

Aline Radimersky

**Ein Beitrag zur Potenzialausschöpfung
von Systemarchitekturmodellen durch
methodische Verknüpfung zu
Berechnungsmodellen**

A contribution to the exploitation of the
potential of system architecture models by
methodical linkage to computational models

Band 174

Systeme ■ Methoden ■ Prozesse

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

Forschungsberichte



Aline Radimersky

**Ein Beitrag zur Potenzialausschöpfung von
Systemarchitekturmodellen durch methodische
Verknüpfung zu Berechnungsmodellen**

A contribution to the exploitation of the potential of system
architecture models by methodical linkage to computational
models

Band 174

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 WIRmachenDRUCK GmbH, Backnang
0711 995 982 - 20

ISSN 1615-8113

Ein Beitrag zur Potenzialausschöpfung von Systemarchitekturmodellen durch methodische Verknüpfung zu Berechnungsmodellen

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

angenommene

DISSERTATION

von

Dipl.-Ing. Aline Radimersky

Tag der mündlichen Prüfung:	10.05.2024
Hauptreferent:	Univ.-Prof. Dr.-Ing. Dr. h. c. Albert Albers
Korreferent:	Prof. Dr.-Ing. Marcus Geimer

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 174

Die Potenziale der Verknüpfung von Maschinenbau, Elektrotechnik und Informatik zu mechatronischen Lösungen für Produkte und Maschinen stellt einen Megatrend dar, der ganz erhebliche Chancen auch zur Differenzierung im globalen Wettbewerb erlaubt. Die durch diese Verknüpfung entstehende Komplexität der Synthese- und Analyseprozesse in der Produktentwicklung sind dabei allerdings immens. Ursache ist die traditionelle, sehr leistungsfähige, disziplinentorientierte Vorgehensweise und die, den jeweiligen Disziplinen zugrunde liegenden und genutzten Methoden, Prozesse, Modellbildungen und auch Berechnungsmodelle die nun in die Verknüpfung einbezogen werden müssen. Die zur Hebung der Synergiepotenziale einer integrativen, parallelen und gleichzeitigen Entwicklung von mechatronischen Lösungen aus allen drei Domänen erforderlichen neuen Vorgehensweisen und Methoden auf der Basis der Systemtheorie sind hier ein zwingender Weg, um die Zukunft anzugehen.

Ein zentraler Ansatz, der das Advanced Systems Engineering in der Modellierung auf ein neues Level bringen kann, ist das Modell der Systemgenerationsentwicklung – SGE nach ALBERS. In Kombination mit dem Model Based Systems Engineering – MBSE als Werkzeug ergeben sich bisher nicht mögliche Verknüpfungslösungen. Eine wichtige Aufgabenstellung bleibt es hierbei aber immer auch, die disziplinentorientierten Vorgehensweisen und zum Beispiel auch Berechnungsmodelle in dieses Vorgehen zu integrieren. Die Karlsruher Schule für Produktentwicklung – KaSPro mit ihren Methoden und Prozessen will hier neue Wege gehen. Die auf der übergeordneten Ebene des Advanced System Engineering formulierten System-Architektur-Modelle sollen als Basis für die Verknüpfung mit den Vorgehensweisen, Modellen und Ansätzen in den Disziplinen genutzt werden. Genau an dieser Stelle setzt die Arbeit von Frau Dr.-Ing. Aline Radimersky an. Sie hat in ihrer wissenschaftlichen Arbeit untersucht, wie Berechnungsmodelle und Systemarchitekturmodelle strukturiert verknüpft werden können. Sie wählt hierzu das Beispiel eines elektrifizierten Antriebstrangs und untersucht daran die notwendigen Modellbildungen, Verknüpfungsansätze und entwickelt eine erste strukturierte Vorgehensweise, um diese Fragestellungen systematisch anzugehen. Die Arbeit leistet einen wichtigen Beitrag um das Verständnis und die Akzeptanz der Ansätze des Advanced System Engineering in der Praxis der Unternehmen zu unterstützen.

Mai, 2024

Albert Albers

Kurzfassung

Komplexe, mechatronische Produkte stellen besondere Herausforderungen an interdisziplinäre Entwicklungsteams hinsichtlich eines gemeinsamen Systemverständnisses und der Konkretisierung des Zielsystems unter Berücksichtigung der gesamtsystemischen Zusammenhänge. Systems Engineering bietet Ansätze und Methoden, diesen Herausforderungen zu begegnen, zentraler Aspekt ist dabei der Wandel von dokumentenbasierter hin zu modellbasierter Entwicklung. Ein Ansatz ist die Nutzung eines qualitativen Systemarchitekturmodells zur zentralen Dokumentation der Systemstruktur und –zusammenhänge. Dieses Modell kann als Rückgrat der Produktentwicklung eine Basis für die Konsistenz weiterer Systemmodelle darstellen. Aktuelle Studien legen nahe, dass die Umsetzung von modellbasiertem Systems Engineering in der Praxis hinter den forschungsseitig aufgezeigten Potenzialen zurückbleibt.

Ziel dieser Arbeit ist es daher, einen Ansatz zur methodischen Unterstützung der Verknüpfung zwischen Artefakten eines Systemmodells und eines Berechnungsmodells zur Zielsystemkonkretisierung zu entwickeln um dadurch Akzeptanz und Nutzung von Systemarchitekturen zu unterstützen. Der Arbeit liegt dabei die Hypothese zugrunde, dass durch einen Ansatz zur systematischen Nutzung der Systemarchitektur Faktoren reduziert werden können, die zu geringer Akzeptanz und Anwendung dieser führen. Insbesondere wird die Nutzung der Systemarchitektur für die Modellbildung eines Berechnungsmodells und für die Dokumentation der daraus abgeleiteten Entscheidungen und Anforderungen betrachtet.

Um die Hypothese zu untersuchen, werden auf Basis aktueller Studien und semistrukturierter Interviews Faktoren identifiziert, die einen Hinderungsgrund für Akzeptanz und Nutzung von Architekturmodellen darstellen. Die Arbeit zeigt auf, dass wesentliche Hinderungsfaktoren der Aufwand der Architekturerstellung in Kombination mit der geringen Nutzbarkeit für die Erstellung quantitativer Systemmodelle und die Nachvollziehbarkeit der daraus konkretisierten Anforderungen sind. Darauf aufbauend wird analysiert, wie Erstellung und Nutzung von Architekturmodellen besser in bestehende Entwicklungsprozesse – insbesondere in die Erstellung quantitativer Modelle - integriert werden können. Ergebnis ist ein methodischer Ansatz, der am Beispiel eines elektrifizierten Fahrzeugantriebs experimentell angewendet wird. Aus den Erkenntnissen der experimentellen Anwendung werden Potenziale und Risiken für die Implementierung des Ansatzes in die Praxis abgeleitet.

Abstract

Complex, mechatronic products pose special challenges to interdisciplinary development teams with regard to a common understanding of the system and the concretization of the system of objectives, taking into account the overall system interrelationships. Systems engineering offers approaches and methods to meet these challenges, a central aspect is the shift from document-based to model-based development. One approach is the use of a qualitative system architecture model for central documentation of system structure and interrelationships. As the backbone of product development, this model can provide a basis for the consistency of further system models. But current studies show that the implementation of model-based systems engineering in practice falls short of the potential shown by research.

Therefore, this thesis aims to develop an approach to methodically support the linkage between artefacts of a system model and a computational model in order to support acceptance and use of system architectures. The work is based on the hypothesis that an approach to systematically use the system architecture for modelling a computational model and documenting the decisions and requirements derived from it can reduce factors that lead to low acceptance and use of the system architecture.

To investigate the hypothesis, factors that are barriers to acceptance and use of architecture models are identified based on recent studies and semi structured interviews. The thesis shows that significant hindering factors are the effort of architecture creation in combination with the low usability of the architecture model for the creation of quantitative system models and the traceability of the requirements concretized from them. Based on this, it is analysed how the creation and use of architecture models can be better integrated into existing development processes - in particular into the creation of computational models. Result is a methodological approach that is applied experimentally using the example of an electrified vehicle powertrain. Potentials and risks for the implementation of the approach in practice are derived from the findings of the experimental application.

Danksagung

Die vorliegende Arbeit entstand auf Basis meiner Tätigkeit als akademische Mitarbeiterin am IPEK – Institut für Produktentwicklung am Karlsruher Institut für Technologie (KIT). Mein besonderer Dank gilt meinem Doktorvater Herrn Univ.-Prof. Dr.-Ing. Dr. h. c. Albert Albers für die Betreuung meiner Arbeit und die konstruktive Begleitung während des Entstehungsprozesses. Er hat am IPEK ein Arbeits- und Forschungsumfeld geschaffen, das Selbstverantwortung, Leistung und Teamwork fordert und fördert sowie gleichzeitig mit einem großen Gestaltungsfreiraum einhergeht. In diesem positiven Spannungsfeld konnte ich mich fachlich und persönlich weiterentwickeln und profitiere bis heute in meinem jetzigen industriellen Arbeitsumfeld davon.

Herrn Prof. Dr.-Ing. Marcus Geimer danke ich herzlich für die Übernahme des Korreferats. Der Austausch zu meiner Arbeit und die konstruktiven Anmerkungen waren in der Fertigstellung sehr wertvoll.

Dem gesamten IPEK-Team danke ich für das kollegiale Miteinander und die motivierende Zusammenarbeit. Auch über meine aktive Zeit am Institut hinaus konnte ich jederzeit auf uneingeschränkte Unterstützung bauen – vielen Dank dafür! Insbesondere gilt mein Dank Herrn Sascha Ott für seine Unterstützung und den jederzeit offenen, vertrauensvollen Austausch. Herrn Dr.-Ing. Constantin Mandel danke ich für die konstruktiven fachlichen Diskussionen während der Entstehung der Arbeit und für das Korrekturlesen. Herrn Dr.-Ing. Simon Rapp gilt mein Dank ebenfalls für das Korrekturlesen und die damit verbundenen wertvollen Anmerkungen zur Arbeit.

Herrn Dr.-Ing. Jens Gerstenberger und Herrn Stefan Reck danke ich für das entgegengebrachte Vertrauen und die Motivation zur Fertigstellung der Arbeit.

Mein größter Dank gilt meiner Familie. Meiner Frau Katharina danke ich dafür, dass sie der ruhende Pol in meinem Leben ist und für das Schaffen des privaten Freiraums, der nötig war, um die Arbeit fertigzustellen. Meinen Eltern und Großeltern danke ich für ihre verlässliche, fortwährende Unterstützung und die Möglichkeiten, die sie mir aufgezeigt und gegeben haben. Nicht zuletzt danke ich meinen Geschwistern, dass sie mit gemeinsamen Aktivitäten und Spaß für den nötigen Ausgleich nebenher gesorgt haben.

„Komplexe Ingenieuraufgaben sind nur zu bewältigen, wenn zahlreiche Spezialisten verschiedener Qualifikation in einem Team zusammenwirken.“
(Ropohl, 1975)

Inhaltsverzeichnis

Kurzfassung	i
Abstract	iii
Abbildungsverzeichnis	xiii
Tabellenverzeichnis	xix
Abkürzungsverzeichnis	xxi
Formelzeichen	xxiii
1 Einleitung	1
2 Stand der Forschung	3
2.1 Modelle des Produktentstehungsprozesses	3
2.1.1 Systeme der Produktentstehung.....	4
2.1.2 Produktentstehung als Problemlösung.....	6
2.1.3 Generationen-Charakter der Produktentstehung	8
2.1.4 Integriertes Produktentstehungs-Modell (iPeM)	12
2.2 Ziele und Anforderungen in der Produktentstehung	16
2.2.1 Begriffsverständnis Zielsystem, Ziel und Anforderung	16
2.2.2 Bedeutung der Modellierung von Zielsystemen im Anforderungsmanagement.	18
2.2.3 Methodische Unterstützung der Ziel- und Anforderungsermittlung	20
2.2.4 Prozessuale Normen und Vorgaben in der Automobilentwicklung	21
2.3 Validierung und Verifikation im Produktentstehungsprozess.....	25
2.3.1 Begriffsverständnis Validierung und Verifikation	25
2.3.2 Validierung im Kontext der Produktentstehung und Teilaktivitäten der Validierung.....	26
2.3.3 IPEK-X-in-the-Loop-Ansatz und –Framework	29
2.3.4 Virtuelles, modellbasiertes Testen.....	33
2.4 Modellbasiertes Systems Engineering	34
2.4.1 Idee des Modellbasierten Systems Engineering (MBSE).....	35
2.4.2 Modellierung von Systemarchitekturen.....	37
2.4.3 Einsatz von modellbasiertem Systems Engineering in der Praxis	40
2.4.4 Ansätze zur Verwendung der Architekturinhalte in Aktivitäten und Modellen während der Produktentwicklung	43
2.5 Zusammenfassung des Stands der Forschung	55
3 Zielsetzung und Forschungsvorgehen	57
3.1 Forschungsbedarf.....	57
3.2 Ziel und Forschungshypothese	58

3.3	Forschungsvorgehen	61
4	Interviewstudie zur Akzeptanz von Methoden des Systems Engineering	63
4.1	Design der Interviewstudie.....	63
4.2	Ergebnisse der Interviewstudie	65
4.3	Ergänzende Beobachtungen aus der Praxis	69
4.4	Zusammenfassung und Reflektion der Ergebnisse	70
5	Ansatz zur Unterstützung quantitativer Modellbildung durch die Systemarchitektur	75
5.1	Anforderungen an den Ansatz	75
5.2	Herleitung und Definition des Ansatzes	76
5.2.1	Struktur des Architekturmodells und Berücksichtigung von Variationsanteilen des Modells der SGE	77
5.2.2	Aufbau des Berechnungsmodells unter Nutzung des Architekturmodells	80
5.2.3	Beschreibung der Nutzung eines Architekturmodells in der Produktentstehung.....	83
5.2.4	Zusammenspiel des Ansatzes mit der Verwendung des IPEK-XiL-Ansatzes und -Frameworks	87
5.2.5	Berechnung und Bewertung alternativer Systemvarianten	92
5.2.6	Dokumentation von Berechnungsergebnissen und Entscheidungen in der Systemarchitektur	94
5.3	Zusammenfassung des Ansatzes	97
6	Experimentelle Anwendung des Ansatzes: Fahrzeugsimulation mit Parametervariation	101
6.1	Beispiel für die experimentelle Anwendung des Ansatzes	103
6.2	Verknüpfung von qualitativer und quantitativer Modellbildung	104
6.2.1	Kontextanalyse – Abbildung des Kontextes in Architektur- und Berechnungsmodell	104
6.2.2	Systemarchitekturmodell und Berechnungsmodell für das System in Development erstellen	112
6.2.3	Zusammenfassung und Schlussfolgerung zur Nutzung des Architekturmodells als Basis für Erstellung und Nutzung eines Berechnungsmodells	131
6.3	Vergleich alternativer Systemvarianten	133
6.3.1	Definition alternativer Systemvarianten durch Parametrierung....	134
6.3.2	Analyse verschiedener Systemvarianten und Ableitung von Subsystemanforderungen.....	140
7	Bewertung der Implementierung des Ansatzes auf Basis der experimentellen Anwendung	161

7.1	Indikation des Einflusses auf Akzeptanz und Nutzung von Systemarchitekturmodellen	161
8	Zusammenfassung, Fazit und Ausblick	167
8.1	Zusammenfassung und Fazit	167
8.2	Ausblick	169
	Literaturverzeichnis	I
	Glossar	XVII

Abbildungsverzeichnis

Abbildung 2.1:	Analyse- und Syntheseschritte im erweiterten ZHO-Modell nach (Albers et al., 2011; Lohmeyer, 2013, S. 142).....	5
Abbildung 2.2:	Schritte der Problemlösungsmethode SPALTEN (Albers, Reiß, Bursac & Breitschuh, 2016, S. 5).....	7
Abbildung 2.3:	Beispiele für unterschiedliche Referenzsysteme und die daraus abgeleiteten Produktgenerationen (Albers & Rapp, 2022, S. 30).....	11
Abbildung 2.4:	iPeM - integriertes Produktentstehungs-Modell nach (Albers, Reiß, Bursac & Richter, 2016, S. 104).....	13
Abbildung 2.5:	Aktivitäten der Produktentstehung und Aktivitäten der Problemlösung innerhalb einer Ebene des iPeM nach (Albers, Reiß, Bursac & Richter, 2016, S. 104).....	14
Abbildung 2.6:	ASPICE Referenzprozessmodell Version 3.1 im Überblick (VDA QMC Working Group 13 / Automotive SIG, 2017).....	23
Abbildung 2.7:	Übersicht über die Teile der ISO 26262 (ISO 26262-2).	24
Abbildung 2.8:	Validierung als Teil des erweiterten ZHO-Modells nach (Albers, Behrendt, Schroeter, Ott & Klingler, 2013).	26
Abbildung 2.9:	Design und Validierung im Produktentstehungsprozess (Albers, Behrendt et al., 2016, S. 545).....	27
Abbildung 2.10:	Elemente und Zusammenhänge des IPEK-X-in-the-Loop-(XiL)-Ansatzes (Albers, Behrendt et al., 2016, S. 559) basierend auf (Albers & Düser, 2008)	29
Abbildung 2.11:	IPEK-X-in-the-Loop-(XiL)-Framework (Albers, Behrendt et al., 2016, S. 560) basierend auf (Albers & Düser, 2008).....	31
Abbildung 2.12:	System in Development (SiD) und System under Investigation (Sul) (Albers, Behrendt et al., 2016, S. 561).	32
Abbildung 2.13:	Korrelation zwischen Produktsystem und Validierungssystem nach (Mandel et al., 2020, S. 2).....	33
Abbildung 2.14:	SPES Modellierungsframework für Systemarchitekturen (Pohl et al., 2012, S. 38).....	39
Abbildung 2.15:	Identifizierung der Variationsarten im SysML-Modell und Berechnung der Variationsanteile(Wäschle et al., 2020, S. 2431).....	44

