messgrößen für die Bewertung der Testergebnisse vor. Diese sind (ISO 26262-4:2018(E)):

- Kontrollierbarkeit mit der Untersuchung von (un)beabsichtigten Nutzungen
- Effektivität gegenüber externen Messungen

- Effektivität gegenüber Elementen anderer Technologien
- Annahmen, insbesondere welche, die die Gefahrenbewertung beeinflussen

Weiterhin werden im Rahmen des IPEK-XiL-Ansatzes Testfälle eingeführt wie z.B. Testwiederholungen, Langzeittests, Praxistests (ISO/PAS 21448:2019(E)). Hierbei sind das SiD sowie weitere Systeme wie die Umwelt oder die Fahrbahn zu berücksichtigen. Es ist darauf zu achten, dass die Auswahl und Kalibrierung der Systeme zur Erstellung von repräsentativen Tests führt (ISO/PAS 21448:2019(E)). Durch Variation von Parametern entstehen Tests, die unbekannte, unsichere Szenarien berücksichtigen (Wäschle, Wolter et al., 2020). Im Kontext von SoS sind Integrationstests wesentlich, die das Systemverhalten aufgrund der Integration einer Änderung überprüfen. Die Analyse der Testergebnisse kann mit Reviews von Experten oder einer FMEA erfolgen. Die Testergebnisse werden evaluiert, in dem die Erfüllung der Anforderungen geprüft wird. Zusätzlich evaluieren Praxistests, Befragung von Experten und weiteren Stakeholdern die Ergebnisse. Bei der Validierung werden neben den Anforderungen auch die Bedarfe der Stakeholder und die Bedarfe der Validierung evaluiert.

6.3.5 Zusammenfassung

In Kapitel 6.3 wird die neue Methode MEFS4SoS eingeführt. Sie basiert auf wesentlichen Standards der Funktionalen Sicherheit und der Szenarienbeschreibung. Die in MEFS4SoS enthaltenen vier Schritte führen dabei zum Einsatz von etablierten Aktivitäten der Funktionalen Sicherheit wie in HARA und STPA zur Erreichung der Funktionalen Sicherheit. MEFS4SoS wird im folgenden Kapitel 7 angewendet.

6.4 Diskussion & Zwischenfazit

Kapitel 5 stellt sprachliche Elemente und die Methode MEFS4SoS vor, welche die Validierung von V2I ermöglichen. Die Elemente und Methode sind dabei abhängig von generischer, bereichsspezifischer und systemischer Ebene. Somit existieren generische, sprachliche Elementtypen, die für die Modellierung von SoS eingesetzt werden können. Bereich und System schränken den Einsatzrahmen der Sprache ein und fokussieren in der Arbeit auf V2I und auf die Funktionale Sicherheit.

6.4.1 Limitierungen

Es werden mit HARA, STPA, SOTIF und FMEA wesentliche Methoden der Funktionalen Sicherheit betrachtet. Sie dienen als Grundlage für die entwickelte

Methode MEFS4SoS. Für weitere Methoden wie HAZOP (Redmill, Chudleigh & Catmur, 1999) und FTA (Ericson, 1999) wird auf die Literatur verwiesen.

Die Gefahr und das Risiko werden anhand der Risikoprioritätszahl, sowie des ASIL in Kapitel 7.2 quantifiziert. Für weitere Berechnungen, welche statistische Kenngrößen wie die Auftretenswahrscheinlichkeit und Ausfallraten mit einbeziehen, wird auf Takeichi, Sato, Suyama und Kawahara (2011) verwiesen.

In der vorliegenden Arbeit wird die Unterstützung der Validierung untersucht und dementsprechend die iPeM-Layer Produktgeneration und Validierungssystem betrachtet. Die weiteren Layer des iPeM „Produktion“ und „Strategie“ werden demnach nicht betrachtet, können jedoch beispielsweise im Sinne des Produkt-Produktion-CoDesign relevant sein.

6.4.2 Zwischenfazit zur Beantwortung der Forschungsfrage 2

Forschungsfrage 2 greift die zwei Säulen des MBSE Sprache und Methode mit dem Fokus auf die Validierung von V2I auf. Kapitel 6 beantwortet hierzu folgende Forschungsfrage 2:

FF2: Wie kann die durchgängige und kontinuierliche Validierung von V2I durch Modellierungssprache und Methode im Sinne von MBSE unterstützt werden?

Eine Modellierungssprache für die Produktentwicklung spannt einen großen Sprachraum auf. Mit Hilfe der fünf Perspektiven und vier MBSE-Layer lässt sich dieser strukturieren. Die semi-formale, konsistente Sprache SysML dient dabei als Basis der Sprache und unterstützt die Verifikation von Elementtypen und deren Beziehungen. Sie ist im Meta-Systemmodell beschrieben und durch SoS spezifische Sprachelemente wie SoS-Kontext und Contracts erweitert.

Das Referenz-Systemmodell konkretisiert die Elementtypen und beinhaltet beispielhafte Use Cases des hochautomatisierten Fahrens.

Eine Methode zur Erfüllung der Funktionalen Sicherheit für SoS wurde basierend auf bestehenden Normen und Standards der Funktionalen Sicherheit entwickelt. Diese nutzt die Modellierungssprache von Meta- und Referenz-Systemmodell. MEFS4SoS beschreibt den Aufbau eines Systemmodells und die Validierung von konkreten Szenarien in einer Validierungsumgebung. Sie wird in Kapitel 7 angewendet. Die Unterstützung der Effekte der Durchgängigkeit und Kontinuität durch die Sprache und Methode wird im folgenden Kapitel 6.4.3 untersucht, um die Effekte zur Forschungsfrage 2 zu untersuchen.

6.4.3 Diskussion der Effekte und Erfolgsfaktoren

Die **Durchgängigkeit der Validierung** wird erreicht ...

... durch die Vernetzung innerhalb und zwischen den vier MBSE-Layern, drei Abstraktionsebenen und fünf Perspektiven von Meta- und Referenz-Systemmodell. Insbesondere die in Abbildung 6.1 dargestellte Systematik mit Abstraktionsgraden verdeutlicht durch Induktion und Deduktion einen modellübergreifenden Austausch. Speziell für die Validierung von V2I ist eine Erweiterung der Layer mit dem MBSE-Layer Funktionale Sicherheit notwendig. Die Layer sind somit bedarfsgerecht einsetz- und erweiterbar und ermöglichen eine Modellierung zur Unterstützung der Validierung von V2I. Die Methode MEFS4SoS beschreibt die Erstellung und den Einsatz von Elementen und Beziehungen und führt zu einer Modellbildung und der Anwendung der Modelle. Hierbei wird für die Funktionale Sicherheit der Fokus auf die Validierung von V2I gelegt. Es wird deutlich, dass eine enge Verknüpfung zwischen der Methode und den sprachlichen Elementen notwendig ist und die Verknüpfung eine Durchgängigkeit ermöglicht. Beispielsweise sind Begrifflichkeiten durchgängig in der Sprache in allen fünf Perspektiven und MBSE-Layer zu verwenden, als auch in MEFS4SoS. Beispielsweise führt eine Modifikation des Elementtyps <Gefahr> zu Auswirkungen auf die Sprache durch Perspektiven und MBSE-Layer und auf Methoden, die die Sprache einsetzen.

Die **Kontinuität der Validierung** wird erreicht

... durch die modellierten Elemente wie Anforderungen und Funktionen, welche zu unterschiedlichen Zeitpunkten der Validierung eingesetzt werden und demnach auch modelliert werden. Aufgrund der Perspektiven, welche den Problem- und Lösungsraum umspannen, werden in verschiedenen Phasen die Elemente betrachtet und die Modelle erweitert. Weiterhin verdeutlicht die Strukturierung der Modelle nach den Layern im Sinne des iPeM und von MBSE eine Kontinuität über verschiedene Phasen des Produktentwicklungsprozesses. In diesen Phasen werden Aktivitäten der Systemgenerationsentwicklung und Validierung eingesetzt, welche die Sprachelemente kontinuierlich nutzen. Die Elemente in MEFS4SoS lassen sich keiner festen Phase zuordnen. Die Methode lässt sich somit zu unterschiedlichen Zeitpunkten in der Produktentwicklung ausführen. Insbesondere durch die Nutzung von Elementen wie beispielsweise Referenzsystemelemente wird der kontinuierliche Methodeneinsatz verdeutlicht.

Im Rahmen des Forschungsdesigns werden darüber hinaus die Kriterien Vollständigkeit und Konsistenz genannt.

Die **Vollständigkeit** kann im Bereich der Modellbildung schwer überprüft werden. Automatisierte und bedarfsgerechte sprachliche Validierungsregeln prüfen zwar die korrekte Verwendung der Modellierungssprache. Es ist jedoch wichtig die Modelltiefe bedarfsgerecht zu wählen. Dies trifft für SoS insbesondere bei den Schnittstellen auf. Hierbei ermöglicht eine systematische Modellierung angelehnt an den C&C²-Ansatz eine in den Systemgrenzen definierte vollständige Beschreibung. Jedoch ist die Anzahl der zu modellierten Sichten bedarfsgerecht und abhängig von den Elementen. Bei der Verwendung von Meta- und Referenz-Systemmodell werden demnach die Elemente eingeführt und somit fehlende Elemente über Verknüpfungen identifiziert. Daher unterstützt die Modellierung die Vollständigkeit zwischen Elementen, jedoch nicht die Vollständigkeit zur benötigten Anzahl an Elementen z.B. Gefahren. Daher kann die Modellierung die Vollständigkeit nicht zeigen. Methoden der Funktionalen Sicherheit prüfen gezielt einzelne Elemente oder (Kontroll-)Strukturen im Modell, um das Modell zu prüfen und bei Bedarf zu erweitern. Dies kann dazu führen, dass fehlende Elemente identifiziert werden und die Unvollständigkeit verringert wird (vgl. Reduzierung des Bereichs Unknown in Abbildung 2.33).

Die **Konsistenz** ist in der Sprache SysML mit den aufgezeigten sprachlichen Elementen gegeben. Verdeutlicht wird dies beispielsweise durch die Anpassung eines Elements, was zu zahlreichen Veränderungen führt. So sind Effekte wie kaskadierende Fehler zu sehen oder redundante Begriffe, welche im Bereich der Funktionalen Sicherheit identifiziert werden.

Als weiterer Effekt wird die **Konsistenz** von einer durchgängigen und vollständigen Betrachtung abgeleitet. Diese ist durch die semi-formale Modellierungssprache SysML sowie die Erweiterungen gegeben, welche nicht gegen die Konsistenz verstoßen, sondern die Sprache generisch, bereichs- und systemspezifisch erweitern.

7 MBSE-unterstützte Validierung von V2I auf Basis des IPEK-XiL-Ansatzes

In Kapitel 6 wird das Meta- und Referenz-Systemmodell ausgehend von den Implikationen aus Kapitel 5 vorgestellt. In Kapitel 7 wird die Forschungsfrage 3 beantwortet: „Wie sollte eine Validierungsumgebung zur durchgängigen und kontinuierlichen Validierung der Funktionalen Sicherheit von V2I realisiert sein?“ Daher beinhaltet Kapitel 7 nun deduktiv die zwei Abstraktionsgrade Systemmodell und realem System zur Validierung von V2I. Somit soll die Sprache und die Methode MEFS4SoS auf ihre Anwendbarkeit in der Validierung überprüft werden.

Basierend auf dem IPEK-XiL-Ansatz wird das SiD – hier V2I – in seiner Umwelt und in unterschiedlichen, bedarfsgerechten Ausprägungen in einem Systemmodell dargestellt. Abbildung 7.1 visualisiert das IPEK-XiL-Framework für die Validierung von V2X. Für das SoS sind die zwei CS Fahrzeug und Infrastruktur „in-Development“ bzw. in Entwicklung, was durch blaue-Kästen angezeigt wird. Zwischen den zwei CS existiert der Block V2I, welcher die Modellelemente herausstellt, die die Interaktion der Systeme ermöglichen. Das Fahrzeug kann in unterschiedlichen Reifegraden vorliegen: beginnend mit Modell, Software¹, Hardware bis hin zum Gesamtfahrzeug. Das Rest-Fahrzeug wird dabei je nach benötigtem „Rest“ zur Betrachtung des Gesamtfahrzeugs dargestellt. Das Fahrzeug wird von einem Fahrer oder einem Ersatzsystem betrieben, was durch ein Modell, einen Fahrroboter oder einer fahrenden Person realisiert wird. Konkrete Szenarien, Manöver und Testfälle werden in Schritt 4 von MEFS4SoS abgeleitet.

¹ Hierbei wird die Software nicht als fertiges Produkt verstanden, sondern als digitaler, ausführbarer Prototyp und somit als Zwischenschritt zum Gesamtfahrzeug

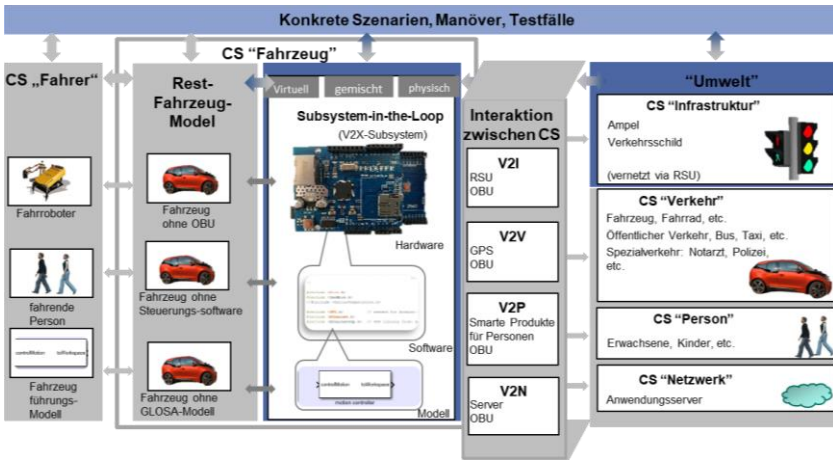


Abbildung 7.1: IPEK-XiL-Framework für V2I (in Anlehnung an Albers, Behrendt et al., 2016; Wäschle, Wang et al., 2022).

Um eine durchgängige Validierung ausgehend von Elementen des Problemraums bis hin zu Testergebnissen zu ermöglichen, werden mehrere Tools eingesetzt und in einer Toolchain verbunden. Abbildung 7.2 veranschaulicht die Verknüpfung des Systemmodells in dem Tool Cameo Systems Modeler über der Testplanungs- und Analyseumgebung in dem Tool ModelCenter mit Ausprägungen des realen Systems basierend auf dem IPEK-XiL-Ansatz in Tools wie MATLAB-Simulink und ns-3.

Die Toolchain kann beginnend auf der linken Seite in Abbildung 7.2 mit dem Systemmodell ausgeführt werden, welches in den Schritten 1 bis 3 von MEFS4SoS (siehe Kapitel 6.3) erstellt wird. Anschließend erfolgt die Validierung im Schritt 4 von MEFS4SoS, welche mit der im Tool ModelCenter implementierten Testplanung beginnt. Ausgehend von einem Validierungsziel werden konkrete Szenarien, Manöver und Testfälle geplant und parametrisiert. Dies kann durch Methoden wie Design of Experiments (Fisher, 1971) oder Monte-Carlo (Metropolis, 1987) erfolgen. Szenarien, Manöver und Testfälle werden anschließend an das reale System in einer Ausprägung von virtuell, gemischt und physisch gesendet. Nach der Testdurchführung werden die Ergebnisse zur Analyse vom ModelCenter gelesen. Die Testergebnisse werden anschließend zur Verifikation von Anforderungen in das Systemmodell übertragen. (Wäschle et al. 2022 basierend auf Mandel et al. 2021)

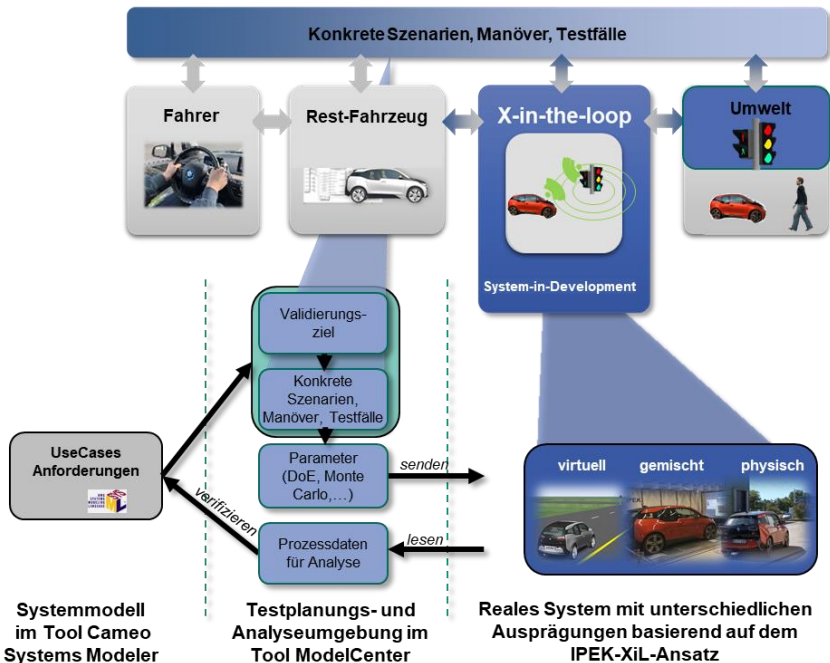


Abbildung 7.2: Konzeptdarstellung der Toolchain für die Verknüpfung von Systemmodell mit realem System (Wäschle et al. 2022 basierend auf Mandel et al. 2021).

Beginnend bei der Systembeschreibung im Systemmodell erfolgen die ersten drei Schritte von MEFS4SoS zur Modellbildung im Cameo Systems Modeler, welche in den Kapiteln 7.1 bis 7.3 vorgestellt werden (siehe Abbildung 7.3). Eine geforderte Flexibilität der Validierung von V2I bezogen auf SoS Anforderungen und die verschiedenen Zeitpunkte in der Produktentwicklung führen zu dem Bedarf, verschiedene Ausprägungen der Validierungsumgebung aufzubauen. Ausgehend von den UseCases, den zugehörigen Anforderungen und Validierungszielen werden die Modellteile im IPEK-XiL-Framework erstellt und in Konfigurationen gekoppelt. Dies führt in Kapitel 7.4 und 7.5 zu den zwei Ausprägungen der Validierungsumgebung im Virtuellen und Virtuell-Physischen. Hierbei wird Schritt vier von MEFS4SoS für den repräsentativen UseCase GLOSA angewendet (siehe Abbildung 7.3).

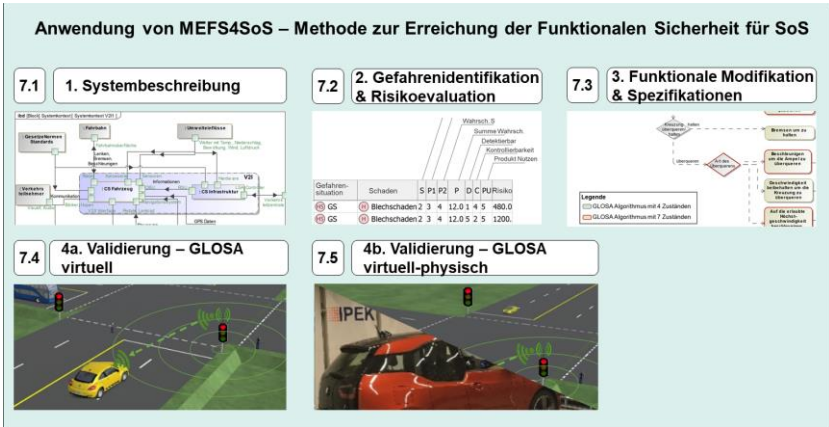


Abbildung 7.3: Übersicht der Kapitel mit der Anwendung der vier Schritten von MEFS4SoS und den zwei Ausprägungen der Validierung in den Schritten 4a & 4b.

7.1 Systembeschreibung

Zu Beginn erfolgt eine Systembeschreibung basierend auf den in Referenz- und Meta-Systemmodell eingeführten fünf Perspektiven (PS-R-F-L-P). In Abbildung 7.4 wird in einem Internen Blockdefinitionsdiagramm der Systemkontext für V2I mit weiteren Systemen und Stakeholdern dargestellt. Es sind basierend auf dem IPEK-XiL-Ansatz die wesentlichen Systeme Fahrzeug, Infrastruktur und Fahrer modelliert. Weiterhin sind in der Umwelt interagierende Systeme sowie weitere Verkehrsteilnehmer relevant. Aufgrund des dynamischen Charakters eines SoS können auch mögliche Erweiterungen wie zu Fahrzeugen (V2V) oder Netzwerkanwendungen (V2N) modelliert werden. Abbildung 7.4 wendet angelehnt an den C&C²-Ansatz mit seinen Wirkflächenpaaren als ProxyPorts zwischen Systemen an. Somit kann schon in dieser Phase eine detailliertere Betrachtung und Diskussion zu den Schnittstellen erfolgen.

Für eine Identifikation von Gefahren gilt es außerdem bereits in der Systembeschreibung Kontrollstrukturen je nach Detaillierungsebene in den Perspektiven Funktionen, Logische Elemente oder Physische Elemente zu modellieren. Für eine Black Box Modellierung eignen sich Variablen als Input- und Output Größen. Bei detaillierter Betrachtung kann insbesondere in den Perspektiven Logische Elemente und Physische Elemente eine Modellierung angelehnt an den C&C²-Ansatz mit ProxyPorts und Konnektoren erfolgen.

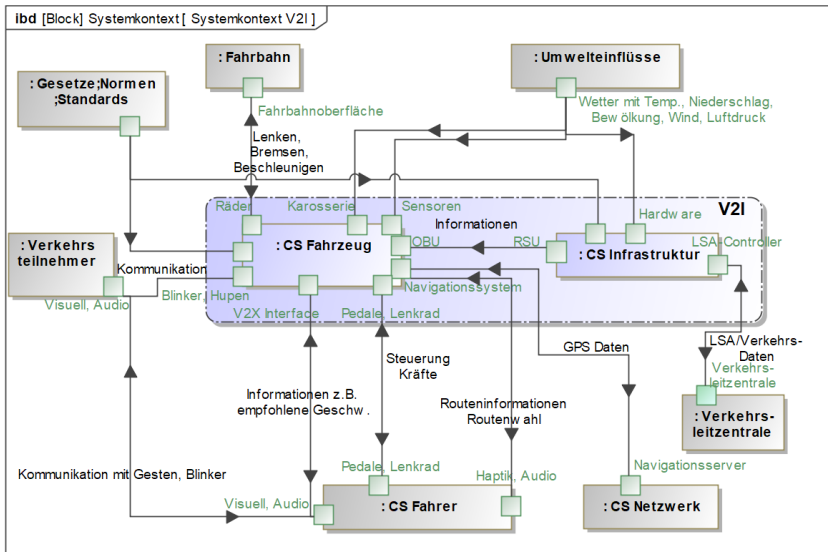


Abbildung 7.4: V2I-Systemkontext als White Box Ansatz mit Schnittstellen.

In Abbildung 7.4 lassen sich Regelkreise bereits identifizieren, beispielsweise zwischen Fahrzeug, Infrastruktur und Umwelteinflüssen. Dies lässt sich nun in Schritt zwei der Gefahrenidentifikation und Risikoevaluation betrachten. Weiterhin lassen sich detaillierte Systembeschreibungen basierend auf den fünf Perspektiven des Meta- und Referenz-Systemmodells erstellen. Ein Auszug der Elemente zeigt die Wirkkette von CS Fahrer zu Systemanforderungen in Abbildung 7.5. Der Auszug visualisiert beispielhaft die Wirkkette für den Sub-UseCase der Fahrt mit GLOSA innerorts.

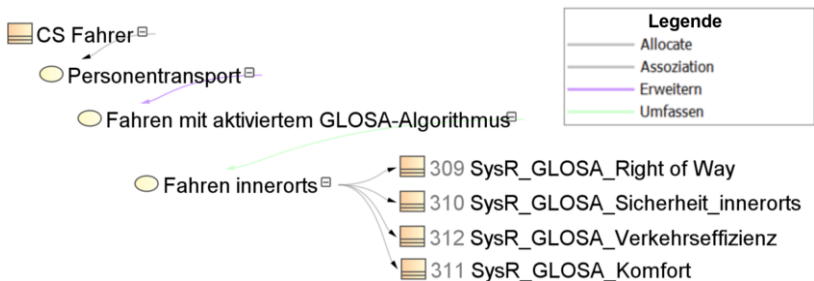


Abbildung 7.5: Auszug der Wirkkette von CS Fahrer zu Systemanforderungen mit unterschiedlichen Beziehungen.

In der vorliegenden Arbeit liegt der Fokus auf dem repräsentativen UseCase GLOSA (vgl. Andersen et al., 2016, S. 3 zu repräsentativer Auswahl eines Rotlicht-Assistenten an einer Verkehrskreuzung) (vgl. Galbras, 2018, S. 5 zur repräsentativen Auswahl eines UseCases mit Verkehrskreuzung). Bei GLOSA (Green Light Optimal Speed Advisory) wird eine Information zur Geschwindigkeitsprädiktion im Targetfahrzeug von einer LSA an das Targetfahrzeug oder den Fahrer des Fahrzeugs gesendet. Diese kann verschiedene Informationen wie Signalphase und verbleibende Zeit in der jeweiligen Signalphase enthalten. Das betrachtete Szenario wird innerorts bei Geschwindigkeiten bis 50km/h untersucht.

Tabelle 7.1 zeigt hierzu die wesentlichen Elemente an, welche in der Systembeschreibung entstehen. In den folgenden Schritten von MEFS4SoS wird diese Tabelle entsprechend erweitert.

Tabelle 7.1: Auszug der zu modellierenden Elemente für den UseCase GLOSA	
Element	Beschreibung
SoS-Kontext	Siehe Abbildung 7.4
Bedarfe	Zuverlässige V2I-Kommunikation für eine sichere, komfortable und verkehrseffiziente Verkehrskreuzungsüberfahrt.
UseCase	Fahren mit aktiviertem GLOSA-Algorithmus
Validierungsbedarfe	VB1: Evaluation Zuverlässigkeit der Signalübertragung bei V2I. VB2-4: Evaluation Sicherheit, Komfort, Verkehrseffizienz bei Passier- und Haltemanövern mit V2I.
Systemanforderungen	SysR1 „Right of Way“: Das Fahrzeug hält bei Rot über mechanisches oder elektrisches Bremsen an der Haltelinie mit einer Toleranz [-2,1]m an ohne Verkehrsregeln zu verletzen und fährt bei einer Überquerung der Haltelinie bei Grün oder Gelb mit einer Toleranz von [0,2]m über die Verkehrskreuzung. SysR2 „Sicherheit“: Das Fahrzeug fährt nicht schneller als die Geschwindigkeits-Begrenzung innerorts von 50km/h [+1]km/h und es hält nicht hinter der Haltelinie [+0,1]m und nicht in größerer Entfernung als 5m [+1]m vor der Haltelinie. SysR3 „Komfort“: Das Fahrzeug beschleunigt nicht im Betrag stärker als 2m/s ² [+0,1]m/s ² . SysR4 „Verkehrseffizienz“: Das Fahrzeug hält bei der Haltelinie [+1]m und fährt beim Passieren nicht langsamer als 15km/h [+1]km/h.

Initiale Validierungsaktivitäten können für das Systemmodell bereits in diesem Schritt erfolgen, mittels Verifikation von SysML-Elementen (vgl. FAS-Plugin für Funktionsstrukturen (No Magic Product Documentation, 2022)) und bedarfsgerechten Validierungsregeln (vgl. Wäschle, Behrendt et al., 2021, S. 105).

7.2 Gefahrenidentifikation und Risikoevaluation

Zu Beginn von Schritt 2 werden unsichere Kontrollaktionen innerhalb des Systems und in der Interaktion mit weiteren Systemen identifiziert. In Abbildung 7.6 wird eine Funktionsstruktur zur Realisierung von GLOSA modelliert. Zwischen den Systemen Fahrzeug und Infrastruktur können beispielsweise fehlende, falsche oder zu falscher Zeit übertragener Daten zu einer unsicheren Kontrollaktion führen, beispielsweise zu einer zu spät erkannten Funktionseinschränkung der Nachrichtenübertragung von der LSA mit ihrer RSU zu OBU.

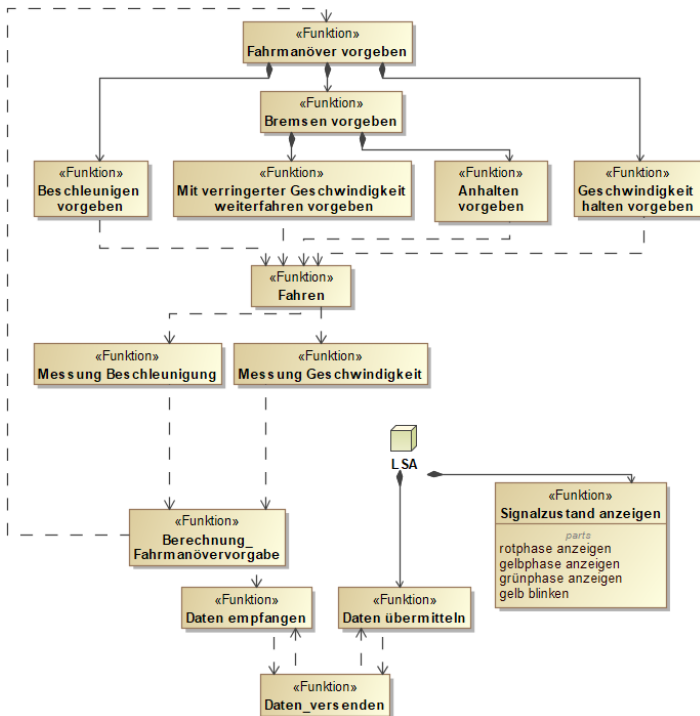


Abbildung 7.6: Funktionsstruktur zur Realisierung des UseCase GLOSA.

Hierzu eignen sich detailliertere Betrachtungen der Funktionen und der realisierenden Logischen Elemente. Abbildung 7.7 zeigt die Übertragung einer CAM-Nachricht an, welche kabellos und über BUS übermittelt und transformiert wird.

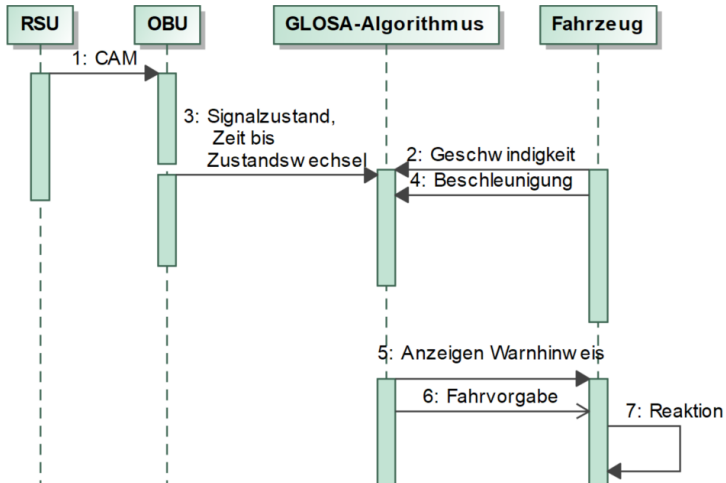


Abbildung 7.7: Sequenzdiagramm zur Nachrichtenübertragung für GLOSA.

Aufgrund der Abhängigkeiten wird deutlich, dass eine verzögerte Nachricht zu einer Gefahr führen kann. Weiterhin fällt die unklare Kombination mit weiteren Systemen beispielsweise für ACC auf oder das fehlende Feedback von anderen Sensoren auf die Fahrvorgabe, was zu einer ungewünschten Positionierung des Fahrzeugs führen kann.

Eine systematische Betrachtung von Einflussfaktoren auf die Nachrichtenübertragung bei GLOSA zeigt Abbildung 7.8. Außerhalb des Fahrzeugs ist die Signalübertragung abhängig von Sender, Signalausbreitung und Empfänger. Einflüsse können zu Gefahren führen wie beispielsweise die Gefahr einer Kollision aufgrund einer zu geringen Übertragungreichweite, die zu einer zu späten Reaktion des Fahrzeugs führt.

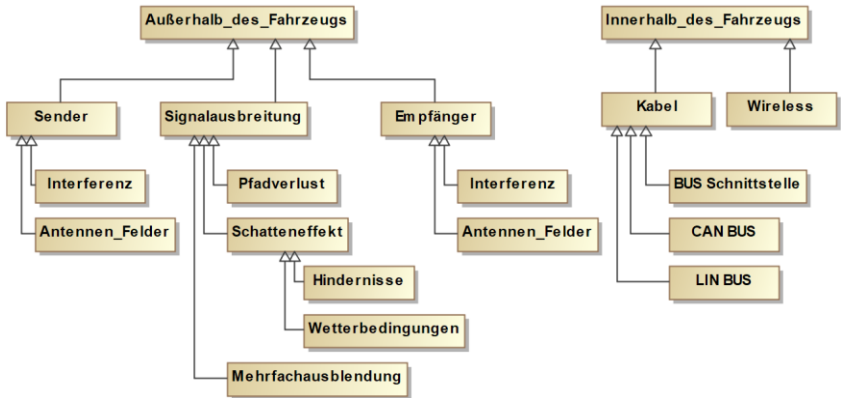


Abbildung 7.8: Auszug der Einflussfaktoren auf die Nachrichtenübertragung bei GLOSA (Yi, 2019)¹.

Demzufolge wird Tabelle 7.1 durch die wesentlichen Elemente der Funktionalen Sicherheit in Tabelle 7.2 erweitert.

Tabelle 7.2: Übersicht – Elemente der Funktionalen Sicherheit für die beschriebene Gefahrensituation	
Element	Beschreibung
Gefahrensituation	GS: Überfahrt Verkehrskreuzung mit LSA & aktivem GLOSA
Gefahr	G1: Fahrzeug kollidiert mit temporärem Teilsystem (siehe Abbildung 2.32) G2: Fahrzeug hält in gefährlicher Position
Risikopriorisierung (RPZ, ASIL)	Für G1: RPZ: 480, ASIL: C Für G2: RPZ: 510, ASIL: C

Eine modellbasierte Darstellung in Anlehnung an das Referenz-Systemmodell befindet sich in Tabelle 7.3, in der das Risiko berechnet wird und ein ASIL benannt werden kann.

¹ Abschlussarbeit (unveröffentlicht)

Tabelle 7.3: Risikotabelle (basierend auf ISO 26262-3:2018(E), S. 19; No Magic Product Documentation, 2022)

Schadensursache	Gefahr	Gefahrensituation	Schaden	Risikofaktoren							
				S	P1	P2	P	D	C	PU	Risiko
(R) GLOSA-Algorithmus	(HZ) O G2	(HS) GS	(H) Blechschaden	2	3	4	12.0	1	4	5	480.0
(R) Übertragungsreichweite	(HZ) M G1	(HS) GS	(H) Blechschaden	2	3	4	12.0	5	2	5	1200.0



Die Berechnung des Risikos erfolgt dabei nach der Formel (basierend auf ISO 26262-3:2018(E), S. 19; No Magic Product Documentation, 2022):

$$Risiko_G(\text{Schaden } (TS)) = S_{\text{Schaden}} * E_G * A_{GS} * A_{\text{Schaden}} * K * Nutzen_{TS}$$

- G: Gefahr
 - GS: Gefahrensituation
 - S: Schwere
 - A: Auftretenswahrscheinlichkeit
 - E: Einfluss
 - K: Kontrollierbarkeit
 - TS: Teilsystem
- 7.1

Das ASIL lässt sich aus den Faktoren Schwere (S), Auftretenswahrscheinlichkeit (A) und Kontrollierbarkeit (K) bestimmen (basierend auf ISO 26262-3:2018(E), S. 10):

$$ASIL = (S\{0,1,2,3\}, A\{0,1,2,3,4\}, K\{0,1,2,3\})$$

7.2

z. B. ASIL Level D → (S3, A4, K3),
 z. B. ASIL Level C → (S3, A4, K2); (S3, A3, K3); (S2, A4, K3)

Für die Evaluation des Risikos einzelner Teilsysteme kann nach der ISO 26262 eine Dekomposition des ASIL eines (Teil-)systems durchgeführt werden. Außerdem können Validierungsaktivitäten nach dem „Pull-Prinzip der Validierung“ erfolgen (vgl. Abbildung 2.25), um Bewertungsgrößen wie die Schwere der Gefahr zu verifizieren.

7.3 Funktionale Modifikation und Spezifikationen

Schritt 3 von MEFS4SoS kann ausgehend von der Betrachtung des Referenzsystems im Meta-Systemmodell in Kapitel 6.1.5 erfolgen. Tabelle 7.4 benennt wesentliche Modifikationen von Referenzsystemelementen der vorangegangenen Produktgeneration G_{n-1} als Teil für die Perspektive Funktionen, jedoch auch für die angrenzenden Perspektiven Anforderungen und Logische Elemente. Funktionale Anforderungen beeinflussen Funktionen, welche durch Logische Elemente realisiert werden. Hierbei kommen Ausprägungsvariation (AV) und Prinzipvariation (PV) zum Einsatz. Zudem werden die Änderungen spezifiziert, sodass Parameter benannt werden.

Tabelle 7.4: Auszug der zu modellierenden veränderten Elemente für den UseCase GLOSA mit Variationsarten des Modells der PGE	
Element	Beschreibung
Veränderte Anforderungen	AV: Geschwindigkeitsgrenzwert $\{R_n;50, G_n;30\}$ km/h
Veränderte Funktionen	AV: Signalphasensequenz $\{R_n;(gr,g,r,rg,gr) ;G_n:$ unter anderem $(r,rg,gr,g,r)\}$
Veränderte logische Elemente	AV: Zustandsänderung des GLOSA Algorithmus von 4 auf 7 AV: Übertragungreichweite der RSU PV: automatisiertes Fahren anstatt Fahrer
Validierungsbedarfe	Evaluation der Signalübertragungreichweite bei V2I Evaluation der Sicherheit, Komfort, Verkehrseffizienz bei Passier- und Haltemanöver

Beispielhaft wird die Änderung der Zustände im GLOSA-Algorithmus in einem Baumdiagramm in Abbildung 7.9 dargestellt. Hierbei führt die größere Anzahl an Zuständen zu differenzierteren Aktionen des GLOSA Algorithmus.

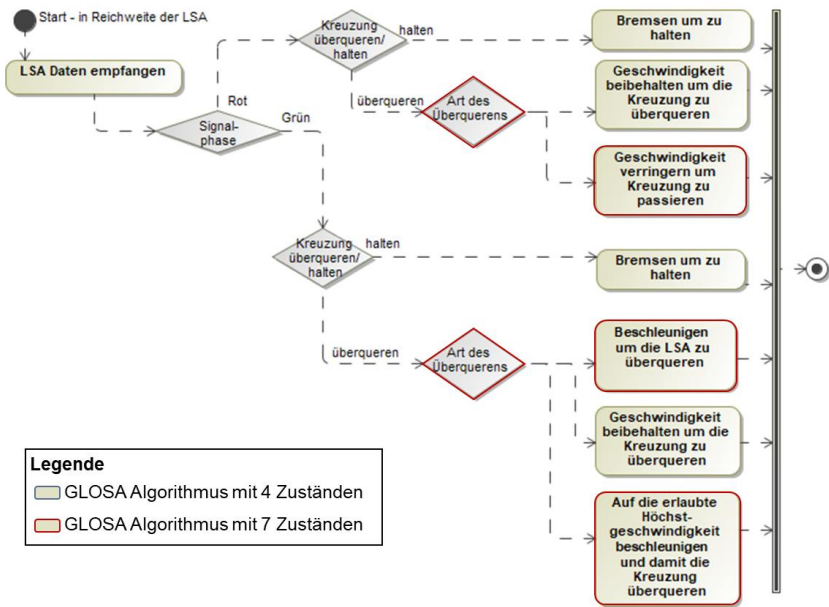


Abbildung 7.9: GLOSA Algorithmen mit vier und sieben Systemzuständen (angelehnt an Glimm & Zadek, 2021).

Nachfolgend werden mit Hilfe der sprachlichen Elemente aus Meta- und Referenz-Systemmodell (siehe Kapitel 6), die ausführbaren realen Systeme innerhalb der Validierungsumgebung vorgestellt und die Validierung durchgeführt. Hierbei wird das Vorgehen zur Erstellung von konkreten Szenarien (siehe Abbildung 6.17) angewendet.

7.4 Validierung – GLOSA virtuell

Die Validierung des realen Systems kann in einer virtuellen Validierungsumgebung in verschiedenen Tools erfolgen. Hierbei muss eine Verknüpfung zum Systemmodell realisiert werden, um die Nachvollziehbarkeit und Validierung von Bedarfen zu prüfen. Die Verknüpfung der Modelle mit Testfällen und ausführbaren quantitativen Modellen (vgl. Bungartz, Zimmer, Buchholz & Pflüger, 2013) oder Prüfständen erfolgt über die Software ModelCenter (siehe Abbildung 7.2 und Abbildung 7.10). Diese ermöglicht über Protokolle wie Telnet (Postel 1980) die Kommunikation verschiedener Programme. Zudem wird eine probabilistische Analyse der Messdaten ermöglicht. Mit Hilfe der Tools können somit Testergebnisse

analysiert werden und mit Anforderungen, welche im Layer der Systemgeneration modelliert sind, abgeglichen werden.

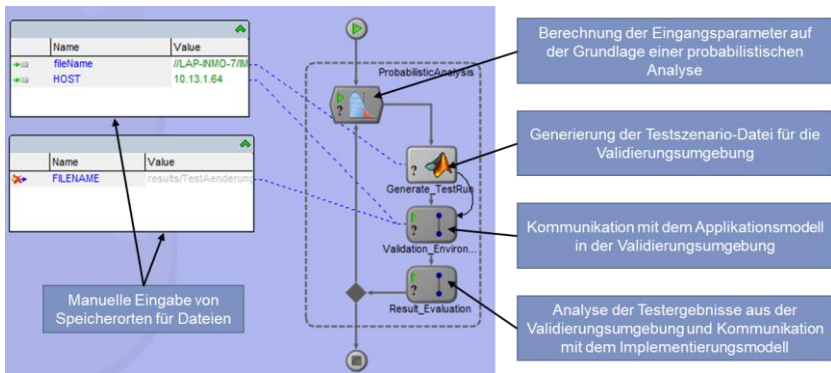


Abbildung 7.10: Iteratives Prozessablaufdiagramm zur Testplanung und -auswertung für die Anwendung des IPEK-XiL-Ansatzes (Mandel, Wäschle et al., 2021).

7.4.1 Beschreibung des Validierungssystems

Für die Validierung wird ein Systemmodell und reales System eingeführt (siehe Kapitel 6). Gemäß der Grundannahme 2 der vorliegenden Arbeit in Kapitel 3.2 werden die Modelle basierend auf dem IPEK-XiL-Ansatz erstellt und angewendet.

Beginnend mit der Perspektive des Problemraums sind zunächst Bedarfe und Ziele im Validierungssystem zu bestimmen. Abgeleitet aus den Elementen des Layers Produktgeneration: Stakeholder-, Validierungsbedarfe, Anforderungen sowie den Elementen des MBSE-Layers der Funktionalen Sicherheit sind die Elemente Validierungsziel, Hypothese und Validierungsumgebung in Tabelle 7.5 aufgelistet. Die in Schritt 3 von MEFS4SoS durchgeführten Funktionale Modifikationen und Spezifikationen gilt es abzusichern. Hierzu werden Parameter innerhalb von Hypothesen bestimmt, deren Tests neben dem Wissensgewinn im Sinne des explorativen Testens (Kaner, Falk & Nguyen, 1999) explizit zur Überprüfung der Modifikationen dienen.

Tabelle 7.5: Validierungsziele, -umgebung und Hypothesen für den UseCase GLOSA innerorts	
Element	Beschreibung
Validierungsziel 1	Ermöglicht der GLOSA-Algorithmus dem Fahrzeug, die Kreuzung unter Einhaltung der Sicherheits-, Komfort- und Verkehrseffizienzvorschriften zu passieren?
Hypothese 1.1	Variation der Zustände des GLOSA-Algorithmus von 4 auf 7 führt zu Änderungen >10% bei der Erfüllung des Sicherheitskriteriums ² für Passiermanöver.
Hypothese 1.2	Variation der Startgeschwindigkeit hat keinen Einfluss auf die Erfüllung des Sicherheitskriteriums ² für Passiermanöver bei 7 GLOSA-Zuständen.
Hypothese 1.3	Variation der Startgeschwindigkeit führt zu <95% Erfüllung des Komfortkriteriums für Passiermanöver ohne Halten bei 7 GLOSA-Zuständen.
Hypothese 1.4	Variation der Startgeschwindigkeit führt zu <95% Erfüllung der Verkehrseffizienz kriteriums ² für Passiermanöver ohne Halten bei 7 GLOSA-Zuständen
Validierungsziel 2	Wie groß ist der Einfluss der Übertragungsreichweite, Latenz und des Verlustmodells auf die Halteposition?
Hypothese 2.1	Variation der Übertragungsreichweite um 50% führt zu Änderungen $\geq 10\%$ bei der Halteposition.
Hypothese 2.2	Variation der Latenz um 10% führt zu Änderungen $\geq 5\%$ bei der Halteposition.
Hypothese 2.3	Variation des Verlustmodells von RPLM auf LPDM führt zu Änderungen $\leq 5\%$ bei der Halteposition.
Validierungs-umgebung	Siehe Abbildung 7.11

Zur Beschreibung der Validierungsumgebung veranschaulicht Abbildung 7.11 ein Systemmodell des IPEK-XiL-Ansatzes. Es werden die virtuellen Modelle V2X-Modul (als OBU), RSU und Umwelt über virtuelle Koppelsysteme verbunden.

² Siehe Erfüllungskriterien in Tabelle 7.7: Übersicht - Test zum Systemverhalten
148

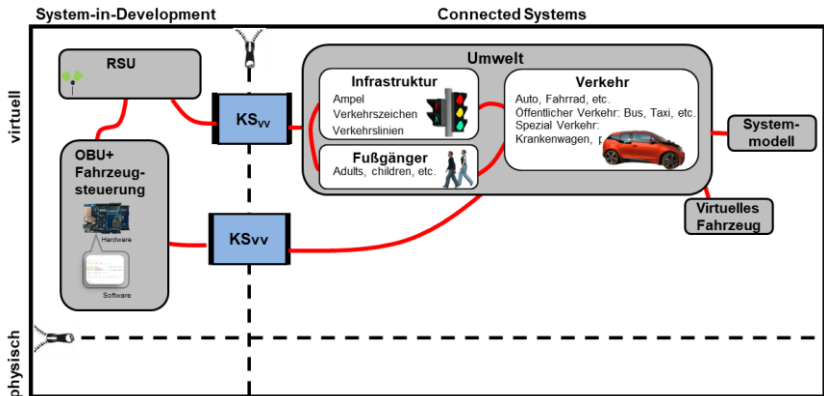


Abbildung 7.11: Systemmodell des IPEK-XiL-Ansatzes für V2I.

Die Modelle werden hierbei in den Tools MATLAB-Simulink und ns-3 erstellt und über die Prozessautomatisierung (siehe Abbildung 7.2, Abbildung 7.10) ausgeführt. Das Systemmodell greift dabei auf Modellbibliotheken zu, um flexibel je nach UseCase und Systeme im SoS die Modelle zu erstellen. Abbildung 7.12 veranschaulicht die Bibliothek im Tool MATLAB-Simulink, welche aus mehreren Modellen besteht, die zu Modulen zusammengefasst sind. Mittels einer Schnittstellenbeschreibung angelehnt an den C&C²-Ansatz sind die Modelle geclustert und geeigneten Modulen zugeordnet. (Chen, 2020)³

Die Validierung erfolgt abhängig vom UseCase in einem realen System. Für den UseCase GLOSA wird dies beispielhaft im nachfolgenden Kapitel durchgeführt. Für eine Erweiterung der MATLAB-Simulink Bibliothek um eine realistische Beschreibung der Kommunikation ist die physikalische und Netzwerkschicht des IEEE-Schichtenmodells in einer Co-Simulation von MATLAB und ns-3 modelliert und durchgeführt (siehe Wäschle, Behrendt et al., 2021).

³ Abschlussarbeit (unveröffentlicht)

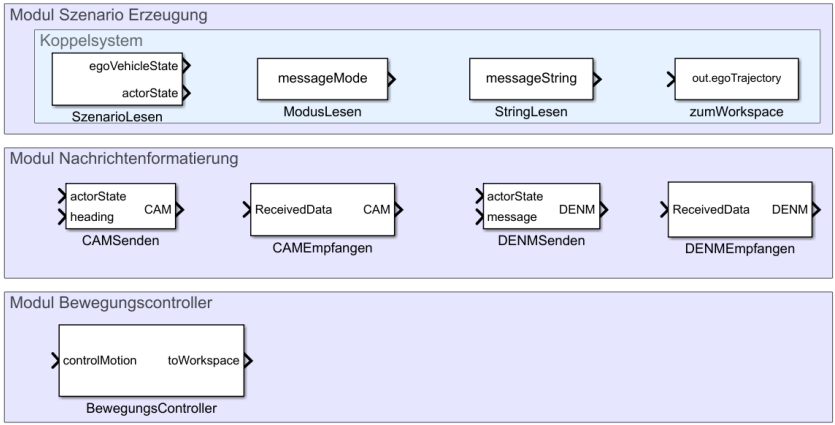


Abbildung 7.12: Auszug der Modellbibliothek des Systemmodells zum Aufbau von realen Systemen in MATLAB-Simulink und ns-3 mit Referenz UseCases in der Kategorie Demos (Chen 2020)⁴.

7.4.2 Erstellung von konkreten Szenarien

Tabelle 7.6 listet die Elemente Test, Testumgebung und logische Szenarien als „Zwischenprodukte“ zur Erstellung der konkreten Szenarien auf. Es werden zwei Tests mit verschiedenen Parametervariationen erstellt, welche im nachfolgenden Kapitel mit deren Testergebnissen beschrieben werden.

Tabelle 7.6: Test zum Systemverhalten für den UseCase GLOSA innerorts

Element	Beschreibung	
Test	Test zum Systemverhalten	Test zur Signalübertragung
Testumgebung	Cameo Systems Modeler, ModelCenter, MATLAB	MATLAB, ns-3
Szenarien katalog	aus Projekte in Kapitel 2 ⁵	
Logisches Szenario	Passier- und Haltemanöver mit Variation der Fahrzeuganzahl, Startgeschwindigkeit, Signalphasen, GLOSA-Zustände	Auswirkungen auf Verkehrseffizienz mit Variation von Verlustmodell,

⁴ Abschlussarbeit (unveröffentlicht)

⁵ Zum Beispiel 5GAA (2019); AdaptiVe Consortium (2017); Böhm (2020).

		Verzögerung, Frequenz, Übertragungreichweite
Konkretes Szenario	Normalverteilung von 1000 Testfällen <u>Parametervariationen:</u> <ul style="list-style-type: none"> • Startgeschwindigkeit [15,30]km/h, • Zustände GLOSA-Algo. {4,7} 	Gleichverteilung von 4 Testfällen <u>Parametervariationen:</u> <ul style="list-style-type: none"> • Verlustmodell {LDPLM, RPLM}, • Sendefrequenz {1,2}Hz, • Sendereichweite {60,100}m

7.4.3 Durchführung der Validierung

In diesem Kapitel werden die zwei Tests zum Systemverhalten und zur Signalübertragung vorgestellt.

Test zum Systemverhalten

In einem ersten Test (siehe Tabelle 7.7) wird untersucht, ob das Fahrzeug sich an der Verkehrskreuzung entsprechend den Verkehrsregeln verhält (engl. „Right of Way“) sowie Kriterien einhält für die Sicherheit (Geschw. < 50km/h), den Komfort ($|$ Beschleunigung $|$ < 2m/s²) sowie das effiziente Halten an der Haltelinie (-5m < Halteposition < 0m). Als Parametervariationen werden mit einer Monte-Carlo-Simulation⁶ die Anzahl der Fahrzeuge, Startgeschwindigkeit, Signalphase und Zustände im GLOSA-Algorithmus untersucht. (Wäschle, Wang et al., 2022)

Element	Beschreibung
Erfüllungskriterien	„Right of Way“, Passieren ⁷ : Sicherheit (Geschw. ≤ 50km/h), Komfort ($ $ Beschleunigung $ $ ≤ 2m/s ²), Verkehrseffizienz (Geschw. ≥ 15km/h) Halten ⁸ : Sicherheit (0 ≤ Halteposition ≤ 2m), Komfort ($ $ Beschleunigung $ $ ≤ 2m/s ²), Verkehrseffizienz (-5m < Halteposition < 0m)
Testfälle	1000 Testfälle mit Passier- und Haltemanöver
Testergebnis	Die Anzahl der Zustände des GLOSA-Algorithmus beeinflusst schwach die Erfüllung des Sicherheitskriteriums (siehe Abbildung 7.15)

⁶ Monte-Carlo-Simulation wird abhängig von einer statistischen Verteilung (hier Normalverteilung) Zufallstestfälle ausgewählt. Das Projekt AdaptiVe Consortium (2017) verwendet die Monte-Carlo-Simulation ebenfalls im Bereich des hochautomatisierten Fahrens.

⁷ Geschwindigkeit für Sicherheit innerstädtisch nach §3 Abs. 3 Bundesministerium für Justiz (2022); komfortable Beschleunigung nach Bae, Moon und Seo (2019) und Geschwindigkeit für Verkehrseffizienz nach Yang, Wang, Ding, Tan und Ran (2021).

⁸ Halteposition 0m nach §49 Abs. 3 Bundesministerium für Justiz (2022)

Abbildung 7.13 veranschaulicht einen beispielhaften Testfall im zeitlichen Verlauf. Es wird gezeigt, dass das Fahrzeug bremst, wenn die Phase auf Gelb wechselt, und die Verzögerung erhöht, wenn die Lichtsignalanlage auf Rot wechselt.

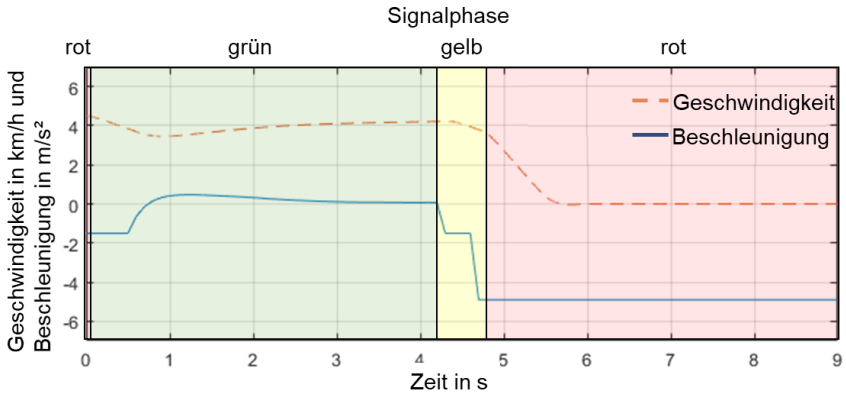


Abbildung 7.13: Exemplarische Messergebnisse für GLOSA (zeitlich) (übersetzt nach Wäschle, Wang et al., 2022).

Die durchgeführten 1000 Testfälle lassen sich unterteilen in 800 Passiermanöver und 200 Haltemanöver. Bei einem Passiermanöver fährt das Fahrzeug über die Verkehrskreuzung ohne zu halten. Ein Scatterplot in Abbildung 7.14 stellt die Erfüllung der Kriterien Sicherheit für 800 Testfälle des Passiermanövers dar (vgl. Wäschle, Wang et al., 2022). Es werden je nach Art und Achse des Plots die Kriterien Komfort, Sicherheit und Verkehrseffizienz als Grenzl意思en visualisiert.

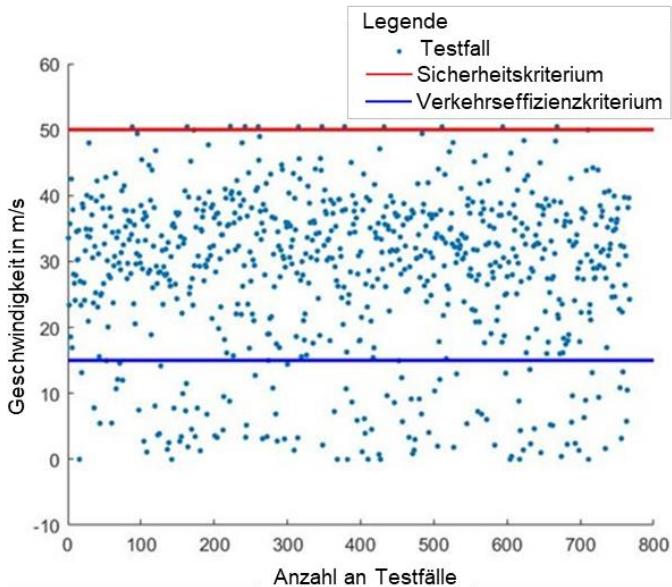


Abbildung 7.14: Scatter Plot für GLOSA mit Sicherheits- und Verkehrseffizienzgrenzwert (N=800) (übersetzt nach dem Vortrag der Publikation Wäschle, Wang et al., 2022).

Die Halte- und Passiermanöver können mit Hilfe von statistischen Untersuchungen analysiert werden. Es zeigt sich in Abbildung 7.15, dass eine größere Anzahl an Zuständen des GLOSA-Algorithmus zwar die Kriterien nicht zu 100% erfüllt, die Abweichungen jedoch auf Potenziale hinweisen. Mit der Entwicklung eines optimalen Algorithmus könnten die GLOSA-Algorithmen weiter verbessert werden (vgl. Stevanovic, Stevanovic & Kergaye, 2013).

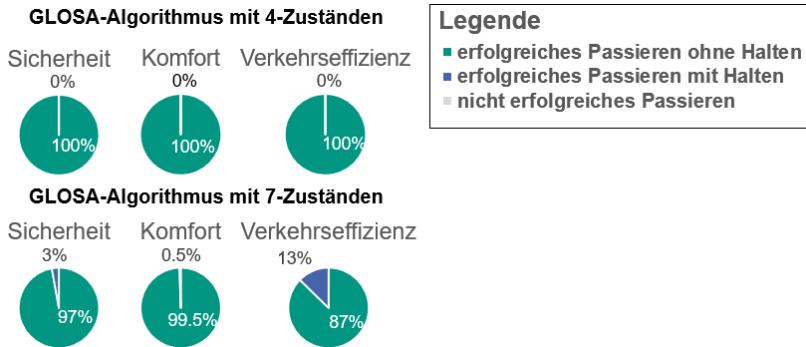


Abbildung 7.15: Überprüfung der Erfüllung der drei Kriterien Sicherheit, Komfort und Verkehrseffizienz beim Passiermanöver für GLOSA (N=800).

Basierend auf den Testergebnissen lassen sich die Hypothesen (siehe Tabelle 7.5) evaluieren. Hierzu wird die Evaluation beispielhaft für Passiermanöver und für den GLOSA-Algorithmus mit 7 Zuständen vorgenommen.

- Hypothese 1.1 ist falsifiziert, da für die Sicherheit der Unterschied zwischen 4 und 7 Zustände des GLOSA-Algorithmus weniger als 10% sind.
- Hypothese 1.2 ist verifiziert, da die Variation der Startgeschwindigkeit zur vollständigen Erfüllung des Sicherheitskriteriums führt.
- Hypothese 1.3 ist falsifiziert, da es zu einer Erfüllung von 99,5% beim Komfortkriterium für Passiermanöver ohne Halten kam.
- Hypothese 1.4 ist verifiziert, da zu 87% das Verkehrseffizienz Kriterium ohne Halten erfüllt ist.

Test zur Signalübertragung

In einem zweiten Test wurde eine Simulation in der für die Netzwerktechnik spezialisierten Linux-basierten Software ns3 durchgeführt (vgl. Wäschle, Behrendt et al., 2021). Das hierbei eingesetzte Wave-Modell beschreibt die Signalübertragung nach IEEE 802.11p (siehe van Heesch, 2016).

Tabelle 7.8: Übersicht - Test zur Signalübertragung für GLOSA

Element	Beschreibung
Erfüllungskriterien	Verkehrseffizienz (-5m<Halteposition<0m)
Testfall	4 Testfälle zur Signalübertragung
Testergebnis	Es existieren große Unterschiede in der Halteposition abhängig von der Frequenz und Übertragungsreichweite. Mit zunehmender Latenz werden die Unterschiede zwischen den Verlustmodellen LDPLM und RPLM kleiner.

Als wesentliche Variation werden zwei Verlustmodelle bei der Signalübertragung betrachtet. Das „Log Distance Propagation Loss Model (LDPLM)“ verwendet eine logarithmische Abhängigkeit zwischen Signalübertragungsverlusten und Distanz zwischen Sender und Empfänger (siehe Formel 7.3). (Maygua-Marcillo, Urquiza-Aguilar & Paredes, 2018)

$$L = L_0 + 10n \log\left(\frac{d}{d_0}\right)$$

L : Pfadverlust

L_0 : Pfadverlust bei Referenzdistanz = 46.67dB⁹

7.4

n : Pfadverlust Distanz Exponent = 2 (Rappaport, 2002)

d : Distanz

d_0 : Referenzdistanz = 1m⁹

Das „Range Propagation Loss Model“ (RPLM) beschreibt den Signalübertragungsverlust in Abhängigkeit der Reichweite zwischen Sender und Empfänger. Bei einer überschrittenen maximalen Reichweite von 100m wird ein Verlust von -1000dBm festgelegt. (Maygua-Marcillo et al., 2018)

In der Simulation wird die Verzögerungszeit für die Signalübertragung von der RSU zum Fahrzeug stufenweise auf 0ms, 50ms, 100ms und 200ms festgelegt¹⁰ (siehe Abbildung 7.16). Jeder Simulationsaufbau wird zehn Mal wiederholt. Die durchschnittliche Entfernung zwischen der Halteposition des Fahrzeugs und der LSA wird als Ergebnis angezeigt. Abbildung 7.16 zeigt den Halteabstand über die

⁹ Default-Wert in LDPLM in ns-3

¹⁰ Untersuchungen von Guerrero-Ibanez, Flores, Damian-Reyes, Barba und Reyes (2011) deuten auf eine Verzögerungszeit kleiner 100ms für 802.11p hin.

Latenz bei Variation des Verlustmodells, der Frequenz und der Reichweite der Signalübertragung¹¹.

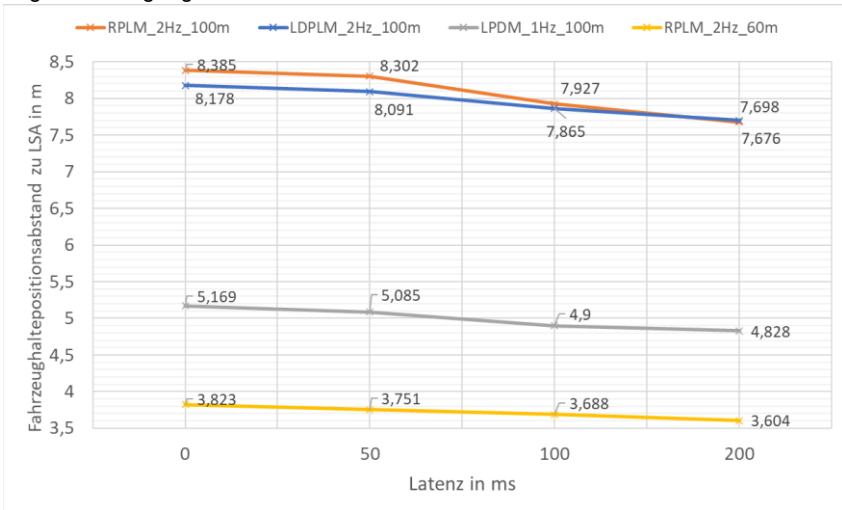


Abbildung 7.16: Fahrzeughaltposition in Abhängigkeit der Latenz für zwei Verlustmodelle der Signalübertragung (RPLM und LPDM) mit Übertragungsfrequenzen (1,2 Hz) und Abständen (60,100 m) für GLOSA.

Die Abbildung verdeutlicht große Unterschiede in der Halteposition abhängig von der Frequenz und Übertragungsreichweite. Mit zunehmender Latenz werden die Unterschiede zwischen den Verlustmodellen LDPLM und RPLM kleiner. In zehn Testfällen führen die innerhalb der ETSI Norm (ETSI EN 302 637) eingestellten Frequenzen von 1Hz und 2Hz im LPDM zu den unterschiedlichen durchschnittlichen Entfernungen zwischen Fahrzeughaltposition und Ampel von 8,2m und 5,2m.