Abbildung 2.16:	Verknüpfung eines neuen Anforderungsprofils mit dem Modell der vorigen Produktgeneration nach (Albers, Kühn et al., 2017).	45
Abbildung 2.17:	Aufruf und Kommunikation zwischen dem SysML-Tool, MechML Visualizer und CAD (Moeser et al., 2016).	47
Abbildung 2.18:	SysML-Stereotypen zur Abbildung von Produktsystem und Validierungssystem (Mandel, Böning et al., 2021, S. 4).	48
Abbildung 2.19:	Rahmenwerk der Bibliothek und Verknüpfte Komponenten (Mandel et al., 2020, S. 4).	50
Abbildung 2.20:	Schematische Darstellung des Rahmenwerks der genutzten Modelle (Mandel, Wäschle et al., 2021).	51
Abbildung 2.21:	Zusicherungsdiagramm für einen Aktuator als Schnittstelle zu MATLAB/Simulink (Vidal & Villota, 2018, S. 151)	52
Abbildung 2.22:	Ansatz zur Transformation eines SysML-Modells in ein Modelica-Modell unter Berücksichtigung Werkzeugabhängiger und –unabhängiger Randbedingungen. (Herzig et al., 2013, S. 121)	54
Abbildung 3.1:	Darstellung des Forschungsvorgehens in Anlehnung an das Forschungsdesign nach (Marxen & Albers, 2012, S. 1014)	62
Abbildung 4.1:	Schema zur Akzeptanzbewertung von Methoden, Werkzeugen und Ansätzen nach (Albers, Lohmeyer et al., 2012).	64
Abbildung 4.2:	Nennung von Methoden, Ansätzen und Werkzeugen in den Interviews (Albers, Lohmeyer et al., 2012).	67
Abbildung 5.1:	Elemente alternativer Systemarchitekturen eingeordnet in die Darstellung des SPES-Modeling-Framework von (Pohl et al., 2012).	78
Abbildung 5.2:	Zusammenhang zwischen Berechnungsmodell und Architekturmodell.	81
Abbildung 5.3:	Beschreibung der Erstellung und Nutzung des Architekturmodells durch das erweiterte ZHO-Modell unter Berücksichtigung von Validierung und Kreation in den Analyse- und Syntheseschritten	85
Abbildung 5.4:	Übertragung der Elemente des IPEK-XiL-Ansatz und -Framework in den zu entwickelnden Ansatz.	89
Abbildung 5.5:	Zusammenhang zwischen IPEK-XiL-Framework und Ansatz in tieferen Systemebenen.	91

Abbildung 5.6:	Dimensionen alternativer Systemvarianten im Architekturmodell.	93
Abbildung 5.7:	Schritte und Ausgangsgrößen des Handlungssystems bei der Ableitung von Anforderungen.	95
Abbildung 5.8:	Verlinkungskonzept von Ergebnissen und Entscheidungen mit und innerhalb der Systemarchitektur.	96
Abbildung 5.9:	Ansatz zur Entwicklung konsistenter, nachvollziehbarer und quantitativer Anforderungen durch inhaltliche Verknüpfung von Architektur- und Berechnungsmodell.....	99
Abbildung 6.1:	Strukturierung der experimentellen Anwendung des Ansatzes anhand des Beispiels elektrifizierter Fahrzeugantrieb.....	102
Abbildung 6.2:	Wirknetz eines elektrischen Antriebssystems dargestellt im C&C ² -Ansatz nach (Albers, Radimersky & Brezger, 2015)	103
Abbildung 6.3:	Kontextelemente des EV-Antriebs im Blockdefinitionsdiagramm der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells.	105
Abbildung 6.4:	Anwendungsfalldiagramm für die Hauptanwendung „Insassen und Gepäck bewegen“ des EV-Antriebs in der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells.....	106
Abbildung 6.5:	Funktionaler Ablauf des Anwendungsfalls „Insassen und Gepäck bewegen“ im Aktivitätsdiagramm der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells.....	107
Abbildung 6.6:	Logische Zusammenhänge und Schnittstellen des SiD „EV-Antrieb“ mit den umgebenden Systemen im internen Blockdiagramm der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells.....	107
Abbildung 6.7:	Geschwindigkeitsprofile für einen sportlichen und einen normalen Fahrer bei Fahrsituation „Autobahnfahrt“ auf Basis (Heinrich, 2016).	108
Abbildung 6.8:	Vergleich der Modellierung der umgebenden Systeme und deren Zusammenhänge mit dem SiD „EV-Antrieb“ im Architekturmodell und im Berechnungsmodell, Fokus: Fahrerwunsch	109
Abbildung 6.9:	Vergleich der Modellierung der umgebenden Systeme und deren Zusammenhänge mit dem SiD „EV-Antrieb“ im Architekturmodell und im Berechnungsmodell, Fokus: Leistungsbedarf an der Achse	110

Abbildung 6.10:	Repräsentation der Anwendungsfälle des SiD „EV-Antrieb“ als Anforderungen im Architekturmodell, die die im Berechnungsmodell verwendeten Zyklen beinhalten.....	111
Abbildung 6.11:	Blockdefinitionsdiagramm des SiD „EV-Antrieb“ in der logischen Sicht auf Ebene 1 des Architekturmodells.....	113
Abbildung 6.12:	Aktivitätsdiagramm zur Detaillierung der Funktion „Leistung an der Achse bereitstellen“ in der funktionalen Sicht auf Ebene 1 des Architekturmodells.	114
Abbildung 6.13:	Darstellung der Zusammenhänge zwischen den Subsystemen des EV-Antriebs in einem internen Blockdiagramm der logischen Sicht auf Ebene 1 des Architekturmodells.	115
Abbildung 6.14:	Berechnung des im aktuellen Fahrzustand notwendigen Drehmoments und der Drehzahl der E-Maschine für jede mögliche Getriebeübersetzung.....	117
Abbildung 6.15:	Beispiel für die Abbildung unterschiedlicher Getriebevarianten in der technischen Sicht auf Ebene 1 des Architekturmodells.	118
Abbildung 6.16:	Ermitteln der optimalen Übersetzung für den gegebenen Fahrzustand.....	119
Abbildung 6.17:	Berechnung des Energiebedarfs und des resultierenden Stromprofils an der Batterie.....	121
Abbildung 6.18:	Subsysteme des Level-2-Systems „Getriebe“, dargestellt im Blockdefinitionsdiagramm der logischen Sicht auf Ebene 2 des Architekturmodells.	123
Abbildung 6.19:	Detaillierung der Funktionen „Getriebeeingangsleistung in Leistung an der Achse wandeln“ und „Info aktueller Gang senden“ durch Darstellung in einem Aktivitätsdiagramm in der funktionalen Sicht auf Ebene 2 des Architekturmodells.	124
Abbildung 6.20:	Zusammenhänge zwischen den Subsystemen des Getriebes dargestellt in einem internen Blockdiagramm in der logischen Sicht auf Ebene 2 des Architekturmodells.....	126
Abbildung 6.21:	Berechnung der notwendigen Energie zur Aktuierung des Schaltelements je Gangwechsel.	128
Abbildung 6.22:	Berechnung der Verluste im Schaltelement je Gangwechsel.	130
Abbildung 6.23:	Varierte Parametrierung auf Ebene eins des Antriebssystems, dargestellt im qualitativen Systemmodell.	137

Abbildung 6.24:	Reichweitenpotenziale der Antriebsvarianten auf Ebene 1 im angenommenen Nutzungsprofil.....	142
Abbildung 6.25:	Reichweitenpotenziale der Antriebsvarianten auf Ebene 1 in den einzelnen Fahrzyklen.....	143
Abbildung 6.26:	Verknüpfung von Analyseergebnis und Entscheidung im Architekturmodell am Beispiel Traktionsmaschine.....	144
Abbildung 6.27:	Vergleich der EM 1-Antriebsvarianten bezüglich Reichweitenpotenzial gegenüber 1-Gang-Referenz im angenommenen Nutzungsprofil.....	145
Abbildung 6.28:	Vergleich der EM 1-Antriebsvarianten bezüglich Reichweitenpotenzial gegenüber 1-Gang-Referenz in den einzelnen Fahrzyklen.....	146
Abbildung 6.29:	Ergänzung der Entscheidungsdokumentation durch Hinzufügen von beschreibendem Text und zusätzlicher Anbindung des logischen Elements „Getriebe“ an den Kommentar.....	147
Abbildung 6.30:	Reichweitenpotenziale der Antriebsvariante EM 1 [18, 10] bei unterschiedlichen Getriebewirkungsgraden in den einzelnen Fahrzyklen.....	149
Abbildung 6.31:	Ergänzung der Entscheidungsdokumentation durch Konkretisieren des Textes und zusätzlicher Anbindung der Analyseergebnisse der Wirkungsgradvariation an den Kommentar.....	150
Abbildung 6.32:	Verknüpfung zwischen den Ergebnissen des Berechnungsmodells und den Elementen des Systemarchitekturmodells.....	152
Abbildung 6.33:	Vergleich der Auswirkung unterschiedlicher Schaltabstände (hyst) und Schaltdauern (dt) im angenommenen Nutzungsprofil.....	154
Abbildung 6.34:	Vergleich der Auswirkung unterschiedlicher Schaltabstände (hyst) und Schaltdauern (dt) für effizienzoptimale Gangwahl in den einzelnen Fahrzyklen.....	155
Abbildung 6.35:	Vergleich der Auswirkung unterschiedlicher Schaltabstände (hyst) und Schaltdauern (dt) für suboptimale Gangwahl in den einzelnen Fahrzyklen.....	156
Abbildung 6.36:	Verknüpfung zwischen Analyseergebnissen auf Ebene zwei, Entscheidungen und Subsystemanforderungen im Architekturmodell.....	157

Abbildung 6.37: Modellelement der Anforderung „Betriebspunkte der Traktionsmaschine“ im Architekturmodell. 159

Tabellenverzeichnis

Tabelle 6.1:	Nutzungsprofil des Fahrzeugs charakterisiert durch Anteil von Fahrzyklen an Betriebsdauer des Fahrzeugs.....	135
Tabelle 6.2:	Varierte Parameter im Berechnungsmodell auf Ebene 1	136
Tabelle 6.3:	Varierte Parameter im Berechnungsmodell auf Ebene 2	140
Tabelle 6.4:	Referenz für die Reichweitenbetrachtung auf Ebene 1 mit Antriebsvariante EM 1, $i=10$, 94% Wirkungsgrad, 80% Rekuperationsanteil	141

Abkürzungsverzeichnis

bdd	<i>Block Definitions Diagramm</i>
Diff	<i>Differenzial</i>
EM	<i>E-Maschine, Traktionsmaschine</i>
EV	<i>elektrifiziertes Fahrzeug, electric vehicle</i>
HV	<i>Hochvolt, High Voltage</i>
ibd	<i>internes Block Diagramm</i>
iPeM	<i>integrierte Produktentstehungs-Modell</i>
LE	<i>Leistungselektronik</i>
LV	<i>Low Voltage</i>
MBE	<i>Model-Based Engineering</i>
MBSE	<i>Modellbasiertes Systems Engineering</i>
PGE	<i>Produktgenerationsentwicklung</i>
SGE	<i>Systemgenerationsentwicklung</i>
SiD	<i>System in Development</i>
Sul	<i>System under Investigation</i>
SysML	<i>System Modeling Language</i>
WLTP	<i>Worldwide Harmonized Light-Duty Vehicles Test Procedure</i>
XiL	<i>X-in-the-Loop</i>
ZHO	<i>Zielsystem Handlungssystem Objektsystem</i>

Formelzeichen

R_n	Referenzsystem
G_n	Produktgeneration
$\ddot{U}V_n$	Übernahmevariation
AV_n	Ausprägungsvariation
PV_n	Prinzipvariation
M	Drehmoment
n	Drehzahl
U	Spannung
I	Strom

1 Einleitung

Elektrifizierung und Vernetzung beeinflussen aktuell insbesondere die Automobilindustrie. Die Integration von Komfort- und Assistenzfunktionen im Fahrzeug, die Kommunikation in Flottenverbänden und die Begeisterung der Nutzer durch Car-Multimedia-Funktionen führen dazu, dass der Softwareanteil in Fahrzeugen stetig zunimmt. (Albert Albers, 2023, S. 7; Charette, 2021) Gleichzeitig müssen Absicherungsmaßnahmen im System umgesetzt werden, beispielsweise gegen zufällige Elektronikfehler oder gegen unzulässigen Zugriff auf die Fahrzeugkommunikation von außen. (Charette, 2021) Die Umsetzung dieser Funktionalitäten bedarf einem komplexen Zusammenspiel von Software-, Hardware- und Mechanik-Anteilen. Dabei muss der Zielkonflikt zwischen fundierter, disziplinspezifischer Entwicklung und ganzheitlichem Systemdenken gelöst werden. (Gausemeier, Meyer & Steglich, 2022) Entwicklungsteams stehen vor der Herausforderung, die Funktionen und Merkmale des Gesamtsystems interdisziplinär zu entwickeln und über Produktgenerationen hin anzupassen und zu optimieren. (Albert Albers, 2023, S. 5) Modellbasierte Entwicklung kann zu Durchgängigkeit, Transparenz und Zielorientierung innerhalb des Entwicklungsprozesses beitragen. (Brinkschulte, Engelman, Siebert, Iwanicki & Geimer, 2017, S. 178) Es bedarf jedoch neuer Ansätze in der Entwicklung, um bei der gegebenen Produktkomplexität das Systemverständnis und das Entwicklungsvorgehen der beteiligten Domänen zu harmonisieren und zu steuern. (Dumitrescu, Albers, Riedel, Stark & Gausemeier, 2021)

Systems Engineering umfasst Methoden und Ansätze, um mit den genannten Herausforderungen umzugehen. Ein zentraler Aspekt dabei ist der Wandel von der dokumentenbasierten Entwicklung hin zu einer durchgängigen, modellbasierten Entwicklung: Modellbasiertes Systems Engineering (MBSE). (Albert Albers, 2023, S. 11)

„When we apply MBSE properly, the model becomes the collected knowledge associated with the project or system and, ideally, should be considered the single source of truth.“ (Holt & Perry, 2018, S. 4)

Die Relevanz von Prozessen und Methoden des Systems Engineering wird von Unternehmen laut Umfragen als hoch eingeschätzt. (Stützel & Paul, 2021) Dennoch zögern viele Unternehmen MBSE einzuführen. (Albert Albers, 2023, S. 11) Die Umsetzung von MBSE erfolgt bislang maßgeblich in einzelnen Pilotprojekten der Automobil- und Flugzeugindustrie. (Gausemeier et al., 2022) In vielen anderen Branchen

finden sich kaum Ansätze zur Nutzung von MBSE. Unter anderem lässt sich das auf fehlende Ansätze zur Vernetzung des Architekturmodells mit anderen Modellen zurückführen. (Dumitrescu et al., 2021)

Ziel dieser Arbeit ist daher anhand eines elektrischen Fahrzeugantriebs zu untersuchen, wie ein Architekturmodell als zentrale Informationsquelle für andere Modelle genutzt werden kann und einen methodischen Ansatz dafür abzuleiten.

Zur Klärung des Forschungsbedarfs wird dazu in Kapitel 2 der Stand der Forschung zu aktuellen Methoden und Modellen der Produktentwicklung sowie zu Systems Engineering analysiert. In Bezug auf MBSE werden insbesondere Erkenntnisse zur aktuellen Verbreitung in der Praxis und forschungsseitig bereits bestehende Ansätze zur Nutzung von Architekturmodellen betrachtet. Aufbauend auf dem Stand der Forschung wird in Kapitel 3 die Zielsetzung der Arbeit konkretisiert, basierend auf der Forschungshypothese werden sieben Forschungsfragen abgeleitet. Kapitel 4 beleuchtet im Rahmen einer Interviewstudie wesentliche Faktoren, die den Wandel zum durchgängigen, modellbasierten Entwicklungsvorgehen behindern. In Kapitel 5 wird ein Ansatz zur Nutzung der Systemarchitektur eines E-Fahrzeugs für die Ableitung von Zielen und Anforderungen an Subsysteme des Antriebssystems hergeleitet. Hierfür wird zunächst auf theoretischer Basis analysiert, wie ein Architekturmodell strukturiert sein muss, um die Ableitung eines Berechnungsmodells daraus zu ermöglichen und welche Anteile der Systemarchitektur in das Berechnungsmodell übernommen werden. Dabei wird berücksichtigt, dass in beiden Modellen Systemvarianten basierend auf vorangegangenen Produkt- oder Systemgenerationen abbildbar sein sollen und, dass im Sinne einer virtuellen Validierung das IPEK-XiL-Framework zur Strukturierung des Berechnungsmodells genutzt werden kann. Ein weiterer Aspekt ist die Nachvollziehbarkeit der Ziele und Anforderungen, hierfür wird ein Verlinkungskonzept erarbeitet, das die Zugänglichkeit von Berechnungsergebnissen und Entwicklungsentscheidungen aus der Systemarchitektur heraus ermöglicht. Der Ansatz wird in Kapitel 6 am Beispiel des elektrifizierten Antriebssystems umgesetzt. In Kapitel 7 wird bewertet, wie der Ansatz die identifizierten Hinderungsfaktoren adressiert und welche Potenziale zur besseren Akzeptanz von MBSE aber auch welche Risiken der Ansatz dahingehend birgt. Kapitel 8 fasst die Ergebnisse zusammen und gibt einen Ausblick auf weiterführende Fragestellungen.

2 Stand der Forschung

Die Umsetzung und effektive Nutzung von Systems Engineering umfasst nicht nur die Einführung einzelner Methoden und Ansätze, sondern bedarf auch einem Gesamtverständnis verschiedener Aspekte in Produktentstehungsprozessen. Daher werden in Kapitel 2.1 zunächst Beschreibungsmodelle für diese Aspekte analysiert und dargelegt, die im integrierten Produktentstehungsmodell (iPeM) zusammenfließen. Grundlage für die Entwicklung eines erfolgreichen Produkts sind dessen Ziele und Anforderungen, die im Laufe der Produktentwicklung durch Validierungsaktivitäten kontinuierlich konkretisiert werden. Es wird somit in Kapitel 2.2 der Stand der Forschung zur Anforderungsermittlung und Zielsystemmodellierung – auch im Kontext aktueller Prozessvorgaben in der Automobilindustrie - sowie in Kapitel 2.3 zu Validierung und Verifikation dargelegt. In Kapitel 2.4 werden Idee und Anspruch von MBSE und dessen Verbreitung auf Basis aktueller Forschungsergebnisse beschrieben. Weiter werden forschungsseitige Ansätze zur umfassenden Nutzung von Architekturmodellen analysiert. Abschließend fasst Kapitel 2.5 den Stand der Forschung kurz zusammen.

2.1 Modelle des Produktentstehungsprozesses

Aktuelle Produkte sind häufig durch eine hohe Funktionsintegration und eine starke Vernetzung charakterisiert. Die Produkte sind zunehmend interaktiv mit umgebenden Systemen verbunden und umfassen dadurch auch eine steigende Anzahl von Software- und Elektronikkomponenten. (Albert Albers, 2023, S. 5) Mehr Schnittstellen zu umgebenden sozio-technischen Systemen, mehr innere Subsysteme unterschiedlicher Domänen und der Bedarf, flexibel und individualisierbar zu sein, erhöht die Produktkomplexität und damit auch die Herausforderungen in der Entwicklung dieser Produkte. Es bedarf methodischer und prozessualer Unterstützung von Entwicklungsteams, um effizient erfolgreiche Produkte mit hohem Kundenwert zu fairen Preisen zu entwickeln. (Albert Albers, Reiß, Bursac & Richter, 2016, S. 101) Im Laufe der letzten Jahrzehnte ist eine Vielzahl an Vorgehensmodellen für die Aktivitäten und deren Ablauf in der Produktentstehung entstanden. Beispiele sind das Münchner Vorgehensmodell (Lindemann, 2009), das Stage-Gate-Modell (Cooper, 1990) oder das V-Modell (VDI/VDE-Richtlinie 2206), um nur einige zu nennen. Vorgehensmodelle werden angewendet, um Prozesse zu planen, während der Entwicklung den Prozess zu navigieren und zu reflektieren. (VDI-Richtlinie 2221-1)

Ein explizierbares Verständnis der Abläufe ist die Basis für die Entwicklung unterstützender Methoden. Im Folgenden wird daher auf den Stand der Forschung zu Beschreibungsmodellen für die Abläufe der Produktentstehung eingegangen.