¹¹ Die Latenz von 1 bis 4ms befinden sich nach der Domänenanforderung in Guerrero-Ibanez et al. (2011) und McKinley (2021) im erlaubten Bereich. Die Frequenz zur wiederholten Übertragung der Signale ist für CAM-Nachrichten nach Breu, Brakemeier und Menth (2014) zwischen 1 und 10Hz zu wählen. Die Reichweite von 60m bis 100m ist plausibel, Guerrero-Ibanez et al. (2011) formulieren sogar 250m Reichweite.

Basierend auf den Testergebnissen lassen sich die mit Hypothesen nummeriert für Testfall 2 aus Tabelle 7.5 evaluieren:

- Hypothese 2.1 ist verifiziert, da die Verringerung der Übertragungsreichweite um 50% zu einer Verringerung der Halteposition um $(8,178\text{m}-3,823\text{m})/ 8,178\text{m} >53\%$ führt.
- Hypothese 2.2 ist falsifiziert, da die Verringerung der Sendefrequenz von 2 zu 1 bei der Halteposition zu einer Veränderung $<1\text{m}$ Veränderung im Schnitt führt.
- Hypothese 2.3 ist verifiziert, da die Variation des Verlustmodells bei der Halteposition zu einer Veränderung $<1\%$ führt.

7.4.4 Zusammenfassung und Nutzung der Testergebnisse

In Kapitel 7.4 wird eine virtuelle Validierung durchgeführt. Abgeleitet aus zwei Validierungszielen wird zu Beginn die Validierungskonfiguration vorgestellt, mit welcher in zwei Tests zu Systemverhalten und Signalübertragung sieben Hypothesen untersucht werden. Mit Hilfe von Passier-, Haltemanöver und Manöver zur Verkehrseffizienz konnten die nachfolgenden Testergebnisse realisiert werden:

1. Die Zustandsänderung des GLOSA-Algorithmus hat keinen großen Einfluss ($<10\%$) auf die Sicherheit.
2. Die Startgeschwindigkeit hat keinen Einfluss (0%) auf die Sicherheit.
3. Die Startgeschwindigkeit hat keinen großen Einfluss ($<10\%$) auf den Komfort.
4. Die Startgeschwindigkeit hat einen Einfluss ($>10\%$) auf die Verkehrseffizienz.
5. Die Übertragungsreichweite hat einen Einfluss ($>10\%$) auf die Halteposition.

Die Testergebnisse können nun mit den Stakeholderanforderungen und -bedarfen abgeglichen werden (siehe Tabelle 7.9). So werden die Unterschiede der Parameter angezeigt und die Erfüllung der Anforderungen geprüft. Beispielsweise wird die maximale Beschleunigung (a_{max}) zwischen Referenzwert (engl. Baseline) und gemessenem Wert (engl. Value) verglichen (Delta). Das Delta wird mit der Anforderung an eine komfortable, maximale Beschleunigung (GLOSA_ac_komf_max) geprüft und somit die Anforderung verifiziert.

Tabelle 7.9: Analyse im Tool ModelCenter basierend auf Testergebnissen im Systemmodell zu Variablen mit Abweichung zur Baseline (oben) und zu Anforderungen mit Verifizierung und Einhaltung von Grenzwerten (unten)

Name	Baseline	Value	Units	Change	Delta	Delta %																																				
GLOSA <ul style="list-style-type: none"> a_max <table border="0" style="margin-left: 20px;"> <tr> <td>2,0000</td> <td>0,95109</td> <td></td> <td></td> <td>1,0489</td> <td>52,446</td> </tr> </table> b_max <table border="0" style="margin-left: 20px;"> <tr> <td>-2,0000</td> <td>-1,5623</td> <td></td> <td></td> <td>0,43771</td> <td>-21,885</td> </tr> </table> s_stop <table border="0" style="margin-left: 20px;"> <tr> <td>0,0</td> <td>-5,1686</td> <td></td> <td></td> <td>5,1686</td> <td></td> </tr> </table> stop <table border="0" style="margin-left: 20px;"> <tr> <td>2,0000</td> <td>3,0000</td> <td></td> <td></td> <td>1,0000</td> <td>50,000</td> </tr> </table> v_min <table border="0" style="margin-left: 20px;"> <tr> <td>15,000</td> <td>6,4963E-7</td> <td></td> <td></td> <td>15,000</td> <td>100,000</td> </tr> </table> v_max <table border="0" style="margin-left: 20px;"> <tr> <td>50,000</td> <td>21,099</td> <td></td> <td></td> <td>28,901</td> <td>57,802</td> </tr> </table> 	2,0000	0,95109			1,0489	52,446	-2,0000	-1,5623			0,43771	-21,885	0,0	-5,1686			5,1686		2,0000	3,0000			1,0000	50,000	15,000	6,4963E-7			15,000	100,000	50,000	21,099			28,901	57,802						
2,0000	0,95109			1,0489	52,446																																					
-2,0000	-1,5623			0,43771	-21,885																																					
0,0	-5,1686			5,1686																																						
2,0000	3,0000			1,0000	50,000																																					
15,000	6,4963E-7			15,000	100,000																																					
50,000	21,099			28,901	57,802																																					

Requirements		
Name	Satisfied	Margin
34 GLOSA_s_stop_min		0,16862
32 GLOSA_dec_comf_max		-0,43771
40 GLOSA Komfort		
39 GLOSA Effizienz		
33 GLOSA_s_stop_max		-7,1686
30 GLOSA_v_pass_min		15,000
31 GLOSA_ac_comf_max		1,0489
29 GLOSA_v_pass_max		-28,901
38 GLOSA Sicherheit		

Hierzu werden die Testergebnisse im Systemmodell (teil-)automatisiert als Value Properties in ein Internes Blockdefinitionsdiagramm eingefügt. Es wird folgende beispielhafte Nachvollziehbarkeitskette erzielt (siehe Abbildung 7.17). Die Kette zeigt basierend von einem Testergebnis eine qualitative Wirkung auf die Testfahrt und die Testvariable Startgeschwindigkeit im UseCase GLOSA. Die Testvariable hat wiederum Wirkungen zu verschiedenen Variablen und weiteren Systemelementen. Zum Schluss kann auf den Bedarf z.B. nach Komfort oder Sicherheit geschlossen werden.



Abbildung 7.17: Traceability von Testergebnis zu Bedarfen realisiert mit Hilfe von unterschiedlichen Beziehungen.

Im Kontext des Produktentstehungsprozesses ist neben der virtuellen Validierung die physische Validierung von Prototypen und des Gesamtsystems relevant. Daher stellt das folgende Kapitel eine gemischt virtuell-physische Validierung basierend auf dem IPEK-XiL-Ansatz vor.

7.5 Validierung – GLOSA virtuell-physisch

Wie bereits bei der virtuellen Validierung erfolgt die Verknüpfung der Tools über den Aktivitäten Systemmodellierung, Testplanung und -durchführung. Für eine Anbindung an physische Teilsysteme muss jedoch eine Anpassung zur Integration der Prüfstandsarchitektur mit dessen Echtzeitsystem (hier: AVL InMotion powered by IPG CarMaker) erfolgen. Die Interaktion mit der Prüfstandshardware ist ortsunabhängig möglich, jedoch sollte sie nur mit Prüfstandspersonal und unter Sicherheitsvorkehrungen durchgeführt werden. Bezüglich der generellen Vernetzung von Prüfständen wurden Sicherheitszustände definiert (siehe Wäschle und Wolter et al. (2021), Pecha et al. 2022), welche in MATLAB-Simulink einen sicheren Betrieb und eine sichere Kommunikation zwischen Prüfstandsmodellen realisieren. Nachfolgend wird das Validierungssystem mit der integrierten Prüfstandshardware vorgestellt.

7.5.1 Beschreibung Validierungssystem

Physische (Teil-)Systeme lassen sich in einer virtuell-physisch gemischten Validierungsumgebung evaluieren. Dies ist insbesondere in der frühen Phase des Produktentstehungsprozesses sowie in SoS notwendig, um flexibel und schnell Validierungsergebnisse in den Produktentstehungsprozess zurückzugeben.

Tabelle 7.10 listet analog zur virtuellen Validierung Validierungsziele, -umgebung und Hypothesen im Validierungssystem auf.

Tabelle 7.10: Übersicht – Validierungsziele und Hypothesen für den UseCase GLOSA innerorts	
Element	Beschreibung
Validierungsziel 1	Ermöglicht der GLOSA-Algorithmus dem Fahrzeug, die Kreuzung unter Einhaltung der Sicherheits-, Komfort- und Verkehrseffizienzvorschriften zu passieren?
Hypothese 1.1	Variationen der Signalphasensequenz hat einen Einfluss <10% auf erfolgreiches Passieren.
Hypothese 1.2	Die Fahrt mit dem Fahrer führt zu >10% zu nicht erfolgreichem Passieren.
Validierungsziel 2	Wie groß ist der Einfluss des Fahrers, des Prüfstands / der Straße auf die Halteposition bzw. auf die erfolgreiche Überquerung der Kreuzung?
Hypothese 2.1	Variation der Signalverzögerung um 40% hat einen Einfluss von $\geq 10\%$ auf erfolgreiches Passieren.
Hypothese 2.2	Die Fahrt mit dem Fahrer führt zu <10% zu nicht erfolgreichem Passieren.
Validierungs-umgebung	Siehe Abbildung 7.19, Abbildung 7.20, Abbildung 7.21

Die Validierungsziele und Hypothesen führen dazu, dass unterschiedliche Validierungskonfigurationen notwendig sind, um die verschiedenen Ausprägungen von virtuell/physisch, Straße/Rollenprüfstand und Fahrer/Fahrroboter zu realisieren.

Abbildung 7.18 verdeutlicht den prinzipiellen Aufbau der Validierungskonfiguration mit dem Einsatz eines Rollenprüfstands. Die Abbildung visualisiert Datenpfade und verarbeitende (Teil-)Systeme, welche in einem geschlossenen Regelkreis (engl. closed-Loop) eingesetzt werden.

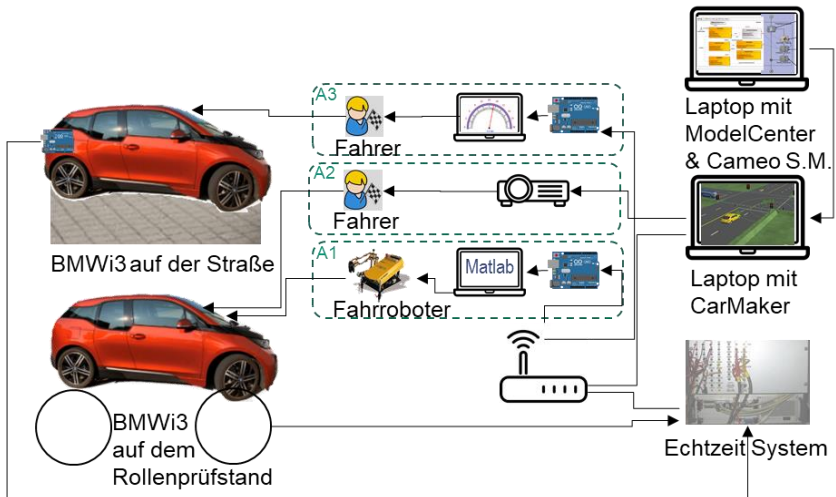


Abbildung 7.18: Prinzipdarstellung der virtuell-physischen Validierungsumgebung mit drei Validierungskonfigurationen dargestellt in den alternativen Pfaden A1 bis A3 (übersetzt nach Wäschle, Wang et al., 2022).

Die in der Validierungsumgebung generierten alternativen Datenpfade A1 bis A3 führen zu unterschiedlichen realen Systemen. Diese lassen sich in drei Modellen mit einer Architektur nach dem IPEK-XiL-Ansatz strukturieren und bieten die Möglichkeit, diese Modelle mit ihren Ausprägungen schnell zu überblicken. Nachfolgend werden die entsprechenden drei Validierungskonfigurationen mit Hilfe der nummerierten Datenübertragung beschrieben und dargestellt.

① Die V2I-Anwendungsszenarien werden in Cameo Systems Modeler im Systemmodell modelliert, anschließend in ModelCenter Testpläne erstellt und in CarMaker das ausführbare Szenario (siehe Abbildung 7.2). Das Szenario läuft auf dem AVL InMotion Echtzeitsystem. Anschließend sind drei alternative Abläufe möglich:

Alternative 1 (A1) – Fahrroboter fährt auf dem Prüfstand (siehe Abbildung 7.19): ②

Die UDP-Daten der Lichtsignalanlage werden über einen Router gesendet und vom ESP8266 empfangen. ③ Nachdem ESP8266 die UDP-Daten verarbeitet und entpackt hat, werden die Ergebnisse an der seriellen Schnittstelle ausgegeben. ④ MATLAB liest die Zeichenkette und die Steuerbefehle werden an den Fahrroboter gesendet. ⑤ Das Fahrzeug wird durch eine skalierte Kraft auf das Beschleunigungs- und das Verzögerungspedal des Fahrroboters betrieben. ⑥ Schließlich werden die Daten über ein Prüfstands-CAN an die virtuelle Umgebung

zurückgegeben, welche auf einem Echtzeitsystem ausgeführt wird. ⑦ Zum Schluss werden die Daten an ModelCenter und Cameo Systems Modeler übertragen und ausgewertet.

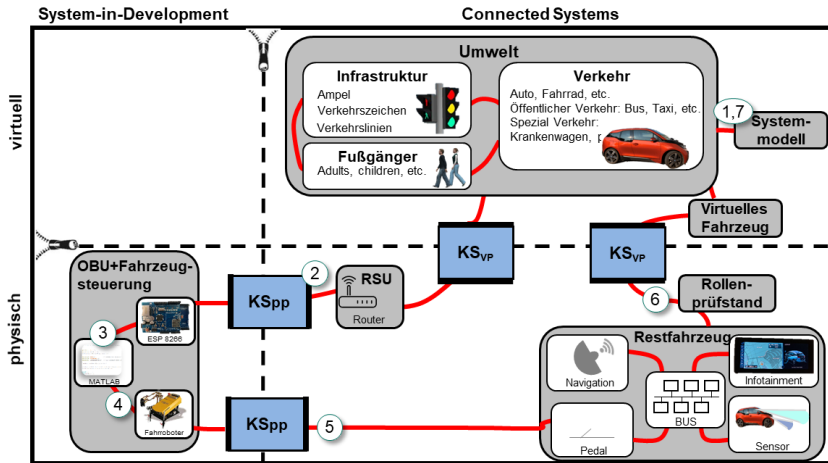


Abbildung 7.19: Alternative 1 – Fahrroboter fährt auf dem Prüfstand.

Alternative 2 (A2) – Fahrer fährt auf dem Prüfstand (siehe Abbildung 7.20):

② Die UDP-Daten der Lichtsignalanlage werden über einen Router per WiFi übertragen und vom ESP8266 empfangen. ③ Nachdem ESP8266 die UDP-Daten verarbeitet und entpackt hat, werden die Ergebnisse in einem GUI angezeigt. ④ Der Fahrer steuert das Fahrzeug unter der Anweisung des GUI. ⑤ Anschließend werden die CAN-Daten des Fahrzeugs über LAN zurück an das Echtzeitsystem gesendet, auf welchem die virtuelle Umgebung mit der Lichtsignalanlage simuliert wird. ⑥ Zum Schluss werden die Daten an ModelCenter und Cameo Systems Modeler übertragen und ausgewertet.

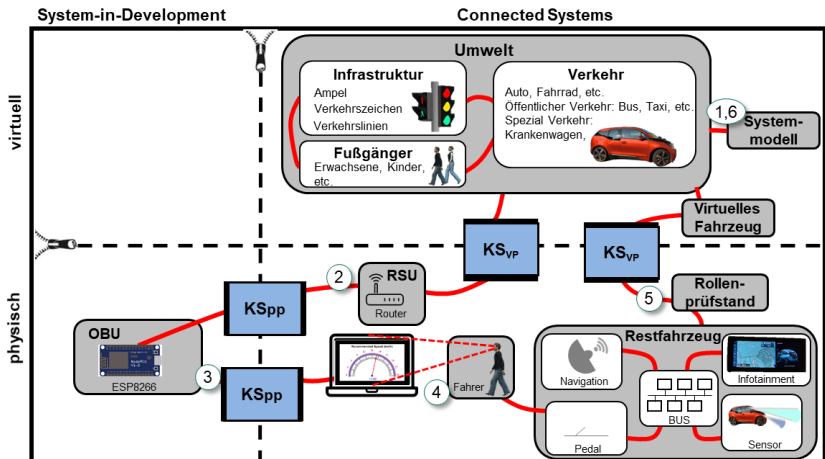


Abbildung 7.20: Alternative 2 – Fahrer fährt auf dem Prüfstand.

Alternative 3 (A3) – Fahrer fährt auf der Straße (siehe Abbildung 7.21):

② Die UDP-Daten werden über einen Router übertragen und vom ESP8266 empfangen. ③ Nachdem ESP8266 die UDP-Daten verarbeitet und entpackt hat, werden die Ergebnisse in das GUI angezeigt. ④ Nachdem der Fahrer das Fahrzeug unter der Anweisung des GUI steuert, werden die CAN-Daten des Fahrzeugs in UDP umgewandelt und ⑤ gesendet. ⑥ Der Router empfängt die Daten und ein Arduino Uno wandelt sie in CAN-Daten um. ⑦ Danach werden die CAN-Daten über LAN zurück an das AVL InMotion Echtzeitsystem gesendet. ⑧ Zum Schluss werden die Daten an ModelCenter und Cameo Systems Modeler übertragen und ausgewertet.

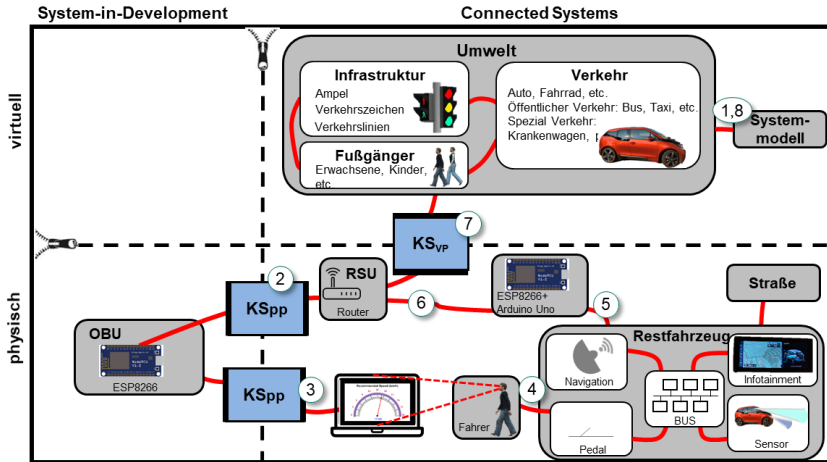


Abbildung 7.21: Alternative 3 – Fahrer fährt auf der Straße.

In den drei Alternativen können Koppelsysteme die Grenze zwischen physisch und virtuell sowie SiD und Connected Systems überschreiten. Koppelsysteme haben einen Einfluss auf die angeschlossenen Modelle. In dem Systemmodell ist es möglich zu bewerten, welches Koppelsystem zwischen verschiedenen Modellen benötigt wird. Zum Beispiel besteht ein Koppelsystem zwischen einem physischen und einem virtuellen Modell aus einer Verbindung zwischen einem Fahrzeug und einem Softwaretool. Tabelle A.1 in Anhang A gibt einen Überblick zu den eingesetzten Koppelsystemen und deren Funktionen.

Basierend auf den Validierungskonfigurationen können nun im nachfolgenden Kapitel konkrete Szenarien zur Validierung erstellt werden.

7.5.2 Erstellung von konkreten Szenarien

Tabelle 7.11 listet analog zur virtuellen Validierung in Kapitel 7.4.2 die Elemente Test, Testumgebung, logische- und konkrete Szenarien auf. Zur Erreichung der Validierungsziele werden zwei Tests mit verschiedenen Parametervariationen erstellt, welche im nachfolgenden Kapitel mit deren Testergebnissen vorgestellt werden.

Tabelle 7.11: Übersicht - Tests zu Passiermanöver für den UseCase GLOSA innerorts		
Element	Beschreibung	
Test	Test zu Passiermanöver auf der Straße und am Rollenprüfstand	Test zu Passiermanöver am Rollenprüfstand
Testumgebung	Cameo Systems Modeler, ModelCenter, CarMaker	Cameo Systems Modeler, ModelCenter, CarMaker
Szenarienatalog	Siehe Projekte in Kapitel 2 ¹²	
Logisches Szenario	Passiermanöver bei Variation der Signalstartphase	Passiermanöver bei Variation der Signalstartphase und Signalverzögerung
Konkretes Szenario	Gleichverteilung von 35 Testfällen <u>Parametervariation:</u> <ul style="list-style-type: none"> • Signalstartphase {rot,gelb,grün,rot-gelb} 	Gleichverteilung von 8 Testfällen <u>Parametervariation:</u> <ul style="list-style-type: none"> • Signalstartphase {rot,gelb,grün} • Signalverzögerung {400,1000}ms¹³

7.5.3 Durchführung der Validierung

In diesem Kapitel werden die zwei Tests zu Passiermanöver auf der Straße mit Fahrer und am Rollenprüfstand ohne Fahrer vorgestellt.

Test zu Passiermanöver mit Fahrer auf dem Rollenprüfstand und auf der Straße

Aus der virtuellen Umweltsimulation werden Messwerte wie der longitudinale Abstand zwischen Targetfahrzeug und Haltelinie, Geschwindigkeit und Beschleunigung ausgegeben. Das empfohlene longitudinale Fahrzeugverhalten wird in Abhängigkeit der Messwerte Geschwindigkeit, Signalphase, Dauer in aktueller Signalphase und dem Abstand zur Haltelinie bestimmt. Die physische Umgebung des Fahrzeugrollenprüfstands oder das BUS-System des Fahrzeugs liefert die aktuelle Fahrzeuggeschwindigkeit. Der Fahrer bekommt die in Abbildung 7.22 dargestellten Informationen „empfohlene Geschwindigkeit“, „aktuelle Geschwindigkeit“ und „kalkulierte Geschwindigkeit“ über ein Display angezeigt. Die „empfohlene Geschwindigkeit“ wird von dem GLOSA-Algorithmus mit 4 Zuständen berechnet. Die „aktuelle Geschwindigkeit“ wird aus dem CAN-Signal des Fahrzeugs

¹² Z.B.: 5GAA (2019); Adaptive Consortium (2017); Böhm (2020).

¹³ Abgeleitet aus Angaben zu Infotainment und Verkehrseffizienz in MacHardy, Khan, Obana und Iwashina (2018, S. 1864).

bestimmt, wohingegen die „kalkulierte Geschwindigkeit“ aus der virtuellen Umgebung stammt.



Abbildung 7.22: GUI zur Anzeige für den Fahrer.

In Tabelle 7.12 sind Erfüllungskriterien, die Testfälle und das Testergebnis beschrieben.

Tabelle 7.12: Übersicht – Test zu Passiermanöver mit Fahrer auf dem Rollenprüfstand und auf der Straße für den UseCase GLOSA	
Element	Beschreibung
Erfüllungskriterien	Kein Halten und keine Verletzung von Verkehrsregeln „Right of Way“
Testfälle	35 Testfälle
Testergebnis	Das Fahrzeug konnte in 89 % der Testfälle die LSA bei Grün oder Gelb passieren ohne anzuhalten.

Bei Variation der Signalstartphase wird geprüft, ob das Fahrzeug die Verkehrskreuzung passieren kann, ohne aufgrund einer roten Signalphase anzuhalten.

Abbildung 7.23 veranschaulicht die Ergebnisse, bei denen beim erfolgreichen Passieren einer nicht roten Signalphase die verbleibende Zeit in der Signalphase beim Überfahren der Haltelinie aufgezeichnet wird. Drei von vier Signalphasensequenzen werden 10-mal getestet und die Mittelwerte der Einzelmessungen gebildet. Insgesamt konnte das Fahrzeug in 89 % der Fälle die Lichtsignalanlage bei Grün oder Gelb passieren.

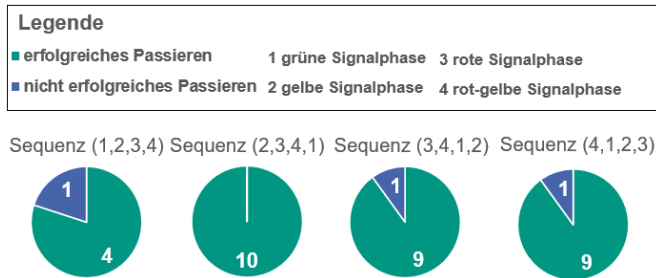


Abbildung 7.23: Testergebnisse zu erfolgreichen Passiermanövern ohne Halten mit Fahrer für GLOSA.

Exemplarisch sind in Abbildung 7.24 die Messdaten eines Testfalls im zeitlichen Verlauf dargestellt. Die Reaktionszeit des Fahrers (typischerweise 0,18s) und das manuelle Regeln einer empfohlenen Geschwindigkeit führen zu Schwankungen im Geschwindigkeitsverlauf. Dies deutet auf einen Einfluss des Fahrers auf den Geschwindigkeitsverlauf hin, da ein exaktes Einhalten der empfohlenen Geschwindigkeit aufgrund von Reaktionszeiten und Toleranzen nicht möglich ist.

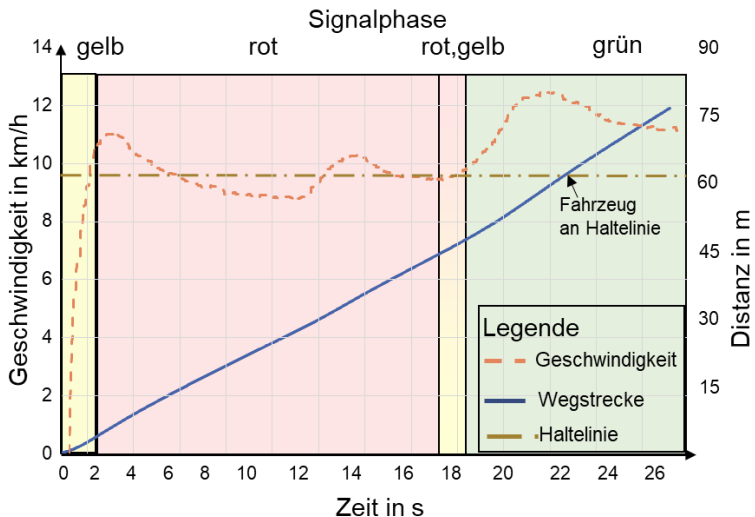


Abbildung 7.24: Exemplarische Messergebnisse Passieren ohne Halten mit Fahrer (zeitlich) (übersetzt nach Wäschle, Jiang, Behrendt & Albers, 2020).

Die Ergebnisse in Abbildung 7.23 zeigen, dass das virtuelle Auto in den meisten Testfällen die Kreuzung passiert. Daher kann das Passiermanöver in der Validierungsumgebung realisiert werden.

Basierend auf den Testergebnissen lassen sich die Hypothesen evaluieren:

- Hypothese 1.1 ist falsifiziert, der Einfluss der Signalphasensequenz auf das erfolgreiche Passieren ist nicht geringer als 10%.
- Hypothese 1.2 ist verifiziert, da der Fahrer einen Einfluss (>10%) auf das erfolgreiche Passieren hat.

Test zu Passiermanöver ohne Fahrer am Rollenprüfstand

Im zweiten Test (siehe Tabelle 7.13) wurde ein Fahrroboter eingesetzt, um die longitudinale Fahrempfehlung automatisiert an das Fahrzeug weiterzugeben (vgl. Wäschle, Behrendt et al., 2021). Aufgrund von Sicherheitsrichtlinien ist der Einsatz des Fahrroboters nur in Verbindung mit dem Rollenprüfstand möglich, sodass diese Tests nicht auf der Straße durchgeführt werden.

Element	Beschreibung
Erfüllungskriterien	Keine Verletzung von Verkehrsregeln „Right of Way“
Parametervariation	Signalstartphase {rot,gelb,grün}, Zeitverzögerung (Delay)
Testfall	Gleichverteilung von 6 Testfällen mit kleiner Verzögerung und 2 Testfällen mit großer Verzögerung
Testergebnis	Das Fahrzeug konnte in 100 % der Testfälle mit eingestellter Verzögerung der OBU Datenverarbeitung von 400ms die Kreuzung passieren.

Abbildung 7.25 veranschaulicht die Ergebnisse bei unterschiedlichen Startpunkten der Signalphasen bei Verzögerung von 400ms. Das Fahrzeug konnte in 100 % der Fälle die Lichtsignalanlage passieren.

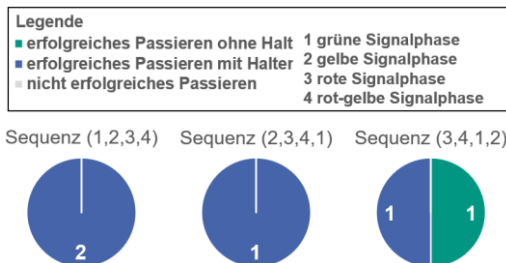


Abbildung 7.25: Testergebnisse für erfolgreiche Passiermanöver mit und ohne Halten mit Fahrroboter für GLOSA.

Abbildung 7.26 illustriert den zeitlichen Verlauf exemplarischer Messergebnisse. Hierbei wird ersichtlich, dass durch den Einsatz des Fahrroboters ein geglätteter Geschwindigkeitsverlauf dargestellt wird, im Gegensatz zur Fahrt mit einem Fahrer. Die Abbildung veranschaulicht, dass das Fahrzeug nach 53,6m anhält, wenn die Haltelinie noch 3,7m vor dem Fahrzeug liegt. Nach dem Ende der roten Signalphase beschleunigt das Fahrzeug wieder, um die Verkehrskreuzung zu überqueren.

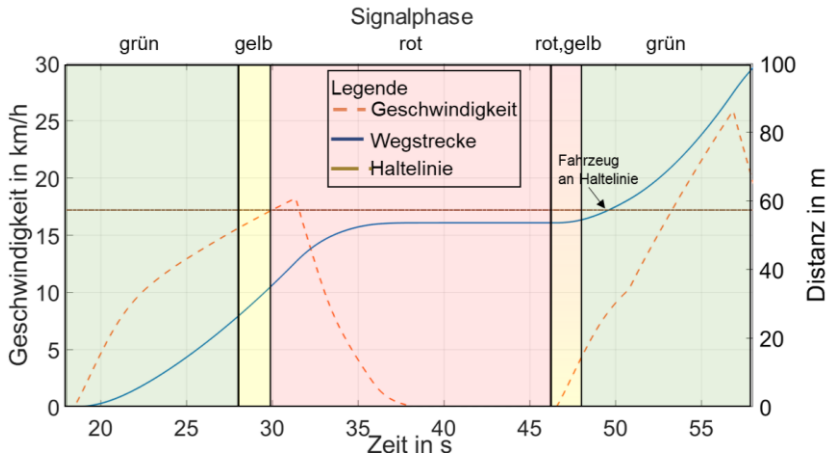


Abbildung 7.26: Exemplarische Messergebnisse Passieren mit Halten mit Fahrroboter (zeitlich) (vgl. Wäschle, Behrendt et al., 2021).

Zusätzlich wurde die Gefahr einer erhöhten Signalverzögerung getestet. Diese führt zu einer Störung des Geschwindigkeitsregelungsalgorithmus und damit zum vorzeitigen Abbruch.

Basierend auf den Testergebnissen lassen sich die Hypothesen evaluieren:

- Hypothese 2.1 ist verifiziert, der Einfluss bei einer Steigerung der Signalverzögerung ist deutlich zu erkennen. Einen Einfluss und den UseCase-spezifischen Einsatz bestätigen MacHardy, Khan, Obana und Iwashina (2018).
- Hypothese 2.2 ist verifiziert, da die Fahrt mit dem Fahrroboter immer zu erfolgreichen Passiermanövern führt bei der Signalverzögerung von 400ms.

7.5.4 Zusammenfassung und Nutzung der Testergebnisse

Kapitel 7.5 führt eine gemischt virtuell-physische Validierung durch. Abgeleitet aus zwei Validierungszielen werden zu Beginn drei alternative Validierungskonfigurationen vorgestellt, mit welchen eine Fahrt auf dem Rollenprüfstand, auf der Straße sowie mit Fahrer oder Fahrroboter möglich ist. Mit Hilfe von Passiermanövern konnten die nachfolgenden Testergebnisse realisiert werden:

1. Die Signalphasensequenz hat keinen geringen Einfluss (>10%) auf das erfolgreiche Passieren.
2. Der Fahrer hat einen Einfluss (>10%) auf das Passiermanöver.
3. Die Signalverzögerung hat einen Einfluss (>10%) auf das Passiermanöver.
4. Der Fahrroboter konnte reproduzierbar das Passiermanöver erfolgreich durchführen (100%).

Die Testergebnisse können nun mit Stakeholderanforderungen und -bedarfen abgeglichen werden (siehe Tabelle 7.9). Hierzu werden diese im Systemmodell (teil-)automatisiert eingefügt und mit den entsprechenden Systemelementen verknüpft (siehe Abbildung 6.17). Hierdurch wird die Validierung von V2I erreicht.

7.6 Diskussion & Zwischenfazit

In Kapitel 7.6 werden zu Beginn Limitierungen der Anwendung von MEFS4SoS benannt. Anschließend wird die Forschungsfrage 3 beantwortet. Zum Schluss werden die in Kapitel 4.1 postulierten Effekte und Erfolgsfaktoren geprüft. Zusätzlich werden in Kapitel 5.2.4 vier Anforderungen an Sprache und Methode formuliert, die es zu verifizieren gilt.

7.6.1 Limitierungen

In Kapitel 7 wird die Methode MEFS4SoS für die Validierung von V2I am repräsentativen UseCase GLOSA ausgeführt. Hierbei werden deduktiv von Meta- und Referenz-Systemmodell das Systemmodell und das reale System für V2I vorgestellt.

Im ersten Schritt erfolgte eine Systembeschreibung von V2I. Diese wird modellbasiert durchgeführt und zeichnet sich durch die Schnittstellenbetrachtung

mit SysML-Elementen angelehnt an den C&C²-Ansatz aus. Aufgrund der Größe des SoS V2I, wie auch der konkreten Betrachtung einzelner Funktionen, zeigt die vorliegende Arbeit in Auszügen die Systembeschreibung. Die Auszüge enthalten einzelne Sichten in Form von Diagrammen und Tabellen auf V2I. Die Sichten sind dabei abhängig von der modellierenden Person, da es keine eindeutige Lösung gibt und SysML verschiedene Sichtweisen erlaubt. Beispielsweise werden die Schnittstellen mit Elementen des C&C²-Ansatzes fokussiert, jedoch lassen sich diese auch mit anderen Systemelementen darstellen. Das gesamte Modell enthält über 100 Sichten, sodass eine rollen- und/oder aktivitätenbasierte Auswahl der Sichten für eine anwendungsfreundliche Nutzung erforderlich ist (vgl. Mandel, Behrendt & Albers, 2021).

Im zweiten Schritt werden Gefahren mit zugehörigen Risiken identifiziert, wobei weitere Elemente wie Schaden und Schadensursache tabellarisch modelliert werden. Mit Hilfe der Metriken Risikoprioritätszahl und ASIL erfolgt die Bewertung des Risikos. Wie bereits bei der Systembeschreibung sind auch hier weitere Sichten darstellbar. Weiterführende Betrachtungen beispielsweise durch den Einsatz von Fehlerbäumen sind möglich und werden bereits durch Plugins in der Modellierung unterstützt.

Im dritten Schritt werden funktionale Modifikationen und Spezifikationen basierend auf der Risikopriorisierung durchgeführt. Es werden Elemente der Perspektiven Anforderungen, Funktionen und Logische Elemente basierend auf dem Modell der SGE — Systemgenerationsentwicklung variiert und spezifiziert. Wie bereits in Kapitel 6.1.5 durch den Begriff des Systemtupels eingeführt, lassen sich weitere Elemente betrachten, welche in diesem Schritt einen indirekten Bezug zur funktionalen Betrachtung haben.

Im vierten Schritt erfolgt die Validierung basierend auf der Beschreibung des Validierungssystems, der Herleitung von konkreten Szenarien und der Durchführung der Validierung und Nutzung der Ergebnisse. Hierzu werden zwei verschiedene Validierungsumgebungen betrachtet und zu Beginn das Validierungssystem mit den Validierungsbedarfen beschrieben. Es werden Validierungsziele und weitere Elemente basierend auf dem Meta- und Referenz-Systemmodell modelliert um konkrete Szenarien abzuleiten. Aufgestellte Hypothesen resultieren in unterschiedliche Tests. Die Testergebnisse werden in das Systemmodell (teil-)automatisiert eingefügt. Eine vollumfängliche Validierung ist aufgrund der Systemgröße von V2I und der Vielzahl von UseCases und Tests nicht durchgeführt. Weiterhin ist die Validierung basierend auf Modellen eine unvollständige Abbildung der Realität. Hierzu sind die Faktoren zu benennen und zu quantifizieren, welche die Unterscheidung zwischen Modell und Realität kennzeichnen. Die Unterscheidung lässt sich anhand Abbildung 7.27 in fünf Punkten darstellen.

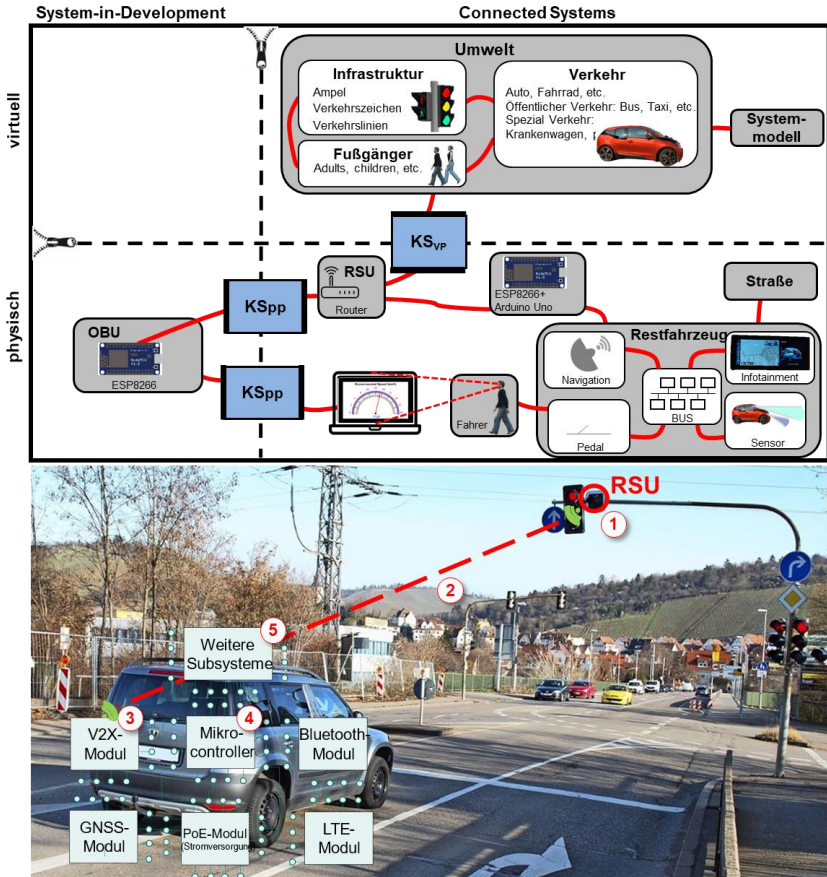


Abbildung 7.27: Einteilung der Untersuchung auf Plausibilisierung zu realen Messungen im Straßenverkehr für GLOSA (vgl. Yi (2019)¹⁴, Hintergrundbild unten: Hauptmann (2019)).

- Teil ①: Die Signalinformationen werden über eine drahtgebundene Verbindung zur straßenseitigen Einheit übertragen. In der Validierungsumgebung wird dieser Teil jedoch über ein UDP-Projekt abgewickelt. Da es sich um ein Kopplungssystem handelt, kann es bei diesem Teil zu einer Systemlatenz kommen, die mit der physischen

Verbindung im realen Verkehr verglichen werden muss. Zudem werden weitere Einflussfaktoren auf den Standard wie Interferenzen und Hindernisse im realen Verkehr in der Validierungsumgebung nicht berücksichtigt. In der Realität ist die Datenübertragungsrate durch die Time-to-Live begrenzt (Zhang et al., 2012). Dies kann zu einer niedrigen Datenübertragungsrate führen, was es den Fahrzeugen erschwert, miteinander zu kommunizieren

- Teil ②: Nach ETSI TS 103 301 basiert die drahtlose Kommunikation auf der Norm IEEE 802.11p. In der aufgebauten Validierungsumgebung basiert hingegen die drahtlose Kommunikation auf der Norm IEEE 802.11g/IEEE 802.11b, die eine geringere Übertragungreichweite als IEEE 802.11p spezifiziert.
- Teil ③: Der Algorithmus zur Berechnung der empfohlenen Geschwindigkeit in der Validierungsumgebung ist exemplarisch und nicht für eine Straßenzulassung geeignet. Es ist ein verbesserter Algorithmus notwendig, welcher beispielsweise die Reaktionszeit des Fahrers, die Beschleunigungszeit des Testfahrzeugs berücksichtigt. Außerdem besteht in der Validierungsumgebung ein Zeitunterschied zwischen der Übertragung der Daten vom Router und dem Empfang der Daten durch den ESP8266 (Teil ②+③). Untersuchungen ergaben einem Grenzwert dieser Latenz von 19ms (L. Eberle, 2019)¹⁵.
- Teil ④: Die Verbindung zwischen V2X-System und Infotainment-System in der realen Umgebung erfolgt über den CAN-Bus. In der Validierungsumgebung wird diese Verbindung jedoch über ein Netzkabel und eine virtuelle GUI-Schnittstelle an einem Monitor hergestellt.
- Teil ⑤: Nachdem der Fahrer die empfohlene Geschwindigkeit im Infotainment-System angezeigt bekommt, fährt das Fahrzeug mit der empfohlenen Geschwindigkeit. Diese kann über die Pedalstellungen des Fahrers oder über die automatisierte Fahrt des Fahrzeugs erfolgen. Bei der Fahrt des Fahrers gibt es keinen Unterschied zwischen der Validierungsumgebung und der realen Umgebung. Ein Unterschied existiert bei der Nutzung eines Fahrroboters, der im Gegensatz zur realen Nutzung des CAN-Bus, die Pedale automatisiert ansteuert und betätigt. Aufgrund der geringen Stellzeit des Fahrroboters von max. 0,55ms (Stähle GmbH, 2015) ist der Einfluss als gering anzusehen.

Die Plausibilisierung von Annahmen, Parametern und Testergebnissen wurden in der vorliegenden Arbeit durch einen Vergleich mit Literaturergebnissen

¹⁵ Abschlussarbeit (unveröffentlicht)

durchgeführt. Die Plausibilisierung erfolgt dabei direkt in den Kapiteln 7.4 und 7.5.

7.6.2 Zwischenfazit zur Beantwortung der Forschungsfrage 3

Der repräsentative UseCase GLOSA wird in Kapitel 7 validiert und ermöglicht somit eine Bewertung der Effekte und Erfolgsfaktoren der Methode MEFS4SoS sowie der sprachlichen Elemente von Meta-, Referenz-Systemmodell, Systemmodell und realem System. Forschungsfrage 3 lautet:

FF3: Wie sollte eine Validierungsumgebung zur durchgängigen und kontinuierlichen Validierung der Funktionalen Sicherheit von V2I realisiert sein?

Zur Beantwortung der Frage wurde auf den Ergebnissen von Forschungsfrage 2 aufgebaut. Die in Meta- und Referenz-Systemmodell beschriebene Sprache deduziert das Systemmodell und die realen Systeme in der Validierungsumgebung von V2I. Unter Anwendung von MEFS4SoS wurde zu Beginn das Systemmodell basierend auf einer Systembeschreibung erstellt (siehe Kapitel 7.1). Die anschließenden Schritte 2 „Gefahrenidentifikation und Risikoevaluation“ (siehe Kapitel 7.2) und 3 „Funktionale Modifikation und Spezifikationen“ (siehe Kapitel 7.3) sind hierbei explizit für die Erfüllung der Funktionalen Sicherheit erforderlich. In diesen wird das Systemmodell konkretisiert und durch spezifische Elemente und Sichten zur Funktionalen Sicherheit ergänzt. Somit sind die in Meta- und Referenz-Systemmodell beschriebenen Sprachelemente im Systemmodell enthalten. Im letzten Schritt 4 „Validierung“ (siehe Kapitel 7.4 und 7.5) wird nun das Validierungssystem mit seiner Validierungsumgebung aufgebaut. Es enthält Elemente der Validierung und ergänzt das Systemmodell. Weiterhin wird das Systemmodell mit realen Systemen bidirektional verknüpft. Somit kann die Testplanung basierend auf Elementen des Systemmodells erfolgen. Das Testen findet mit realen Systemen basierend auf dem IPEK-XiL-Ansatz statt. In der vorliegenden Arbeit werden hierzu mehrere Konfigurationen ausgeführt, die die Nutzbarkeit des Ansatzes nachweisen. Ergebnisse des Tests werden wieder in das Systemmodell übertragen, sodass eine Rückverfolgbarkeit hin zu Anforderungen und Stakeholderbedarfen ermöglicht wird.

Das nachfolgende Kapitel überprüft nun die Effekte Durchgängigkeit und Kontinuität, die in der Validierungsumgebung erzielt werden. Zusätzlich werden weitere in Kapitel 5.2.4 postulierte Effekte und Erfolgsfaktoren untersucht.

7.6.3 Diskussion der Effekte und Erfolgsfaktoren sowie der Anforderungen aus Kapitel 5.2.4

Nachfolgend werden die postulierten Effekte und Einflussfaktoren (siehe Abbildung 4.3) sowie die Anforderungen aus Forschungsfrage 1 (siehe Abbildung 5.15) überprüft.

Die **Durchgängigkeit der Validierung** wird erreicht ...

...durch die Vernetzung der Elemente innerhalb und zwischen den Modellen. Hierbei sind Meta-Systemmodell, Systemmodell und reale Systeme für eine Durchgängigkeit notwendig. Im Meta-Systemmodell werden die Sprache und somit die Elementtypen mit Beziehungen festgelegt. Das Systemmodell enthält die notwendigen Elemente angefangen bei der Beschreibung des Systems, über die Schritte zwei und drei von MEFS4SoS bis hin zur Durchführung der Validierung. Hierbei ist die bidirektionale Vernetzung der Modelle zur Erreichung der Durchgängigkeit wesentlich. Deduktiv und induktiv sind somit Meta-Systemmodell mit Systemmodell innerhalb der Sprache SysML vernetzt. Weiterhin ist das Systemmodell mit realen Systemen verknüpft, wobei die (teil-)automatisierte Verknüpfung im Tool ModelCenter durchgeführt wird. Somit sind auch unterschiedliche Konfigurationen der Validierungsumgebung mit demselben Systemmodell vernetzt und ermöglichen eine durchgängige Validierung.

Die **Kontinuität der Validierung** wird erreicht

... durch den Einsatz von Modellen auf Basis des IPEK-XiL-Ansatzes. Dieser erlaubt eine (Teil-)Systembetrachtung (auch SiD) zu unterschiedlichen Zeitpunkten innerhalb der Produktentwicklung. Somit werden frühzeitig in Modellen wie in MATLAB-Simulink erste Ergebnisse analysiert und synthetisiert. Weiterhin zeigt die Verwendung von Softwarecode, Hardware und Gesamtfahrzeug die Kontinuität in verschiedenen Systemebenen. Zusätzlich werden verschiedene Ausprägungen der Systeme, virtuell und virtuell-physisch, untersucht. Die Validierungsumgebung ist somit kontinuierlich in der Produktentwicklung einsetzbar.

Weitere zu untersuchende Effekte sind **Flexibilität** und **Erweiterbarkeit**, welche in der nachfolgenden Anforderung 2 evaluiert werden.

Aus Flexibilität und Erweiterbarkeit folgt die **Effizienz** (siehe Abbildung 4.3). In der vorliegenden Arbeit ist die Effizienz bei der Ausführung weiterer Tests erzielt worden. Aufgrund des modularen Konzepts der Validierungsumgebung werden

Erweiterungen mit geringem Ressourcenaufwand¹⁶ durchgeführt und validiert. Ein positiver Effekt auf Effizienz ist daher insbesondere bei Änderungen realisiert. Dies verdeutlichen virtuell und gemischt virtuell-physische Validierungskonfigurationen, die in wenigen Tagen aufgebaut werden und Messungen ermöglichen. Im Gegensatz dazu sind Straßentests (siehe Kapitel 2.3.2) deutlich aufwändiger (vgl. Eberle & Jütten, 2017; vgl. Galbras, 2018).

Schlussendlich ist der Erfolgsfaktor **Praxisnutzen** von MEFS4SoS und der sprachlichen Unterstützung in Form von Modellen zu prüfen. Der postulierte Praxisnutzen konnte anhand einer quantitativen Befragung in der Lehrveranstaltung IP22/23 des IPEK durchgeführt werden. Bei einer Skala von 1 (sehr schlecht) bis 5 (sehr gut) wurden folgende Bewertungen für die Unterstützung von MEFS4SoS für die Modellbildung und Validierung abgegeben (N=14, m=3,6, $\sigma=0,7$, Kontrollgruppe: N=14, m=2,9, $\sigma=1,1$). Dies deutet auf das Potenzial der Methode hin. Eine qualitative Befragung von Experten der Funktionalen Sicherheit erfolgte im Rahmen des Meeting Events der Functional Safety Community (Thaler & Wäschle, 2023). Die Ergebnisse sind insbesondere im Sinne des Breitentransfers der Methode anzusehen und zeigen das Potenzial der Methode auf (N=5, m=3,0). Der erfolgreiche Einsatz von MEFS4SoS erfolgte in Kapitel 7 anhand des V2I-UseCases GLOSA, welcher Standards und Anforderungen aus der Praxis enthält. Die Methode wurde somit in virtuellen und gemischt virtuell-physischen Konfigurationen am realen Fahrzeug durchgeführt. Eine quantitative Untersuchung einer Anwendung bei einem Unternehmen gilt als wesentlicher nächster Schritt. Aufgrund der Anlehnung der Methoden an bestehende und notwendige Standards der Funktionalen Sicherheit ist eine Operationalisierung und damit Berücksichtigung in Unternehmen erleichtert.

Anforderung 1: Schnittstellenbetrachtung

Bereits in der Systemmodellierung werden die Schnittstellen im Sinne des C&C²-Ansatzes beschrieben. Die Schnittstellenbetrachtung ist in allen vier Schritten von MEFS4SoS enthalten. Es zeigt sich, dass durch MBSE die Vernetzung der Elemente und somit die Schnittstellen modelliert, dokumentiert und analysiert werden können.

¹⁶ Basierend auf der Dauer von Hardware- und Softwareanpassungen für den Aufbau der Validierungsumgebung (vgl. unveröffentlichte Abschlussarbeiten von: Yi, 2019; Qi, 2019; Shi, 2020; Klein, 2022).

Insbesondere im Schritt vier von MEFS4SoS zur Durchführung der Validierung sind Schnittstellen wesentlich. In der virtuell ausgeprägten Validierungsumgebung lassen sich Schnittstellen durch vordefinierte modulare Strukturen wie in Bibliotheken frühzeitig berücksichtigen. In der virtuell-physisch ausgeprägten Validierungsumgebung werden Schnittstellen über Koppelsysteme berücksichtigt. Beispielsweise ist das Echtzeitsystem mit WiFi, CAN oder Ethernet mit weiteren (Teil-)systemen verbunden. Koppelsysteme ermöglichen einen flexiblen Aufbau. Aus diesem Grund sollten zentrale Systeme möglichst viele Schnittstellen aufweisen und Koppelsysteme gezielt mitentwickelt werden, um flexibel auf Änderungen der Bedarfe zu reagieren. Dies kann über Schnittstellenverträge (engl. contracts) modellbasiert erfolgen. Der Einfluss von nicht idealen Koppelsystemen und weiteren Schnittstellen muss in der Validierungsumgebung berücksichtigt werden. Bei einer geeigneten Berücksichtigung von Schnittstellen kann die Validierung teil-automatisiert mit Verknüpfung zum Systemmodell und möglichen weiteren Modellen (z.B. Anforderungsmanagement) erfolgen.

Anforderung 2: Flexible, erweiterbare Validierungsumgebung

Die Flexibilität folgt aus der Vielfalt an Ausprägungen der Validierungsumgebung, Validierungszielen und weiteren Elementen des Validierungssystems. Hierbei wurde ein Fokus auf die Modularisierbarkeit, (siehe Abbildung 7.12), Skalierbarkeit mit Hilfe von X-in-the-Loop (siehe Vorgehensweise in 2.3.2) und Virtualisierung (siehe Vorgehensweise in 2.3.2) gelegt. Neben den Anwendungen des repräsentativen UseCase GLOSA wurde die Erweiterung um andere UseCases und die weitere V2X Konfigurationen V2V und V2P untersucht. Für eine detaillierte Betrachtung der nachfolgenden UseCases, sei auf (Wäschle, Behrendt et al., 2021; Wäschle, Wang et al., 2022) und Arbeiten in Zusammenarbeit mit Yi (2019), Chen (2022) und Qi (2019)¹⁷ verwiesen. Eine Übersicht der untersuchten UseCases zeigt Tabelle D.1 in Anhang D auf.

- „V2I – Road Work Warning – virtuell“,
- „V2V – Verkehrskreuzung mit Fahrzeugen – virtuell“,
- „V2V & V2P – Verkehrskreuzung mit Fußgänger – virtuell-physisch“
- „V2I & V2V – Netzwerk VANET – virtuell“

¹⁷ Abschlussarbeit (unveröffentlicht)

Anforderung 3: Systemanalyse mit Stakeholderanalyse

Bereits im ersten Schritt von MEFS4SoS wird das System modelliert. Hierbei wurden die Elementtypen Stakeholder und Bedarf eingeführt und mit weiteren Elementtypen vernetzt. Es zeigt sich, dass eine Vielzahl von Standards sowie Stakeholder wie Lieferanten zu einer Vielzahl von Vernetzungen führt. Die Analyse kann daher durch die Abbildung in Modellen erleichtert werden und eine Bewertung der Auswirkung und des Einflusses eines Stakeholders ermöglichen. Hierzu wurden im Forschungsprojekt SmartLoad Verknüpfungen untersucht (vgl. Albers, Klingler & Ebel, 2013; vgl. Wäschle, Wolter et al., 2021). Als ein Treiber gelten neue Technologien, die durch mehrere Systeme mit unterschiedlichen Lebenszyklen Einzug in SoS halten und zu „neuen Kooperationen und veränderten Rollenverteilungen [führen]“ (Gasser et al., 2015, S. 19). Die vorliegende Arbeit fokussiert auf den Stakeholder, spezifische Rollen des Stakeholders werden in Mandel et al. (2021) untersucht.

Anforderung 4: Berücksichtigung der Funktionalen Sicherheit

MEFS4SoS ist aus Standards der Funktionalen Sicherheit abgeleitet und berücksichtigt somit wesentliche Elemente der Funktionalen Sicherheit wie die Gefahr. Demzufolge wird die Vorgehensweise der funktionalen Sicherheitsabsicherung über die Gefahr und deren Verknüpfungen genutzt (siehe Vorgehensweise in 2.3.2). Die Überprüfung von Kriterien der Funktionalen Sicherheit ist über die (teil-)automatisierte Rückführung ins Systemmodell realisiert.

8 Zusammenfassung, Gesamtfazit und Ausblick

Kapitel 8.1 fasst das Forschungsziel mit den zugehörigen Forschungsannahmen, -these und -fragen zusammen. Kapitel 8.2 stellt die wesentlichen Erkenntnisse der Arbeit in einem Gesamtfazit zusammen. Kapitel 8.3 gibt einen Ausblick auf weiterführende Untersuchungen im Hinblick auf die drei Forschungsfragen.

8.1 Zusammenfassung

Die vorliegende Arbeit untersucht das Forschungsziel:

Forschungsziel

Durchgängige, kontinuierliche Sprache und Methode zur Validierung der Funktionalen Sicherheit von Vehicle-to-Infrastructure als SoS durch MBSE-Unterstützung unter Einsatz des IPEK-XiL-Ansatzes.

Zur Erreichung des Forschungsziels wird eine Vorgehensweise (Kapitel 5.2), Sprache (Kapitel 6.1 und 6.2) und Methode (Kapitel 6.3) entwickelt sowie deren Anwendung (Kapitel 7) geprüft. Mit Hilfe der Vorgehensweise wird das Zielsystem für V2I als SoS beschrieben. Eine systematische Beschreibung der Sprache durch Abstraktionsgrade von Modellen und die Anwendung der Modelle für die Validierung von V2I zeigt die Unterstützung von MBSE auf. Die Validierung erfolgt auf Basis einer Sprache und Methode für V2I für die Funktionale Sicherheit. Hierbei wird der IPEK-XiL-Ansatz in der Validierung erfolgreich eingesetzt und die Effekte der Durchgängigkeit und Kontinuität werden bestätigt.

Zu Beginn wurden zwei Grundannahmen und eine Grundthese aufgestellt:

Grundannahmen und Grundthese

GA1: SoS lassen sich unter Einsatz von MBSE modellieren.

GA2: SoS lassen sich unter Einsatz des IPEK-XiL-Ansatzes validieren.

GT: Mittels eines Beschreibungsmodells lassen sich SoS-Charakteristiken identifizieren, die die Validierung von SoS-Anwendungsfällen unterstützen.

Die erste Grundannahme wird bestätigt, da sich V2I mit MBSE modellieren lässt. Die systematische Beschreibung von Modellen wird erzielt durch die Abstraktionsgrade Meta-, Referenz-Systemmodell, Systemmodell und reales System sowie der generischen, bereichsspezifischen und systemspezifischen Betrachtung (siehe Abbildung 6.1 und die Beschreibung und Anwendung in den Kapitel 6, 7).

Mit dem IPEK-XiL-Ansatz wird das Validierungssystem für V2I in Kapitel 7 beschrieben und somit die zweite Grundannahme bestätigt. Der Ansatz unterstützt dabei den Systemgedanken aufgrund der Aufstellung von Systemgrenzen, der Umweltbetrachtung und des zugrunde liegenden hierarchischen Konzeptes. MEFS4SoS nutzt somit in dem Schritt der Validierung den IPEK-XiL-Ansatz zur Beschreibung des Validierungssystems mit seinen virtuell und virtuell-physischen Konfigurationen.

Die Grundthese hat sich als umsetzbar erwiesen. Die in Kapitel 5 vorgestellte Beschreibung des Zielsystems verdeutlicht die Herleitung von Charakteristiken aus der Literatur. Zudem werden diese bewertet und Implikationen für die Validierung abgeleitet. Im Rahmen der Modellbildung werden V2I-Anwendungsfälle genutzt (siehe Kapitel 6, 7).

Weiterhin werden die drei aufgestellten Forschungsfragen beantwortet:

Forschungsfragen

- FF1: Wie lässt sich das Zielsystem von V2I als SoS verständlich beschreiben sowie zu verteilten Systemen abgrenzen, um spezifische Herausforderungen in der Validierung von V2I zu identifizieren?
- FF2: Wie kann die durchgängige und kontinuierliche Validierung von V2I durch Modellierungssprache und Methode im Sinne von MBSE unterstützt werden?
- FF3: Wie sollte eine Validierungsumgebung zur durchgängigen und kontinuierlichen Validierung der Funktionalen Sicherheit von V2I realisiert sein?

Forschungsfrage 1 wird in Kapitel 5 beantwortet. Das Literaturreview und die empirische Studie zeigen die Bedarfe nach der Zielsystembeschreibung auf. Die Vorgehensweise enthält die Profilfindung, die Charakterisierung und die Ableitung von Implikationen in Form von Herausforderungen. In der Profilfindung wird das Produktprofil eingesetzt, um Zweck, Nutzen und weitere Zielsystemelemente zu beschreiben. Innerhalb der Profilfindung werden weitere Begriffe mit Hilfe eines Venn-Diagramms eingeordnet und abgegrenzt. Die Charakterisierung erlaubt die Beschreibung von Eigenschaften eines SoS und ermöglicht den Vergleich

verschiedener SoS. Es zeigt sich, dass SoS unterschiedliche Ausprägungen der Charakteristiken haben, beispielsweise beziehen sich verteilte Systeme auf die Charakteristik der Verteilung (meist Standortverteilung). Die Ableitung von Implikationen wird in einer Tabelle zur Herleitung von Herausforderungen für das zu untersuchende SoS realisiert. Hieraus lassen sich konkrete Anforderungen an die Validierung von SoS ableiten.

Forschungsfrage 2 wird in Kapitel 6 unter Berücksichtigung der Sprache in den Abstraktionsgraden der Modelle Meta-, Referenz-Systemmodell, Systemmodell und des realen Systems beantwortet. Zudem wird die Methode MEFS4SoS eingeführt, welche im Kontext der Funktionalen Sicherheit Aktivitäten zur Validierung des SoS ermöglicht. Die Kontinuität und Durchgängigkeit kann durch die systematische Modellierung und Modellkopplung sowie die verschiedenen Levels innerhalb eines Modells realisiert werden. Die konsistente Sprache, welche in Modellen und in MEFS4SoS zum Einsatz kommt, ermöglicht eine Berücksichtigung verschiedener Layer für V2I.

Forschungsfrage 3 wird in Kapitel 7 mit der Anwendung der Validierung für den spezifischen UseCase GLOSA beantwortet. Die entwickelte Methode MEFS4SoS wird durchgeführt. Im ersten Schritt der Methode wird das System mit seinen Schnittstellen modelliert, um im zweiten Schritt Gefahren zu identifizieren und das Risiko zu priorisieren. Daraufhin lässt sich das Systemmodell modifizieren, um das Risiko geeignet zu adressieren. In der anschließenden Validierung wird das Validierungssystem im Systemmodell beschrieben, um virtuelle und virtuell-physisch gemischte Validierungskonfigurationen einzuführen. Die erzielten Testergebnisse lassen sich wiederum in das Systemmodell integrieren. Somit werden Hypothesen evaluiert, Anforderungen verifiziert und das SoS gegenüber den Bedarfen validiert. Eine Anwendung der Methode erfolgt mit Hilfe von Modellen für den UseCase GLOSA, wobei weitere UseCases und SoS exemplarisch untersucht werden.

8.2 Gesamtfazit

Die drei wesentlichen Erkenntnisse der vorliegenden Arbeit, die sich aus den drei Forschungsfragen angelehnt an die drei Stadien der DRM ergeben: DS1, PS und Initiale DS2 (siehe Abbildung 8.1), sind in Kapitel 8.2.1 dargestellt. Die Forschung ist dabei ein Teil der KaSPro, sodass in Kapitel 8.2.2 die drei Erkenntnisse als Beitrag in der KaSPro beschrieben werden.

8.2.1 Erkenntnisse

Der systemtheoretische Begriff des SoS und dessen Implikationen sind von Produktentwickelnden schwer zu verstehen. Diese Erkenntnis wurde in einer empirischen Studie identifiziert, wobei ein erstelltes Workshopkonzept das Begriffsverständnis nachweislich erhöht hat. Hierzu dient eine Vorgehensweise zur Beschreibung des Zielsystems, welche zu Anforderungen an die Validierung führt.

Mithilfe dreier Abstraktionsgrade (generisch, bereichs-, produktspezifisch) lassen sich Modelle mittels Deduktion und Induktion verknüpfen und zeitlich in die Produktentwicklung einordnen. Für V2I sind bereichsspezifische Sprachelemente notwendig, welche in der vorliegenden Arbeit für die Funktionale Sicherheit untersucht werden. MBSE ist geeignet, die Beziehungen der Sprachelemente formalisiert durchgängig und kontinuierlich darzustellen. MEFS4SoS nutzt diese Verlinkungen und untersucht die Funktionale Sicherheit für V2I.

Die Anwendung von MEFS4SoS für den UseCase GLOSA zeigt die Notwendigkeit einer flexiblen, erweiterbaren Validierungsumgebung, welche Modelle, Software, Hardware und das Gesamtfahrzeug enthält. MEFS4SoS mit der genutzten Sprache ist hierbei anwendbar und nützlich.