2.1.1 Systeme der Produktentstehung

Nach Ropohl ist eine Systemvorstellung ein Modell, das von menschlichem Denken konstruiert ist und sich auf beliebige Gegebenheiten beziehen kann. Demzufolge lässt sich auch die Produktentstehung mit Hilfe der Systemtheorie beschreiben. Ropohl identifiziert dabei bereits drei wesentliche Systeme: Das Sachsystem, das die durch die Ingenieurstätigkeit entstehenden Objekte beinhaltet. Das Handlungssystem, das die Maßnahmen und Einrichtungen beinhaltet, die für die Ingenieurstätigkeit von Nöten sind. Und das Zielsystem, das die Menge an Zielvorgaben enthält, an denen sich die Ingenieurstätigkeit orientiert. Mit diesem Verständnis beschreibt er die Ingenieurstätigkeit insgesamt als das Zusammenwirken von Sachsystem, Handlungssystem und Zielsystem. (Ropohl, 1975)

Albers greift dieses Verständnis auf und konkretisiert es im Rahmen der Forschung zum Zielsystem-Handlungssystem-Objektsystem-Modell (ZHO-Modell). Die zweite Hypothese der Produktentstehung nach Albers lautet: *„Produktentstehung ist die Überführung eines Zielsystems in ein konkretes Objektsystem durch ein Handlungssystem.“* Das Handlungssystem ist dabei das einzig aktive System, es leitet aus verfügbarem Wissen Ziele ab und überführt diese wiederum in konkrete Objekte. (Albert Albers, 2010)

Albers und Lohmeyer erweitern das ZHO-Modell um eine Beschreibung der Interaktion der drei Systeme als Transformationsschritte in Form von Analyse und Synthese. Abbildung 2.1 zeigt die Analyse- und Syntheseschritte im erweiterten ZHO-Modell. Produktentstehung wird dabei als kontinuierliche Interaktion der drei Systeme beschrieben, bei der das Handlungssystem simultan Ziel- und Objektsystem entwickelt. Im Laufe dieser Interaktion wird Wissen erzeugt auf dessen Basis neue Objekte entstehen, diese wiederum ermöglichen die Erzeugung neuer Informationen und neuen Wissens. (Albert Albers, Lohmeyer & Ebel, 2011) Lohmeyer definiert Analyse im Kontext des erweiterten ZHO-Modells als das Verstehen eines existierenden Systems mit dem Resultat einer Erkenntnis. Synthese bezweckt nach Lohmeyer das Erschaffen eines bis dato nichtexistierenden Systems, ihr Ergebnis ist ein Element des Zielsystems oder ein Element des Objektsystems. (Lohmeyer, 2013, S. 128)

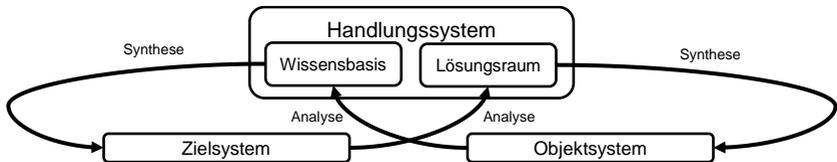


Abbildung 2.1: Analyse- und Syntheseschritte im erweiterten ZHO-Modell nach (Albert Albers et al., 2011; Lohmeyer, 2013, S. 142).

Innerhalb des Handlungssystems werden in diesem Modell die Aspekte Wissensbasis und Lösungsraum unterschieden. Die Wissensbasis umfasst die Menge an verfügbarem Domänen- und fallspezifischem Wissen innerhalb eines bestimmten Produktentstehungsprozesses. Wissen ist personengebunden und kann daher durch Erkenntnisgewinn oder durch Integration zusätzlicher Personen erweitert werden. (Albert Albers, Ebel & Lohmeyer, 2012) Eine Teilmenge der Wissensbasis ist der Problemraum, dieser entspricht dem subjektiven Verständnis, das eine Person von einem Problem auf Basis der gesetzten Ziele und erkannten Randbedingungen hat. Der Lösungsraum entspricht dagegen dem subjektiven Verständnis, das die Person von den zulässigen Lösungen des Problems hat. Er hängt damit von allen von den durch den Problemraum definierten Freiheitsgraden ab. (Albert Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013, S. 119)

Die Analyse des Zielsystems umfasst die subjektive Wahrnehmung und Interpretation des Zielsystems durch eine Person zu einem individuell wahrgenommenen Lösungsraum. In Entwicklungsteams entstehen daher trotz desselben explizierten Zielsystems unterschiedliche individuelle Lösungsräume. (Albert Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013, S. 138) In der Synthese des Objektsystems findet die Modellierung von Prinzip und Gestalt einer oder mehrerer Lösungen in Form expliziter Modelle statt. Die Modelle können virtuell, physisch oder gemischt physisch-virtueller Art sein. (Albert Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013, S. 139) In der Analyse des Objektsystems werden die durch Synthese entstandenen Modelle unter definierten Bedingungen und Eingangsgrößen untersucht und ihr Verhalten ausgewertet. Durch diese Analyse wird die Wissensbasis des Handlungssystems vergrößert und dadurch bestehende Wissenslücken und Unsicherheiten reduziert. (Albert Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013, S. 136) In der Synthese des Zielsystems entscheidet das Handlungssystem auf Grundlage seiner Wissensbasis, wie das Zielsystem verändert - z. B. erweitert oder konkretisiert - wird. (Albert Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013, S. 137)

Zwischenfazit

Mit Hilfe des erweiterten ZHO-Modells lassen sich Denk- und Handlungsabläufe der Produktentstehung allgemein als kontinuierliche Abfolge von Analyse- und Syntheseschritten beschreiben. Es bietet damit auch eine theoretische Grundlage, zur Einordnung, Beschreibung und Weiterentwicklung konkreter Methoden und Prozesse, ohne diese vorzugeben oder einzuschränken.

2.1.2 Produktentstehung als Problemlösung

Dörner beschreibt ein Problem als Kombination aus einem unerwünschten Anfangszustand, einem gewünschten Endzustand und einer Barriere, die eine simple Transformation von Anfangs- in Endzustand verhindert. (Dörner, 1979) Diese Definition lässt sich auf die bereits eingeführten Begriffe des Ziel- und Objektsystems übertragen. Das Zielsystem beschreibt den gewünschten Endzustand, das Objektsystem den aktuellen Zustand. Die Überführung des Zielsystems in ein Objektsystem ist in diesem Verständnis das Lösen eines Problems. Produktentstehung kann also grundsätzlich als Problemlösung verstanden werden, wobei sich das Gesamtproblem entsprechend der Systemtheorie in Teilprobleme strukturieren und aufspalten lässt. (Albert Albers, 2010)

SPALTEN ist ein allgemeiner methodischer Ansatz zur Problemlösung, basierend auf der Idee des Systems Engineering. (Albert Albers, Reiß, Bursac & Breitschuh, 2016, S. 411) Das Ziel von SPALTEN ist die systematische Unterstützung zur Lösung von Problemen unterschiedlicher Komplexitätsgrade. (Albert Albers, Saak & Burkardt, 2002, S. 83)

Abbildung 2.2 zeigt die Schritte des SPALTEN-Prozesses, diese umfassen: Das Sammeln und Analysieren von Informationen zur aktuellen Situation (Situationsanalyse). Die Konkretisierung des vorliegenden Problems auf Basis der gesammelten Informationen, z. B. durch identifizierte Abhängigkeiten (Problemeingrenzung). Das Erarbeiten verschiedener Lösungsmöglichkeiten für das konkretisierte Problem (Alternative Lösungen). Vergleich und Auswahl einer Lösung anhand definierter Kriterien (Lösungsauswahl). Bewertung der möglichen Konsequenzen in Form von Chancen und Risiken bei Umsetzung der gewählten Lösung (Tragweitenanalyse). Operationalisierung der Umsetzung durch Verabschieden der gewählten Lösung und Planung der notwendigen Schritte zur Umsetzung (Entscheiden und Umsetzen). Reflektion der vorangegangenen Schritte und derer Ergebnisse um die dabei gesammelte Erfahrung in späteren Problemlösungsprozessen berücksichtigen zu können (Nachbereiten und Lernen). (Albert Albers, Reiß, Bursac & Breitschuh, 2016, S. 415)

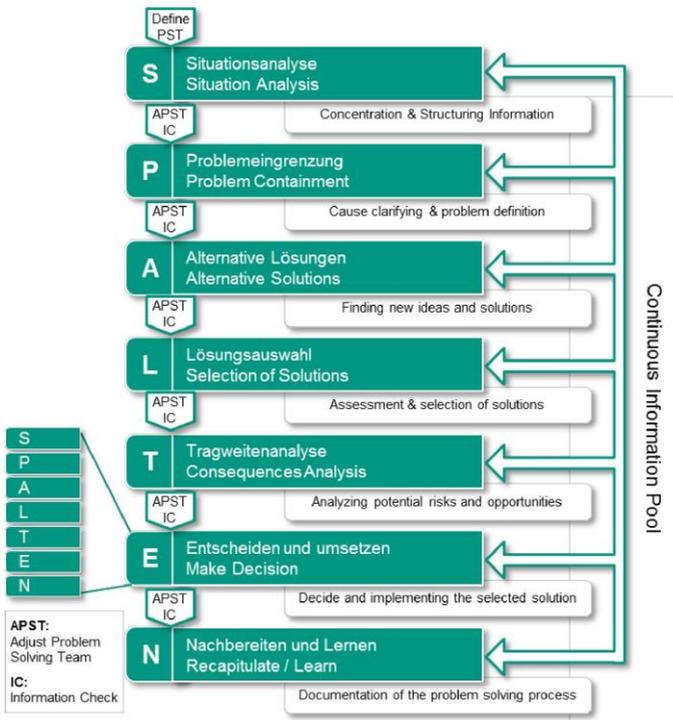


Abbildung 2.2: Schritte der Problemlösungsmethode SPALTEN (Albert Albers, Reiß, Bursac & Breitschuh, 2016, S. 5).

Im Laufe der SPALTEN-Schritte werden immer wieder Informationen und Ideen erarbeitet (Situationsanalyse, Alternative Lösungen, Tragweitenanalyse, Nachbereiten und Lernen), um darauf aufbauend zielgerichtet zu fokussieren (Problemeingrenzung, Lösungsauswahl, Entscheiden und Umsetzen). So entsteht über die Schritte hinweg ein "atmender" Zyklus. (Albert Albers, Burkhardt, Meboldt & Saak, 2005, S. 5) Wissensbasis und Lösungsraum des Handlungssystems werden also kontinuierlich erweitert. Im Rahmen des „IC – Information Check“ werden die verfügbaren Informationen geprüft um Änderungen in der Folge mit einzubeziehen. Albers et. al. stellen außerdem die besondere Bedeutung des Problemlösungsteams

heraus, das Problem- und situationspezifisch gewählt und auch während des Problemlösungsprozesses angepasst werden muss. (Albert Albers et al., 2002, S. 84) in Abbildung 2.2 ist dieses hinterfragen und anpassen des Problemlösungsteams als „APST – Adjust Problem Solving Team“ gekennzeichnet.

Eine Studie zur Anwendung der SPALTEN-Methodik in der Praxis zeigt, dass sich die Methodik auf Probleme unterschiedlichster Rahmenbedingungen und Komplexitätsgrade pragmatisch anwenden lässt. Das systematische Vorgehen bei der Problemlösung wird durch den roten Faden, die Fragmentierung in Teilprobleme und die fokussierte Herangehensweise unterstützt. Die SPALTEN-Schritte selbst können wiederum durch eine Vielzahl von Methoden situationsangepasst unterstützt werden. (Albert Albers, Reiß, Bursac & Breitschuh, 2016, S. 419)

Zwischenfazit

Produktentstehung kann als Problemlösung verstanden werden, wobei sich das Gesamtproblem (Überführung eines Zielsystems in ein Objektsystem) gemäß der Systemtheorie in Teilprobleme aufspalten lässt. SPALTEN bietet eine universell anwendbare Vorgehensweise für diese Teilprobleme, unabhängig von Detaillierungsebene und Komplexitätsgrad.

2.1.3 Generationen-Charakter der Produktentstehung

Albers et. al. stellen fest, dass sich die Mehrzahl der Entwicklungsprojekte in der Praxis nicht eindeutig Kategorien wie Neukonstruktion, Anpassungskonstruktion oder Variantenkonstruktion zuordnen lassen. Vielmehr werden bei der realen Entwicklung von Produkten sowohl vorhandene Teilsysteme übernommen und angepasst als auch Teilsysteme in ihrer Gestalt oder dem Lösungsprinzip neu entwickelt. Um bei moderatem Entwicklungsrisiko die Innovationschancen zu erhöhen, bietet es sich beispielsweise an, Basisanforderungen¹ durch geringe Gestaltvariation zu realisieren und die Entwicklungsressourcen auf die Leistungs¹- und Begeisterungsmerkmale¹ zu fokussieren. Die Produkte basieren somit auf vorangegangenen Produkten oder deren Anteilen und können als neue Generation dieser beschrieben werden. Auf Basis dieser Feststellung entwickeln Albers et. al. das Beschreibungsmodell der Produktgenerationsentwicklung nach Albers (PGE). (Albert Albers,

¹ Entsprechend der Anforderungskategorisierung nach dem Kano-Modell Kano, Seraku, Takahashi und ichi Tsuji (1984).

Bursac & Wintergerst, 2015) Das Beschreibungsmodell basiert auf zwei zentralen Hypothesen.

Erste Hypothese: *„Die Entwicklung neuer Produktgenerationen G_n basiert auf einem Referenzsystem R_n , welches aus Elementen existierender oder geplanter sozio-technischer Systeme besteht.“* (Albert Albers et al., 2022)

Zweite Hypothese: *„Neue Produkte werden basierend auf diesem Referenzsystem R_n durch die Aktivitäten Übernahme- $\ddot{U}V_n$, Ausprägungs- AV_n und Prinzipvariation PV_n entwickelt und gestaltet.“* (Albert Albers et al., 2022)

$$R_n \xrightarrow{V} G_n = \ddot{U}V_n \cup AV_n \cup PV_n \quad 2.1$$

Im Folgenden werden die zentralen Begriffe dieser Hypothesen näher erläutert.

Nach VDI-Richtlinie 2221 ist ein Produkt *„ein Erzeugnis oder Leistung materieller wie immaterieller Art, das oder die allein oder als System angeboten wird, um den Bedarf am Markt sowie die Bedürfnisse von Nutzern zielgruppengerecht zu befriedigen.“* (VDI-Richtlinie 2221-1) Eine **Produktgeneration** beschreibt dementsprechend ein Produkt, das für einen bestimmten Markt zu einem bestimmten Zeitpunkt entwickelt wird. Die Beschreibung der Übernahme und Neuentwicklungsanteile lassen sich jedoch nicht nur auf das Endprodukt beziehen, sondern auch auf dessen Derivate oder Varianten, die im Laufe des Entwicklungsprozesses entstehen (Albert Albers, Bursac & Wintergerst, 2015), diese werden als Entwicklungsgenerationen bezeichnet. Entwicklungsgenerationen können beispielsweise unterschiedliche Reifegradstufen oder Sprint-Inkrementen sein. (Albert Albers et al., 2022) Des Weiteren führen Entwicklungsaktivitäten nicht ausschließlich zu am Markt verfügbaren Produktgenerationen und deren Entwicklungsgenerationen. Es entstehen auch Systeme, die nicht am Markt vertrieben werden aber als Referenz für spätere Produktgenerationen dienen - beispielsweise Prototypen oder Demonstratoren aus der Vorausentwicklung. Ebenso entstehen Systeme, die einem zu entwickelnden Produkt übergeordnet sind und auch gemeinsam mit mehreren Produktgenerationen verwendet werden können - beispielsweise Validierungsumgebungen. Die Entwicklung dieser Systeme kann ebenfalls durch das Modell der PGE nach Albers beschrieben werden. Aufgrund des erweiterten Kontextes wird die Beschreibung der Zusammenhänge dabei als Modell der **Systemgenerationsentwicklung nach Albers (SGE)** bezeichnet. (Albert Albers et al., 2022).

Das **Referenzsystem** bildet den Ausgangspunkt für die Entwicklung des neuen Produkts oder Systems. Die Subsysteme (Elemente) des Referenzsystems selbst basieren wiederum auf bereits bestehenden oder geplanten soziotechnischen Systemen und der dazugehörigen Dokumentation. (Albert Albers et al., 2019; Albert Albers & Rapp, 2022, S. 29) Sie geben einen Großteil der Struktur einer neuen Systemgeneration vor. Der Begriff Struktur umfasst dabei die Aspekte funktionale Struktur, Aufbaustruktur, Lösungsprinzipien und Gestalt. (Albert Albers, Behrendt, Klingler, Reiss & Bursac, 2017, S. 2) Elemente des Referenzsystems können unterschiedlicher Art und Ursprungs sein. Es kann sich beispielsweise um Vorgängerprodukte des gleichen Unternehmens, Wettbewerbsprodukte, existierende Produkte anderer Branchen oder Sparten, Prototypen der Vorentwicklung usw. handeln. Die systematische Suche nach Elementen eines Referenzsystems spielt in der Entwicklung einer ersten Produktgeneration (G_1) eine besondere Rolle – auch wenn das in der Praxis nur selten der Fall ist. (Albert Albers et al., 2022) Im Vorwort zu Yan stellt Albers fest: „Wird [...] ein Entwicklungspfad in einem Unternehmen völlig neu begonnen, so wird [...] von einer Produktgeneration 1 gesprochen, die dadurch gekennzeichnet ist, dass im Referenzsystem als Element keine Vorgänger-Generation vorhanden ist“. (Yan, 2020) Das bedeutet, dass sich das Referenzsystem dieser ersten Generation zwar aus Produkten und Systemen von Wettbewerbern, anderen Branchen und eigenen Lösungen zusammensetzt, jedoch keinen Vorgänger des zu entwickelnden Produkts enthält. (Yan, 2020) Nach Albers et. al. lässt sich eine Generation unter anderem anhand der folgenden Aspekte als G_1 identifizieren: wenig Vertrautheit mit Stakeholdern/-Nutzen & Rahmenbedingungen des Systems, hoher Anteil an Ausprägungs- und Prinzipvariationen sowie deren Beitrag zur Funktion des Systems und geringe Erfahrung mit dem verknüpften Validierungssystem und Produktionssystem. (Albert Albers, Ebertz et al., 2020) Fehlendem Erfahrungswissen und technologischem Wissen gepaart mit einer hohen Unsicherheit bezüglich der Anforderungen an das Produkt wird im ersten Schritt durch die systematische Suche nach Elementen des Referenzsystems begegnet. Im Laufe der Entwicklung auftretende Fragestellungen werden neben Validierungs- und Entwicklungsaktivitäten durch kontinuierliche Erweiterung des Referenzsystems adressiert. (Albert Albers et al., 2022). Eine neue Produktgeneration muss ausreichende Differenzierungsmerkmale zu bestehenden Generationen bzw. Referenzsystemen haben um wirtschaftlich erfolgreich zu sein. (Albert Albers, Bursac & Wintergerst, 2015) Abbildung 2.3 zeigt Beispiele für Produktgenerationen und deren zugrundeliegende Referenzsysteme.