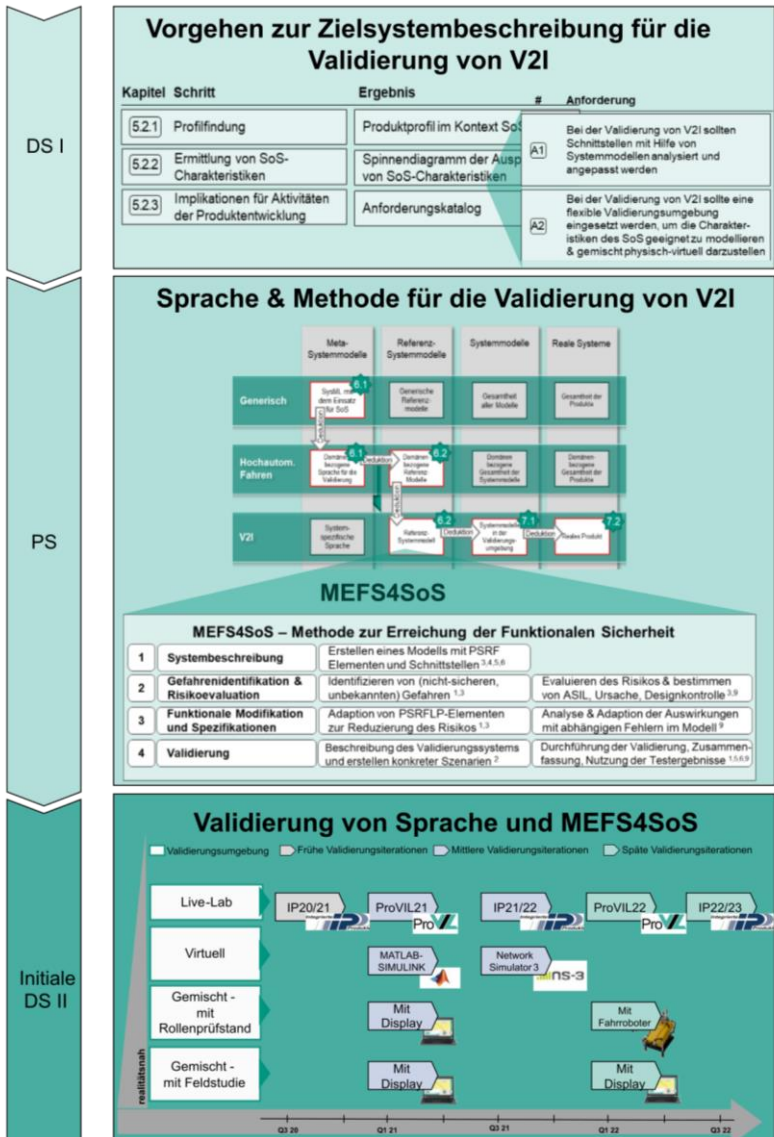


Abbildung 8.1: Übersicht der wesentlichen Ergebnisse der vorliegenden Arbeit eingeteilt nach den drei Stadien DS1, PS und Initiale DS2.

8.2.2 Beitrag in der KaSPro – Karlsruher Schule für Produktentwicklung

Die vorliegende Arbeit ist im Rahmen einer Forschungstätigkeit am Institut für Produktentwicklung entstanden und Teil der Karlsruher Schule für Produktentwicklung (KaSPro). Diese basiert auf den drei Aussagen:

- „Jeder Produktentstehungsprozess ist einzigartig und individuell“ (Albers, 2010, S. 347; Albers, Braun & Muschik, 2010b; Albers et al., 2011, S. 256)
- „Der Mensch steht im Mittelpunkt der Produktentwicklung“ (Albers et al., 2011, S. 260; A. M. Maier et al., 2007)
- „Produkte werden in Generationen entwickelt“ (Albers, Bursac, Urbanec, Lüdcke & Rachenkova, 2014; Albers, Bursac & Wintergerst, 2015; Feldhusen & Grote, 2013)

Bezogen auf die erste Aussage muss die Beschreibung des Zielsystems und der Sprache auf jedes Produkt angepasst werden. Dabei wird der Systembegriff zur Beschreibung von Produkten verwendet. Es wird das SoS betrachtet, welches spezifische Charakteristiken aufweist. Demzufolge wird aufgezeigt, wie Sprache generisch, bereichsspezifisch und systemspezifisch auftritt und mittels Induktion und Deduktion Modelle in unterschiedlichen Abstraktionsgraden gebildet werden. Eine Methode zur Erfüllung der Funktionalen Sicherheit wurde entwickelt, die die Produktentwicklung unterstützt.

Befragungen und die Betrachtung von Stakeholdern und deren Bedarfe verdeutlichen die Notwendigkeit das SoS als soziotechnisches System wahrzunehmen und entsprechend zu beschreiben. Hierbei ist der Mensch wesentlich, sodass durch die Befragung von Entwicklern, die Berücksichtigung der Stakeholder mit ihren Bedarfen und die Untersuchung mit Fahrern erfolgt ist. Dies zeigt Kapitel 5 mit der Beschreibung des Zielsystems sowie Wirkketten innerhalb von aufgestellten Modellen mit dem Ziel Testergebnisse mit Bedarfen abzugleichen.

Das SoS ist veränderlich mit weicher Systemgrenze. Nach Albers und Peglow et al. (2018) ist die Entwicklung eines SoS nie abgeschlossen. Es muss somit die Entwicklung in Generationen betrachtet und berücksichtigt werden. Demzufolge sind Elemente des Referenzsystems in Sprache und Methode enthalten.

Sprache und Methode werden in das IPEK-MBSE-Framework (Kubin et al., 2023) integriert, welches weitere MBSE-Layer aufweist. So wird eine menschenzentrierte, ganzheitliche und generationsübergreifende Systemmodellierung ermöglicht. Die vorliegende Arbeit liefert hierzu mit Sprache und Methode für die Validierung des SoS V2I und die Funktionale Sicherheit einen Beitrag, wobei weitere Forschungsarbeiten das IPEK-MBSE-Framework bedarfsorientiert erweitern.

Daher liefert die vorliegende Arbeit insbesondere einen Beitrag zu dem in der KaSPro enthaltenen Forschungspfad der „Validierung als zentrale Aktivität zur Wissensgenerierung im Produktentstehungsprozess“ (Paraphrasiert nach Albers, 2010, S. 348).

8.3 Ausblick

In diesem Kapitel werden ausgehend von den drei Forschungsfragen der vorliegenden Arbeit zu: „Zielsystembeschreibung von V2I“, „Sprache und Methode für V2I“ und „Validierung von V2I“ weiter zu erforschende Themen und Aktivitäten benannt.

8.3.1 Zielsystembeschreibung von V2I

Das Zielsystem kann aufgrund der Komplexität, Größe sowie weiterer Charakteristiken von SoS weiter erforscht werden. Insbesondere die weichen Systemgrenzen führen zu einer dynamischen Systementwicklung, die durch gesellschaftliche und technologische Treiber beeinflusst wird.

Beschreibung des Zielsystems für soziotechnische SoS

Die vorliegende Arbeit beschreibt das Zielsystem für die Validierung von V2I. Darauf aufbauend können zukünftig Erweiterungen und eine Generalisierung auf SoS erfolgen. Hierbei ist eine Einschränkung auf Arten von SoS wie soziotechnische SoS notwendig, um die Vielzahl an unterschiedlichen Anwendungsgebieten zu berücksichtigen, welche im Literaturreview in Kapitel 5.1.1 identifiziert wurden. Weiterhin wird innerhalb der Produktentwicklung das initiale Zielsystem durch Analyse- und Synthese-Aktivitäten weiterentwickelt (Ebel, 2015; Richter, Witt et al., 2019b). Nachfolgende Untersuchungen des Zielsystems für SoS mit weiteren Zielsystemelementen wären daher notwendig, um eine vollständige Beschreibung zu realisieren (vgl. Kapitel 5.3.1).

Perspektiven und Kontext des SoS

Ein SoS kann nicht unabhängig von seiner Umwelt betrachtet werden. Beispielsweise ist das V2I Teil der Mobilität mit zahlreichen Systemen. Der Einfluss solcher Systeme wurde im Rahmen der Erweiterung der Validierungsumgebung auf V2P und V2V exemplarisch untersucht. Eine weiterführende Betrachtung des Einflusses der Systeme und zwischen SoS gilt es zu erforschen. Zudem sind Einflussfaktoren wie der gesellschaftliche Wandel (z.B. in Gesellschaftsstrukturen) und technologische (disruptive) Innovationen (z.B. von 4G auf 5G) zu beachten, die

Umwelt sowie SoS beeinflussen und als Treiber für Änderungen innerhalb und außerhalb des SoS gelten. Änderungen im SoS sind als Herausforderungen zu verstehen, die geeignet zu beherrschen sind. Die Identifikation, Bewertung und der Umgang mit Änderungen werden als weiterer Beitrag zum IPEK-MBSE-Framework im Rahmen des Projektes MoSyS untersucht (vgl. Martin, 2022).

8.3.2 Sprache und Methode für V2I

In Bezug auf Sprachen sind generische Rahmenwerke, bereichs- und systemspezifische Standards zu erarbeiten. Für Methoden von V2I können diese in ein übergreifendes IPEK-MBSE-Framework integriert werden und mit Hilfe von Aktivitäten gezielt eingesetzt werden.

Weiterführende Untersuchung von Modellen für die Menschorientierte Gestaltung komplexer System of Systems

Die Sprache lässt sich über verschiedenartige Modelle einsetzen. Modelle können unterschiedlichen Abstraktionsgraden zugeordnet werden. Zukünftig können zu jedem Abstraktionsgrad weitere Forschungsaktivitäten erfolgen. Im generischen Meta-Systemmodell wird zukünftig SysML 2.0 genutzt. Diese Sprache erweitert die in der vorliegenden Arbeit genutzte Sprache SysML 1.6 um Elemente wie eine benutzerfreundliche Programmierung von Diagrammen und Wertetypen (Object Management Group, 2023).

Speziell für generische Modelle in Bezug zu SoS gilt es an Standards zu arbeiten, um eine durchgängige Beschreibung über Unternehmensgrenzen hinweg zu realisieren. Hierzu werden im Rahmen des Forschungsprojektes MoSyS gezielt Lösungen für eine unternehmensweite Sprache und eine unternehmensspezifische Instanziierung erarbeitet (siehe Mandel, Guenther et al., 2022). Die in der vorliegenden Arbeit entwickelte Sprache und die Modelle werden im Forschungsprojekt MoSyS diskutiert und eingesetzt. Zum Beispiel ist das Sprachelement Contract für die vernetzten Systeme innerhalb eines SoS in SoS-Mustern relevant. Auch in Bezug zu bereichsspezifischen Modellen sind Standards anzustreben. Dies zeigen Ergebnisse der ASAM zu den OpenX Standards, welche eine einheitliche Beschreibung von ausgewählten Elementen wie Szenarien und Fahrhahnoberflächen ermöglichen (vgl. ASAM e. V., 2022a). Dabei sind geeignete Austauschformate zu nutzen, um einen Austausch über bereichsspezifische Tools zu ermöglichen. Erste Ansätze verwenden FMI, OpenSimulationInterface und XML. Für den Breitentransfer wurde die Methode MEFS4SoS dem Fachpublikum „Functional Safety Community“ vorgestellt und diskutiert. Hierbei wurden insbesondere der Link zu Standards begrüßt, wobei eine Erweiterung um Themen

der Security und der Künstlichen Intelligenz benannt wurden. (Thaler & Wäschle, 2023)

Untersuchung von Prozessen für SoS

Ergänzend zu der sprachlichen und methodischen Betrachtung sollten zukünftige Aktivitäten in Prozessen untersucht werden. Diese ermöglichen einen bedarfsorientierten, aktivitätenbezogenen Methodeneinsatz und müssen neben der Produktentwicklung den gesamten Lebenszyklus eines SoS einbeziehen. Eine Validierung von SoS ist während der Entwicklung und Anwendung vorzunehmen. Erste Ansätze zur Berücksichtigung der Agilität in MBSE finden sich in Heimicke et al. (2021) wieder, wobei einzelne Aspekte und Charakteristiken von SoS wie der hohe Grad an Veränderungen über Updates auf Basis des Modells der SGE untersucht werden sollten.

8.3.3 Validierung von V2I

Zukünftig sollten Erweiterungen der durchgeführten Untersuchungen von UseCases und Tests sowie des aufgebauten Validierungssystems betrachtet werden. Dabei könnten die wesentlichen Themenbereiche Security, 5G und KI betrachtet werden.

Erweiterung der V2I-Validierungsumgebung bis zur Integration in das Industrial Metaverse

Die vorliegende Arbeit fokussiert auf eine indirekte Systemkommunikation beispielsweise mit Hilfe eines Fahrroboters, der Geschwindigkeitsregelung am Rollenprüfstand oder durch Informationen an den Fahrer. Untersuchungen von einzelnen Manövern am Fahrzeugrollenprüfstand zeigen die Möglichkeit Tests unter Laborbedingungen durchzuführen. Daher ist weiterhin zu untersuchen, inwieweit Lösungen mit direkter Systemkommunikation mit dem Fahrzeug über CAN-Daten (sog. „E-Gas“) eingesetzt werden können. In Kombination mit der virtuellen Umgebung lässt sich die Validierungskonfiguration, je nach Anforderung an das Testergebnis (z.B. Testgütekriterien (siehe Tool Confidence Level in ISO 26262-8:2018(E), 25,26), geeignet auswählen. Erweiterungen des Validierungssystems sind zur Betrachtung von unterschiedlichen Übertragungsprotokollen von C-V2X und 802.11p zu wählen. Zahlreiche weitere Tests können bedarfsgerecht untersucht werden. Hierbei können aus Modellen die Stakeholderbedarfe z.B. an Gesetze, Normen oder unternehmensspezifische Vorgaben sowie gesellschaftliche Treiber untersucht werden.

Untersuchungen zu V2I sollten zukünftig ausgehend von repräsentativen UseCases wie GLOSA auf weitere UseCases ausgeweitet werden. Hierbei ist kontinuierlich während des Produktlebenszyklus eine Überprüfung der UseCases erforderlich, um

aufgrund des evolutionären Charakters eines SoS neue und veränderte UseCases zu identifizieren. Die Nutzung der Validierungsumgebung kann hierbei zukünftig als Business Case verstanden werden, bei dem komplexe Validierungsumgebungen für SoS bedarfsgerecht genutzt werden können. Themen wie die Digitalisierung ermöglichen somit eine serviceorientierte Nutzung der Validierungsumgebung z.B. im Industrial Metaverse.

Integrierte Betrachtung von Safety und Security

In der vorliegenden Arbeit wurde das Thema Safety mit zwei der drei Sicherheitsdimensionen (Funktionale Sicherheit und SOTIF) betrachtet (Schneider & Hosse, 2019). Das Thema Security (ISO-SAE AWI 21434) wurde aufgrund des Fokus der vorliegenden Arbeit und notwendigen umfangreichen Forschungsaktivitäten auf dem Gebiet nicht untersucht. Demzufolge sind Ergebnisse unter dem Vorbehalt zu betrachten, dass Security weitere Einschränkungen, beispielsweise in Form einer längeren Signalverarbeitungszeit, mit sich bringen. Ansätze wie „Private Key Identifier“ (Siehe Villarreal-Vasquez et al., 2017) oder Künstliche Intelligenz (siehe Prinkle, Liu, Wang & Zhang, 2017) zur Sicherstellung der Security sind zukünftig einzubeziehen. Ein gemeinsames Design von Safety und Security wird notwendig, was nur durch eine Integration und intensive Zusammenarbeit der Disziplinen gelingen kann.

KI als Teil von SoS und als Unterstützung in der Entwicklung von SoS

Ein wesentlicher technologischer Treiber im Bereich des hochautomatisierten Fahrens ist das Themenfeld Künstliche Intelligenz. Ausgehend eines systematischen Literaturreviews in Wäschle, Thaler, Berres, Pözlbauer und Albers (2022) sind drei Aspekte zu beachten:

1. Potenziale beim Einsatz von KI im System (Humpert, Wäschle, Horstmeyer, Dumitrescu & Albers, 2023; Kharatyan et al., 2022)
2. Validieren von SoS, welche KI enthalten (vgl. Wäschle, Thaler et al., 2022),
3. Einsatz von KI zur Validierung von SoS (vgl. Raj, Jha, Ramanathan & Pullum, 2017).

Erste Ansätze zu Aktivitäten von SoS mit KI werden in Wäschle und Thaler et al. (2022) und Kharatyan et al. (2022) sowie voraussichtlich zukünftig in der ISO/IEC 5469 zusammengefasst. Zudem finden sich in der Literatur Themen wie die sichere Nachrichtenübertragung mittels KI (Prinkle et al., 2017) und die Modellierung in MBSE zum Umgang bei der Integration von KI und Digitalem Zwilling (Wilking, Sauer, Schleich & Wartzack, 2022). Fedullo, Morato, Tramarin, Cattini und Rovati (2022) zeigen Potenziale von KI für die Messtechnik auf. Demzufolge ist die

Validierung mittels Methoden, welche KI einsetzen, als zukunftssträchtiges Forschungsfeld anzusehen.

Die vorliegende Arbeit zeigt die Unterstützung der Validierung von V2I als SoS. Es wird deutlich, dass sich daraus weitere Forschungsaktivitäten ergeben.

Literaturverzeichnis

- 5GAA (Eds.). (2019). *C-V2X Use Cases, Methodology, Examples and Service Level Requirements. Whitepaper* (1.0). Working Group 1. Zugriff am 26.11.23 unter: https://5gaa.org/wp-content/uploads/2019/07/5GAA_191906_WP_CV2X_UCs_v1-3-1.pdf
- Abboud, K., Omar, H. A. & Zhuang, W. (2016). Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey. *IEEE Transactions on Vehicular Technology*, 65(12), 9457–9470.
<https://doi.org/10.1109/tvt.2016.2591558>
- AdaptiVe Consortium (Eds.). (2017, Juni). *Deliverable D1.0 Final project results* (1.0). Zugriff am 26.11.23 unter: https://www.adaptive-ip.eu/files/adaptive/content/downloads/AdaptiVe-SP1-v1-0-DL-D1-0-Final_Report.pdf
- Adler, P. S., Goldoftas, B. & Levine, D. I. (1999). Flexibility Versus Efficiency? A Case Study of Model Changeovers in the Toyota Production System. *Organization Science*, 10(1), 43–68. <https://doi.org/10.1287/orsc.10.1.43>
- Albers, A. (2010). Five Hypotheses about Engineering Processes and their Consequences. In I. Horvath, F. Mandorli & Z. Rusak (Eds.), *Proceedings of TMCE 2010 Symposium* (Bd. 1, S. 343–356). Ancona, Italien.
- Albers, A. . (2023). Engineering neu denken und gestalten. Herausforderungen, Anwendungsszenarien und das neue Leitbild Advanced Systems Engineering. *acatech Impuls*. https://doi.org/10.48669/aca_2023-7
- Albers, A., Behrendt, M., Klingler, S. & Matros, K. (2016). Verifikation und Validierung im Produktentstehungsprozess. In U. Lindemann (Hrsg.), *Handbuch Produktentwicklung* (S. 541–569). München: Carl Hanser Verlag.
<https://doi.org/10.3139/9783446445819.019>
- Albers, A., Braun, A. & Muschik, S. (2010a). Ein Beitrag zum Verständnis des Aktivitätsbegriffs im System der Produktentstehung. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems-Engineering. München, Freising, 10. - 12. November 2010* (S. 87–96). München: Hanser.

- Albers, A., Braun, A. & Muschik, S. (2010b). Uniqueness and the Multiple Fractal Character of Product Engineering Processes. In P. Heisig, P. J. Clarkson & S. Vajna (Eds.), *1st International Conference on Modelling and Management Engineering Processes* (S. 15–26). London: Springer London.
https://doi.org/10.1007/978-1-84996-199-8_2
- Albers, A., Bursac, N., Urbanec, J., Lüdcke, R. & Rachenkova, G. (Hrsg.). (2014). *Knowledge Management in Product Generation Development – an empirical study*. In Krause, Dieter; Paetzold, Kristin; Wartzack, Sandro (Hrsg.), *Beiträge zum 25. DfX-Symposium*. Bamberg
- Albers, A., Bursac, N. & Wintergerst, E. (2015). Produktgenerationsentwicklung - Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In H. Binz, B. Bertsche, W. Bauer & D. Roth (Hrsg.), *Stuttgarter Symposium für Produktentwicklung. Entwicklung smarter Produkte für die Zukunft* (Bd. 5, S. 1–10). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.
- Albers, A., Dumitrescu, R., Gausemeier, J., Lindow, K., Riedel, O. & Stark, R. (2022). Strategie Advanced Systems Engineering. Leitinitiative zur Zukunft des Engineering und Innovationsstandorts Deutschland. *acatech KOOPERATION*.
- Albers, A., Dumitrescu, R., Gausemeier, J., Riedel, O. & Stark, R. (2018). Advanced Systems Engineering – Eine Leitlinie zur Stärkung der Innovationskraft. *acatech KOOPERATION*.
- Albers, A., Dumitrescu, R., Marthaler, F., Albers Alexander A., Strauch, M., Kuehfuss, D. et al. (2018). PGE-Produktgenerationsentwicklung und Zukunftsvorausschau: Eine systematische Betrachtung zur Ermittlung der Zusammenhänge. Symposium für Vorausschau und Technologieplanung. In J. Gausemeier, W. Bauer & R. Dumitrescu (Hrsg.), *Vorausschau und Technologieplanung. 14. Symposium für Vorausschau und Technologieplanung* (n.p.). Paderborn: Universität Paderborn Heinz Nixdorf Institut.
- Albers, A. & Düser, T. (2008, 23. September). *Integrierte Validierungsumgebung für energieeffiziente Fahrerassistenzsysteme am Rollenprüfstand*. IPG Technology Conference, Ettlingen. <https://doi.org/10.5445/IR/1000011736>

- Albers, A., Düser, T., Sander, O., Roth, C. & Henning, J. (2010). X-in-the-Loop-Framework für Fahrzeuge, Steuergeräte und Kommunikationssysteme. *ATZelektronik*, 5(5), 60–65. <https://doi.org/10.1007/BF03224034>
- Albers, A., Ebertz, J., Rapp, S., Heimicke, J., Kürten, C., Zimmermann, V. et al. (2020). Produktgeneration 1 im Modell der PGE – Produktgenerationsentwicklung. Verständnis, Zusammenhänge und Auswirkungen in der Produktentwicklung. *KIT Scientific Working Papers*, 149. ISSN 2194-1629. <https://doi.org/10.5445/IR/1000127971>
- Albers, A., Fischer, J., Behrendt, M. & Lieske, D. (2014). Identification and definition of acoustic relevant limit values for electric vehicles. 14th Stuttgarter Symposium – Automotive and Engine Technology. In M. Bargende, H.-C. Reuss & J. Wiedemann (Hrsg.), *14. Internationales Stuttgarter Symposium* (Internationales Stuttgarter Symposium, Bd. 14). Wiesbaden: Springer Vieweg.
- Albers, A. & Gausemeier, J. (2012). Von der fachdisziplinentorientierten Produktentwicklung zur vorausschauenden und systemorientierten Produktentstehung. In *Smart Engineering* (S. 17–29). Heidelberg: Springer. https://doi.org/10.1007/978-3-642-29372-6_3
- Albers, A., Heimicke, J., Hirschter, T., Richter, T., Reiß, N., Maier, A. et al. (2018). Managing Systems of Objectives in the agile Development of Mechatronic Systems by ASD – Agile Systems Design. In The Design Society (Eds.), *Proceedings of NordDesign 2018. Design in the Era of Digitalization* (n. p.). Linköping, Schweden: The Design Society.
- Albers, A., Heimicke, J., Walter, B., Basedow, G., Reiß, N., Heitger, N. et al. (2018). Product Profiles. Modelling customer benefits as a foundation to bring inventions to innovations. In F. Laroche & A. Bernard (Eds.), *Proceedings of the 28th CIRP Design Conference*. Nantes, Frankreich. ISSN: 2212-8271 (pp. 253–258). Elsevier B.V.
- Albers, A., Klingler, S. & Ebel, B. (2013). Modeling systems of objectives in engineering design practice. In U. Lindemann, S. Venkataraman, Y. Kim, S. W. Lee, J. Clarkson & G. Cascini (Eds.), *Proceedings of the 19th International Conference on Engineering Design (ICED) 2013*. ISSN 2220-4342 (Bd. 1, pp. 379–388).
- Albers, A., Kürten, C., Rapp, S., Birk, C., Hünemeyer, S. & Kempf, C. (2022). *SGE – Systemgenerationsentwicklung. Analyse und Zusammenhänge von*

Entwicklungspfad in der Produktentstehung (KIT Scientific Working Papers 199). <https://doi.org/10.5445/IR/1000151151>

- Albers, A. & Lohmeyer, Q. (Eds.). (2012). Advanced systems engineering – towards a model-based and human-centered methodology. *9th International Symposium on Tools and Methods of Competitive Engineering TMCE 2012*, Karlsruhe.
- Albers, A., Lohmeyer, Q. & Ebel, B. (2011). Dimensions of objectives in interdisciplinary product development projects. In S. J. Culley, B. J. Hicks, T. C. McAloone, T. J. Howard & Y. Reich (Eds.), *Proceedings of the 18th International Conference on Engineering Design (ICED 11). Impacting Society through Engineering* (ICED, Bd. 2, Bd. 2, pp. 256–265). Lyngby/ Kopenhagen, Dänemark: The Design Society.
- Albers, A., Lutz, S., Behrendt, M., Düser, T. & Hettel, R. (2017). Validation environment for function development and coverage of ADAS in the car development process. In M. Bargende, H.-C. Reuss & J. Wiedemann (Hrsg.), *17. Internationales Stuttgarter Symposium*. Wiesbaden: Springer Fachmedien Wiesbaden.
- Albers, A., Mandel, C., Yan, S. L. & Behrendt, M. (2018). SYSTEM OF SYSTEMS APPROACH FOR THE DESCRIPTION AND DESIGN OF VALIDATIONENVIRONMENTS. In D. Marjanovic, M. Štorga, S. Škec, N. Bojčetić & N. Pavković (Eds.), *DS92: Proceedings of the DESIGN 2018 15th International Design Conference*. Glasgow, Schottland: The Design Society.
- Albers, A., Matros, K., Behrendt, M. & Jetzinger, H. (2015). Das Pull-Prinzip der Validierung – Ein Referenzmodell zur effizienten Integration von Validierungsaktivitäten in den Produktentstehungsprozess. *VDI Konstruktion*, 67(6), 74–81. <https://doi.org/10.37544/0720-5953-2015-06-74>
- Albers, A. & Matthiesen, S. (2002). Konstruktionsmethodisches Grundmodell zum Zusammenhang von Funktion und Gestalt technischer Systeme. *Konstruktion*. In *Konstruktion Zeitschrift für Produktentwicklung* (Bd. 54, 7-8/2002, S. 55–60). ISSN 0720-5953, 0373-3300. Düsseldorf: Springer-VDI-Verl.
- Albers, A., Matthiesen, S., Bursac, N., Moeser, G., Schmidt, S. & Lüdcke, R. (2015). Abstraktionsgrade der Systemmodellierung – von der Sprache zur Anwendung. In M. Maurer, S.-O. Schulze & J. Abulawi (Hrsg.), *Tag des*

- Systems Engineering* (S. 183–192). München: Hanser.
<https://doi.org/10.13140/2.1.4350.8007>
- Albers, A., Peglow, N., Powelske, J., Birk, C. & Bursac, N. (2018). Coping with Complex Systems-of-Systems in the Context of PGE – Product Generation Engineering. In F. Laroche & A. Bernard (Eds.), *Procedia CIRP*, 70, 457–462. <https://doi.org/10.1016/j.procir.2018.01.004>
- Albers, A., Pinner, T., Yan, S. L., Hettel, R. & Behrendt, M. (2016). Koppelsystems: Obligatory Elements within Validation Setups. In *14th International Design Conference, DESIGN 2016* (DS 84, pp. 791–800). The Design Society.
- Albers, A. & Rapp, S. (2022). Model of SGE: System Generation Engineering as Basis for Structured Planning and Management of Development. In D. Krause & E. Heyden (Eds.), *Design Methodology for Future Products. Data Driven, Agile and Flexible* (pp. 27–46). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-78368-6_2
- Albers, A., Rapp, S., Spadinger, M., Richter, T., Birk, C., Marthaler, F. et al. (2019). The Reference System in the Model of PGE: Proposing a Generalized Description of Reference Products and their Interrelations. *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1), 1693–1702. <https://doi.org/10.1017/dsi.2019.175>
- Albers, A., Reiß, N., Bursac, N. & Richter, T. (2016). iPeM – Integrated Product Engineering Model in Context of Product Generation Engineering. In L. Wang & T. Kjellberg (Eds.), *Proceedings of the 26th CIRP Design Conference (CIRP) 2016*, 50(1), 100–105. Stockholm, Schweden: Elsevier Procedia. <https://doi.org/10.1016/j.procir.2016.04.168>
- Albers, A., Reiß, N., Bursac, N., Schwarz, L. & Lüdcke, R. (2013). Modelling technique for knowledge management, process management and method application - A Formula student exploratory study. In M. Schabacker, K. Gericke, N. Szélig & S. Vajna (Eds.), *Proceedings of the 3rd International Conference for Modelling and Management of Engineering Processes (MMEP) 2013* (pp. 151–162). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-44009-4_13
- Albers, A., Walter, B., Wilmsen, M. & Bursac, N. (2018). Live-Labs as Real-World Validation Environments for Design Methods. In D. Marjanovic, M. Storga, S. Skec, N. Bojčetić & N. Pavkovic (Eds.) (2018). *Proceedings of the DESIGN*

2018 15th International Design Conference (pp. 13–24). Dubrovnik, Kroatien: The Design Society, Glasgow, UK. <https://doi.org/10.21278/idc.2018.0303>

Albers, A. & Wintergerst, E. (2014). The Contact and Channel Approach (C&C²-A) – relating a system's physical structure to its functionality. In A. Chakrabarti & L. Blessing (Eds.), *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations* (pp. 151–172). London: Springer London. https://doi.org/10.1007/978-1-4471-6338-1_8

Alt, O. (2012). *Modellbasierte Systementwicklung mit SysML*. München: Carl Hanser Fachbuchverlag.

Andersen, C., Parikh, J., Kailas, A., Bitar, G., Konet, H., Meier, J. et al. (2016). Development of Vehicle-to-Infrastructure Safety Applications in the United States. In *ITS - World Congress* .

ASAM e. V. (Hrsg.). (2022, 20. Juli). *ASAM OpenSCENARIO@ 2.0.0*. Zugriff am 26.11.23 unter: https://www.asam.net/static_downloads/public/asam-openscenario/2.0.0/welcome.html

ASAM e.V. (Hrsg.). (2022, 2. Februar). *Evolving landscapes of collaborative testing for ADAS & AD. ASAM Test Specification Study Group Report 2022* (1.0.0). Zugriff am 26.11.23 unter: https://report.asam.net/wp-content/uploads/2022/02/ASAM-Test-Specification-Study-Group-Report-2022_V1-0-0.pdf

Bae, I., Moon, J. & Seo, J. (2019). Toward a Comfortable Driving Experience for a Self-Driving Shuttle Bus. *Electronics*, 8(9), 943. <https://doi.org/10.3390/electronics8090943>

Bagschik, G., Menzel, T., Reschka, A. & Maurer, M. (2017). Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen. In *11. Workshop Fahrerassistenzsysteme und automatisiertes Fahren* (S. 125–135). Uni-DAS e. V.

Balduf, F., Heinrich, J., Becker, M. & Adler, R. (2022). System of Systems Engineering in Deutschland: Bestandsaufnahme und Ausblick. In W. Koch, D. Wilke, S. Dreiseitel & R. Kaffenberger (Hrsg.) (2022). *Tag des Systems Engineering 2022* (S. 26–30). München: Carl Hanser Verlag.

- Baldwin, K. J. (Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, Hrsg.). (2008). *Systems Engineering Guide for Systems of Systems*. version 1.0. Zugriff am 26.11.23 unter: <https://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf>
- Balic, J. & Bulaja, M. (2021, 16. Juni). *Setting the Standards for Scenario-Based ADAS/AD Testing, Design and Editing*. Zugriff am 26.11.23 unter: <https://mobex.io/webinars/setting-the-standards-for-scenario-based-adas-ad-testing-design-and-editing/>
- Bartels, A., Eberle, U. & Knapp, A. (2015). *AdaptIVe - System Classification and Glossary* (AdaptIVe - Automated Driving Applications and Technologies for Intelligent Vehicles, Eds.) (AdaptIVe-SP2-v12-DL-D2.1).
- BCS SIGIST. (2001, 27. April). *Standard for component software testing. Working Draft 3.4* (British Computer Society Specialist Interest Group in Software Testing (BCS SIGIST), Eds.). Zugriff am 26.11.23 unter: <http://www.testingstandards.co.uk/Component%20Testing.pdf>
- DIN SAE SPEC 91381:2019-06 (2019). *Begriffe und Definitionen in Bezug auf die Prüfung automatisierter Fahrzeugtechnologien*. Berlin: Beuth Verlag GmbH.
- Bertalanffy, L. von (1950). An Outline of General System Theory. In *The British Journal for the Philosophy of Science* 1(2), 134–165.
<https://doi.org/10.1093/bjps/i.2.134>
- Bezai, N. E., Medjdoub, B., Al-Habaibeh, A., Chalal, M. L. & Fadli, F. (2021). Future cities and autonomous vehicles: analysis of the barriers to full adoption. *Energy and Built Environment*, 2(1), 65–81. ISSN 2666-1233.
<https://doi.org/10.1016/j.enbenv.2020.05.002>
- Bilal, M., Daclin, N. & Chapurlat, V. (2014). System of Systems Design Verification: Problematic, Trends and Opportunities. In K. Mertins, F. Bénaben, R. Poler & J.-P. Bourrières (Eds.), *Enterprise Interoperability VI. Interoperability for Agility, Resilience and Plasticity of Collaborations* (pp. 405–415). ISSN: 2199-2533. Basel, Schweiz: Springer International Publishing. https://doi.org/10.1007/978-3-319-04948-9_34
- Blessing, L. T. & Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. London: Springer London. <https://doi.org/10.1007/978-1-84882-587-1>

- Bock, J., Krajewski, R., Eckstein, L., Klimke, J., Sauerbier, J. & Zlocki, A. (2018). Data basis for scenario-based validation of HAD on highways. In *27th Aachen colloquium automobile and engine technology* (pp. 8–10).
- Böhm, M. (2020). *C-Roads - The platform of harmonised C-ITS deployment in Europe*. C-ROADS. Zugriff am 26.11.23 unter: https://www.c-roads.eu/fileadmin/user_upload/media/Dokumente/C-Roads_fact_sheet_2020.pdf
- Botte, M., Pariota, L., D’Acierno, L. & Bifulco, G. N. (2018). C-ITS Communication: An Insight on the Current Research Activities in the European Union. In P. Bouchner (Eds.), *International Journal of Transportation Systems*, 3, 52–63. ISSN 2534-8876. Zugriff am 26.11.23 unter: [https://www.iaras.org/iaras/filedownloads/ijts/2018/019-0007\(2018\).pdf](https://www.iaras.org/iaras/filedownloads/ijts/2018/019-0007(2018).pdf)
- Brade, T. (2022, 15. März). *How to derive “Top Goals” systematically?* Mid-Term Presentation 15 / 16 March 2022. Zugriff am 26.11.23 unter: https://www.vvm-projekt.de/fileadmin/user_upload/Mid-Term/Presentations/VVM_HZE_S1_P1_20220315_TopGoals.pdf
- Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M. & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4), 571–583. ISSN 0164-1212. <https://doi.org/10.1016/j.jss.2006.07.009>
- Breu, J., Brakemeier, A. & Menth, M. (2014). A quantitative study of Cooperative Awareness Messages in production VANETs. *EURASIP Journal on Wireless Communications and Networking*, 2014(1), 98. <https://doi.org/10.1186/1687-1499-2014-98>
- Bryans, J., Fitzgerald, J., Payne, R., Miyazawa, A. & Kristensen, K. (2014). SysML Contracts for Systems of Systems. In IEEE (Eds.), *International Conference on System of Systems Engineering (SOSE)*, 9, 73–78. o.O.: IEEE. <https://doi.org/10.1109/sysose.2014.6892466>
- Bundesministerium für Digitales und Verkehr (Hrsg.). (2020). *Digitale Testfelder für das automatisierte und vernetzte Fahren im Realverkehr in Deutschland*. Zugriff am 26.11.23 unter: https://www.bmdv.bund.de/SharedDocs/DE/Anlage/DG/Digitales/uebersicht-digitale-testfelder-avf-bmvi.pdf%3F__blob%3DpublicationFile&ved=2ahUKEwiiij-

ek2KzvAhXH16QKHTS7BmgQFjAAegQIARAC&usg=AOvVaw1_vWssnWPslu
UOZQc1pLPu

Bundesministerium für Justiz (n.d.). Straßenverkehrs-Ordnung (StVO). BGBl. I,
(108) StVO, 367.

Bundesministerium für Verkehr und digitale Infrastruktur (Hrsg.). (2017). *So fahren wir morgen* (kompakt2). Zugriff am 26.11.23 unter:
https://bmdv.bund.de/SharedDocs/DE/Publikationen/G/kompakt-automatisiertes-fahren.pdf?__blob=publicationFile

Bundestag. Gesetz zur Änderung des Straßenverkehrsgesetzes und des Pflichtversicherungsgesetzes – Gesetz zum autonomen Fahren. BGBl I Nr. 48. *Bundesanzeiger Verlag* (Nr. 48), 3108–3114. Zugriff am 26.11.23 unter:
https://bmdv.bund.de/SharedDocs/DE/Anlage/Gesetze/Gesetze-19/gesetz-aenderung-strassenverkehrsgesetz-pflichtversicherungsgesetz-autonomes-fahren.pdf?__blob=publicationFile

Bungartz, H.-J., Zimmer, S., Buchholz, M. & Pflüger, D. (2013). *Modellbildung und Simulation*. Berlin, Heidelberg: Springer Berlin Heidelberg.
<https://doi.org/10.1007/978-3-642-37656-6>

Bursac, N. (2016). Model Based Systems Engineering zur Unterstützung der Baukastenentwicklung im Kontext der Frühen Phase der Produktgenerationsentwicklung. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 93). Karlsruhe: Karlsruher Institut für Technologie (KIT).
<https://doi.org/10.5445/IR/1000054484>

Bussler, A. (2022). *A METHOD TO SELECT RELEVANT SCENARIOS FOR A V&V PROCESS. From Data and Knowledge to Scenarios for Testing*. Projekt „VV-Methoden“ Validierung und Verifikation für hochautomatisiertes Fahren.

CAR 2 CAR Communication Consortium (Eds.). (2021). *Guidance for day 2 and beyond roadmap* (1.2). 2072. Zugriff am 26.11.23 unter: https://www.car-2-car.org/fileadmin/documents/General_Documents/C2CCC_WP_2072_RoadmapDay2AndBeyond_V1.2.pdf

Carvalho Pinheiro, H. de, Stanke, D. C., Ferraris, A., Carello, M., Gabiati, G., Camuffo, I. et al. (2021). Autonomous Driving Scenario Generation in Overtake

- Manoeuvres Through Data Fusion. In Niola, V., Gasparetto, A. (Eds.), *Advances in Italian Mechanism Science* (IFTToMM ITALY 2020. Mechanisms and Machine Science,, Bd. 91, o.O.). Springer, Cham.
https://doi.org/10.1007/978-3-540-69293-5_9
- Chen, L., Jiao, J. & Zhao, T. (2020). A Novel Hazard Analysis and Risk Assessment Approach for Road Vehicle Functional Safety through Integrating STPA with FMEA. *Applied Sciences*, 10(21), 7400.
<https://doi.org/10.3390/app10217400>
- Christie, J. (2008). The seductive and dangerous V model. The Magazine for Professional Testers. *Testing Experience* (ISSN: 1866-5705).
- Clarivate. (2022, 6. Oktober). *Web of Science Core Collection - Clarivate*. Zugriff am 26.11.23 unter: <https://clarivate.com/products/scientific-and-academic-research/research-discovery-and-workflow-solutions/web-of-science/web-of-science-core-collection/>
- Coleman, J. W., Malmos, A. K., Larsen, P. G., Peleska, J., Hains, R., Andrews, Z. et al. (2012). COMPASS tool vision for a system of systems Collaborative Development Environment. In *7th International Conference on System of Systems Engineering (SoSE)*, 7, 451–456). o.O.: IEEE.
<https://doi.org/10.1109/sysose.2012.6384150>
- Commission Mobility and Transport (Eds.). (2016). *C-ITS Platform. Final report*. Zugriff am 26.11.23 unter: <https://transport.ec.europa.eu/system/files/2016-09/c-its-platform-final-report-january-2016.pdf>
- Connected Motorcycle Consortium (Eds.). (2020). *Application Specification. Hazardous Location Notification (HLN)*. 1.0. Zugriff am 26.11.23 unter: https://www.cmc-info.net/uploads/1/2/1/4/121453783/4.2_-_hln_-_use_case_description___technical_description.pdf
- Convex-project.de. (2020, 14. Juli). *Use Cases | convex-project.de*. Zugriff am 26.11.23 unter: <https://convex-project.de/usecases.html>
- Costello, R. J. & Liu, D.-B. (1995). Metrics for requirements engineering. *Journal of Systems and Software*, 29(1), 39–63. ISSN 0164-1212.
[https://doi.org/10.1016/0164-1212\(94\)00127-9](https://doi.org/10.1016/0164-1212(94)00127-9)

- Cox, A. E., Mueller, A. S. & Cicchino, J. B. (2023). *Safety potential of crash avoidance features, improved headlights, and V2V-enhanced technologies for older drivers*. Journal of Safety Research.
<https://doi.org/10.1016/j.jsr.2023.02.004>
- Dahmann, J. (2014). System of Systems Pain Points. *INCOSE International Symposium*, 24(1), 108–121. Session 1 Track 4: SoS.
<https://doi.org/10.1002/j.2334-5837.2014.tb03138.x>
- Dahmann, J. & Baldwin, K. J. (2008). Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering. In *2008 2nd Annual IEEE Systems Conference*. o.O.: IEEE.
<https://doi.org/10.1109/systems.2008.4518994>
- Dahmann, J., Lane, J. A., Rebovich, G. & Lowry, R. (2010). Systems of systems test and evaluation challenges. In *2010 5th International Conference on System of Systems Engineering (SoSE)*. o.O.: IEEE.
<https://doi.org/10.1109/sysose.2010.5543979>
- Dahmann, J. & Roedler, G. (2016). SYSTEMS OF SYSTEMS ENGINEERING STANDARDS. *INSIGHT*, 19(3), 23–26. <https://doi.org/10.1002/inst.12102>
- Dandashi, F. & Hause, M. C. (2015). UAF for system of systems modeling. In *2015 10th System of Systems Engineering Conference (SoSE)*. o.O.: IEEE.
<https://doi.org/10.1109/sysose.2015.7151944>
- Dänzer, M., Kleiner, S., Lamm, J. G., Moeser, G., Morant, F., Munker, F. et al. (2014). Funktionale Systemmodellierung nach der FAS-Methode: Auswertung von vier Industrieprojekten. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering 2014* (S. 75–84). München: Carl Hanser Verlag.
- Delligatti, L. (2014). *SysML distilled. A brief guide to the systems modeling language*. o. O.: Pearson Education (US).
- Deutscher Bundestag (n.d.). Gesetz zur Änderung des Straßenverkehrsgesetzes. 18/11776. (Deutschland). BGBl. I, (20). Zugriff am 26.11.23 unter:
<https://dserver.bundestag.de/btd/18/113/1811300.pdf>

- Dona, R. & Ciuffo, B. (2022). Virtual Testing of Automated Driving Systems. A Survey on Validation Methods. *IEEE Access*, 10, 24349–24367.
<https://doi.org/10.1109/ACCESS.2022.3153722>
- Dudenredaktion. (n.d.). „Haus“ auf Duden online. Berlin: Cornelsen Verlag GmbH.
Zugriff am 26.11.23 unter: <https://www.duden.de/node/63917/revision/697830>
- Dumitrescu, R., Albers, A., Riedel, O., Stark, R. & Gausemeier, J. (2021). *Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft. Ein Beitrag zum Advanced Systems Engineering*. Paderborn.
- Düser, T. (2010). X-in-the Loop, ein durchgängiges Validierungsframework für die Fahrzeugentwicklung am Beispiel von Antriebsstrangfunktionen und Fahrerassistenzsystemen. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 47). Karlsruhe: Karlsruher Institut für Technologie (KIT).
<https://doi.org/10.5445/IR/1000020671>
- Düser, T., Olms, H., Haase, A., Gavel, R. von, Schmidt, C. & Schmidt, U. (2011). *Fahrzeugrollenprüfstände. Von der Zertifizierung bis hin zur mechatronischen Entwicklungsplattform* (Die Bibliothek der Technik, Bd. 341). Landsberg/Lech: Verl. Moderne Industrie.
- Ebel, B. (2015). Modellierung von Zielsystemen in der interdisziplinären Produktentstehung. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung, Systeme, Methoden, Prozesse* (Bd. 85). Karlsruhe: Karlsruher Institut für Technologie (KIT).
<https://doi.org/10.5445/IR/1000048334>
- Eberle, U. & Jütten, V. (2017, 28. Juli). *Validating the safety of automated driving. Adaptive Final Event*, Aachen. Zugriff am 26.11.23 unter:
https://www.adaptive-ip.eu/files/adaptive/content/downloads/Deliverables%20&%20papers/Adaptive_Final%20Event_Safety%20Validation_Vera%20Juetten_Ulrich%20Eberle_20170628.pdf
- EnBW Energie Baden-Württemberg AG. (2022, 30. März). *BMW i3 | EnBW E-Auto Finder*. Zugriff am 26.11.23 unter: <https://www.enbw.com/blog/cars/bmw-i3s/>

- Ericson, C. A. (1999). Fault Tree Analysis – A History. In *17th International System Safety Conference* (pp. 1–9). International System Safety Society (ISSS).
Zugriff am 26.11.23 unter:
<https://ftaassociates.files.wordpress.com/2018/12/C.-Ericson-Fault-Tree-Analysis-A-History-Proceedings-of-the-17th-International-System-Safety-Conference-1999.pdf>
- Estefan, J. (2008). Survey of Model-Based Systems Engineering (MBSE) Methodologies. *INCOSE MBSE Focus Group*, 25.
- ETSI, EN 302 637 (09-2014). *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative; Awareness Basic Service*.
- ETSI, EN 302 663 (05-2019). *Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band*.
- ETSI, TS 103 301 (03-2021). *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services; Release 2*.
- Euro NCAP (Eds.). (2017). *Euro NCAP 2025 Roadmap. in pursuit of vision zero*.
Zugriff am 26.11.23 unter: <https://www.euroncap.com/en/for-engineers/technical-papers/>
- Euro NCAP (Eds.). (2019). *EUROPEAN NEW CAR ASSESSMENT PROGRAMME. TEST PROTOCOL – AEB Car-to-Car systems Version 3.0.2*.
Zugriff am 26.11.23 unter: <https://cdn.euroncap.com/media/56143/euro-ncap-aeb-c2c-test-protocol-v302.pdf>
- Europäische Kommission. DURCHFÜHRUNGSBESCHLUSS (EU) 2020/1426 DER KOMMISSION. C(2020) 6773. Zugriff am 26.11.23 unter: <https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32020D1426&from=DE>
- EUROPEAN PARLIAMENT & COUNCIL (Eds.). (2010). *DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 7 July 2010. on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport*.
Legislative acts (Official Journal of the European Union Document

02010L0040). Strasbourg, Frankreich. Zugriff am 26.11.23 unter: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02010L0040-20180109>

- Fagen, M. D. (Eds.). (1978). *A History of engineering and science in the Bell system* (1st printing). New York: Bell Telephone Laboratories.
- Faldik, O., Payne, R., Fitzgerald, J. & Buhnova, B. (2017). Modelling System of Systems Interface Contract Behaviour. *Electronic Proceedings in Theoretical Computer Science*, 245(5), 1–15. <https://doi.org/10.4204/EPTCS.245.1>
- Fedullo, T., Morato, A., Tramarin, F., Cattini, S. & Rovati, L. (2022). Artificial Intelligence - Based Measurement Systems for Automotive: a Comprehensive Review. In *2022 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)* (pp. 122–127). o.O.: IEEE. <https://doi.org/10.1109/metroautomotive54295.2022.9855154>
- Feldhusen, J. & Grote, K.-H. (2013). *Pahl/Beitz Konstruktionslehre - Methoden und Anwendung erfolgreicher Produktentwicklung* (8. vollständig überarbeitete Auflage 2013). Berlin Heidelberg: Springer Vieweg.
- Fisher, R. (1971). *The Design of Experiments* (9. Aufl.). London: Collier MacMillan Canada Ltd.
- Friedenthal, S., Beihoff, B., Oster, C., Paredis, C., Kemp, D., Stoewer, H. et al. (2014). *INCOSE Systems Engineering Vision 2025*. International Council on Systems Engineering (INCOSE) , Eds.). Zugriff am 26.11.23 unter: https://www.incose.org/about-systems-engineering_2/se-vision-2025.
- Friedenthal, S., Davey, C., Nielsen, P., Schreinemakers, P., Oster, C., Stoewer, H. et al. (2021). *Systems Engineering Vision 2035. Engineering solutions for a better world* (International Council on Systems Engineering (INCOSE), Eds.). Zugriff am 26.11.23 unter: https://www.incose.org/docs/default-source/se-vision/incose-se-vision-2035.pdf?sfvrsn=e32063c7_10
- Friedenthal, S., Griego, R. & Sampson, M. (2007, 29. Juli). *INCOSE Model Based Systems Engineering (MBSE) Initiative*. INCOSE 2007 Symposium, San Diego, CA, USA.
- Galbras, R. (2018, 3. September). *Projekt „VV-Methoden“ Validierung und Verifikation für hochautomatisiertes Fahren*. Symposium „Testen -

Automatisiertes und Vernetztes Fahren“. Zugriff am 26.11.23 unter:
<https://docplayer.org/130536932-Projekt-vv-methoden-validierung-und-verifikation-fuer-hochautomatisiertes-fahren.html>

- Gardner, K., D'Souza, C., Hounsell, N., Shrestha, B. & Bretherton, D. (UITP WORKING GROUP: Interaction of buses and signals at road crossings, Eds.). (2009). *Review of Bus Priority at Traffic Signals around the World. FINAL REPORT Version 2.0*. Working Program Bus Committee 2007-2009 – Technical Cluster, Nationale Plattform Mobilität der Zukunft; NPM. Zugriff am 26.11.23 unter: <https://content.tfl.gov.uk/interaction-of-buses-and-signals-at-road-crossings.pdf>
- Gasser, T. M., Schmidt, E. A., Bengler, K., Chiellino, U., Diederichs, F., Eckstein, L. et al. (2015, 19. Januar). *Bericht zum Forschungsbedarf. Runder Tisch Automatisiertes Fahren – AG Forschung* (BMVI, Hrsg.).
- Gausemeier, J., Gaukstern, T. & Tschirner, C. (2013). Systems Engineering Management Based on a Discipline-Spanning System Model. *Procedia Computer Science*, 16, 303–312. ISSN 1877-0509.
<https://doi.org/10.1016/j.procs.2013.01.032>
- Glimm, T. & Zadek, H. (2021). Konzept zur Realisierung eines GLOSA-Systems ohne Datenverbindung zu einem Verkehrssteuergerät. *Logistics Journal : Proceedings*. https://doi.org/10.2195/lj_Proc_glimm_de_202112_01
- Götz, G., Stich, P., Backhaus, J. & Reinhart, G. (2016). Design Approach for the Development of Format-Flexible Packaging Machines. *International Journal of Mechanical and Mechatronics Engineering*, (Vol. 10), 108–117.
<https://doi.org/10.5281/zenodo.1338736>
- Grundel, M., Abulawi, J., Moeser, G., Weilkiens, T., Scheithauer, A., Kleiner, S. et al. (2014). FAS4M - No more: “Please mind the gap!”. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering 2014* (S. 65–74). München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446443761.007>
- Guerrero-Ibanez, A., Flores, C., Damian-Reyes, P., Barba, A. & Reyes, A. (2011). A Performance Study of the 802.11p Standard for Vehicular Applications. In J. E. Guerrero (Hrsg.), *2011 Seventh International Conference on Intelligent Environments*, 17, 165–170. o.O.: IEEE. <https://doi.org/10.1109/ie.2011.26>

- Gühmann, C. (2002). Testautomatisierung und Modellbildung für die Hardware-in-the-Loop Simulation. In *IIR Tagung Versuch, Test und Simulation*. Zugriff am 26.11.23 unter: <http://www.iea.tu-berlin.de/uploads/media/Guehmann2002.pdf>
- Haas, M. (Bitkom Research, Hrsg.). (2023). *Smartphone-Markt: Konjunktur & Trends*. Zugriff am 26.11.23 unter: <https://www.bitkom.org/sites/main/files/2023-02/BitkomPraesentationPressekonferenzSmartphoneMarkt.pdf>
- Hakim, M. & Kreuer, D. (2014). Verfahren der Mobilfunktechnik für Car-to-X-Prüfstände. *ATZelektronik*, 9(6), 12–19. <https://doi.org/10.1365/s35658-014-0500-y>
- Hall, A. D. (1962). *A methodology for systems engineering* (1. publ., repr). Princeton, N.J. [u.a.]: D. Van Nostrand Company, Inc.
- Hauptmann, E. (2019, 20. Januar). Autofahrer sehen Rot. *Stuttgarter Zeitung* (Foto: Sebastian Steegmüller). Zugriff am 26.11.23 unter: <https://www.stuttgarter-zeitung.de/inhalt.ampel-in-stuttgart-obertuerkheim-autofahrer-sehen-rot.7180a171-b86c-4480-b545-1bf0591a937a.html>
- Heimicke, J., Roebenack, S., Frobieter, C., Tihlarik, A., Albert, B., Bramato, L. et al. (2021). Evaluation of Challenges in the Implementation of Scrum in a large German Plant Engineering Company: Derivation of Hypotheses for an Improved Introduction of Agile Approaches into the Processes of Physical Product Development. In *R&D Management Conference 2021 - Innovation in an Era of Disruption* (n.p.). online.
- Heiser, P. (2018). *Meilensteine der qualitativen Sozialforschung*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-18557-2>
- Henshaw, M. (2016). SYSTEMS OF SYSTEMS, CYBER-PHYSICAL SYSTEMS, THE INTERNET-OF-THINGS...WHATEVER NEXT? *INSIGHT*, 19(3), 51–54. <https://doi.org/10.1002/inst.12109>
- Herman elektronika (Eds.). (o.J.) Unit for components of transport infrastructure [Themenheft].
- Herman elektronika (Eds.). (2022, 24. November). *Case studies – V2X-PRIORITY*. Zugriff am 26.11.23 unter: <https://www.v2x-priority.com/en/use-cases/>

- Hieber, L. & Kammeyer, H.-U. (2014). *Verantwortung von Ingenieurinnen und Ingenieuren*. In *Verantwortung von Ingenieurinnen und Ingenieuren* (S. 15–31) Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-05530-1>
- Holt, J. & Perry, S. (2013a). *Guidelines for Systems of Systems Requirements. Comprehensive Modelling for Advanced Systems of Systems*. COMPASS White Paper WP01. Zugriff am 26.11.23 unter: <http://www.compass-research.eu>
- Holt, J. & Perry, S. (2013b). *SysML for systems engineering. A model based approach* (Professional applications of computing series, Bd. 10). London: Institution of Engineering and Technology.
- Holt, J. & Perry, S. (2018). *SysML for systems engineering. A model-based approach* (Computing and Networks, vol. 20, Third edition). Stevenage: Institution of Engineering and Technology. <https://doi.org/10.1049/PBPC020E>
- Hsu, C.-H., Wang, S., Zhou, A. & Shawkat, A. (Eds.). (2016). *Internet of vehicles - technologies and services. Third international conference, IOV 2016, Nadi, Fiji, December 7-10, 2016, Proceedings*. o. O.: Springer.
- Humpert, L., Wäschle, M., Horstmeyer, S., Dumitrescu, R. & Albers, A. (2023). Stakeholder-oriented Elaboration of Artificial Intelligence use cases using the example of Special-Purpose engineering. *Procedia CIRP*, 119, 693–698. <https://doi.org/10.1016/j.procir.2023.02.160>.
- ISO/IEC, 5469 (in Entwicklung). *ISO/IEC CD TR 5469: Artificial intelligence — Functional safety and AI systems*. Berlin: Beuth Verlag GmbH.
- ISO, 26262-1:2018(E) (2018-12). *ISO 26262-1:2018(en) Road vehicles — Functional safety — Part 1: Vocabulary*. Berlin: Beuth Verlag GmbH.
- ISO, 26262-2:2018(E) (2018-12). *ISO 26262-2:2018(en) Road vehicles — Functional safety — Part 2: Management of functional safety*. Berlin: Beuth Verlag GmbH.
- ISO, 26262-3:2018(E) (2018-12). *ISO 26262-3:2018(en) Road vehicles — Functional safety — Part 3: Concept phase*. Berlin: Beuth Verlag GmbH.

- ISO, 26262-4:2018(E) (2018-12). *ISO 26262-4:2018(en) Road vehicles — Functional safety — Part 4: Product development at the system level*. Berlin: Beuth Verlag GmbH.
- ISO, 26262-5:2018(E) (2018-12). *ISO 26262-5:2018(en) Road vehicles - Functional safety - Part 5: Product development at the hardware level*. Berlin: Beuth Verlag GmbH.
- ISO, 26262-6:2018(E) (2018-12). *ISO 26262-6:2018(en) Road vehicles - Functional safety - Part 6: Product development at the software level*. Berlin: Beuth Verlag GmbH.
- ISO, 26262-8:2018(E) (2018-12). *ISO 26262-8:2018(en) Road vehicles — Functional safety — Part 8: Supporting processes*. Berlin: Beuth Verlag GmbH.
- ISO, 26262-9:2018(E) (2018-12). *ISO 26262-9:2018(en) Road vehicles — Functional safety — Part 9: Automotive safety integrity level (ASIL)-oriented and safety-oriented analyses*. Berlin: Beuth Verlag GmbH.
- BS ISO/IEC/IEEE 21839:2019 (25.07.2019). *ISO/IEC/IEEE 21839:2019 Systems and software engineering — System of systems (SoS) considerations in life cycle stages of a system*. Berlin: Beuth Verlag GmbH.
- BS ISO/IEC/IEEE 21841:2019 (25.07.2019). *ISO/IEC/IEEE 21841:2019 Systems and software engineering. Taxonomy of systems of systems*. Berlin: Beuth Verlag GmbH.
- ISO/PAS 21448:2019(E) (2019-01). *ISO/PAS 21448:2019 Road vehicles — Safety of the intended functionality*. Berlin: Beuth Verlag GmbH.
- ISO/SAE 21434:2021 (2021-08). *ISO/SAE 21434:2021 Road vehicles — Cybersecurity engineering*. Berlin: Beuth Verlag GmbH.
- Jamshidi, M. (2008). *System of systems engineering. Innovations for the 21st century* (Wiley series in systems engineering and management). Hoboken, N.J, USA: John Wiley & Sons, Inc. <https://doi.org/10.1002/9780470403501>
- Kaner, C., Falk, J. & Nguyen, H. Q. (1999). *Testing computer software* (2. ed.). New York, Weinheim: Wiley.

- Kang, D. J. & Park, S. (2016). MAS based Approach to HEMS modeling: Application of social interaction mechanism to demand-side dynamics. In *2016 11th System of Systems Engineering Conference (SoSE)*. o.O.: IEEE.
<https://doi.org/10.1109/sysose.2016.7542894>
- Keysight Technologies, Inc. (Eds.). (26. Juli / 2018). *How 5G Will Influence Autonomous Driving Systems. ADAS and Autonomous Driving Improve Safety and Save Lives* (5992-2989EN). USA. Zugriff am 26.11.23 unter:
<https://www.keysight.com/us/en/assets/7018-06173/white-papers/5992-2998.pdf>
- Kharatyan, A., Humpert, L., Anacker, H., Dumitrescu, R., Wäschle, M., Albers, A. et al. (2022). Künstliche Intelligenz im Engineering: Menschorientierte Analyse von Potenzialen am Beispiel vom Sondermaschinenbau. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 117(6), 427–431. Berlin: Walter de Gruyter GmbH. <https://doi.org/10.1515/zwf-2022-1074>
- Klare, H., Kramer, M. E., Langhammer, M., Werle, D., Burger, E. & Reussner, R. (2021). Enabling consistency in view-based system development — The Vitruvius approach. *Journal of Systems and Software*, 171, 110815.
<https://doi.org/10.1016/j.jss.2020.110815>
- Kleiner, S., Husung, S., Mandel, C., Albers, A. & Behrendt, M. (2017). (Model-Based) Systems Engineering für die Digitalisierung der Produktentwicklung. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering* (S. 135–144). München: Carl Hanser Verlag GmbH & Co. KG. <https://doi.org/10.3139/9783446455467.015>
- Knight, E. C., Sòlymos, P., Scott, C. & Bayne, E. M. (2020). Validation prediction: a flexible protocol to increase efficiency of automated acoustic processing for wildlife research. *Ecological Applications*, 30(7).
<https://doi.org/10.1002/eap.2140>
- Kraftfahrt-Bundesamt (Hrsg.). (2021). *Durchschnittsalter der Personenkraftwagen wächst* (Fahrzeugstatistik). Flensburg. Zugriff am 26.11.23 unter:
https://www.kba.de/DE/Statistik/Fahrzeuge/Bestand/Fahrzeugalter/2021/2021_b_kurzbericht_fz_alter_pdf?__blob=publicationFile&v=2#:~:text=Das%20durchschnittliche%20Alter%20der%20zugelassenen,%3A%2018%2C4%3B%201.