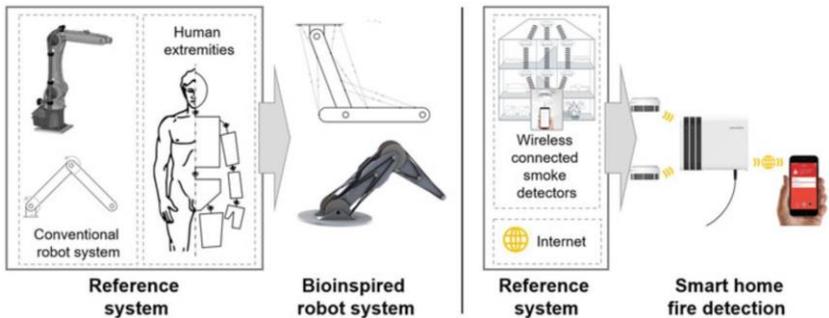


Abbildung 2.3: Beispiele für unterschiedliche Referenzsysteme und die daraus abgeleiteten Produktgenerationen (Albert Albers & Rapp, 2022, S. 30).

Die Gesamtheit eines Systems lässt sich entsprechend Formel 2.1 durch seine Übernahme-, Ausprägungsvariations- und Prinzipvariations-Anteile beschreiben. Bei einer **Übernahmevariation** wird ein Element aus dem Referenzsystem in die neue Systemgeneration übertragen. Das Innere dieses Elements wird als "Black Box" betrachtet, sodass zur Systemintegration nur Änderungen an den Schnittstellen - entsprechend den Anforderungen und Randbedingungen der umgebenden Subsysteme - vorgenommen werden. (Albert Albers, Rapp et al., 2020, S. 2241) Bei einer **Ausprägungsvariation** werden die Verknüpfungen der Elemente innerhalb des Referenzsystems in der neuen Systemgeneration beibehalten. Das Lösungsprinzip bleibt also im Vergleich zum Referenzsystem unverändert. Allerdings werden die Ausprägungen der Elemente variiert. Der Begriff Ausprägung trägt dabei der Tatsache Rechnung, dass nicht nur Gestalteigenschaften, sondern - insbesondere bei mechatronischen Systemen - auch andere Eigenschaften wie z. B. Funktionsparameter einer Variation zugrunde liegen können. (Albert Albers, Rapp et al., 2020, S. 2241) Bei einer **Prinzipvariation** werden Elemente des Referenzsystems und deren Verknüpfung variiert. Elemente und Verknüpfungen können dabei entfernt und hinzugefügt werden, sodass im Vergleich zum Referenzsystem ein neues Lösungsprinzip umgesetzt wird. (Albert Albers, Rapp et al., 2020, S. 2241) Eine Prinzipvariation geht immer auch einher mit einer Ausprägungsvariation, da die Ausprägungen der angepassten Prinziplösung spezifiziert werden müssen. (Albert Albers & Rapp, 2022, S. 31)

Die mathematische Modellbildung der PGE bzw. SGE hat unter anderem zum Ziel, auf Basis der Variationsanteile projekt- und situationsspezifisch die Analyse von

Entwicklungsrisiko und Tragweite eines geplanten Entwicklungsprozesses zu unterstützen. (Albert Albers, Bursac & Wintergerst, 2015) Daran wird deutlich, dass der Ansatz nicht nur die Beschreibung von Produktentstehungsprozessen ermöglicht, sondern durch die Klassifizierung von Variationsarten und -anteilen auch die Basis für die Weiterentwicklung methodischer Unterstützung bietet. (Albert Albers, Rapp et al., 2020, S. 2243)

Zwischenfazit

Das Modell der Systemgenerationsentwicklung nach Albers beschreibt das Verständnis, dass die Einteilung einer Produktentwicklung als Ganzes in Kategorien wie Neuentwicklung oder Anpassungsentwicklung reale Entwicklungsprozesse nicht beschreibt und daher nicht zielführend ist. Jede Produktentwicklung beinhaltet Übernahme-, Ausprägungs- und Prinzipvariationen, die auf Referenzsystemen beruhen. Das bezieht sich nicht nur auf die Planung und Synthese der Endprodukte für den Markt, sondern auch auf deren Derivate und umgebende Systeme, wie z. B. Validierungsumgebungen. Das Modell der SGE nach Albers ermöglicht die Identifikation dieser Variationsanteile und bietet damit die Basis, Produktentstehungsprozesse realitätsnah zu beschreiben und passende Methoden zu entwickeln.

2.1.4 Integriertes Produktentstehungs-Modell (iPeM)

Das integrierte Produktentstehungs-Modell (iPeM) führt die in Kapitel 2.1.1, Kapitel 2.1.2 und Kapitel 2.1.3 beschriebenen Beschreibungsmodelle zusammen und stellt somit ein umfassendes Metamodell zur Beschreibung von Produktentstehungsprozessen zur Verfügung. Abbildung 2.4 zeigt das Modell und seine Bestandteile im Überblick.

Das Systemtripel aus Zielsystem, Handlungssystem und Objektsystem (vgl. Kapitel 2.1.1) ist die Basis für das integrierte Produktentstehungsmodell. (Albert Albers & Meboldt, 2007, S. 5) In Abbildung 2.4 befindet sich das Zielsystem links als eines der verbindenden Elemente zwischen den Ebenen. Die Zielsysteme der Ebenen beeinflussen sich gegenseitig. Auf der rechten Seite sind die Objektsysteme der Ebenen dargestellt, deren Objekte eigenständige Systeme oder Modelle darstellen. Zwischen Ziel- und Objektsystem ist das Handlungssystem dargestellt. Es ist in mehrere Aspekte unterteilt. Unter dem Verständnis, dass Produktentstehung grundsätzlich als Problemlösung verstanden werden kann, wobei sich das Gesamtproblem entsprechend der Systemtheorie in Teilprobleme aufspalten lässt (vgl. Kapitel 2.1.2), werden innerhalb des Handlungssystems Aktivitäten beschrieben.

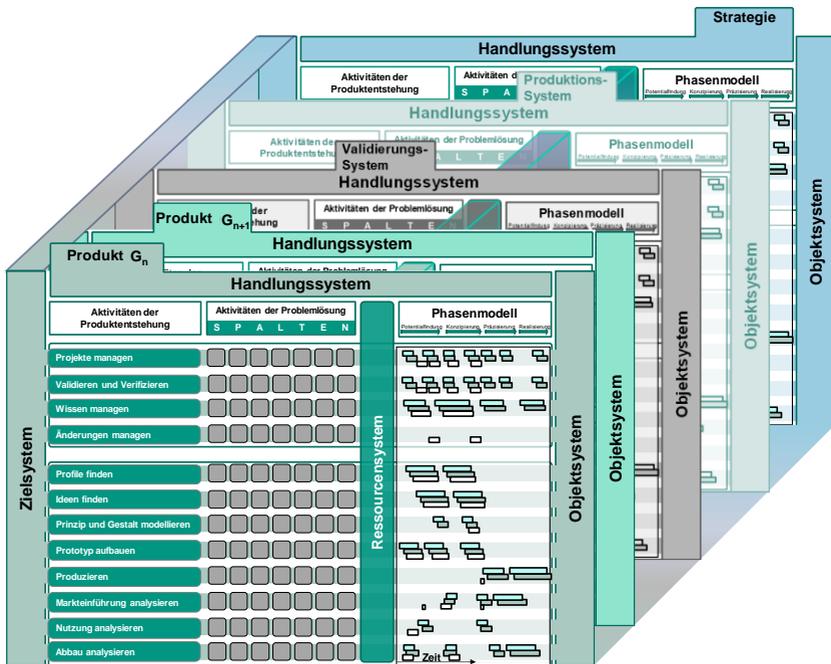


Abbildung 2.4: iPeM - integriertes Produktentstehungs-Modell nach (Albert Albers, Reiß, Bursac & Richter, 2016, S. 104).

Abbildung 2.5 zeigt die Aktivitäten innerhalb einer Ebene des iPeM. Die **Aktivitäten der Produktentstehung** repräsentieren Tätigkeiten zur Lösung typischer Herausforderungen während des Lebenszyklus eines Produkts. (Albert Albers, 2010) **Profile finden**: Anwendungsfälle der Nutzer werden identifiziert und analysiert, auf dieser Basis werden die gewünschten Eigenschaften des zukünftigen Produkts beschrieben. **Ideen finden**: Es werden Ideen, Ansätze und Lösungen für die Erfüllung des Profils bzw. des konkretisierten Zielsystems erarbeitet. **Prinzip und Gestalt modellieren**: Die Produktidee wird unter Berücksichtigung technischer und wirtschaftlicher Aspekte in Funktions-Gestaltzusammenhänge umgesetzt. **Prototyp aufbauen**: Das System inklusive der Funktions-Gestalt-Zusammenhänge wird in Form von physischen, virtuellen oder gemischt physisch-virtuellen Modellen zweckorientiert abgebildet. **Produzieren**: Das Produkt wird durch die parallel zum Produkt

erarbeiteten Produktionsprozesse und -anlagen hergestellt. **Markteinführung analysieren:** Das Produkt wird vermarktet, u.a. beinhaltet dies den Aufbau eines Vertriebsnetzes sowie die Festlegung einer Marketingstrategie. **Nutzung analysieren:** Identifizieren von Verbesserungspotenzial des Produkts und antizipieren des zukünftigen Nutzerverhaltens durch Beobachtung des aktuellen Nutzerverhaltens. **Abbau analysieren:** Identifizieren und antizipieren von Recycling-Möglichkeiten am Ende der Produktlebensdauer. (Albert Albers, Behrendt et al., 2017, S. 5)

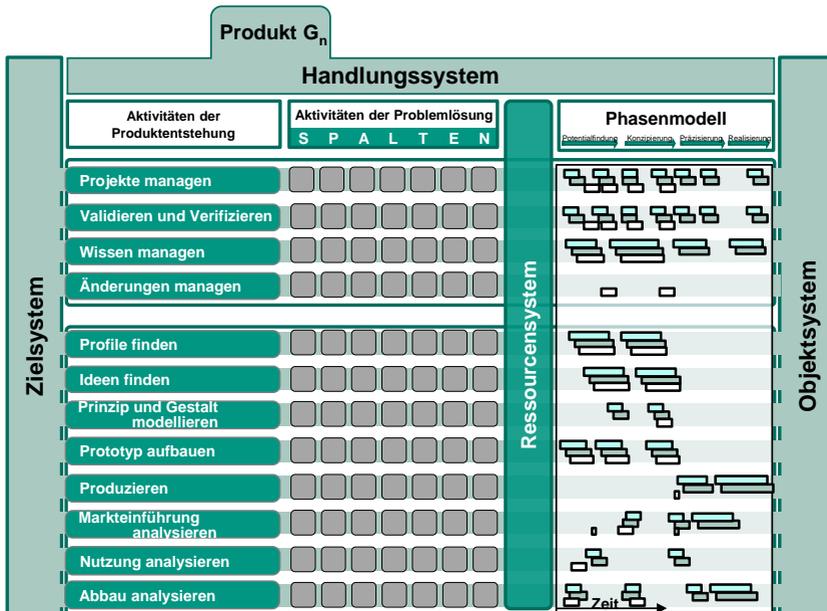


Abbildung 2.5: Aktivitäten der Produktentstehung und Aktivitäten der Problemlösung innerhalb einer Ebene des iPeM nach (Albert Albers, Reiß, Bursac & Richter, 2016, S. 104)

Parallel zu diesen Aktivitäten gibt es Basisaktivitäten, die ebenfalls Teil der Arbeit eines Entwicklungsteams sind. **Wissen managen:** Verfügbare interne und externe Daten, Informationen und Fähigkeiten werden strukturiert und zur Verfügung gestellt. (Albert Albers, Reiß, Bursac & Richter, 2016, S. 103) **Änderungen managen:** Änderungsbedarfe, z. B. aufgrund erkannter Fehler, Optimierungspotenzialen oder geänderter Kundenanforderungen werden erfasst, bewertet und die Umsetzung in das Projekt eingeplant. (Albert Albers, Reiß, Bursac & Richter, 2016, S. 103) **Projekte managen:** Der Projektverlauf wird vom Kick-Off bis zur Produktion geplant,

überwacht und gesteuert. Dazu gehört u. a. definieren des Entwicklungsteams, Kommunikation mit den Stakeholdern sowie die Überwachung von Zeit, Kosten und Qualität. (Albert Albers & Braun, 2011, S. 9) **Validieren und Verifizieren:** Der aktuelle Entwicklungsstand wird kontinuierlich mit dem im Zielsystem beschriebenen gewünschten Zielzustand unter Berücksichtigung der Bedarfe der Anwender abgeglichen. (Albert Albers & Braun, 2011, S. 9) Siehe auch Kapitel 2.3.2.

Die Aktivitäten der Produktentstehung und deren Ergebnisse stehen nicht für sich alleine, sondern dienen der Umsetzung anderer Aktivitäten. (Albert Albers, Reiß, Bursac & Richter, 2016, S. 103) Jede Aktivität der Produktentstehung kann in Teilprobleme strukturiert und wiederum durch die **Aktivitäten der Problemlösung** – also die Schritte der Problemlösungsmethodik SPALTEN (siehe Kapitel 2.1.2) - beschrieben werden. (Albert Albers, 2010)

Die **Ebenen des iPeM** repräsentieren die Bereiche Unternehmensstrategie, Produktionssystem, Validierungssystem und Produktentwicklung unterschiedlicher Generationen. Dabei kann jede Ebene durch eine Kombination der vorgestellten Aktivitäten beschrieben werden. Die Ebene der **Unternehmensstrategie** umfasst die Entwicklung unternehmensspezifische Ziele, Prozesse und Randbedingungen. Dazu gehört beispielsweise das Geschäftsmodell des Unternehmens aber auch das Produktportfolio, die Anzahl der angebotenen Produktvarianten, unternehmensinterne Baukästen etc. (Albert Albers, Reiß, Bursac & Richter, 2016, S. 104) Die Entwicklung des **Produktionssystems** für ein Produkt wird in einer eigenen Ebene repräsentiert. Diese umfasst die Entwicklung und Qualifizierung geeigneter Produktionsprozesse sowie die Entwicklung der benötigten Produktionsanlagen. Dabei ist es möglich, dass das Produktionssystem eines Unternehmens gleichzeitig der Lieferumfang - also das Produkt - eines anderen Unternehmens ist. (Albert Albers, Reiß, Bursac & Richter, 2016, S. 104) Ebenso ist die Entwicklung des **Validierungssystems** in einer eigenen Ebene dargestellt. In diesem Rahmen werden unter Berücksichtigung des zu validierenden Systems und der Validierungsziele die Validierungsstrategien inklusive der zur Umsetzung notwendigen Validierungsumgebungen entwickelt. (Albert Albers, Reiß, Bursac & Richter, 2016, S. 104) Die Ebene des **Produkts** beschreibt die eigentliche Entwicklung eines Produkts oder einer Produkt- bzw. Entwicklungsgeneration. Die Entwicklung jeder Produktgeneration wird dabei in einer eigenen Ebene repräsentiert, in Abbildung 2.4 durch die Indizes der ersten beiden Ebenen angedeutet. (Albert Albers, Reiß, Bursac & Richter, 2016)

Zwischenfazit:

Das iPeM kombiniert unterschiedliche Beschreibungsmodelle der Produktentstehung zu einem universalen Meta-Modell mit dessen Hilfe reale Produktentstehungsprozesse aus unterschiedlichen Blickwinkeln beschrieben werden können. Die Beschreibung mittels iPeM ermöglicht es, Produktentstehungsprozesse zu analysieren und darauf aufbauend zielgerichtet unterstützende Methoden und Prozesse zu entwickeln.