- Kretschmer, R., Khelladi, D. E., Lopez-Herrejon, R. E. & Egyed, A. (2021). Consistent change propagation within models. *Software and Systems Modeling, 20*(2), 539–555. <https://doi.org/10.1007/s10270-020-00823-4>
- Krusche, T. (2000). *Strukturierung von Anforderungen für eine effiziente und effektive Produktentwicklung. Bericht / Institut für Konstruktionslehre, Maschinen-und Feinwerkelemente TU Braunschweig*. Aachen: Mainz.
- Kubin, A., Wagenmann, S., Reichert, F., Mandel, C. & Albers, A. (2023). Stakeholder Needs in Systems Engineering: A Proposal for a Formal Definition. In IEEE (Eds.), *The 17th Annual IEEE International Systems Conference*. o.O.: IEEE. <https://doi.org/10.1109/syscon53073.2023.10131072>
- Kurrle, A. (2017). Durchgängige Dokumentation von verteilten Zielsystemen in der Produktentwicklung durch Verwendung semantischer Metainformationen am Beispiel Connected Car. Dissertation. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung, Systeme, Methoden, Prozesse* (Bd. 108). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000080059>
- Tiefbau- und Verkehrsamt Erfurt Abteilung Verkehr. (2014, 29. Januar). *Technische Anforderungen an Lichtsignalanlagen. des Tiefbau- und Verkehrsamtes Erfurt*. Forderungen zur Planung und Errichtung von Lichtsignalanlagen, in Trägerschaft der Stadt Erfurt (Landeshauptstadt Erfurt Stadtverwaltung, Hrsg.). Erfurt. Zugriff am 26.11.23 unter: https://www.erfurt.de/mam/ef/leben/arbeit_und_beruf/ausschreibungen/vorschriften/technische_anforderungen_lsa.pdf
- Lane, J. A. & Epstein, D. J. (2013). *What is a System of Systems and Why Should I Care?* Los Angeles, CA, USA (Report USC-CSSE-2013-001).
- Lawrenz, W. (2011). *Controller area network : CAN. Grundlagen, Design, Anwendungen, Testtechnik*. Obermüller, Nils (Hrsg.). ISBN 978-3-8007-3332-3. Vde Verlag GmbH, Berlin - Offenbach.
- Lee, E. (2015). The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors (Basel, Schweiz), 15*(3), 4837–4869. <https://doi.org/10.3390/s150304837>

- Leveson, N. & Thomas, J. (2018). *STPA Handbook*. Zugriff am 26.11.23 unter: https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf
- Likert, R. (1932). A technique for measurement of attitudes. *Archives of Psychology*, (140), 5–55. New York.
- LimeSurvey Project Team / Carsten Schmitz. (2012). *LimeSurvey: An Open Source survey tool*. Hamburg. Zugriff am 26.11.23 unter: <http://www.limesurvey.org>
- Lohmeyer, Q. (2013). Menschzentrierte Modellierung von Produktentstehungssystemen unter besonderer Berücksichtigung der Synthese und Analyse dynamischer Zielsysteme. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung, Systeme, Methoden, Prozesse* (Bd. 59). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000035102>
- Lollini, P., Mori, M., Babu, A. & Bouchenak, S. (2016). AMADEOS SysML Profile for SoS Conceptual Modeling. In A. Bondavalli, S. Bouchenak & H. Kopetz (Eds.), *Cyber-Physical Systems of Systems. Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy* (pp. 97–127). ISSN: 1611-3349. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-47590-5_4
- Luna, S., Lopes, A., Tao, H. Y. S., Zapata, F. & Pineda, R. (2013). Integration, Verification, Validation, Test, and Evaluation (IVVT&E) Framework for System of Systems (SoS). In V. Alexandrov, M. Lees, V. Krzhizhanovskaya, J. Dongarra & P. M. Soot (Eds.), *Procedia Computer Science* (Bd. 20, pp. 298–305). ISSN 1877-0509. Amsterdam, Niederlande: Elsevier B.V. <https://doi.org/10.1016/j.procs.2013.09.276>
- Lutz, S., Behrendt, M. & Albers, A. (2020). Continuous development environment for the validation of autonomous driving functions. In M. Bargende, H.-C. Reuss & A. Wagner (Hrsg.), *20. Internationales Stuttgarter Symposium* (Proceedings, S. 327–343). Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-30995-4_32
- MacHardy, Z., Khan, A., Obana, K. & Iwashina, S. (2018). V2X Access Technologies: Regulation, Research, and Remaining Challenges. *IEEE Communications Surveys & Tutorials*, 20(3), 1858–1877. <https://doi.org/10.1109/COMST.2018.2808444>

- Maier, A. M., Hepperle, C., Kreimeyer, M., Eckert, C., Lindemann, U. & Clarkson, P. J. (2007). ASSOCIATIONS BETWEEN FACTORS INFLUENCING ENGINEERING DESIGN COMMUNICATION. In J. C. Bocquet (Eds.), *Proceedings of ICED 2007, the 16th International Conference on Engineering Design* (ICED, pp. 1–14). Paris, Frankreich.: The Design Society.
- Maier, M. (1996). Architecting Principles for Systems-of-Systems. *INCOSE International Symposium*, 6(1), 565–573. <https://doi.org/10.1002/j.2334-5837.1996.tb02054.x>
- Maier, M. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, 1(4), 267–284. [https://doi.org/10.1002/\(SICI\)1520-6858\(1998\)1:4<267::AID-SYS3>3.0.CO;2-D](https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D)
- Maier, M. (2021, 14. Juni). *Systems-of-Systems and the Spectrum of System Classes*. Keynotes, Referat auf der 16th Annual System of Systems Engineering Conference. online. Zugriff am 26.11.23 unter: <http://sosengineering.org/2021/keynotes/#Maier>
- Mandel, C., Behrendt, M. & Albers, A. (2021). Nutzer- und Aufgabengerechte Unterstützung von Modellierungsaktivitäten im Kontext des MBSE-Model-Based Systems Engineering. In R. Stelzer & J. Krzywinski (Hrsg.), *EEE2021 - Entwerfen Entwickeln Erleben* (S. 727–736). Dresden: TUDpress Verlag der Wissenschaften. <https://doi.org/10.25368/2021.64>
- Mandel, C., Böning, J., Behrendt, M. & Albers, A. (2021). A Model-Based Systems Engineering Approach to Support Continuous Validation in PGE - Product Generation Engineering. In *2021 IEEE International Symposium on Systems Engineering (ISSE)*. o. O.: IEEE. <https://doi.org/10.1109/isse51541.2021.9582475>
- Mandel, C., Guenther, M., Martin, A., Windisch, E., Bursac, N., Rapp, S. et al. (2022). Towards a System of Systems Engineering Architecture Framework. In *2022 17th Annual System of Systems Engineering Conference (SOSE)* (pp. 221–226). o.O.: IEEE.
- Mandel, C., Martin, A. & Albers, A. (2022). Addressing Factors for User Acceptance of Model-Based Systems Engineering. In L. Bitetti, I. Bitran, S. Conn, J. Fishburn, E. Huizingh, M. Torkkeli et al. (Eds.), *Proceedings of the XXXIII ISPIM Innovation Conference "Innovating in a Digital World"* (n.p.). Lappeenranta-Lahti, Finland: LUT Scientific and Expertise Publications.

Zugriff am 26.11.23 unter:

https://www.conferencesubmissions.com/ispim/copenhagen2022/documents/1232749706_Paper.pdf

- Mandel, C., Wäschle, M., Lutz, S. & Behrendt, M. (2021). A Model-Based approach for automation and traceability of validation activities – clarified for Advanced Driver Assistance Systems. In *9th Int. Symposium on Development Methodology* (n.p.). Wiesbaden. doi:10.5445/IR/1000142322.
- Mandel, C., Wolter, K., Bause, K., Behrendt, M., Hanf, M. M. & Albers, A. (2020). Model-Based Systems Engineering methods to support the reuse of knowledge within the development of validation environments. In IEEE (Eds.), *SysCon 2020. 14th Annual IEEE International Systems Conference*. o.O.: IEEE. <https://doi.org/10.1109/syscon47679.2020.9275919>
- Martin, A. (2022). Advanced Engineering Change Impact Approach (AECIA) – Towards a model-based approach for a continuous Engineering Change Management. In IEEE (Eds.), *8th IEEE ISSE International Symposium on Systems Engineering 2022*. o.O.: IEEE.
- Marxen, L. & Albers, A. (2012). Supporting Validation in the Development of Design Methods. In *Proceedings of DESIGN 2012 - the 12th International Design Conference, Dubrovnik, Croatia, May 21-24, 2012*. Ed.: D. Marjanovic (Bd. 3, pp. 1009–1018). University of Zagreb.
- Marxen, L. (2014). A Framework for Design Support Development based on the integrated Product Engineering Model iPeM. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung* (Bd. 74). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000045823>
- Maschmeyer, H., Beidl, C., Düser, T. & Schick, B. (2016). RDE-Homologation – Herausforderungen, Lösungen und Chancen. *MTZ - Motortechnische Zeitschrift*, 77(10), 84–91. <https://doi.org/10.1007/s35146-016-0110-8>
- Matthiesen, S. (2002). Ein Beitrag zur Basisdefinition des Elementmodells Wirkflächenpaare & Leitstützstrukturen zum Zusammenhang von Funktion und Gestalt technischer Systeme. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des Instituts für Maschinenkonstruktionslehre und Kraftfahrzeugbau* (Bd. 6). Karlsruhe: Universität Karlsruhe (TH). <https://doi.org/10.5445/IR/12092002>

- Maygua-Marcillo, L., Urquiza-Aguiar, L. & Paredes, C. (2018). *Wireless Channel 802.11*. In NS-3. Preprints 2018, 2018090367.
<https://doi.org/10.20944/preprints201809.0367.v1>
- McKinley, B. (2021). *Enabling Enhanced Safety and Advanced Use Cases with 3GPP R16 NR-V2X*. Referat 19.10.2021 innerhalb Webinarreihe Mobex by Automotive World. Zugriff am 26.11.23 unter:
<https://mobex.io/webinars/enabling-enhanced-safety-and-advanced-use-cases-with-3gpp-r16-nr-v2x/>
- McKinsey. (2018). *Reaktion gegenüber dem Gebrauch von autonomen Fahrzeugen durch Familienangehörige in China, Deutschland und den USA im Jahr 2018*. [Graph]. Statista. Zugriff am 26.11.23 unter:
<https://de.statista.com/statistik/daten/studie/959443/umfrage/umfrage-zur-akzeptanz-selbstfahrender-autos-in-laendern-weltweit/>
- McKinsey (Statista, Hrsg.). (2021). *Wie hoch sind die Entwicklungskosten für verschiedene Stufen des autonomen Fahrens?* Zugriff am 26.11.23 unter:
<https://de.statista.com/statistik/daten/studie/1294457/umfrage/umfrage-autonomes-fahren-entwicklungskosten/>
- Messerle, M. (2016). *Methodik zur Identifizierung der erfolgversprechendsten Produktideen in den frühen Phasen des Produktentwicklungsprozesses*. Dissertation. Universität Stuttgart, Stuttgart. <https://doi.org/10.18419/opus-8920>
- Metropolis, N. (1987). THE BEGINNING of the MONTE CARLO METHOD. *Los Alamos Science*, (15), 125–130. Special Issue. Zugriff am 26.11.23 unter:
<https://library.lanl.gov/cgi-bin/getfile?00326866.pdf>
- Moeser, G. (2016). *Kurzversion FAS4M Methodenguideline. Methodisches Vorgehen im FAS4M-Ansatz; Kurzversion der 24-seitigen Methodenguideline (konkrete Handlungsanleitungen) zur Veröffentlichung* (IPEK - Institut für Produktentwicklung am Karlsruher Institut für Technologie, Hrsg.). Karlsruhe. Zugriff am 26.11.23 unter: <http://digbib.ubka.uni-karlsruhe.de/volltexte/1000056208> <https://doi.org/10.5445/IR/1000056208>
- Monschein, D., Mazkattli, M., Heinrich, R. & Koziol, A. (2021). Enabling Consistency between Software Artefacts for Software Adaption and Evolution.

- In *2021 IEEE 18th International Conference on Software Architecture (ICSA)*.
o.O.: IEEE. <https://doi.org/10.1109/icsa51549.2021.00009>
- Moosbrugger, H. & Kelava, A. (2012). *Testtheorie und Fragebogenkonstruktion*
(SpringerLink Bücher, 2., aktualisierte und überarbeitete Auflage). Berlin,
Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-20072-4>
- Muschik, S. (2011). Development of Systems of Objectives in Early Product
Engineering, Entwicklung von Zielsystemen in der frühen Produktentstehung.
In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für
Produktentwicklung, Systeme, Methoden, Prozesse* (Bd. 50). Karlsruhe:
Karlsruher Institut für Technologie (KIT).
<https://doi.org/10.5445/IR/1000023768>
- Nassir, N., Tong, J., Lavieri, P., Sarvi, M., Sweatman, P., Ryan, S. et al. (2020,
Oktober). *Putting the Connectivity in C-ITS – Investigating pathways to
accelerate the uptake of road safety and efficiency technologies*. *Victoria Road
Safety Data Analysis* (ITS AUSTRALIA, Eds.). The University of Melbourne.
Zugriff am 26.11.23 unter: [https://cms.its-
australia.com.au/assets/images/PDFs/3-C-ITS-Crash-Data-Analysis-
Report.pdf](https://cms.its-australia.com.au/assets/images/PDFs/3-C-ITS-Crash-Data-Analysis-Report.pdf)
- Nationale Plattform Mobilität der Zukunft (Federal Ministry of Transport and Digital
Infrastructure (BMVI), Hrsg.). (2019). *Progress report 2019 of the National
Platform Future of Mobility*, Nationale Plattform Mobilität der Zukunft; NPM.
Zugriff am 26.11.23 unter: [https://www.plattform-zukunft-mobilitaet.de/wp-
content/uploads/2020/03/NPM-Progress-report-2019.pdf](https://www.plattform-zukunft-mobilitaet.de/wp-content/uploads/2020/03/NPM-Progress-report-2019.pdf)
- Nationale Plattform Mobilität der Zukunft. (2020). *HANDLUNGSEMPFEHLUNGEN
ZUR TYPGENEHMIGUNG UND ZERTIFIZIERUNG FÜR EINE VERNETZTE
UND AUTOMATISIERTE MOBILITÄT. ARBEITSGRUPPE 6 NORMUNG,
STANDARDISIERUNG, ZERTIFIZIERUNG UND TYPGENEHMIGUNG*,
Nationale Plattform Mobilität der Zukunft; NPM. Zugriff am 26.11.23 unter:
[https://www.plattform-zukunft-mobilitaet.de/wp-content/uploads/2020/06/NPM-
AG-6-Handlungsempfehlungen-zur-Typgehmigung-und-Zertifizierung-
f%C3%BCr-eine-vernetzte-und-automatisierte-Mobilit%C3%A4t.pdf](https://www.plattform-zukunft-mobilitaet.de/wp-content/uploads/2020/06/NPM-AG-6-Handlungsempfehlungen-zur-Typgehmigung-und-Zertifizierung-f%C3%BCr-eine-vernetzte-und-automatisierte-Mobilit%C3%A4t.pdf)
- Nelson, D. R., Adger, W. N. & Brown, K. (2007). Adaptation to Environmental
Change: Contributions of a Resilience Framework. *Annual Review of
Environment and Resources*, 32(1), 395–419.
<https://doi.org/10.1146/annurev.energy.32.051807.090348>

- Ngo, A., Bauer, M. P. & Resch, M. (2021). A Multi-Layered Approach for Measuring the Simulation-to-Reality Gap of Radar Perception for Autonomous Driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (pp. 4008–4014). IEEE.
<https://doi.org/10.1109/itsc48978.2021.9564521>
- NHTSA. (2022, 23. November). *Vehicle-to-Vehicle Communication | NHTSA*. Zugriff am 26.11.23 unter: <https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication#nhtsa-in-action>
- Nielsen, C. B., Larsen, P. G., Fitzgerald, J., Woodcock, J. & Peleska, J. (2015). Systems of Systems Engineering. *ACM Computing Surveys*, *48*(2), 1–41.
<https://doi.org/10.1145/2794381>
- No Magic Product Documentation. (2022, 24. November). *Introduction to Cameo Safety and Reliability Analyzer*. Zugriff am 26.11.23 unter: <https://docs.nomagic.com/display/NMDOC/No+Magic+Documentation>
- Object Management Group (Eds.). (2019). *OMG Systems Modeling Language™* (Version 1.6) (formal/2019-10-01). Zugriff am 26.11.23 unter: <https://www.omg.org/spec/SysML/1.6/>
- Object Management Group (Eds.). (2021). *Unified Architecture Framework Modeling Language (UAFML)* (Version 1.2) (Object Management Group, Eds.) (Normative dtc/2021-12-07). Zugriff am 26.11.23 unter: <https://www.omg.org/spec/UAF/1.2/Beta1/UAFML/PDF>
- Object Management Group (Eds.). (2022). *UAF Unified Architecture Framework. UAF SPECIFICATION AT OMG* (Version 1.2) (Object Management Group, Eds.) (Normative dtc/2022-12-07). Zugriff am 26.11.23 unter: <https://www.omg.org/spec/UAF/1.2/About-UAF>
- Object Management Group (Eds.) (2023): *OMG Systems Modeling Language™ (SysML). Part2: SysML v1 to SysML v2 Transformation*. Version 2.0 Beta 1. online. Zugriff am 26.11.23 unter: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/2b-SysML_v1_to_v2_Transformation.pdf
- Olstam, J. J., Lundgren, J., Adlers, M. & Matstoms, P. (2008). A framework for simulation of surrounding vehicles in driving simulators. *ACM Transactions on*

- Modeling and Computer Simulation*, 18(3), 1–24.
<https://doi.org/10.1145/1371574.1371575>
- Pegasus (Hrsg.). (o.J.). *Scenario Description. Requirements & Conditions – Stand 4*. Zugriff am 26.11.23 unter: https://www.pegasusprojekt.de/files/tmp/PDF-Symposium/04_Scenario-Description.pdf
- Pelliccione, P., Knauss, E., Ågren, S. M., Heldal, R., Bergenhem, C., Vinel, A. et al. (2020). Beyond connected cars: A systems of systems perspective. *Science of Computer Programming*, 191, 1–38.
<https://doi.org/10.1016/j.scico.2020.102414>
- Pohl, K., Hönninger, H., Achatz, R. & Broy, M. (2012). *Model-Based Engineering of Embedded Systems. The SPES 2020 Methodology*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-34614-9>
- Posner, B., Keller, A., Binz, H. & Roth, D. (2012). Anforderungen an eine Methode zum leichtbaugerechten Konstruieren. In F. Rieg, J. Feldhusen, R. Stelzer, K.-H. Grote & K. Brökel (eds.), *Beiträge zum 10. Gemeinsamen Kolloquium Konstruktionstechnik* (S. 537–548). Dresden: TUDpress; Sächsische Landesbibliothek.
- Postel, J. (1980). Telnet Protocol specification. o. O.: RFC Editor.
<https://doi.org/10.17487/rfc0764>
- Prinkle, S., Liu, H., Wang, H. & Zhang, S. (2017). Securing wireless communications of connected vehicles with artificial intelligence. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*. O.O.: IEEE. <https://doi.org/10.1109/ths.2017.7943477>
- Raj, S., Jha, S. K., Ramanathan, A. & Pullum, L. L. (2017). Testing autonomous cyber-physical systems using fuzzing features from convolutional neural networks. In *Proceedings of the Thirteenth ACM International Conference on Embedded Software 2017 Companion - EMSOFT '17* (n.p.). New York, NY, USA: ACM Press. <https://doi.org/10.1145/3125503.3125568>
- Rappaport, T. S. (2002). *Wireless communications. Wireless communications* (Prentice Hall communications engineering and emerging technologies series). Upper Saddle River, N.J: Prentice Hall.

- Rau, M. & Dojan, E. (2021, 30. November). *Navigating Through Standards Complexity: Side by Side ISO26262 and SOTIF Comparison. Functional Safety/ SOTIF & Safety for Automated Driving*. Referat in Serie von SGS-TÜV Saar GmbH. Zugriff am 26.11.23 unter:
https://www.youtube.com/watch?time_continue=27&v=Nk7CENQ0ilo&feature=emb_logo
- Redmill, F., Chudleigh, M. & Catmur, J. (1999). *System safety. HAZOP and software HAZOP*. *Computing & Control Engineering Journal*, 10(4), 140.
<https://doi.org/10.1049/cce:19990402>
- Ress, C. & Wiecker, M. (2016). Potenzial der V2X-Kommunikation für Verkehrssicherheit und Effizienz. *ATZ - Automobiltechnische Zeitschrift*, 118(1), 16–21. <https://doi.org/10.1007/s35148-015-0154-y>
- Richter, T., Rapp, S., Kurtz, V., Romanov, V. & Albers, A. . (2019). Creating innovative products with reference system elements - a case study on approaches in practice. *Procedia CIRP*, 84, 804–809.
<https://doi.org/10.1016/j.procir.2019.04.259>
- Richter, T., Witt, J.-H., Gesk, J. W. & Albers, A. . (2019a). Identification of requirements of methods and processes for modeling objectives in predevelopment projects. *Procedia CIRP*, 84, 419–427.
<https://doi.org/10.1016/j.procir.2019.04.203>
- Richter, T., Witt, J.-H., Gesk, J. W. & Albers, A. . (2019b). Systematic modeling of objectives and identification of reference system elements in a predevelopment project. *Procedia CIRP*, 84, 579–585. ISSN 2212-8271.
<https://doi.org/10.1016/j.procir.2019.04.258>
- Ropohl, G. (Hrsg.). (1975). *Systemtechnik - Grundlagen und Anwendung*. München: Carl Hanser Verlag. <https://doi.org/10.1002/cite.330471315>
- Ropohl, G. (2009). *Allgemeine Technologie – Eine Systemtheorie der Technik. Allgemeine Technologie – Eine Systemtheorie der Technik, 3., Überarbeitete Auflage*. Karlsruhe: Universitätsverlag Karlsruhe 2009. Zugriff am 26.11.23 unter: <http://digbib.uibk.uni-karlsruhe.de/volltexte/1000011529>
- Rösener, C., Hennecke, F., Sauerbier, J., Zlocki, A., Kemper, D., Oeser, M. et al. (2019). *BASt-Bericht F 128: Potenzieller gesellschaftlicher Nutzen durch*

zunehmende Fahrzeugautomatisierung (Berichte der Bundesanstalt für Straßenwesen, Heft F 128). Bericht zum Forschungsprojekt FE 82.0626/2015/. Bergisch Gladbach: Fachverlag NW in der Carl Schünemann Verlag GmbH.

SAE, J3016_202104 (30.04.2021). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International. https://doi.org/10.4271/J3016_202104

Schlager, K. J. (1956). Systems engineering-key to modern development. *IRE Transactions on Engineering Management, EM-3*(3), 64–66. <https://doi.org/10.1109/IRET-EM.1956.5007383>

Schmitz, K. & Hoppe, W.-D. (2021). *The Future of Automotive Mobility. Uncertain drivers take global automotive markets to a crossroads*. Global Automotive Mobility Study - 3rd Edition. Arthur D. Little. Zugriff am 26.11.23 unter: https://www.adlittle.de/sites/default/files/reports/ADL_The_future_of_automotive_mobility.pdf

Schnieder, L. & Hosse, R. S. (2019). *Leitfaden Safety of the Intended Functionality*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-25023-2>

SEBoK. (2017). *Guide to the systems engineering body of knowledge. Systems of Systems (SoS)*. SEBoK v. 2.7.,. Zugriff am 26.11.23 unter: [http://www.sebokwiki.org/w/index.php?title=Systems_of_Systems_\(SoS\)&oldid=52771](http://www.sebokwiki.org/w/index.php?title=Systems_of_Systems_(SoS)&oldid=52771)

Shalev-Shwartz, S., Shammah, S. & Shashua, A. (2017). *On a Formal Model of Safe and Scalable Self-driving Cars* (ArXiv, Hrsg.). Zugriff am 26.11.23 unter: <https://arxiv.org/pdf/1708.06374.pdf>

Shenzhen Fama Intelligent Equipment Co., Ltd. (2022, 28. November). *des Ertrag-32W legierter Stahl Shell Vernetzungs-Ampel-des Prüfer-AC110V/220V*. Zugriff am 26.11.23 unter: <https://german.ledtrafficleighting.com/sale-11606860-32w-output-networking-traffic-light-controller-ac110v-220v-alloy-steel-shell.html>

Siemens Industry, Inc. (Hrsg.). (2018). *Connected Vehicle Roadside Unit (RSU)*. USA. Zugriff am 26.11.23 unter: <https://assets.new.siemens.com/siemens/assets/api/uuid:eeb2cf21-ca44-41df->

8ea8-bdab82ee140e/operation:download/Connected-Vehicle-Roadside-Unit-Datasheet.pdf

- Sikora, E., Tenbergen, B. & Pohl, K. (2012). Industry needs and research directions in requirements engineering for embedded systems. *Requirements Engineering*, 17(1), 57–78. <https://doi.org/10.1007/s00766-011-0144-x>
- Silingas, D. & Butleris, R. (2009). Towards implementing a framework for modeling software requirements in MagicDraw UML. *Information technology and control*, 38, 153–164.
- Smirnov, A., Kashevnik, A., Mikhailov, S., Mironov, M. & Petrov, M. (2016). Smart M3-Based Robot Interaction Scenario for Coalition Work. In *Lecture Notes in Computer Science* (S. 199–207). Springer International Publishing. https://doi.org/10.1007/978-3-319-43955-6_24
- Stadt Nürnberg (Hrsg.). (2021). *Zustandsbericht Lichtsignalanlagen 2021*. Zugriff am 26.11.23 unter: https://www.nuernberg.de/imperia/md/soer_nbg/dokumente/grau/lsa_zustandsbericht_2021.pdf
- Stähle GmbH (Hrsg.). (2015). *Autopilot System SAP2000. zum computergesteuerten Fahren auf Rollenprüfständen*. Zugriff am 26.11.23 unter: <https://www.staehle-robots.com/german/produkte/sap2000/>
- Steimle, M., Weber, N. & Maurer, M. (2022). Toward Generating Sufficiently Valid Test Case Results: A Method for Systematically Assigning Test Cases to Test Bench Configurations in a Scenario-Based Test Approach for Automated Vehicles. *IEEE Access*, 10, 6260–6285. <https://doi.org/10.1109/access.2022.3141198>
- Stellet, J. E., Zofka, M. R., Schumacher, J., Schamm, T., Niewels, F. & Zollner, J. M. (2015). Testing of Advanced Driver Assistance Towards Automated Driving: A Survey and Taxonomy on Existing Approaches and Open Questions. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*. 15 - 18 Sept. 2015, Las Palmas, Gran Canaria, Spain (pp. 1455–1462). Piscataway, NJ: IEEE.

- Stevanovic, A., Stevanovic, J. & Kergaye, C. (2013). Green Light Optimized Speed Advisory Systems. *Transportation Research Record: Journal of the Transportation Research Board*, 2390, 53–59. <https://doi.org/10.3141/2390-06>
- Sulaman, S. M., Beer, A., Felderer, M. & Höst, M. (2019). Comparison of the FMEA and STPA safety analysis methods—a case study. *Software Quality Journal*, 27(1), 349–387. <https://doi.org/10.1007/s11219-017-9396-0>
- Takeichi, M., Sato, Y., Suyama, K. & Kawahara, T. (2011). Failure rate calculation with priority FTA method for functional safety of complex automotive subsystems. In *2011 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering* (pp. 55–58). IEEE. <https://doi.org/10.1109/icqr2mse.2011.5976568>
- Tan, J. & Wang, L.. (2010). Flexibility–efficiency tradeoff and performance implications among Chinese SOEs. *Journal of Business Research*, 63(4), 356–362. <https://doi.org/10.1016/j.jbusres.2009.04.016>
- Thaler, F. & Wäschle, M. (2023, 6. März). *AI Safety in highly automated driving A review*. Referat im Rahmen des FuSaCom G2G Event. Zugriff am 26.11.23 unter: https://www.youtube.com/watch?v=9FHQVr9z61E&ab_channel=VIRTUALVEHICLE
- Themann, P., Raudszus, D., Zlocki, A. & Eckstein, L. (2016). Ganzheitliche Bewertung des vernetzten und automatisierten Fahrens. *ATZ - Automobiltechnische Zeitschrift*, 118(1), 26–31. <https://doi.org/10.1007/s35148-015-0159-6>
- Thiernel, A. (2020, Juni). *Welche Hersteller bieten bereits C2X an? ADAC-Umfrage 6/2020* (ADAC e.V., Hrsg.). München. Zugriff am 26.11.23 unter: https://assets.adac.de/image/upload/v1595919606/ADAC-eV/KOR/Text/PDF/Umfrage_Hersteller_Car2X_dl45xm.pdf
- Thorn, E., Kimmel Shawn & Chaka, M. (2018, September). *A Framework for Automated Driving System Testable Cases and Scenarios* (National Highway Traffic Safety Administration, Eds.) (DOT HS 812 623). Washington DC. Zugriff am 26.11.23 unter: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13882-automateddrivingsystems_092618_v1a_tag.pdf

- U.S. Department of Transportation - federal aviation administration (Eds.). (2017). *Simple and Complex Electronic Hardware Approval Guidance (DO-254)*. Zugriff am 26.11.23 unter: https://www.faa.gov/documentLibrary/media/Order/FAA_Order_8110_105A.pdf
- Ulrich, K. T. & Eppinger, S. D. (2012). *Product design and development* (5. ed.). New York, NY: McGraw-Hill.
- Underwriters' Laboratories. (2020). *UL 4600: Standard for Evaluation of Autonomous Products* (Standard for safety). Underwriters Laboratories. Zugriff am 26.11.23 unter: <https://books.google.de/books?id=7fH3zQEACAAJ>
- United Nations Economic and Social Council. (2022). *Guidelines and Recommendations concerning Safety Requirements for Automated Driving Systems (WP.29-187-10)*. Zugriff am 26.11.23 unter: <https://unece.org/sites/default/files/2022-06/WP.29-187-10e.pdf>
- Van Heesch, D. (Eds.). (2016, 4. Oktober). *A Discrete-Event Network Simulator API. ns-3: src/wave/examples/wave-simple-80211p.cc File Reference*. Zugriff am 26.11.23 unter: https://www.nsnam.org/docs/release/3.26/doxygen/wave-simple-80211p_8cc.html
- VDA QMC Working Group 13 / Automotive SIG. (2017). *Automotive SPICE Process Assessment / Reference Model*. Version 3.1. Zugriff am 26.11.23 unter: http://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_PAM_31.pdf
- VDI, 2221 Blatt 1 (2019). *Entwicklung technischer Produkte und Systeme - Modell der Produktentwicklung*. Berlin: Beuth Verlag GmbH.
- VDI/VDE 2206 (2021-11-00). *Entwicklung mechatronischer und cyber-physischer Systeme*. Berlin: Beuth Verlag GmbH.
- Vereinte Nationen. (2016). *Ziele für nachhaltige Entwicklung. Bericht 2016*. Hauptabteilung Wirtschaftliche und Soziale Angelegenheiten (DESA) (Hrsg.). Zugriff am 26.11.23 unter: <https://www.un.org/depts/german/millennium/SDG%20Bericht%202016.pdf>
- Vierhauser, M., Grünbacher, P., Egyed, A., Rabiser, R. & Heider, W. (2010). Flexible and scalable consistency checking on product line variability models.

In *Proceedings of the IEEE/ACM international conference on Automated software engineering - ASE '10* (pp. 63–72). New York, USA: ACM Press.
<https://doi.org/10.1145/1858996.1859009>

- Villarreal-Vasquez, M., Bhargava, B. & Angin, P. (2017). Adaptable Safety and Security in V2X Systems. In *2017 IEEE International Congress on Internet of Things (ICIOT)*. o.O.: IEEE. <https://doi.org/10.1109/ieeee.iciot.2017.12>
- Volkswagen AG (Hrsg.). (2020). *Technical milestone in road safety: experts praise Volkswagen's Car2X technology*. Press Release. Wolfsburg. Zugriff am 26.11.23 unter: <https://www.volkswagen-newsroom.com/en/press-releases/technical-milestone-in-road-safety-experts-praise-volkswagens-car2x-technology-5914>
- Walter, B., Albers, A., Haupt, F. & Bursac, N. (2016). Produktentwicklung im virtuellen Ideenlabor – Konzipierung und Implementierung eines Live-Lab. In D. Kraus & K. P. Dieter Krause (Hrsg.), *Design for X : Beiträge zum 27. DfX-Symposium Oktober 2016* (283--295). TuTech Innovation Verlag.
- Wäschle, M., Behrendt, M., Xing, K., Shi, H. & Albers, A. (2021). Contract-based methods and activities in the validation of interfaces for System of Systems. In *16th International Conference of System of Systems Engineering (SoSE) (SoSE 2021)* (pp. 102–107). virtual: IEEE.
<https://doi.org/10.1109/sose52739.2021.9497476>
- Wäschle, M., Imamovic, A., Rapp, S. & Albers, A. (2022). Model-Based Systems Engineering bei der effektiven Schnittstellengestaltung eines Serious Games als System of Systems. In W. Koch, D. Wilke, S. Dreiseitel & R. Kaffenberger (Hrsg.), *Tag des Systems Engineering 2022* (S. 36–40). München: Carl Hanser Verlag.
- Wäschle, M., Jiang, Y., Behrendt, M. & Albers, A. (2020). Development of an integrated validation environment for System of Systems in the context of V2X using the XiL-Approach. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*. o. O.: IEEE.
<https://doi.org/10.1109/sose50414.2020.9130509>
- Wäschle, M., Martin, A., Radimersky, A., Behrendt, M. & Albers, A. . (2020). SUPPORTING THE MODELLING IN MBSE BY APPLYING PRODUCT GENERATION ENGINEERING USING ELECTRONIC COMPACT

ACTUATORS AS AN EXAMPLE. *Proceedings of the Design Society: DESIGN Conference, 1*, 2425–2434. <https://doi.org/10.1017/dsd.2020.293>

Wäschle, M., Thaler, F., Berres, A., Pözlbauer, F. & Albers, A. . (2022). A review on AI Safety in highly automated driving. *Frontiers in Artificial Intelligence*, 5. <https://doi.org/10.3389/frai.2022.952773>

Wäschle, M., Timucin, M., Radimersky, A., Mandel, C., Fahl, J. & Hirschter, T. (2021). Vorgehen zur systematischen Verknüpfung von Produkthanforderungen und Systemarchitektur unter Berücksichtigung der PGE. In Binz, Hansgeorg, Bertsche, Bernd, D. Spath & D. Roth (Hrsg.), *Stuttgarter Symposium für Produktentwicklung SSP 2021 : Stuttgart, 20. Mai 2021, Wissenschaftliche Konferenz* (Stuttgarter Symposium für Produktentwicklung, Bd. 6, 199-210). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.