2.2 Ziele und Anforderungen in der Produktentstehung

Der Problemlösungsprozess der Produktentstehung verläuft in der Regel nicht linear, sondern ist geprägt von Erfahrung und Qualifikation des Entwicklungsteams, Iterationen und Rekursionen. Die Produktziele werden dadurch kontinuierlich erweitert und konkretisiert. In diesem Zuge besteht das Risiko, dass die ursprünglichen Ziele des Produkts aus den Augen verloren werden und der Prozess dadurch scheitert. Es ist daher von Bedeutung, während der Entwicklung des Objektsystems die Ziele zu berücksichtigen und transparent zu machen. Die vierte Hypothese der Produktentstehung nach Albers lautet daher: *„Die Elemente des Objektsystems [...] müssen in Bezug zu ihren geplanten Funktionen, die Teil des Zielsystems sind, beschrieben werden um die Transparenz der Ziele des Produkts zu erhalten.“* Um Ziele und Anforderungen nachvollziehbar mit den entstandenen Elementen des Objektsystems verknüpfen zu können, bedarf es einer konsistenten Modellierung des Zielsystems. (Albert Albers, 2010)

2.2.1 Begriffsverständnis Zielsystem, Ziel und Anforderung

„Ein Zielsystem beinhaltet alle expliziten Ziele eines zu entwickelnden Produktes, einschließlich derer Abhängigkeiten und Randbedingungen, innerhalb eines definierten Interessenbereichs (d.h. innerhalb eines System-of-Interest) zu einem bestimmten Zeitpunkt.“ (Lohmeyer, 2013, S. 85) Dabei umfasst es auch die Anforderungen an ein Produkt, die - wie das gesamte Zielsystem - im Laufe eines Entwicklungsprozesses auf Basis von erlangtem Wissen ständig erweitert konkretisiert werden. (Albert Albers & Braun, 2011, S. 7–8) Ziele beschreiben also den zukünftigen, vereinbarten Soll-Zustand und begründen damit die Funktion und Gestalt von Objekten. Anforderungen konkretisieren die Ziele indem sie einzelne Produktmerkmale durch einen Wert oder Wertebereich beschreiben. (Albert Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013, S. 81)

Dieses Verständnis von Anforderung lässt sich mit dem der gängigen Literatur und Richtlinien in Einklang bringen. In der VDI-Richtlinie 2221 ist eine Anforderung *„eine Beschaffenheit, Fähigkeit oder Leistung, die ein Produkt, [...] erfüllen oder besitzen*

muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.“ (VDI-Richtlinie 2221-1) Nach VDI/VDE-Richtlinie 2206 ist eine Anforderung eine *„Beschaffenheit, Fähigkeit oder Leistung eines Produkts, die es erfüllen bzw. besitzen muss.“* (VDI/VDE-Richtlinie 2206) Der VDA QMC definiert Anforderung als *„eine Eigenschaft oder Fähigkeit, die von einem System eingehalten werden muss um einen Vertrag, einen Standard oder andere Vorschriften zu erfüllen.“* (VDA QMC Working Group 13 / Automotive SIG, 2017) Eine Anforderung muss demzufolge eindeutig und messbar sein – Also Produktmerkmale durch Werte oder Wertebereiche beschreiben - um die Vertragserfüllung bewerten zu können. Kriterien für „gute“ Anforderungen sind nach Pohl und Rupp Eindeutigkeit, Konsistenz (zu übergeordneten Anforderungen und zu Anforderungen paralleler Subsysteme) und Prüfbarkeit. (Pohl & Rupp, 2021)

Aus Anforderungen auf einer hohen Systemebene lassen sich Anforderungen auf tieferen Detaillierungsebenen ableiten. Als Detailstufen nennt der VDMA beispielhaft das Gesamtsystem, dessen Subsysteme und die Domänenanteile, aus denen sich die Subsysteme zusammensetzen (Hardware, Software, ...). (VDMA, 2002) Insbesondere durch die erhöhte Komplexität mechatronischer und softwarelastiger Produkte bedarf es methodischer Unterstützung zur Ableitung von konsistenten, wiederverwendbaren Anforderungen über mehrere Ebenen. (Bühne et al., 2004, S. 1)

Aktivitäten, die der Erweiterung und Konkretisierung des Zielsystems dienen, werden häufig unter dem englischen Begriff „Requirements Engineering“ zusammengefasst. Nach Baumgart ist der Zweck von "Requirements Engineering", dass alle relevanten Produkthanforderungen und die zugehörigen Verifikations- und Validierungsmaßnahmen bekannt und abgestimmt sind, eindeutig, klar und schriftlich dokumentiert sind, korrekt und im gesamten Produktlebenszyklus gültig sind und rückverfolgbar sind um relevante Zusammenhänge zwischen Anforderungen oder Anforderungen und Lösungen nachvollziehbar zu machen. Dabei unterscheidet sie die Teilbereiche der Aufgabenklärung (erheben, analysieren und dokumentieren von Anforderungen) und des Anforderungsmanagements (freigeben, versionieren, ändern und rückverfolgen von Anforderungen). (Baumgart, 2016, S. 427) Der VDMA übersetzt und beschreibt den Begriff "Requirements Engineering" als "systematische Durchführung einer Anforderungsanalyse". (VDMA, 2002) Windisch stellt fest, dass der Begriff "Requirements Engineering" sowohl das Spezifizieren von Anforderungen als auch das Anforderungsmanagement umfasst. (Windisch, Mandel, Rapp, Bursac & Albers, 2022, S. 551) Anforderungsmanagement umfasst die Aktivitäten zur Erstellung, Abstimmung und Dokumentation der erforderlichen Informationen über das zu entwickelnde Produkt. (Ponn & Lindemann, 2008)

Zwischenfazit

Für die Entwicklung kundenwerter (Sub-)Systeme ist es essenziell, dass die Anforderungen auch auf tieferen Systemebenen sowohl inhaltlich nachvollziehbar aus den Zielen für das Produkt abgeleitet werden als auch zu diesen zurückverfolgt werden können. Das aufgezeigte Verständnis des englischen Begriffs „Requirements Engineering“ macht das Zusammenspiel dieser beiden Komponenten deutlich. Die nächsten Kapitel gehen daher auf den Stand der Forschung zum Anforderungsmanagement durch Zielsystemmodellierung und auf den Stand der Forschung zur Methodischen Unterstützung der inhaltlichen Anforderungsableitung ein.

2.2.2 Bedeutung der Modellierung von Zielsystemen im Anforderungsmanagement.

Die Rückverfolgbarkeit zwischen Elementen des Zielsystems auf verschiedenen Ebenen (z.B. Domänenanforderungen zu Systemanforderungen) wird durch Methoden des Anforderungsmanagements und der Zielsystemmodellierung unterstützt. Die Elemente des Zielsystems haben sowohl untereinander Abhängigkeiten als auch Verbindungen zu Elementen, die nicht primär dem Zielsystem zugeordnet sind. (Windisch et al., 2022, S. 551) Für ein konsistentes Zielsystem müssen diese Wechselwirkungen in Form von modellierten Beziehungen expliziert werden. Mit starren Anforderungslisten ist das nicht möglich, vielmehr sind Modelle notwendig, um die vorhandenen Wechselwirkungen abzubilden und die dynamische Entwicklung des Zielsystems zu erfassen. (Windisch et al., 2022, S. 551) Bereits Lindemann stellt fest, dass es hilfreich sein kann, Anforderungen mit Modellen zu verknüpfen, da eine Verknüpfung ermöglicht, die für ein Modell relevanten Anforderungen jederzeit nachzuverfolgen. Die Lösungsbewertung kann dadurch vereinfacht werden und der Aufwand zur Dokumentation und Aktualisierung verringert. (Ponn & Lindemann, 2008) Die Möglichkeit, durch MBSE auch unterschiedliche Beziehungstypen zu modellieren (hierarchisch, konkurrierend, etc.) unterstützt die Abbildung von Wechselwirkungen und damit die Tragweitenanalyse bei Änderungen. (Scherer, Albers & Bursac, 2017) Ein weiterer Aspekt im Management eines konsistenten Zielsystems ist auch, dass die Rahmenbedingungen für Anforderungen und Entscheidungen dokumentiert werden können um auch nach mehreren Produkt- oder Entwicklungsgenerationen noch die Gültigkeit einer Anforderung bewerten zu können. (Bursac et al., 2021) Schlussfolgerung: Aus diesen Beobachtungen lässt sich schlussfolgern, dass Anforderungen durch die Verknüpfung mit Entscheidungen und den zugrundeliegenden Informationen beständiger sind bzw. die Tragweitenanalyse bei Anpassungen des Zielsystems erleichtert wird.

Windisch et. al. zeigen in einer Studie auf, dass sich das Anforderungsmanagement in der Praxis von einer dokumentenbasierten Vorgehensweise wandelt, hin zu modellbasiertem Anforderungsmanagement. (Windisch et al., 2022, S. 552) Forschungsseitig gehen u. a. die Arbeit von Ebel (Ebel, 2015) und Scherer (Scherer et al., 2017) darauf ein, wie Zielsysteme mit Hilfe unterschiedlicher Ansätze modelliert werden können. In der Industriellen Praxis kommen aktuell häufig Lösungen wie IBM Doors (IBM, 2021) oder Polarion Requirements (Siemens Product Lifecycle Management Software Inc. © 2022) zum Einsatz (Scherer et al., 2017). Diese ermöglichen auch die Verknüpfung zu Systemmodellen des MBSE und deren Artefakte (siehe Kapitel 2.4).

Bühne et. al. heben die Bedeutung einer Ebenen-Struktur von Anforderungen, Zielen und Anwendungsfällen hervor. Sie strukturieren die Anforderungen an das System Fahrzeug in vier Ebenen: Fahrzeug, System, Funktion und Software. Anforderungen auf tieferen Ebenen hängen dabei nicht zwingend direkt von Anforderungen und Zielen der oberen Ebene ab, vielmehr können sie auch notwendig sein um ein gewünschtes Verhalten auf der aktuellen Ebene zu erreichen. Dieses wiederum kann neben den übergeordneten Zielen zusätzlich vom gewählten Lösungsprinzip oder Konzept abhängen. (Bühne et al., 2004, S. 1) Das ACRE-Framework für modellbasiertes Requirements Engineering beinhaltet sieben Sichten: Quellen Sicht (beinhaltet die Informationen zur Quelle der Anforderungen), Beschreibungs-Sicht (beschreibt die Anforderungselemente auf Basis definierter Attribute), Regel Sicht (beinhaltet Regeln zur Umsetzung der Anforderung, z. B. mathematische Gleichungen oder textuelle Vorgaben), Anforderungskontext Sicht (hier wird die Anforderung im Kontext verschiedener Blickwinkel beschrieben um Fehlinterpretationen zu reduzieren), Kontext Definitions-Sicht (definiert die Blickwinkel der Anforderungskontext Sicht, z. B. Stakeholder), Validierungs-Sicht (beinhaltet Kriterien zur Beurteilung ob die Anforderung erfüllt ist), Rückverfolgungs-Sicht (Zeigt die Verknüpfung zu den ursprünglichen Anforderungen auf). (Holt et al., 2012, S. 563–564) Windisch et. al. stellen die Zusammenhänge von Anforderungen, Entwicklungsaufträgen und Testfällen modellbasiert mit Hilfe von Ansätzen des MBSE dar. (Windisch et al., 2022, S. 553)

Zwischenfazit

Modellbasierte Vorgehen im Ziel- bzw. Anforderungsmanagement unterstützen die Handhabung der Komplexität mechatronischer Produkte. Wichtige Aspekte sind dabei die Berücksichtigung unterschiedlicher Detaillierungsebenen, unterschiedlicher Sichten und Verknüpfungen zu Elementen, die außerhalb des Zielsystems verortet sind. Als zielführend zur Umsetzung dieser Aspekte scheint der Ansatz des MBSE.

Die inhaltliche Nachvollziehbarkeit von Anforderungen hängt davon ab, wie die Anforderungen aus den darüber liegenden Zielen und Anforderungen abgeleitet wurden und wie der Weg der Ableitung dokumentiert und damit für die Mitglieder des Entwicklungsteams zugänglich ist. Im folgenden Kapitel wird daher auf die methodische Unterstützung der inhaltlichen Ermittlung von Zielen und Anforderungen eingegangen.

2.2.3 Methodische Unterstützung der Ziel- und Anforderungsermittlung

Zur Klärung und Ermittlung von Anforderungen nennt Baumgart unter anderem Benchmarking, die Auswertung von Beschwerdesystemen, Gebrauchstest, Interviews und Workshops mit Anwendern. Zur Transformation in lösungsneutrale, prüfbare und eindeutige Anforderungen dient nach Baumgart die Methode des "House of Quality" als Hilfsmittel. Die transformierten Anforderungen können mit Hilfe von Klassifizierungsmerkmal-Listen konkretisiert werden. (Baumgart, 2016)

Der VDA stellt ein Vorgehen zur Anforderungsanalyse in drei Schritten vor: Projektidee formulieren, Grobkonzept erarbeiten, Systemspezifikation erstellen. Zu jedem dieser Schritte werden neben Ziel und erwartetem Ergebnis auch mögliche Methoden und Maßnahmen aufgeführt. Für die Formulierung der Projektidee und die Erarbeitung des Grobkonzepts werden Methoden wie Mindmapping, Moderationstechniken, Line of Visibility und natürlichsprachliche Methoden vorgeschlagen. Der Fokus liegt dabei auf der Kommunikation mit den Beauftragenden und Anwendenden des Systems mit dem Ziel, deren Bedarfe zu erfassen. Als wichtigste Anforderungsquellen nennt der VDMA Kund/-innen, Management, System-Anwendende, Service, Marketing/Vertrieb, Entwicklung, Prüfer/-innen und Auditor/-innen. Im Rahmen der Konkretisierung des Zielsystems werden verschiedene Anforderungen erarbeitet, festgehalten und dokumentiert. Dabei kann die Ableitung funktionaler Anforderungen aus Use Cases durch die detaillierte Beschreibung des Ablaufs eines Use Cases erfolgen. Die Ableitung nicht-funktionaler Anforderungen werden aus den Use-Cases und Randbedingungen abgeleitet. Wobei Randbedingungen unterschiedlicher Art sein können, z. B. organisatorisch, technologisch, juristisch, kulturell. (VDMA, 2002) Ponn und Lindemann schlagen mit Bezug auf die jeweiligen Quellen mehrere Methoden zur Identifikation von Anforderungen vor, u. a. die Nutzung von Checklisten, die Analyse des Produktlebenszyklus mittels Suchmatrix, Benchmarking-Aktivitäten, Mindmaps, Wirknetze oder Analyse der Systemstruktur durch strukturierte Fragen (Relationale Iterative Anforderungsklä rung). (Ponn & Lindemann, 2008) In Bezug auf System of Systems (SoS) stellen Holt et. al. ein Vorgehen zur Anforderungsentwicklung vor, das auf der Analyse des Kontextes der Anforderung beruht. (Holt et al., 2012, S. 564) Lewis et. al. schlagen ebenfalls mit

Bezug auf SoS vor, nach der Kontextbeschreibung und Beschreibung der Ziele der einzelnen Systeme, die Wechselwirkungen zwischen den Systemen zu analysieren und zu beschreiben, sodass die benötigten Funktionen und Schnittstellen der Systeme identifiziert werden können. Parallel dazu werden die einzelnen Systeme hinsichtlich ihrer bereits vorhandenen Funktionen und Schnittstellen analysiert. In einem weiteren Schritt werden aus diesen Analysen die Lücken identifiziert, die im Rahmen der Entwicklung geschlossen werden müssen. (Lewis, Morris, Place, Simanta & Smith, 2009) Simpson, Bobuk et al. setzen in Ihrem Ansatz für die Anforderungsentwicklung von Produktfamilien sowohl qualitative (z. B. Marktsegmentierungsraster, GVI) als auch quantitative Methoden (z. B. Mehrzieloptimierung durch Simulation, Gemeinsamkeitsindizes) ein. (Simpson et al., 2012, S. 151)

Zwischenfazit

Viele Ansätze zur Anforderungsermittlung fokussieren auf das Ermitteln, Verstehen und Strukturieren von Markt- und Stakeholderbedürfnissen, deren Erfassung und Analyse ist notwendige Voraussetzung für ein erfolgreiches Produkt ist. Hierbei kommen überwiegend qualitative Methoden zum Einsatz. Allerdings müssen diese Anforderungen auf oberster Ebene systematisch über den Produktentstehungsprozess in eindeutige, testbare und untereinander konsistente Anforderungen an Subsysteme heruntergebrochen werden. Methodische Ansätze hierfür finden sich in quantitativen Methoden wie Berechnung und Simulation in Kombination mit Methoden des Systems Engineering.

2.2.4 Prozessuale Normen und Vorgaben in der Automobilentwicklung

Die Einhaltung von Normen, Richtlinien und weiteren Prozessvorgaben wird in der industriellen Praxis durch Prozessassessments sichergestellt. Ein Assessment erfolgt durch die Analyse und Bewertung der Prozesse einer Organisationseinheit anhand eines Prozessmodells.

Das Automotive SPICE (ASPICE) Prozessassessmentmodell dient der Bewertung der Prozessfähigkeit bei der Entwicklung eingebetteter Systeme für die Automobilindustrie. Für diesen Zweck enthält es neben einem Prozessreferenzmodell (Abbildung 2.6) auch verschiedene Indikatoren in Form von erwarteten Arbeitsprodukten und grundlegenden Vorgehen, die in der Bewertung zu berücksichtigen sind. (VDA QMC Working Group 13 / Automotive SIG, 2017) Das Referenzprozessmodell orientiert sich am V-Modell nach VDI 2206 und ist in mehrere Prozessgebiete strukturiert. Eines der Prozessgebiete ist die Systementwicklung. Hier wird Systems Engineering gefordert und Anforderungen an die konkrete Umsetzung gestellt. Systems

Engineering umfasst hierbei die Erhebung von Stakeholderanforderungen (SYS.1), die darauf aufbauende Ableitung und Verfeinerung von Systemanforderungen (SYS.2), die Erstellung einer Systemarchitektur (SYS.3) sowie die Überprüfung der Anforderungserfüllung auf Subsystemebene in Integrationstests (SYS.4) und auf Gesamtsystemebene in Qualifikationstests (SYS.5). Indikatoren für die Prozesserrückmeldung der Systemarchitektur sind dabei u.a. die Verknüpfung von Systemfunktionen und Systemelementen auf mehreren Detaillierungsebenen und die damit einhergehende Identifikation von Schnittstellen, die Allokation der Systemanforderungen auf die Architekturelemente inklusive bidirektionaler Rückverfolgbarkeit, und die Sicherstellung der Konsistenz zwischen Anforderungen und Systemarchitektur. (VDA QMC Working Group 13 / Automotive SIG, 2017)

Die Norm ISO 26262 stellt sowohl technische Anforderungen als auch Prozessanforderungen an die Entwicklung sicherheitsrelevanter Systeme, die elektrische und/oder elektronische (E/E) Systeme umfassen und in Serien-Straßenfahrzeugen zum Einsatz kommen. Ziel ist, mögliche Gefahren durch Fehlfunktionen von E/E-Systemen zu minimieren. Die Erfüllung der Anforderungen der ISO 26262 wird durch Prozessaudits und durch projektspezifische Assessments bewertet. (ISO 26262-2) Abbildung 2.7 stellt einen Überblick über die zentralen Teile der Norm dar. Teil drei der Norm spezifiziert die Anforderungen für die Konzeptphase der Automobilentwicklung auf Fahrzeugebene, das umfasst die Definition des Betrachtungsraums (engl.: item definition), die Gefahrenanalyse und Risikobewertung (engl.: Hazard analysis and risk assessment (H&R)) zur Bestimmung des Sicherheitslevels und die Definition des funktionalen Sicherheitskonzepts. Zur Beschreibung des Betrachtungsumfanges gehört die Kontextbeschreibung in Form von umgebenden Systemen und Wechselwirkungen mit diesen sowie Anwendungsfälle. Ebenso müssen die gesetzlichen und normativen Anforderungen, Qualitäts- und Leistungsanforderungen identifiziert werden. Aus den im Rahmen der H&R identifizierten Sicherheitszielen werden im Rahmen der Erstellung des funktionalen Sicherheitskonzepts funktionale Sicherheitsanforderungen abgeleitet, die zu Elementen der Systemarchitektur allokiert werden sollen. (ISO 26262-3) Teil vier der Norm beschreibt die Anforderungen an die Systementwicklung. Das umfasst insbesondere die Entwicklung des technischen Sicherheitskonzepts und der entsprechenden technischen Sicherheitsanforderungen, der Entwurf einer Systemarchitektur, die Systemintegration und -prüfung und die Validierung der Sicherheit. (ISO 26262-4)

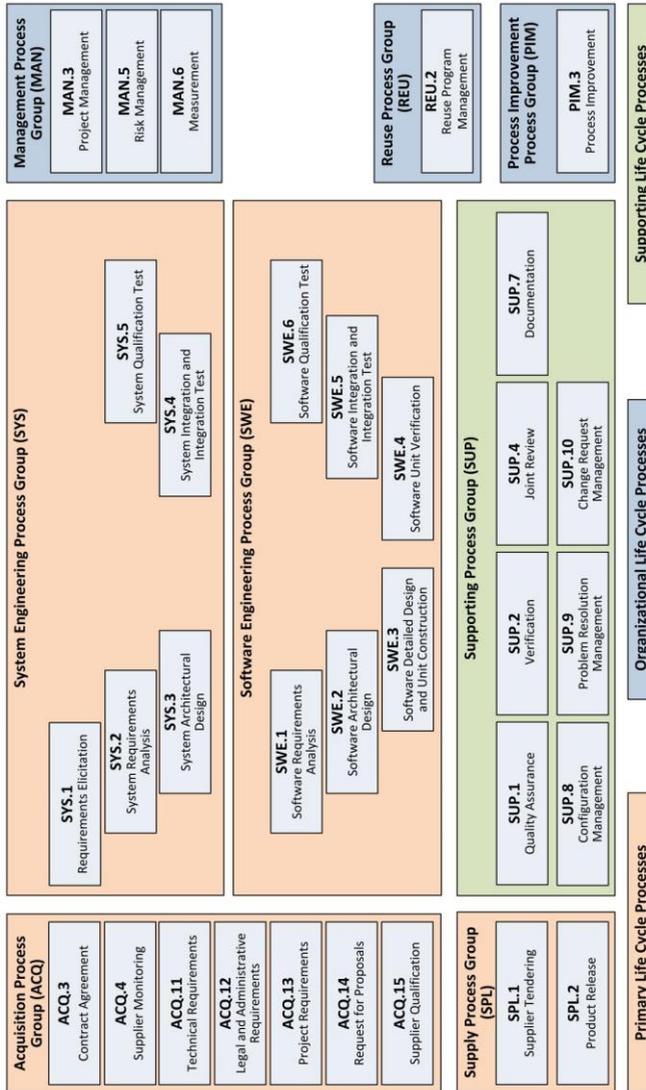


Abbildung 2.6: ASPICE Referenzprozessmodell Version 3.1 im Überblick (VDA QMC Working Group 13 / Automotive SIG, 2017).

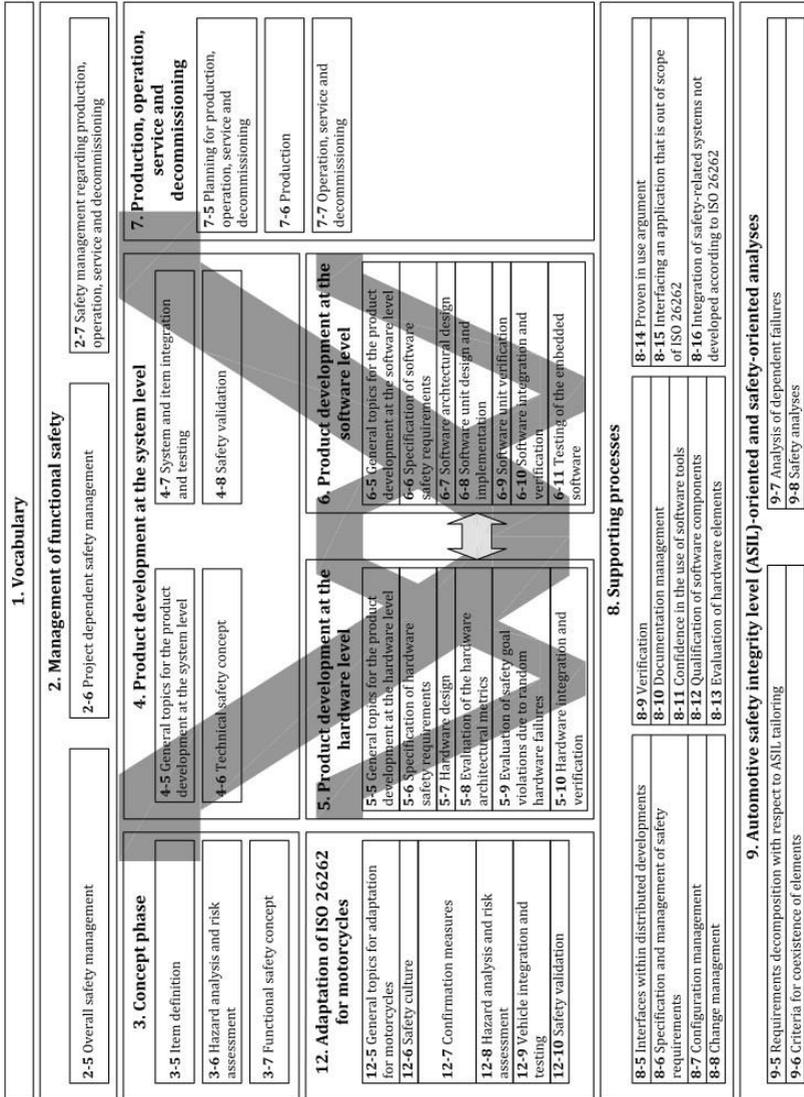


Abbildung 2.7: Übersicht über die Teile der ISO 26262 (ISO 26262-2).

Zwischenfazit:

In der Automobilindustrie kommen in den letzten Jahren normative Regelungen und Prozessstandards dazu, die den Einsatz von Systems Engineering und auch konkret die Erstellung von Systemarchitekturen fordern. Produkthanforderungen sollen über mehrere Detaillierungsebenen hinweg nachverfolgbar abgeleitet und miteinander verknüpft werden. Zentrale Motivation ist dabei ein klares und expliziertes Bild der Systemzusammenhänge. Zusätzlich wird aus den vorangehenden Kapiteln deutlich, dass auch eine intrinsische Motivation der Unternehmen besteht, Systems Engineering Ansätze und insbesondere MBSE in der Entwicklung komplexer mechatronischer und cyberphysischer Systeme einzuführen.

2.3 Validierung und Verifikation im Produktentstehungsprozess

Validierung ist entsprechend der dritten Hypothese der Produktentstehung nach Albers „*die zentrale Aktivität der Produktentstehung*“. (Albert Albers, 2010). Durch Validierung entsteht Wissen über das untersuchte System, das es ermöglicht Entwicklungsentscheidungen zu treffen und dadurch das zu entwickelnde Produkt sowie dessen Ziele und Anforderungen weiter zu konkretisieren. (Albert Albers, Behrendt & Ott, 2010, S. 5)

2.3.1 Begriffsverständnis Validierung und Verifikation

Abbildung 2.8 stellt die Schritte der Validierung und Kreation im Kontext der kontinuierlichen Analyse und Synthese im erweiterten ZHO-Modell (siehe Kapitel 2.1.1 und Abbildung 2.1) dar. Validierung erfolgt dabei durch die Analyse von Elementen des Objektsystems und die folgende Synthese zum Zielsystem. Demzufolge wird Validierung als Abgleich der aktuellen Produkteigenschaften mit den Zielen des Produkts verstanden. (Albert Albers, Behrendt, Klingler & Matros, 2016, S. 541) Dieses Verständnis findet sich auch in der Definition des VDI/VDE wieder, der Validierung als Prüfung der Systemeigenschaften bezüglich des definierten Einsatzzwecks oder Nutzens beschreibt. (VDI/VDE-Richtlinie 2206)

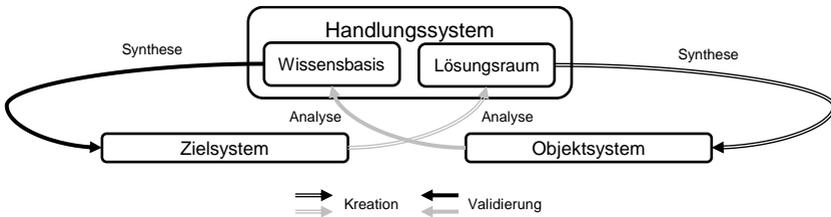


Abbildung 2.8: Validierung als Teil des erweiterten ZHO-Modells nach (Albert Albers, Behrendt, Schroeter, Ott & Klingler, 2013).

Im Unterschied zur Validierung beschreibt der VDI/VDE den Begriff der Verifikation als Prüfung der Systemeigenschaften bezüglich der Spezifikation auf den jeweiligen Systemebenen und Konkretisierungsniveaus. Die Spezifikation ist dabei die formale Beschreibung eines Systems durch prüfbare Anforderungen zum Zweck der Abnahme. (VDI/VDE-Richtlinie 2206) Auch Albers et. al. definieren Verifikation als Abgleich aktueller Produktfunktionen und -eigenschaften mit den spezifizierten Anforderungen, unabhängig davon ob die Spezifikation bezüglich der Produktziele richtig und konsistent ist. (Albert Albers, Behrendt et al., 2016, S. 542)

Der VDI/VDE fasst die Aktivitäten der Verifikation und Validierung unter dem Begriff Eigenschaftsabsicherung zusammen. (VDI/VDE-Richtlinie 2206) Albers et. al. werden konkreter und beschreiben die Aktivitäten im Bezug zueinander. Validierung umfasst den Abgleich von Produkt und Spezifikation (Verifikation) und darüber hinaus die Überprüfung, ob die Spezifikation konsistent mit den Produktzielen ist. Verifikation ist damit - wie auch in Abbildung 2.9 dargestellt - eine Teilaktivität der Validierung. (Albert Albers, Behrendt et al., 2016)

2.3.2 Validierung im Kontext der Produktentstehung und Teilaktivitäten der Validierung

Alle Basisaktivitäten und Aktivitäten der Produktentstehung lassen sich entsprechend Kapitel 2.1.4 als Problemlösungsprozesse beschreiben, die Entscheidungen bedürfen. Die Qualität von Entscheidungen hängt dabei von der verfügbaren Wissensbasis ab, daher muss es Ziel einer effizienten Produktentwicklung sein, möglichst frühzeitig Wissenslücken zu schließen und Unsicherheiten zu reduzieren. Demzufolge sollte Validierung bereits in frühen Phasen der Produktentstehung stattfinden und kontinuierlich über den gesamten Produktentstehungsprozess hinweg

fortgeführt werden. Das Pull-Prinzip der Validierung setzt diesen Anspruch um, indem es bewusste, frühzeitige Validierungsschritte vorsieht. Design- und Validierungsaktivitäten laufen dabei eng verzahnt in kurzen Iterationen ab, wie es auch die Grundidee agiler Entwicklungsansätze ist. Dies führt zu einem "atmenden" Prozess aus Aufweitung der Wissens- und/oder Lösungsvielfalt und Eingrenzung dieser durch Entscheidung. (Albert Albers, Matros, Behrendt & Jetzinger, 2015) Das dadurch früh erarbeitete Wissen dient als Grundlage für die nächsten Entwicklungs-generationen und stößt weitere Entwicklungstätigkeiten an. (Albert Albers, Behrendt et al., 2016, S. 558)

Startpunkt von Validierungsaktivitäten ist nach Albers et. al. immer eine Wissenslücke, diese kann unterschiedlich ausgeprägt sein. Es kann sowohl Ziel sein, Wissen über konkrete technische Anforderungen zu erzeugen, als auch eher vage Hypothesen zu untermauern. (Albert Albers, Behrendt et al., 2016, S. 554) Um die Wissenslücke valide zu verkleinern oder zu schließen, sind mehrere Teilaktivitäten nötig (Abbildung 2.9).

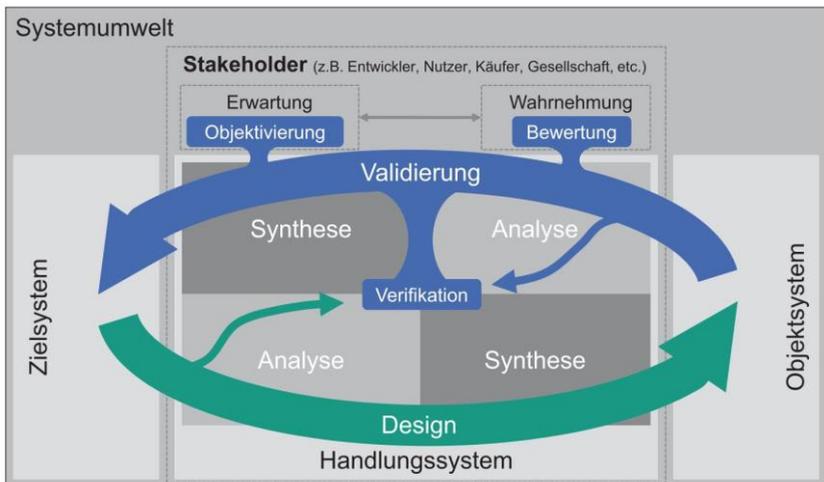


Abbildung 2.9: Design und Validierung im Produktentstehungsprozess (Albert Albers, Behrendt et al., 2016, S. 545).

Wie bereits in Kapitel 2.3.1 dargelegt, ist ein Bestandteil die Verifikation – also der Abgleich, ob das untersuchte System seinen spezifizierten Anforderungen entspricht. Darüber hinaus müssen die umgesetzten Systemmerkmale und –funktionen

aus der subjektiven Wahrnehmung der betroffenen Stakeholder heraus bewertet werden. Das ermöglicht einen Rückschluss auf die empfundene Wertigkeit und Zweckerfüllung des aktuellen Entwicklungsstandes. Im Schritt der Objektivierung werden Zusammenhänge zwischen quantitativen Größen und den Wahrnehmungen der Stakeholder hergestellt. Damit dient die Objektivierung der Überführung der Bewertung in eine Korrektur, Ergänzung oder Verfeinerung der Inhalte des Zielsystems. (Albert Albers, Behrendt et al., 2016) Nur das Zusammenspiel aller drei Aktivitäten – Verifikation, Bewertung und Objektivierung - ermöglicht das Erreichen von Validität. Nur eine oder zwei dieser Aktivitäten sind nicht ausreichend, um die Validität eines Produktes zu beurteilen und in Folge zu verbessern. (Albert Albers, Matros et al., 2015, S. 76)

Die Analyse des Validierungsgegenstandes kann z. B. durch Tests erfolgen. Die VDA QMC Arbeitsgruppe definiert Testen als Aktivität bei der ein Element (System, Hardware oder Software) unter definierten Randbedingungen betrieben bzw. ausgeführt wird mit dem Ziel, die Ergebnisse zu dokumentieren, zusammenzufassen und zu kommunizieren. (VDA QMC Working Group 13 / Automotive SIG, 2017) Ein Test muss dabei nicht zwingend mit physischen Instanzen des Produkts durchgeführt werden, sondern kann auch auf Basis virtueller Modelle oder gemischt physisch-virtuell stattfinden. Der Testfall beschreibt dabei nach Ebel die Eingangsgrößen und die Randbedingungen des Tests, beinhaltet aber auch das erwartete Verhalten des beobachteten Systems. Er bildet damit einen oder mehrere Anwendungsfälle des Produkts ab. (Ebel, 2015) In der Automobilentwicklung können Testfälle z. B. auf Gesamtfahrzeugebene konkret durch Manöver beschrieben werden. Ein Manöver beinhaltet dabei das Fahrerverhalten, den Zustand sowie die Eigenschaften des Restfahrzeugs und der Umwelt. Des Weiteren werden Validierungsziel sowie Analyse- und Bewertungskriterien dem Manöver zugeordnet. (Albert Albers, Behrendt et al., 2016, S. 556)

Die definierten Testfälle müssen in geeigneter Weise umgesetzt werden, das geschieht in der Test- bzw. Validierungsumgebung. Diese stellt alle Modelle (physisch, virtuell oder gemischt virtuell-physisch) bereit, die für die Durchführung des Tests nötig sind. (Ebel, 2015) Die Validierungsumgebung muss dabei abhängig vom Entwicklungsstand eines Produkts und dem jeweiligen Validierungsziel für jeden Test individuell definiert werden. (Albert Albers, Behrendt et al., 2016, S. 547)

Das mittels Tests erfasste Systemverhalten ist die zentrale Eingangsgröße von Verifikation, Bewertung und Objektivierung. In der Interpretation der Testergebnisse wird das erfasste Systemverhalten analysiert und Schlussfolgerungen hinsichtlich der Erfüllung von Zielen, Bedarfen und Anforderungen gezogen. Dadurch können neue Erkenntnisse bezüglich des beobachteten Systems abgeleitet werden. (Ebel,

2015) Die Interpretation der Ergebnisse muss dabei jedoch immer vor dem Hintergrund der eingesetzten Testumgebung erfolgen, da für einen zielführenden Erkenntnisgewinn Einflüsse wie z. B. Latenzzeiten oder Modellverkürzungen berücksichtigt werden müssen. (Albert Albers, Behrendt et al., 2016, S. 554)

2.3.3 IPEK-X-in-the-Loop-Ansatz und –Framework

Dem grundlegenden Verständnis, dass zur Validierung eines Systems, dieses in Kontext mit Gesamtsystem, Umwelt und ggf. weiteren interagierenden Systemen eingebunden werden muss, wird im IPEK-X-in-the-Loop-(XiL)-Ansatz Rechnung getragen. Dieser bietet die Grundlage zur systematischen Erarbeitung geeigneter Testfälle und Testumgebungen sowie zur Testinterpretation. Abbildung 2.10 zeigt die entsprechenden Elemente des IPEK-XiL-Ansatzes am Beispiel eines zu entwickelnden Subsystems im Fahrzeug.