Wäschle, M., Wang, X. [Xi], Xinlu, C. & Albers, A. (2022). Validation of Vehicle-to-Infrastructure scenarios based on the X-in-the-Loop-approach. In *2022 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*. IEEE. <https://doi.org/10.1109/metroautomotive54295.2022.9855027>

Wäschle, M., Wolter, K., Bause, K. & Behrendt, M. (Euroforum Deutschland GmbH, Hrsg.). (2020). *A new safety oriented approach in the validation of highly automated electric vehicle subsystems*. CTI Symposium World Series. Zugriff am 26.11.23 unter: <https://drivetrain-symposium.world/a-new-safety-oriented-approach-in-the-validation-of-highly-automated-electric-vehicle-subsystems/>

Wäschle, M., Wolter, K., Bause, K., Behrendt, M. & Albers, A. (2022). Considering Functional Safety - supporting the development of automated driving vehicles by the use of Model-Based Systems Engineering. In *2022 17th Annual System of Systems Engineering Conference (SOSE)* (pp. 275–280). IEEE. <https://doi.org/10.1109/sose55472.2022.9812686>

Wäschle, M., Wolter, K., Han, C., Pecha, U., Bause, K. & Behrendt, M. (2021). Validation concept for scenario-based connected test benches of a highly automated vehicle. In T. Bertram (Hrsg.), *Automatisiertes Fahren 2021. Vom assistierten zum autonomen Fahren 7. Internationale ATZ-Fachtagung* (1. Aufl., S. 95–109). Wiesbaden: Springer Vieweg Wiesbaden. https://doi.org/10.1007/978-3-658-34754-3_8

- Weilkiens, T. (2015). *SYSMOD - The Systems Modeling Toolbox. - Pragmatic MBSE with SysML* - (MBSE4U Booklet Series, Version 4.0). Fredesdorf: MBSE4U. Zugriff am 26.11.23 unter: <https://mbse4u.com/books/sysmod-the-systems-modeling-toolbox-pragmatic-mbse-with-sysml/>
- Weilkiens, T. (2016). *Variant Modeling with SysML* (MBSE4U Booklet Series). MBSE4U - Tim Weilkiens. Zugriff am 26.11.23 unter: <https://mbse4u.com/2012/05/07/variant-modeling-with-sysml/>.
- Weilkiens, T., Berres, A., Endler, D., Haarer, A., Lalitsch-Schneider, C., Krammer, M. et al. (2015). System Safety in SysML. In S.-O. Schulze & C. Tschirmer (Hrsg.), *Tag des Systems Engineering* (S. 343–354). München: Carl Hanser Verlag GmbH & Co. KG. <https://doi.org/10.3139/9783446447288.034>
- Weilkiens, T. & Soley, R. M. (2014). *Systems Engineering mit SysML/UML. Anforderungen, Analyse, Architektur* (3., überarb. und aktualisierte Aufl.). Heidelberg: dpunkt.verl.
- Wiedner, A. (2013). Feldstudie zur Identifikation der von Konstrukteuren praktizierten Handlungsmuster bei der Funktion-Gestalt-Synthese. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung, Systeme, Methoden, Prozesse* (Bd. 65). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000036525>
- Wilking, F., Sauer, C., Schleich, B. & Wartzack, S. (2022). Integrating Machine Learning in Digital Twins by utilizing SysML System Models. In *2022 17th Annual System of Systems Engineering Conference (SOSE)* (pp. 297–302). IEEE. <https://doi.org/10.1109/sose55472.2022.9812700>
- Wood, M., Robbel, P., Knobel, C., Boymanns, D., Garbacik, N., O'Brien, M. et al. (2019). *Safety First for Automated Driving* (Aptiv Services US, L. L., AUDI AG, Bayerische Motoren Werke AG, Beijing Baidu Netcom Science Technology Co., Ltd & Continental, Eds.). Zugriff am 26.11.23 unter: <https://www.daimler.com/documents/innovation/other/safety-first-for-automated-driving.pdf>
- Yan, S. L. (2020). Vernetzte Validierungsumgebungen – Ein Beitrag zur Validierung im verteilten Produktentwicklungsumfeld auf Basis des IPEK-X-in-the-Loop-Ansatzes am Beispiel der Antriebssystementwicklung. Dissertation.

In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 130). Karlsruhe: Karlsruher Institut für Technologie (KIT).
<https://doi.org/10.5445/IR/1000126974>

Yan, S. L., Mandel, C., Behrendt, M. & Albers, A. (2019). Dokumentation von Erfahrungswissen in der Entwicklung von vernetzten Validierungsumgebungen. In H. Binz, B. Bertsche, W. Bauer, O. Riedel, D. Spath & D. Roth (Hrsg.), *Stuttgarter Symposium für Produktentwicklung*. ISSN 2364-4885 (S. 173–182). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.

Yang, F., Wang, F., Ding, F., Tan, H. & Ran, B. (2021). Identify Optimal Traffic Condition and Speed Limit for Hard Shoulder Running Strategy. *Sustainability*, 13(4), 1822. <https://doi.org/10.3390/su13041822>

Zhang, W., Chen, Y., Yang, Y., Wang, X. [Xiangyang], Zhang, Y., Hong, X. et al. (2012). Multi-Hop Connectivity Probability in Infrastructure-Based Vehicular Networks. *IEEE Journal on Selected Areas in Communications*, 30(4), 740–747. <https://doi.org/10.1109/jsac.2012.120508>

Zingel, C. (2013). Basisdefinition einer gemeinsamen Sprache der Produktentwicklung im Kontext der Modellbildung technischer Systeme und einer Modellierungstechnik für Zielsystem und Objektsystem technischer Systeme in SysML auf Grundlage des ZHO-Prinzips. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 70). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000037421>

Studentische Abschlussarbeiten, die im Kontext dieser Dissertation am IPEK – Institut für Produktentwicklung am Karlsruher Institut für Technologie (KIT) vom Autor co-betreut wurden:

Chai, T. (2020). Investigation of test automation using model-based system engineering for the validation of V2X. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.

Chen, S. (2020). Modellbasierter Ansatz in der Entwicklung einer Validierungsumgebung für V2X. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.

- Chen, X. (2022). Development of a virtual validation environment in the context of the IPEK-X-in-the-loop approach for efficient validation of Vehicle-to-Infrastructure. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Di, K. (2020). Identifikation und Analyse von Einflussfaktoren in der Validierung von V2X basierend auf dem IPEK-X-in-the-Loop-Ansatz. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Klein, S. (2022). Systematische Modellierung zur effizienten Validierung von V2I auf Basis des IPEK-XiL Ansatzes. Unveröffentlichte Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Kozok, M. (2020). Modellierung von Abhängigkeiten in einer Validierungsumgebung für hochautomatisierte elektrische Fahrzeuge mit Hilfe von MBSE-Ansätzen. Unveröffentlichte Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Martin, A. (2019). Entwicklung eines SysML-Modells eines Electronic Compact Actuator (ECA) zur Unterstützung der modellbasierten Systementwicklung im Kontext der Produktgenerationsentwicklung und Ableitung von Handlungsempfehlungen für die Modellbildung. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Mehmedagic, M. (2019). Potenziale der Baukastenentwicklung in der Entwicklung von System-of-Systems am Beispiel Connected Car. Unveröffentlichte Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Metzger, Y. (2019). Untersuchung und Bewertung von Fahrerassistenzsystemen im Rennsport. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Qi, K. (2019). Entwicklung einer neuen Generation einer Validierungsumgebung mit mehreren Informationsquellen für V2X. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Shi, H. (2020). Entwicklung einer V2X-spezifischen Software- Umgebung im Kontext der X-in-the-Loop-Validierung. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.

- Sternke, F. (2019, 14. Juni). Simulative Analyse von Wechselwirkungen zwischen Kunde, Elektrofahrzeug und Energieversorgung. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Timucin, M. (2020). Entwicklung eines Vorgehens zur systematischen Verknüpfung von Produktanforderungen und Systemarchitektur unter Berücksichtigung der PGE am Beispiel ECA. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Xi, W. (2022). Testfallgetriebene Validierung hochautomatisierter Fahrzeuge auf Basis von V2I-Szenarien. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Xing, K. (2020). Schnittstellenbetrachtung von System of Systems am Beispiel einer V2X Validierungsumgebung. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Yi, J. (2019). Entwicklung einer Validierungsumgebung für System-of-systems im Kontext von V2X unter Verwendung des IPEK-XiL-Ansatzes. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.
- Zeng, B. (2020). Entwicklung und Implementierung eines durchgängigen Model-Based Engineering Ansatzes für die Validierung technischer Systeme am Beispiel Fahrerassistenzsysteme. Unveröffentlichte Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.

Studentische Abschlussarbeiten, die im Kontext dieser Dissertation zitiert worden:

- Eberle, L. (2019). Weiterentwicklung einer Validierungsumgebung für die Entwicklung von kamerabasierten Fahrerassistenzsystemen zur Anwendung an verteilten Standorten. Unveröffentlichte Bachelorarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.

Publikationsliste

- Fleischer, J. & Albers, A. [Albert] & Anderl, R. & Aurich, J. & Giese, T. & Glatt, M. & Gönnheimer, P. & Litsche, S. & Mühlbeier, E. & Schlagenhaut, T. & Steglich, S. & Wang, Y. & Wäschle, M. (2021). 5G in der Industrie – Wege in die Technologieführerschaft in Produktentwicklung und Produktion. In Fleischer, J. / Albers, A. /Anderl, R./Aurich, J. (Hrsg.), *acatech IMPULS*, München.
https://doi.org/10.48669/aca_2021-2
- Humpert, L., Wäschle, M., Horstmeyer, S., Dumitrescu, R. & Albers, A. [Albert] (2023). Stakeholder-oriented Elaboration of Artificial Intelligence use cases using the example of Special-Purpose engineering. *Procedia CIRP*, 119, 693–698, o.O.. <https://doi.org/10.1016/j.procir.2023.02.160>.
- Kaiser, L.; Willich, F.; Waschle, M.; Schneeloch, J.; Ardeo, L.F. (2021). Doctoral workshop 'engineering smart cities': A report on a new conference element. In *IEEE European Technology and Engineering Management Summit, E-TEMS 2021 - Conference Proceedings*, 5–9, o.O.: IEEE. <https://doi.org/10.1109/E-TEMS51171.2021.9524895>.
- Kharatyan, A., Humpert, L., Anacker, H., Dumitrescu, R., Wäschle, M., Albers, A. [Albert] et al. (2022). Künstliche Intelligenz im Engineering: Menschorientierte Analyse von Potenzialen am Beispiel vom Sondermaschinenbau. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 117(6), 427–431. Berlin: Walter de Gruyter GmbH. <https://doi.org/10.1515/zwf-2022-1074>
- Mandel, C., Wäschle, M., Lutz, S. & Behrendt, M. (2021). A Model-Based approach for automation and traceability of validation activities – clarified for Advanced Driver Assistance Systems. In *9th Int. Symposium on Development Methodology*. Wiesbaden. doi:10.5445/IR/1000142322.
- Pecha, U., Parspour, N., Wolter, K., Wäschle, M. & Bause, K. (2022). Validation of Fault-Tolerant Control of Converters under Open-Switch Faults on Connected Test Benches. In *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*. o.O.: IEEE.
<https://doi.org/10.1109/IECON49645.2022.9968422>

Thaler, F. & Wäschle, M. (2023, 6. März). *AI Safety in highly automated driving A review*. Referat im Rahmen des FuSaCom G2G Event (o.O.). Zugriff am 26.11.23 unter:

https://www.youtube.com/watch?v=9FHQVr9z61E&ab_channel=VIRTUALVEHICLE

Wäschle, M., Behrendt, M., Xing, K., Shi, H. & Albers, A. [Albert] (2021). Contract-based methods and activities in the validation of interfaces for System of Systems. In *16th International Conference of System of Systems Engineering (SoSE) (SoSE 2021)* (pp. 102–107), o.O.: IEEE.
<https://doi.org/10.1109/sose52739.2021.9497476>

Wäschle, M., Imamovic, A., Rapp, S. & Albers, A. [Albert] (2022). Model-Based Systems Engineering bei der effektiven Schnittstellengestaltung eines Serious Games als System of Systems. In W. Koch, D. Wilke, S. Dreiseitel & R. Kaffenberger (Hrsg.), *Tag des Systems Engineering 2022* (S. 36–40). München: Carl Hanser Verlag.

Wäschle, M., Jiang, Y., Behrendt, M. & Albers, A. [Albert] (2020). Development of an integrated validation environment for System of Systems in the context of V2X using the XiL-Approach. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*. o. O.: IEEE.
<https://doi.org/10.1109/sose50414.2020.9130509>

Wäschle, M., Martin, A., Radimersky, A., Behrendt, M. & Albers, A. [Albert]. (2020). SUPPORTING THE MODELLING IN MBSE BY APPLYING PRODUCT GENERATION ENGINEERING USING ELECTRONIC COMPACT ACTUATORS AS AN EXAMPLE. *Proceedings of the Design Society: DESIGN Conference, 1*, 2425–2434, o.O.. <https://doi.org/10.1017/dsd.2020.293>

Wäschle, M., Thaler, F., Berres, A., Pözlbauer, F. & Albers, A. [Albert]. (2022). A review on AI Safety in highly automated driving. *Frontiers in Artificial Intelligence, 5*. <https://doi.org/10.3389/frai.2022.952773>

Wäschle, M., Timucin, M., Radimersky, A., Mandel, C., Fahl, J. & Hirschter, T. (2021). Vorgehen zur systematischen Verknüpfung von Produktanforderungen und Systemarchitektur unter Berücksichtigung der PGE. In Binz, H., Bertsche, B., Spath, D. & Roth, D. (Hrsg.), *Stuttgarter Symposium für Produktentwicklung SSP 2021 : Stuttgart, 20. Mai 2021, Wissenschaftliche Konferenz* (Stuttgarter Symposium für Produktentwicklung, Bd. 6, 199-210). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.

- Wäschle, M., Wang, X. [Xi], Xinlu, C. & Albers, A. [Albert] (2022). Validation of Vehicle-to-Infrastructure scenarios based on the X-in-the-Loop-approach. In *2022 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*, o.O.: IEEE.
<https://doi.org/10.1109/metroautomotive54295.2022.9855027>
- Wäschle, M., Wolter, K., Bause, K. & Behrendt, M. (Euroforum Deutschland GmbH, Hrsg.). (2020). *A new safety oriented approach in the validation of highly automated electric vehicle subsystems*. CTI Symposium World Series. Zugriff am 26.11.23 unter: <https://drivetrain-symposium.world/a-new-safety-oriented-approach-in-the-validation-of-highly-automated-electric-vehicle-subsystems/>
- Wäschle, M., Wolter, K., Bause, K., Behrendt, M. & Albers, A. [Albert] (2022). Considering Functional Safety - supporting the development of automated driving vehicles by the use of Model-Based Systems Engineering. In *2022 17th Annual System of Systems Engineering Conference (SOSE)* (pp. 275–280), o.O.: IEEE. <https://doi.org/10.1109/sose55472.2022.9812686>
- Wäschle, M., Wolter, K., Han, C., Pecha, U., Bause, K. & Behrendt, M. (2021). Validation concept for scenario-based connected test benches of a highly automated vehicle. In T. Bertram (Hrsg.), *Automatisiertes Fahren 2021. Vom assistierten zum autonomen Fahren 7. Internationale ATZ-Fachtagung* (1. Aufl., S. 95–109). Wiesbaden: Springer Vieweg Wiesbaden.
https://doi.org/10.1007/978-3-658-34754-3_8

Glossar

Begriff	Definition
Produktentstehung	Die Produktentstehung ist ein Teil des Produktlebenszyklus und beschreibt den grundsätzlichen Ablauf von der Produkt- bzw. Geschäftsidee bis zum Serienanlauf. Sie umfasst die drei Hauptaufgabenbereiche strategische Produktplanung, Produktentwicklung und Produktionssystementwicklung. (Albers & Gausemeier, 2012, S. 18)
Produktentwicklung	Die Produktentwicklung ist ein „interdisziplinärer Unternehmensprozess zur Entwicklung eines marktfähigen Produkts, basierend auf der Definition initialer Ziele und Anforderungen an das Produkt, die im Lauf des Prozesses kontinuierlich weiterentwickelt und iterativ angepasst werden. (VDI 2221 Blatt 1, S. 8)
System	Das System ist das Modell einer Ganzheit, die (a) Beziehungen zwischen Attributen (In-/Outputs, Zustände) aufweist (b) Aus miteinander verknüpften Teilen bzw. Subsystemen besteht (c) Von ihrer Umgebung bzw. einem Supersystem abgegrenzt wird. (Ropohl, 2009)
System of Systems	Das System of Systems (SoS) ist ein System, bestehend aus einer Menge interagierender Systeme, von denen jedes einzelne System in sich selbst als System betrachtet werden kann. Die wichtigsten Charakteristika von SoS sind: <ul style="list-style-type: none">• Jedes System kann unabhängig agieren und einen eigenen Zweck besitzen.• Die individuellen Systeme der Menge werden unabhängig organisiert, um ihre Zwecke zu erfüllen.• Die Systemkombination liefert Ergebnisse, die von einzelnen Systemen nicht erreicht werden können. Die Charakteristika können unterschiedlich stark ausgeprägt sein. (angelehnt an BS ISO/IEC/IEEE 21841:2019)
Verwendete Begriffe in der Sprache SysML	
Anforderung	Die Anforderung beschreibt das System mit Werten oder einem Wertebereich. Anforderungen können Ziele konkretisieren jedoch nicht ersetzen. (Lohmeyer, 2013) Die Anforderung muss Betriebsart, Fehlertoleranz, Zeitintervall, sichere Zustände, Notbetriebszeitintervall und funktionale Redundanzen enthalten (ISO 26262-9:2018(E)).
Anwendungsfall / UseCase	Der Anwendungsfall ist ein Nutzungsszenario für ein System, bei dem die Wechselwirkung mit dem Anwender modelliert wird. Er enthält eine Menge an Funktionen eines Systems (angelehnt an Weilkens, 2015; Zingel, 2013).
Block	Der Block ist ein Element der SysML, das den UML-Begriff der Klasse ergänzt. Der Block hat mehrere Eigenschaften wie Parts & Values und gilt als Basiseinheit der SysML. (Vgl. Delligatti, 2014)
Connector	Der Connector ist ein Verknüpfungspunkt für Schnittstellen in einem Internen Blockdefinitionsdiagramm. (Vgl. Zingel, 2013)

Eigenschaften / Property	Die Eigenschaft ist Teil eines Systemelements. (Weilkiens & Soley, 2014)
Funktion	Die Funktion ist die Transformation von Input zu Output. (Albers & Wintergerst, 2014)
Generalisierung	Die Generalisierung zeigt die Vererbung zwischen zwei Systemelementen. (Weilkiens & Soley, 2014)
Item Flow	Fluss von Energie, Material und/oder Information zwischen Elementen. (Angelehnt an Weilkiens & Soley, 2014)
Klasse	Element der UML, das eine Menge von Elementen mit gleichartigen Eigenschaften beschreibt (Zingel, 2013).
Komposition	Element B kann nicht ohne A existieren.
Paket	Das Paket gruppiert Systemelemente. (Weilkiens & Soley, 2014)
Part (Property)	Repräsentiert einen Teil eines Blocks mit Schnittstellen zu weiteren Parts. (Angelehnt an Zingel, 2013)
(Proxy) Port	Der Port ist ein Interaktionspunkt zwischen Systemelementen wie Blöcken und der Umwelt. Port ist eine spezielle Eigenschaft einer Klasse. Der Proxyport stellt Verhalten und Struktur von innen nach außen bereit. (Weilkiens & Soley, 2014)
Profil	Erweiterungsmöglichkeit der UML und SysML. Enthält eine Menge von Stereotypen (Zingel, 2013). Es kann auf ein Paket angewendet werden (Weilkiens & Soley, 2014).
Satisfy	Ein Systemelement erfüllt eine Anforderung. (Weilkiens & Soley, 2014)
Schnittstelle	Ein Interaktionsbereich zwischen zwei Systemen (Zingel, 2013).
Sicht / View	Die Sicht ist eine Darstellung des Systems aus einer bestimmten Perspektive, welche unnötige Dinge weglässt. (Alt, 2012)
Stereotyp	Der Stereotyp ergänzt ein Modellelement um Eigenschaften und Semantik. So kann mit geringem Aufwand neue Eigenschaften definiert werden ohne ein neues Modell definieren zu müssen. (Zingel, 2013)
Viewpoint	Der Viewpoint konkretisiert die Sicht durch spezifische Regeln oder Methoden. (Alt, 2012)

Anhang

A: Übersicht der Aktivitäten in ISO 26262 & ISO/PAS 21448

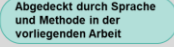
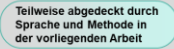
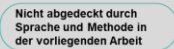

Methode der Funktionalen Sicherheit in der Produktentwicklung auf Systemlevel		Legende    
1. Generelle Themen	Erstellen eines Überblicks	
2. Technisches Sicherheitskonzept	Identifizieren von Anforderungen	
3. System-Integration & testen	Erstellen von Integrations- & Teststrategie	
4. Sicherheitsvalidierung	Validieren der Anforderungen	
Methode der Funktionalen Sicherheit in der Produktentwicklung auf Hardwarelevel		
1. Spezifikation Hardware Anforder.	Spezifizieren der Anforderungen	
2. Hardware Design	Spezifizieren von 1.	
3. Evaluation Hardware Metriken	Verifizieren der Effektivität des Metamodells	
4. Evaluation Anforderungen	Verifizieren Restrisiko für zufällige Hardwarefehler	
4. Hardware Integration & Verifikation	Verifizieren Einhaltung von Hardware zu Anf.	
Methode der Funktionalen Sicherheit in der Produktentwicklung auf Softwarelevel		
1. Generelle Themen Software	Überblick geben Prozess und Umgebung	
2. Spezifikation Software Anf.	Spezifizieren von Anf., Funktionen	
3. Software Metamodell	Erstellen, verifizieren von Software Metamodell	
4. Unit Design & Implementierung	Erstellen eines Unit Designs	
5. Unit Verifikation	Verifizieren des Unit Designs & Sicherheitsmaß.	
6. Integration & Verifikation	Integrieren der Elemente	
7. Test eingebettete Software	Verifizieren eingebettete Software	
Methode der Funktionalen Sicherheit ASIL und sicherheitsorientierte Analysen		
1. Anf. Dekomposition nach ASIL	Überprüfen der Dekomposition nach ASIL	
2. Kriterien für Koexistenz	Anwenden von Kriterien für Koexistenz von Design	
3. Analyse von abh. Fehlern	Überprüfen Elemente unabh. von Interferenzen	
4. Sicherheitsanalyse	Sicherstellen dass Risiko von zufalls- und systemischem Fehler niedrig genug ist	
Methode nach SOTIF		
1. Funktionale- und Systemspezifikation		
2. Gefahrenidentifikation und Risikoevaluation	Identifizieren von nicht-sichere, unbekannte Gefahren	
3. Identifikation von auslösenden Situationen!	Identifizieren der Situationen wie Schnee auf der Straße	
4. Funktionale Modifikation und Spezifikationen	Erstellen von verbesserten Funktionen	
5. Definition Verifikations- & Validierungsstrategie	Erstellen von Testfällen in (un-)bekannten, unsicheren Szenarien	
6. Verifikation von SOTIF	Verifizieren der neuen Funktion- und Systemspezifikation	
7. Validierung von SOTIF	Validieren mit Szenarienkatalog und Dauertests	

Abbildung A1: Übersicht der Aktivitäten in ISO 26262 & ISO/PAS 21448

B: Übersicht der eingesetzten Koppelsysteme

Tabelle B.1: Übersicht der Funktionen einzelner Koppelsysteme

KS / Funktion	Funktion realisiert mit W-LAN	Funktion realisiert mit LAN
KSpv: Router	senden und empfangen von Signalen über WiFi 5	senden und empfangen von Signalen über Ethernet
KSpv: CAN-zu-UDP	MCP2515 liest die Geschwindigkeits-CAN-Daten vom CAN-Bus des Testfahrzeugs. ESP8266 wandelt die CAN-Daten in UDP-Daten um und überträgt sie dann über Wi-Fi an den Router.	MCP2515 liest die Geschwindigkeits-CAN-Daten vom CAN-Bus des Testfahrzeugs. Arduino Mega+ Ethernet Shield wandelt die CAN-Daten in UDP-Daten um und überträgt sie dann über Ethernet an den Router.
KSvp: UDP-zu-CAN	ESP8266 empfängt UDP-Daten über Wi-Fi 5 und wandelt die UDP-Daten in CAN-Daten um. Dann liest MCP2515 die CAN-Daten und sendet sie an den CAN-Bus im Testfahrzeug.	Arduino Mega+ Ethernet Shield empfängt UDP-Daten über Ethernet und wandelt sie in CAN-Daten um. Dann liest MCP2515 die CAN-Daten und sendet sie an den CAN-Bus im Testfahrzeug.
KSpv: Kabel		Ethernet, CAN, VGA Kabel zur Distanzüberbrückung
KSvp: UDP-zu-Fahrroboter-zu-Pedalstellung	ESP8266 empfängt UDP-Daten über Wi-Fi 5 und sendet sie an MATLAB auf einem Messrechner. Über ein Skript werden die Daten in Pedalstellungen umgewandelt und an den mit dem Fahrroboter verbundenen Messrechner über TCP/IP übermittelt. Zudem erhält der Messrechner das Sicherheitssignal „steady state“ über ein Spannungssignal über BNC vom Rollenprüfstand. Die Pedalstellungssignale werden über eine serielle Verbindung an den Fahrroboter versendet. Über Kraftübertragung durch Stangen werden den Fahrpedalen die vorgegebene Pedalstellung eingestellt. (Wäsche et al. 2021)	

C: Fragebogen in den Live-Labs

(1) Haben Sie von dem Begriff System of Systems (SoS) bereits vor dem Workshop gehört? Wenn ja, in welchem Zusammenhang?

Freitext

(2) Wie unterscheidet sich das SoS von einem klassischen System? Gehen Sie auch auf die Rolle der eigenständigen Systeme ein!

Freitext

(3) Wie groß schätzen Sie die Bedeutung von SoS für die Produktentwicklung ein?

1 (sehr gering) 2 (gering) 3 (mittel) 4 (hoch) 5 (sehr hoch)

(4) Wie groß schätzen Sie die Bedeutung von SoS für Ihr Projekt ein?

1 (sehr gering) 2 (gering) 3 (mittel) 4 (hoch) 5 (sehr hoch)

Fragebogen – Finale Befragung

(5) Wiederholung Frage (2)

(6) Inwieweit haben Sie die eingeführten SoS-Typen aus Ihrer Sicht verstanden? Können Sie SoS in SoS-Typen einordnen?

1 (sehr schlecht) 2 (schlecht) 3 (mittel) 4 (gut) 5 (sehr gut)

(7) Was sind aus Ihrer Sicht die größten Herausforderungen im Bereich von SoS?

Freitext

(8) Inwieweit wünschen Sie sich Methoden zur Unterstützung der Entwicklung von SoS / Systeme innerhalb eines SoS?

1 (sehr wenig) 2 (wenig) 3 (mittel) 4 (gern) 5 (sehr gern)

(9) Inwieweit können Ihrer Meinung nach folgende Hilfsmittel zum besseren Verständnis von SoS beitragen?

	wenig	mittel	viel
Leitfragen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Begriffsabgrenzung mittels Diagrammen (z.B. Venn-Diagramm)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Beispiele	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bedarfsgerechte Methoden (z.B. für die Validierung von SoS)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(10) Was halten Sie von den Unterteilung der Modelle in die Abstraktionsgrade Generisch, Bereich und System?

1 (sehr wenig) 2 (wenig) 3 (mittel) 4 (viel) 5 (sehr viel)

Freitext

(11) Inwieweit kann MEFS4SoS (Methode zur Erfüllung der Funktionalen Sicherheit für SoS) die Modellbildung und Validierung unterstützen?

1 (sehr wenig) 2 (wenig) 3 (mittel) 4 (viel) 5 (sehr viel)

(12) Ordnen Sie dem SoS Vehicle to Infrastructure Ausprägungen zu! Bitte wählen Sie die zutreffende Antwort für jeden Punkt aus:

	(1) Trifft völlig zu	(2) Trifft überwiegend zu	(3) Trifft teilweise zu	(4) Trifft eher nicht zu	(5) Trifft überhaupt nicht zu
Operationale Unabhängigkeit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Unabhängiges Management	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Geographische Verteilung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Emergenz	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Evolutionäre Entwicklung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Autonomie	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Zugehörigkeit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vernetzung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vielfältigkeit/Diversität	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Versch. Produktlebenszyklen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interdependenzen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Größe, große Skalierung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(13) Wie groß schätzen Sie die Bedeutung von SoS für die Produktentwicklung ein?

1 (sehr gering) 2 (gering) 3 (mittel) 4 (hoch) 5 (sehr hoch)

(14) Wie groß schätzen Sie die Bedeutung von SoS für Ihr Projekt ein?

1 (sehr gering) 2 (gering) 3 (mittel) 4 (hoch) 5 (sehr hoch)

(15) Inwieweit hilft Ihnen der Workshop für Ihr Projekt?

1 (sehr wenig) 2 (wenig) 3 (mittel) 4 (viel) 5 (sehr viel)

D: Erweiterung der Validierungsumgebung

Tabelle D.1: Erweiterung der Validierungsumgebung auf V2I RoadWorksWarning, V2V und V2P

UseCase Element	V2I Road Works Warning – virtuell (Chen, 2020) ¹	V2V Verkehrskreuzung mit Fahrzeugen – virtuell (Qi, 2019) ¹	V2V & V2P Verkehrskreuzung mit Fußgänger – gemischt (Qi, 2019) ¹	V2I & V2V – Netzwerk VANET – virtuell (Chen, 2022 ¹ ; Wäsche, Wang et al., 2022)
Bedarf	Fahrzeug bremst automatisch bei Baustellenbereich ab	Verkehrskreuzung mit Fahrzeugkommunikation und Vorfahrt für Rettungskräfte	Verkehrskreuzung mit Hinweisen an Fahrzeuge bei Verkehrsschildern und Personen	Netzwerk mit Fahrzeugen und RSU zur vollständigen Kommunikation für Verkehrseffizienz
UseCase	Road Works Warning	Intersection Collision Warning	Intersection Movement Assist with vulnerable road users	VANET
Anforderung	Setzen eines Geschwindigkeitsgrenzwertes für den Baustellenbereich	Fahrzeug bekommt Informationen von anderem Fahrzeug und reagiert, um Unfall zu vermeiden	Fahrzeug bekommt Informationen von weiterer Infrastruktur und Personen	Zwei zufällige Fahrzeuge müssen über RSU und weitere Fahrzeuge kommunizieren können
Verändertes System	-	Hinzukommen des Systems Rettungsfahrzeug	Hinzukommen neuer Systeme Personen und Fahrzeuge	-
Veränderte Anforderung	Änderung Anforderung Übertragungreichweite	-	-	Änderung Anforderung an Übertragungstechnologie

¹ Abschlussarbeit (unveröffentlicht)

Gefahrensituation	Überhöhte Geschwindigkeit in Baustellenbereich führt zu Unfall	Behinderung / Unfall mit Rettungswagen auf Verkehrskreuzung	Unfall mit Fahrzeug und Person	Keine Kommunikation zwischen hochautomatisierten Fahrzeugen führt zu Unfall
Gefahr	Unfall	Behinderung mit gefährlichen Folgen	Unfall mit gefährlichen Folgen	Unfall
ASIL	ASIL C	ASIL D	ASIL D	ASIL C
Validierungsziel	Wie hängt die Übertragsreichweite von der Geschwindigkeit ab?	Kann die Vorfahrt bei Kreuzung mit Rettungsfahrzeug gewährt werden?	Wie kann die Validierung um Quellen und Targets erweitert werden?	Wie beeinflusst die Übertragungstechnologie das VANET?
Hypothese-kriterien	Übertragsreichweite, Geschwindigkeit	Geschwindigkeit, Priorität des Fahrzeugs	Quellen, Targets	Übertragungstechnologie, Verkehrsfluss, -dichte
Validierungskonfiguration	Virtuell	Virtuell	Virtuell & virtuell-physisch	Virtuell
Ergebnisse	Übertragsreichweite beeinflusst die Einhaltung der Geschwindigkeit	Kein Unfall, Prioritätssignal wurde empfangen und im Algorithmus die Vorfahrt gewährt	Kein Unfall aufgrund der erhaltenen Geschwindigkeitsinformationen des Fahrers	Über HMI kann die VANET Simulation erfolgreich ausgeführt werden. Es lassen sich die Hypothese-kriterien einstellen.
Interpretation	Übertragsreichweite beeinflusst Reaktionszeit zur Adaption der Geschwindigkeitseinhaltung	Untersch. Prioritäten sind möglich	Unterschiedliche Systeme können in die Validierungs-umgebung integriert werden	Unterschiedliche Kriterien lassen sich für eine VANET Simulation einstellen.