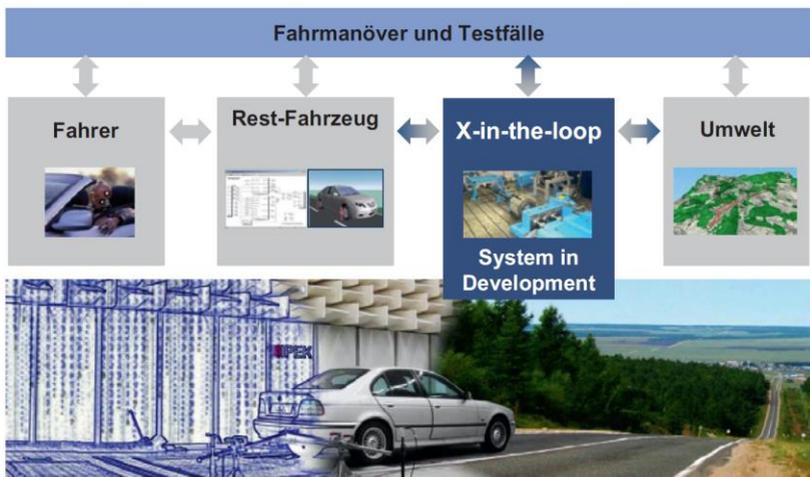


Abbildung 2.10: Elemente und Zusammenhänge des IPEK-X-in-the-Loop-(XiL)-Ansatzes (Albert Albers, Behrendt et al., 2016, S. 559) basierend auf (Albert Albers & Düser, 2008)

Auch wenn nur die Funktion oder die Eigenschaften eines Subsystems untersucht wird, müssen diese im Sinne einer Validierung immer in Kontext zum Gesamtsystem gesetzt werden. Das heißt, die Wechselwirkungen mit den umgebenden Systemen und das Verhalten auf Gesamtsystemebene muss berücksichtigt werden um die Bedarfserfüllung beurteilen zu können. (Albert Albers, Behrendt et al., 2016, S. 559)

Elementar für die Planung und Umsetzung von Validierungsaktivitäten entsprechend dem IPEK-XiL-Ansatz ist das Verständnis über den Systemkontext, die Systemelemente, deren Zusammenhänge und Eigenschaften auf unterschiedlichen Detaillierungsebenen. Düser nutzt zur Systemanalyse und Dokumentation beispielsweise den C&C²-Ansatz und SysML-Diagramme. (Düser, 2010, S. 122)

Das IPEK-XiL-Framework (Abbildung 2.11) zeigt die verschiedenen Ebenen des zu untersuchenden Systems auf und, dass dieses je nach gewähltem Validierungsziel sowohl physisch als auch virtuell oder gemischt virtuell-physisch darstellbar ist. Das gilt auch für die Abbildung des Rest-Modells und die umgebenden Systeme. (Albert Albers, Behrendt et al., 2016, S. 560)

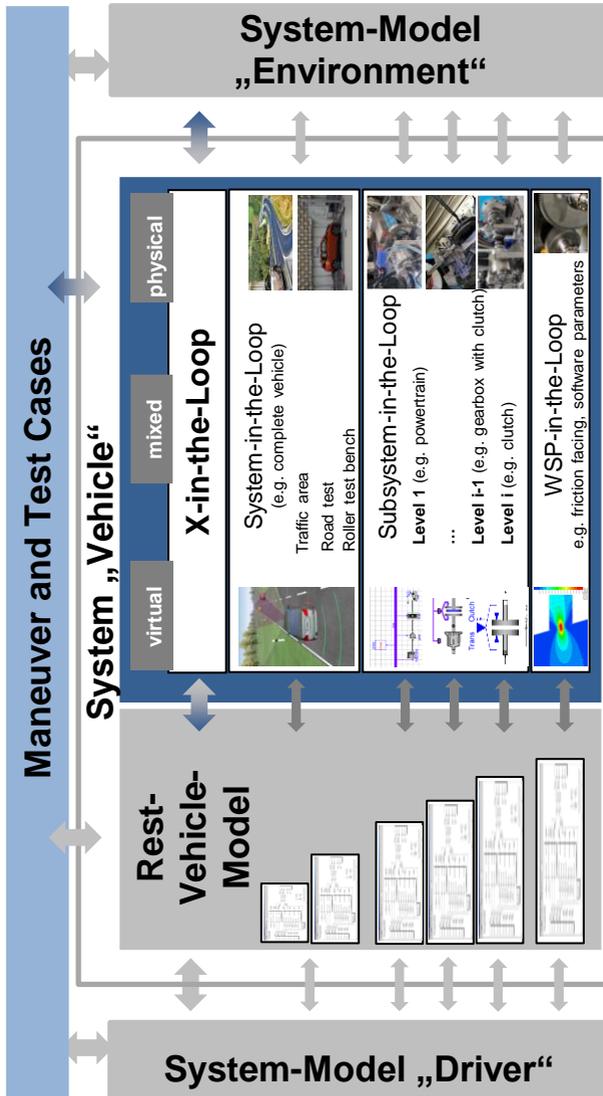


Abbildung 2.11: IPEK-X-in-the-Loop-(XIL)-Framework nach (Albert Albers, Behrendt et al., 2016, S. 560) und (Albert Albers & Düser, 2008).

Das "X" im IPEK-XiL-Ansatz und -Framework ist – in Analogie zu „Software-in-the-Loop (SiL)“ und „Hardware-in-the-Loop (HiL)“ das System, das im Fokus der jeweiligen Validierungsaktivität steht. Dabei können zwei Sichten unterschieden werden: Zum einen kann das "X" das in Entwicklung befindliche (Sub-)System sein (System-in-Development (SiD)), die Validierungsaktivitäten beziehen sich dann hauptsächlich auf die Bewertung der Eigenschafts- und Funktionserfüllung. (Albert Albers, Behrendt et al., 2016, S. 561) Zum anderen kann die Gewinnung von Wissen über ein (Sub-)System zur weiteren Verwendung im Entwicklungsprozess (z. B. zur besseren Modellbildung in der Validierungsumgebung oder in Form von Anforderungen) im Fokus stehen, man spricht dann vom System-under-Investigation (Sul). Ein Sul kann dabei sowohl Bestandteil des eigentlichen SiD sein als auch eines der umgebenden Systeme, wie Abbildung 2.12 beispielhaft zeigt. (Albert Albers, Behrendt et al., 2016, S. 561)

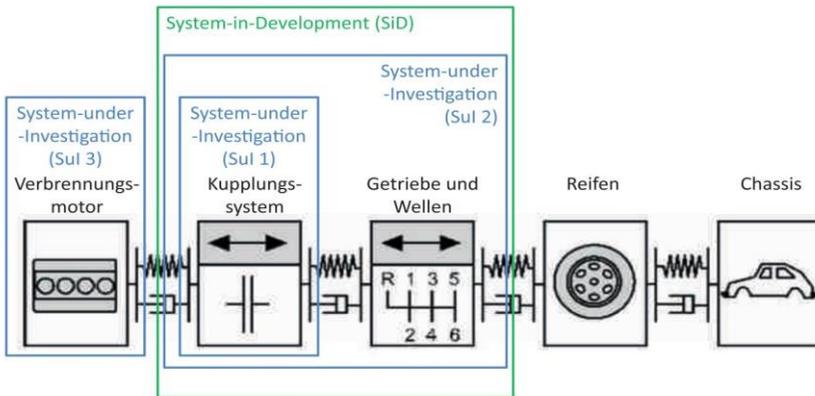


Abbildung 2.12: System in Development (SiD) und System under Investigation (Sul) (Albert Albers, Behrendt et al., 2016, S. 561).

Abbildung 2.13 zeigt die Korrelation zwischen den Elementen des IPEK-XiL-Framework auf der Ebene des Validierungssystems und den Validierungsbedarfen und –ergebnissen auf der Ebene des Produktsystems.

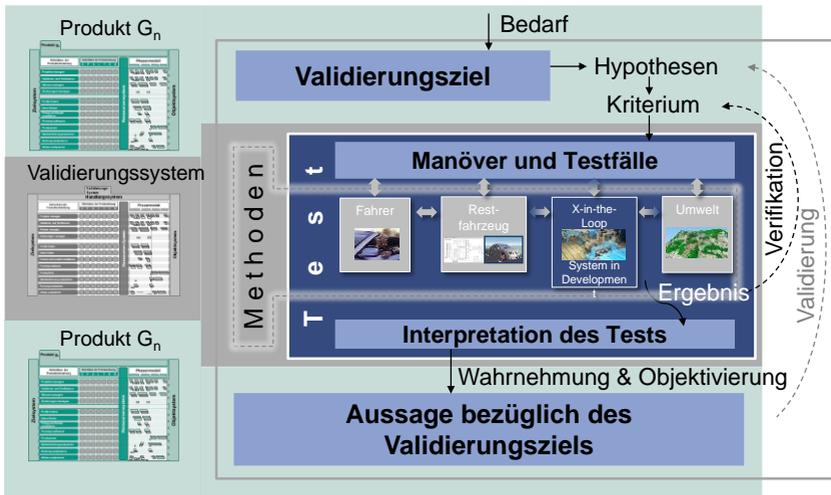


Abbildung 2.13: Korrelation zwischen Produktsystem und Validierungssystem nach (Mandel et al., 2020, S. 2).

Zwischenfazit:

Der IPEK-XiL-Ansatz und das zugehörige Framework ermöglichen, Validierungsaktivitäten unter Berücksichtigung der Kundenanforderungen und Anwendungsfälle, der umgebenden Systeme und des Restsystems systematisch und strukturiert zu planen und zu beschreiben. Dabei können die Modelle der einzelnen Systeme jeweils virtuell, physisch oder gemischt physisch-virtuell sein. Abhängig von der Detailebene, auf der sich das Sul befindet, wird das Restsystem entsprechend dargestellt. Daraus folgt, dass Validierungsaktivitäten auch auf Basis rein virtueller Modelle erfolgen können.

2.3.4 Virtuelles, modellbasiertes Testen

Wie bereits erläutert, kann Validierung nicht nur durch Testen physischer Systeme erfolgen, sondern umfasst auch virtuelle Modelle zur Berechnung und Simulation. (Albert Albers, Behrendt et al., 2016, S. 547) Virtuelle Modelle ermöglichen eine Einschätzung des Systemverhaltens und vereinfachen so Betrachtung und Vergleich von Systemvarianten. (Pohlandt, Brinkschulte & Geimer, 2016) In diesem Zusammenhang ist der Begriff des modellbasierten Testens zu erwähnen. Nach Bringmann und Krämer ist das Ziel von modellbasiertem Testen, das Systemverhalten in

einer virtuellen Umgebung zu prognostizieren und zu analysieren, um so Änderungsbedarfe im Vorfeld der Systemimplementierung effizient identifizieren zu können. Des Weiteren ermöglichen die Ergebnisse einem Entwicklungsteam bereits früh in der Entwicklung ein gemeinsames funktionales und strukturelles Verständnis mechatronischer Systeme aufzubauen. (Bringmann & Krämer, 2008) Nach Utting et. al. birgt modellbasiertes Testen außerdem das Potenzial, verschiedene Eingangsgrößen automatisiert zu variieren und so eine Vielzahl von Testfällen effizient durchzuspielen. (Utting, Pretschner & Legeard, 2006) Auch Albers et. al. stellen fest, dass rein virtuelles Testen früh im Entwicklungsprozess Validierungsaktivitäten ermöglicht. Die Qualität der erreichbaren Aussagen hängt dabei jedoch stark von der Modellbildung ab. Wechselwirkungen und Effekte, die im Modell nicht berücksichtigt sind, sind auch im Ergebnis nicht erfasst. Die Modellbildung für Restsystem, SiD bzw. Sul und umgebende Systeme für eine konkrete Validierungsaktivität geschieht ausgehend vom Verständnis des Entwicklungsteams über die Zusammenhänge im Gesamtsystem. (Albert Albers, Behrendt et al., 2016) Dem entsprechend ist ein gemeinsames und konsistentes Funktions- und Strukturverständnis elementar für die Validierung auf rein virtueller Ebene.

Zwischenfazit:

Virtuelle Modelle spielen insbesondere für Analysen und Validierungsaktivitäten in den frühen Phasen des Entwicklungsprozesses aber auch über den gesamten Prozess begleitend eine große Rolle. Das gleiche System wird je nach Betrachtungsrahmen und –ziel durch unterschiedliche virtuelle Modelle dargestellt. Da alle Modelle dennoch ein Abbild desselben Systems sind ist für Verwendung und Vergleichbarkeit der Ergebnisse die inhaltliche Konsistenz der Modelle wichtig.

2.4 Modellbasiertes Systems Engineering

Die zunehmende Komplexität und Interdisziplinarität von Produkten führen zur Notwendigkeit, diese durch entsprechende Ansätze und Methoden in der Produktentwicklung zu handhaben. (Dumitrescu et al., 2021) Klassische Entwicklungsansätze stoßen in der Entwicklung komplexer Systeme an ihre Grenzen. (Albert Albers, 2023, S. 5) Ein Ansatz, um die Transparenz in der Produktentwicklung zu erhöhen und so mit der Komplexität der zu entwickelnden Produkte besser umgehen zu können ist Systems Engineering. Die Vorteile basieren unter anderem auf einer konsistenten Darstellung und Rückverfolgbarkeit von Zusammenhängen der Systeme. (Dumitrescu et al., 2021)

Das International Council of Systems Engineering definiert den Begriff „Systems Engineering“ wie folgt: „Systems Engineering ist ein interdisziplinärer und integrativer Ansatz, um die erfolgreiche Realisierung, Nutzung und Außerbetriebnahme von technischen Systemen zu ermöglichen. Dabei werden systemorientierte Prinzipien und Konzepte verwendet sowie Methoden aus Wissenschaft, Technologie und Management.“ (INCOSE, 2022)

Dem Systems Engineering liegt ein iteratives Vorgehen auf Basis der Systemtechnik zugrunde, dessen Ziel ein Verständnis über die Zusammenhänge von Anforderungen an Systeme, deren Eigenschaften und Verhaltensweisen ist. (Walden et al., 2015) Modellbildung, Simulation und Visualisierung ermöglichen es, Zusammenhänge komplexer Systeme darzustellen, ihr Verhalten zu antizipieren, darauf aufbauend Lösungen zu erarbeiten und diese zu analysieren. (INCOSE, 2014) Dabei erfordert die Modellierung und Simulation unterschiedlicher Systemaspekte (z. B. dynamisches Verhalten, thermisches Verhalten, Strukturmechanik, ...) jedoch ein hohes Maß an Fachwissen. Daraus ergibt sich die Herausforderung, die kontinuierliche und konsistente Anwendung von Modellen in der Produktentwicklung zu unterstützen. (Albert Albers & Lohmeyer, 2012, S. 412) Daher wird der Stand der Forschung zur Verwendung von Modellen und insbesondere zu modellbasiertem Systems Engineering (MBSE) im Folgenden näher beleuchtet.

2.4.1 Idee des Modellbasierten Systems Engineering (MBSE)

Die Idee von MBSE ist laut Gesellschaft für Systems Engineering (GfSE), dass Modelle als zentrale Quelle aller Artefakte im Entwicklungsprozess dienen. (Rambo, Tschirner, Weber, Zuccaro & Dorsch, 2017, S. 171) Dabei sind Modelle entsprechend der Modelltheorie nach Stachowiak die Repräsentation eines Originals (Abbildungsmerkmal), die immer nur einen Anteil der Attribute dieses Originals darstellt (Verkürzungsmerkmal), nämlich denjenigen Anteil, der für den Zweck, dem das Modell dient, notwendig ist (pragmatisches Merkmal). (Stachowiak, 1973) Ein System setzt sich nach Ropohl aus mehreren, miteinander in Verbindung stehenden Elementen bzw. Subsystemen zusammen (struktureller Aspekt), es lässt sich von seiner Umgebung abgrenzen (hierarchischer Aspekt) und erfüllt Funktionen, indem es Eingangsgrößen aus der Umgebung aufnimmt und Ausgangsgrößen an die Umgebung abgibt (funktionaler Aspekt). (Ropohl, 1975) Mit diesem Verständnis von Modell und System wird deutlich, dass in der Produktentstehung viele verschiedene Modelle von existierenden oder zu entwickelnden Systemen erstellt und verwendet werden, um unterschiedliche Zielsetzungen zu erfüllen. Die Modelle bestehen nicht nur unabhängig voneinander, sondern werden auch miteinander kombiniert. Zum Beispiel zeigen Beiser et. al. den Einsatz eines Mehrkörpermodells zur Beschreibung der Bewegungskinetik eines Portalschreitwerks in Co-Simulation mit einem Modell

des hydraulischen Antriebssystems. (Beiser, Wydra & Geimer, 2023, S. 16) Auch im Rahmen von Validierungsaktivitäten - insbesondere bei kombinierter physisch-virtueller Validierung (vgl. Kapitel 2.3.2) - kommen häufig mehrere Modelle zur Abbildung des Gesamtsystemverhaltens zum Einsatz, die konsistent zusammenspielen müssen. (Brinkschulte et al., 2017, S. 185) Eine Systemarchitektur ist ein qualitatives Modell, mit dem Zweck, ein gemeinsames Verständnis innerhalb des Entwicklungsteams und in der Kommunikation mit den Stakeholdern zu erleichtern, indem die Systemzusammenhänge graphisch modelliert und dargestellt werden. Nach VDI/VDE bildet eine Systemarchitektur die Korrelation zwischen Anforderungen, Funktionen, logischer und physischer Struktur eines technischen Systems über unterschiedliche Konkretisierungsstufen hinweg ab. (VDI/VDE-Richtlinie 2206)

Die Darstellung und Analyse von Systemaspekten in verschiedensten Modellen (CAD-Modelle, Simulationsmodelle, formale Modelle, ...) kann unter dem Begriff Model-Based Engineering (MBE) zusammengefasst werden. MBSE ist in diesem Verständnis ein Teil des MBE. (Dumitrescu et al., 2021; Phoenix Integration, 2020; PivotPoint Technology Corp., 2018)

Nach Albers unterstützt MBSE die Entwicklung komplexer Produkte durch Erhöhung der Transparenz zwischen unterschiedlichen Disziplinen, diese ist Basis für eine bessere Beherrschbarkeit von Komplexität. Grundlegend für die Entwicklung komplexer, interdisziplinärer Systeme ist dabei eine gemeinsame Modellsprache über alle Disziplinen hinweg. Als Vorteile des MBSE identifiziert Albers nicht nur die Visualisierung und Modellierung von Wechselwirkungen im System, sondern auch die Ableitung unterschiedlicher, jedoch konsistenter, stakeholdergerechter Sichten aus dem Modell heraus. Des Weiteren stellt die Verknüpfung des MBSE-Modells mit anderen vorhandenen Systemdaten und -modellen stellt ein großes Potenzial zur Effizienzsteigerung in der Entwicklung Systemen über mehrere Generationen hinweg dar. (Albert Albers, 2023)

Holt und Perry konkretisieren die Idee von MBSE dahingehend, ein Architekturmodell des zu entwickelnden Systems als zentrale Quelle für das gesammelte Wissen, das mit dem Projekt oder System verbunden ist, zu nutzen. Ein solches Architekturmodell umfasst dabei eine kohärente und konsistente Menge von Sichten, die mehrere Blickwinkel auf das System widerspiegeln. (Holt & Perry, 2018) Die Systemarchitektur kann sozusagen als Rückgrat des Entwicklungsprozesses gesehen werden.

Long betont im Vorwort von Weilkiens et al., dass in einer modellbasierten Entwicklung Systemarchitektur und detaillierte Berechnungsmodelle weder voneinander separiert existieren sollten noch ein einziges, großes Modell existieren kann, das alle

Dimensionen erfasst. Vielmehr muss die ganzheitliche Systemperspektive genutzt werden, um Zusammenhänge zu identifizieren, darzustellen und darauf aufbauend eine robuste Toolbox mit unterstützenden Verfahren entwickelt werden. (Weilkiens, Lamm, Roth & Walker, 2015) Dumitrescu et al. stellen fest, dass aktuell noch Ansätze fehlen, um die Informationen aus einem Architekturmodell mit anderen, etablierten Modellen des MBE zu vernetzen. (Dumitrescu et al., 2021) Als Basis für einen "nahtlosen" Übergang zwischen Modellen sehen Pohl et al. eine klare Beziehung der verschiedenen Modellartefakte. Ist das gegeben, besteht auch das Potenzial für automatisierte Analysen und Modelltransformationen bzw. Nutzung einer gemeinsamen Datenbasis für die unterschiedlichen Modelle. (Pohl, Hönniger, Achatz & Broy, 2012, S. 34)

Zwischenfazit:

MBSE im Sinne der Modellierung eines formalen Architekturmodells ermöglicht die qualitative Abbildung der Zusammenhänge von Anforderungen, Funktion und Struktur eines Systems. Damit bietet es nicht nur das Potenzial, ein einheitliches Verständnis im Entwicklungsteam und die interne und externe Kommunikation zu unterstützen, sondern auch als zentrale Quelle für das gesammelte Systemwissen in einem Projekt zu dienen. Um dieses Potenzial eines Rückgrats zu heben, fehlt es aktuell jedoch an Ansätzen, das Architekturmodell mit etablierten Methoden und Modellen des Systems Engineering zu vernetzen.

2.4.2 Modellierung von Systemarchitekturen

Für die Erstellung (d.h. Modellbildung und Modellierung) einer Systemarchitektur sind drei Aspekte notwendig: Eine Modellierungsmethode, eine Modellierungssprache und ein Modellierungswerkzeug. (Rambo et al., 2017, S. 173) Die Methode beschreibt dabei, wie einzelne Artefakte des Modells erarbeitet und modelliert werden können. Beispiele für Methoden sind "Object-Oriented Systems Engineering Method" OOSEM von INCOSE, die Systemmodellierung „SYSMOD“ nach Tim Weilkiens oder die im Rahmen des gleichnamigen BMBF Förderprojekts entwickelte Methode „SPES 2020“. (Rambo et al., 2017, S. 173–174) Die Modellierungssprache definiert die Elemente (Syntax) und Regeln (Grammatik) zur Darstellung des Modells. In der Luft- und Raumfahrt sowie im Fahrzeugbau hat sich die Verwendung der System Modeling Language (SysML²) etabliert. (Dumitrescu et al., 2021) Das

² OMG (2019).

Modellierungswerkzeug ist das Softwaretool in mit dem das Modell erstellt und gepflegt wird. Für die Verwendung von SysML gibt es mehrere etablierte Tools, u. a. Enterprise Architect von Sparx Systems, Cameo Systems Modeler von Dassault Systèmes oder Rational Rhapsody von IBM.

Modellierungsmethoden beschreiben, durch welche Schritte die zu modellierenden Inhalte erarbeitet werden können. Darüber hinaus beinhalten einige Methoden auch Empfehlungen zur übergeordneten Struktur des Architekturmodells in Form von Modellierungs-Rahmenwerken (engl. Frameworks). Unter Beteiligung von Unternehmen aus Luft und Raumfahrt, Softwareentwicklung und Fahrzeugbau ist im Rahmen des Forschungsprojektes „Software Platform Embedded Systems (SPES) 2020“ (Rambo et al., 2017, S. 173–174) ein solches Rahmenwerk entstanden, das heute Einsatz in der Industrie findet. (Martin, 2019; Timucin, 2020; Wäschle, Martin, Radimersky, Behrendt & Albers, 2020; Wäschle et al., 2021)

Motivation der SPES Methodik ist, ein Rahmenwerk für die Modellierung von Systemarchitekturen zu bieten, das die Anknüpfung zwischen verschiedenen Modellen erleichtert und die strukturierte Denkweise bei der Modellbildung unterstützt. Das Framework basiert dabei auf fünf Grundprinzipien (Pohl et al., 2012, S. 34–35): Unterscheidung von Problem und Lösung, explizite Berücksichtigung von Systemdekomposition, Unterstützung eines nahtlosen Übergangs zwischen Modellen, Unterscheidung zwischen logischer und technischer Lösung und kontinuierliche Entwicklung von Querschnittseigenschaften des Systems (z. B. Sicherheitseigenschaften).

Zur Umsetzung dieser Prinzipien ist das Modellierungs-Framework aus zwei Kernkonzepten aufgebaut. Klar getrennte Abstraktionsebenen ermöglichen die Dekomposition des Systems über mehrere Detaillierungsebenen. Neben der Detaillierung ist ein weiterer Vorteil von dezidierten Abstraktionsebenen die Möglichkeit, auf einer definierten Ebene Systemspezifikationen - z. B. zur Weitergabe an andere Organisationseinheiten oder Zulieferer - abzuleiten. Auf diese Weise bleibt die Spezifikation als Schnittstelle unverändert, das Designelement kann dennoch auf neuen Abstraktionsebenen verfeinert werden. Verschiedene Blickwinkel (Sichten, engl. Viewpoints) auf das System dienen dazu, unterschiedliche Teilaspekte der Systemeigenschaften, z. B. Problemraum und Lösungsraum, in den Fokus der Modellierung und Analyse zu stellen. (Pohl et al., 2012, S. 35–36) In Abbildung 2.14 ist das SPES-Framework dargestellt.

Die **Anforderungs-Sicht** (Requirements Viewpoint) hat zum Ziel, das Requirements Engineering zu unterstützen. In dieser Sicht ist der Kontext des Systems dargestellt, die übergeordneten Ziele und die Szenarien und Anwendungsfälle, die mit

diesen Zielen verknüpft sind. Auch die lösungsorientierten technischen Anforderungen, die vom betrachteten System auf der jeweiligen Ebene erfüllt werden müssen sind in dieser Sicht enthalten. (Pohl et al., 2012, S. 39)

Die **funktionale Sicht** (Functional Viewpoint) dient dazu, die Funktionen auf der betrachteten Systemebene zu beschreiben. Über die Systemebenen hinweg ergibt sich so eine Funktionshierarchie, die das Rückverfolgen der funktionalen Zusammenhänge über die Ebenen ermöglicht. (Pohl et al., 2012, S. 41)

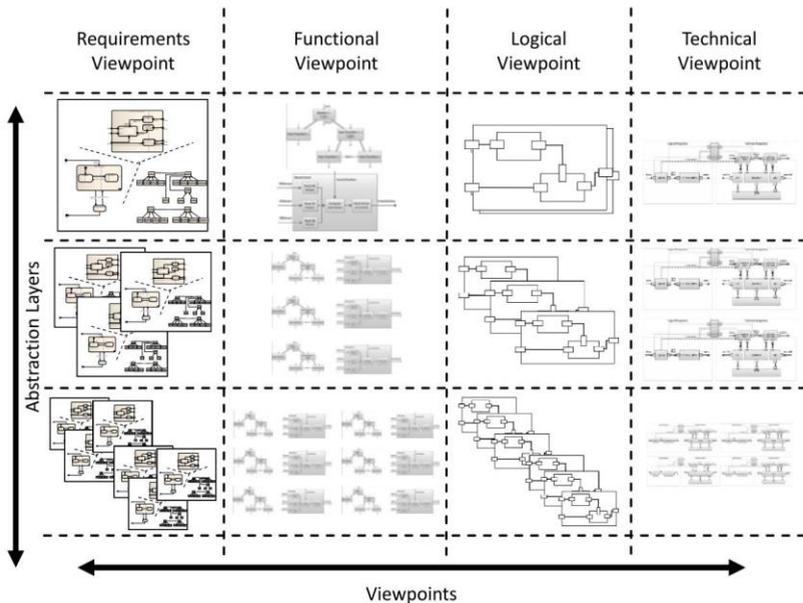


Abbildung 2.14: SPES Modellierungsframework für Systemarchitekturen (Pohl et al., 2012, S. 38).

In der **logischen Sicht** (Logical Viewpoint) wird die "Black-Box" des betrachteten Systems aufgelöst in eine "White-Box", das bedeutet es werden logische Systemkomponenten definiert, die benötigt werden, um die Funktion der "Black-Box" zu erfüllen. Diese logischen Komponenten sind dabei unabhängig von Randbedingungen der technischen Umsetzung. Die Schnittstellen der logischen Elemente und die darüber ausgetauschten Arten von Systemgrößen werden identifiziert und darge-

stellt. Diese Dekomposition der Black-Box ist der Startpunkt für die Anforderungsermittlung und Funktionsdefinition auf der nächsttieferen Detaillierungsebene. Die "White-Box" Elemente einer Ebene sind also gleichzeitig die "Black-Box"-Elemente der nächsten Ebene. (Pohl et al., 2012, S. 41–42)

Die **technische Sicht** (Technical Viewpoint) beschreibt die tatsächliche Umsetzung der Funktionserfüllung durch die logischen Elemente. Das beinhaltet die Beschreibung der physikalisch-technischen Eigenschaften und Parameter der eingesetzten Komponenten. (Pohl et al., 2012, S. 43)

Die konkrete Ausgestaltung der Ebenen ist abhängig vom zu entwickelnden System. Die Sichten des SPES-Frameworks sind dagegen domänenunabhängig gültig und anwendbar. (Pohl et al., 2012, S. 38) Sie sind nicht dogmatisch auf die vier vorgestellten Sichten begrenzt. Entsprechend des abgebildeten Systems und Projektumfelds können situations- und projektabhängig weitere Sichten, wie beispielsweise eine geometrische Sicht eingeführt werden. (Pohl et al., 2012, S. 37)

Zwischenfazit:

Die Modellierung von Systemarchitekturen basiert auf der Anwendung von Methode, Sprache und Tool. Als Sprache ist der OMG Standard SysML in der Industrie etabliert und wird durch verschiedene Tools unterstützt. Methoden unterstützen die Schritte der inhaltlichen Modellbildung, dabei beinhalten einige Methoden auch Frameworks zur Strukturierung der Modellinhalte in Ebenen und Sichten. Ein etabliertes Framework ist das der SPES-Methode, das neben dezidierten Abstraktionsebenen die Sichten Anforderungen, Funktion, logische Struktur und technische Struktur unterscheidet.

2.4.3 Einsatz von modellbasiertem Systems Engineering in der Praxis

Eine Interviewstudie mit 33 Experten der Industrie aus Deutschland, Schweiz und Österreich aus dem Jahr 2015 zeigt, dass in weiten Teilen der Unternehmen ein grundlegendes Verständnis von Systems Engineering besteht. Die Erwartungen an die Unterstützung in der Entwicklung interdisziplinärer Produkte sind hoch, auch wenn die Aktivitäten, Methoden und Tools für die Interviewten schwer überschaubar sind. Teilweise werden Systemarchitekturen in SysML erzeugt, jedoch ist die Pflege und Nutzung der Modelle gering. Oft werden die Modelle als "tote Bilder" bezeichnet. Nur wenige Unternehmen der Automobilindustrie und Luft- und Raumfahrtindustrie erproben interdisziplinäre Systemmodelle als Basis für die modellbasierte Entwicklung. Häufig im Rahmen von Initiativen, z. B. zur funktionalen Sicherheit

oder Qualitätssicherung. Die Autoren stellen fest, dass die Potenziale domänenübergreifender Systemarchitekturen in den Unternehmen nicht ausgeschöpft werden. Die Entwicklungsprozesse sind noch stark dokumentenbasiert und Modelle werden überwiegend unabhängig von den jeweiligen Fachdisziplinen erstellt und weiterverwendet. Ein Abgleich der Modelle über Disziplinen hinweg oder deren Verknüpfung findet kaum statt. (Gausemeier et al., 2015)

Eine internationale Online-Umfrage mit 661 Teilnehmern im Dezember 2018 lässt ebenfalls den Rückschluss zu, dass die Potenziale von Systemarchitekturen nicht ausgenutzt werden. Die Architekturentwicklung von Systemen und Subsystemen wird als größtes Potenzial von MBSE gesehen, gefolgt von Anforderungsanalyse und früher Konzepterstellung. Der Fokus bei der Nutzung von MBSE liegt in den befragten Unternehmen maßgeblich auf der Architekturmodellierung und der Rückverfolgbarkeit von Anforderungen. (Cloutier, 2019)

Ein ähnliches Bild zeichnet sich in einer Umfrage aus dem Jahr 2021 mit 127 Teilnehmern aus unterschiedlichen deutschen Unternehmen ab. Sie zeigt, dass etwa drei Viertel der befragten Unternehmen im Begriff sind, Systems Engineering Ansätze einzuführen. Motivation für die Einführung ist für die befragten Unternehmen die Komplexität und Interdisziplinarität der Systeme. Die Modellierung mit SysML wird dabei als wertvollste Kompetenz bei der Einführung von Systems Engineering angesehen. Die tatsächliche Erstellung und Nutzung von Modellen in SysML ist in den befragten Unternehmen jedoch noch wenig verbreitet. (Stützel & Paul, 2021) Auch Dumitrescu et. al. stellen fest, dass Unternehmen eine hohe Erwartungshaltung an die Unterstützung eines durchgängigen Entwicklungsgeschehens durch MBSE haben, der Leistungsstand jedoch sehr heterogen und abhängig von Unternehmensgröße und Branche ist. (Dumitrescu et al., 2021) Daraus lässt sich ableiten, dass Methoden und Ansätzen des Systems Engineering als relevant eingeschätzt werden, ihre Anwendung in der Praxis – insbesondere die Nutzung von Systemarchitekturen als zentrales Modell – jedoch noch nicht etabliert ist.

Unter Einbeziehung weiterer Literatur lassen sich Rückschlüsse auf die Gründe für die geringe Nutzung von Architekturmodellen ziehen:

- **Hindernisse in der Kommunikation durch geringe Erfahrung im Lesen von SysML-Modellen:** Im Rahmen von Live-Labs mit Studierenden stellen Mandel et. al fest, dass das erstellte Architekturmodell kaum zur Kommunikation mit Stakeholdern und Teammitgliedern, die nicht direkt in die Architekturerstellung eingebunden sind, genutzt wird. Meist werden hierfür Repräsentationsformen, z. B. Powerpoint, verwendet. (Mandel, Martin & Albers, 2022, S. 6) Das deutet darauf hin, dass das Vermitteln und Lesen der im Modell abgebildeten Informationen einen gewissen Erfahrungs-

oder Kenntnisstand braucht. Passend dazu ist ein Resultat der Studie von Stützel und Paul, dass die Erweiterung der Kompetenzen, insbesondere in der Entwicklung und im Management, ein wichtiger Baustein bei der Einführung von Systems Engineering ist. (Stützel & Paul, 2021) Die Studie von Gausemeier et al. identifiziert, dass Expertise und Bewusstsein über Ansätze und Methoden des Systems Engineering stark auf einzelne Personen konzentriert sind. (Gausemeier et al., 2015, S. 32)

- **Mangelnde inhaltliche Nutzung des Architekturmodells für andere Methoden und Modelle in der Produktentstehung:** Holt und Perry beschreiben, dass die Vermittlung der Vorteile von MBSE eine große Hürde ist. (Holt & Perry, 2018, S. 579) Aus der Studie von Cloutier resultiert u. a. geringer erwarteter Nutzen als Schwierigkeit bei der Einführung von MBSE. (Cloutier, 2019) Ebenso stellen Gausemeier et al. fest, dass die Modellerstellung häufig als zu aufwändig und nicht handhabbar im Vergleich zu den gegebenen Vorteilen gesehen wird. (Gausemeier et al., 2015, S. 36) Die Potenziale der Etablierung eines Architekturmodells scheinen in der Praxis somit nicht klar erkennbar zu sein, weil die Verbindung zu anderen Aktivitäten und Modellen der Produktentstehung fehlt. Im Rahmen der Live-Labs beobachten Mandel et al. ein ähnliches Phänomen: Der grundsätzliche Vorteil des Modellierens einer Systemarchitektur ist für die Mitglieder des Entwicklungsteams oft nicht klar erkennbar. (Mandel et al., 2022, S. 10) Jedoch wird die Architektur für die Identifikation und Dokumentation von Variationsanteilen entsprechend dem Modell der SGE als hilfreich empfunden. (Mandel et al., 2022, S. 14) Das gibt einen Hinweis darauf, dass der individuell empfundene Nutzen der Systemarchitektur steigt, wenn sie aktiv zur Umsetzung weiterer Methodischer Ansätze oder Modellierungen verwendet wird.
- **Unzureichende Werkzeugunterstützung in der Nutzung der Informationen aus dem Architekturmodell:** Ein weiteres Hindernis in der Einführung von Systems Engineering erkennen Gausemeier et al. auch in einer unzureichenden Anpassbarkeit. Die Methoden des Systems Engineering werden als schlecht in bestehende Prozesse integrierbar wahrgenommen. (Gausemeier et al., 2015, S. 31) Unter anderem fehlen Werkzeugketten und etablierte Standards zur Modellverknüpfung, konsistente Modelle sind daher aktuell kaum möglich. (Gausemeier et al., 2015, S. 38) Das deutet darauf hin, dass die zuvor identifizierte Schwierigkeit der mangelnden inhaltlichen Verknüpfung verstärkt wird, weil die Nutzung der Informationen aus dem Architekturmodell in anderen Modellen kaum durch Methoden, Prozesse und Werkzeuge unterstützt ist.

Zwischenfazit:

Verschiedene Studien zeigen, dass die Relevanz von Systems Engineering und das theoretische Potenzial von Systemarchitekturmodellen in der industriellen Praxis er-

kannt wurde. Ansätze des Systems Engineering werden in vielen Unternehmen bereits genutzt oder zumindest eingeführt. Insbesondere in der Automobilindustrie und der Luft- und Raumfahrtindustrie werden dabei auch Systemarchitekturen erstellt und genutzt. Die Kompetenz und Akzeptanz der Methoden und Modelle ist jedoch nicht flächendeckend, sondern stark personen- und projektbezogen. Ein Aspekt dabei ist, dass die in der Systemarchitektur dokumentierten Zusammenhänge und Informationen nur unzureichend in den Aktivitäten der Produktentstehung genutzt werden. Einerseits weil das Bewusstsein darüber gering ist, wo und wie genau diese Informationen genutzt werden können, andererseits, weil die toolseitige Anbindung der Systemarchitektur z. B. an andere Modelle nicht ausreichend unterstützt ist. Ansätze zur Nutzung der in der Systemarchitektur dokumentierten Informationen sind jedoch gleichzeitig die Basis für eine erfolgreiche werkzeugseitige Anbindung. Im Folgenden wird daher beleuchtet, welche Ansätze im zur inhaltlichen Nutzung im Stand der Forschung zu finden sind.

2.4.4 Ansätze zur Verwendung der Architekturinhalte in Aktivitäten und Modellen während der Produktentwicklung

Wie in Kapitel 2.4.3 beleuchtet, hängt die Verbreitung und Akzeptanz von MBSE - im Sinne einer Systemarchitektur als zentrale Informationsquelle - davon ab, welchen Nutzen der Aufbau einer Systemarchitektur für andere Aktivitäten und Modelle während der Produktentwicklung hat. Für die Nutzung der Informationen gibt es verschiedene Ansätze, die im Weiteren dargelegt werden.

Unterstützung von Planungsaktivitäten durch Abschätzen von Entwicklungsaufwänden

In der Planung und Steuerung eines laufenden Entwicklungsprojektes, aber auch vorgelagert in der strategischen Produktplanung, besteht die Notwendigkeit voraussichtliche Entwicklungsaufwände und -risiken abzuschätzen. Auf dieser Basis können dann Entscheidungen für oder gegen bestimmte Ausprägungen oder Lösungsprinzipien getroffen werden. Entsprechend des Modells der SGE (siehe Kapitel 2.1.3) kann die Analyse der Elemente des Referenzsystems und die darauf aufbauende Bestimmung von Variationsanteilen solche Aufwands- und Risikoabschätzungen unterstützen. Wäschle et al. zeigen am Beispiel eines elektromechanischen Aktuators auf, wie die Variationsarten auf verschiedenen Ebenen der logischen Architektur des Systemmodells identifiziert und dokumentiert werden können (Abbildung 2.15). Eine Gewichtung erlaubt die projekt- und situationsspezifische Adaption der Variationsanteile. Zur Berechnung und eingängigeren Darstellung der Variationsanteile werden die Daten aus dem SysML-Modell in Excel exportiert. (Wäschle et al., 2020, S. 2430)

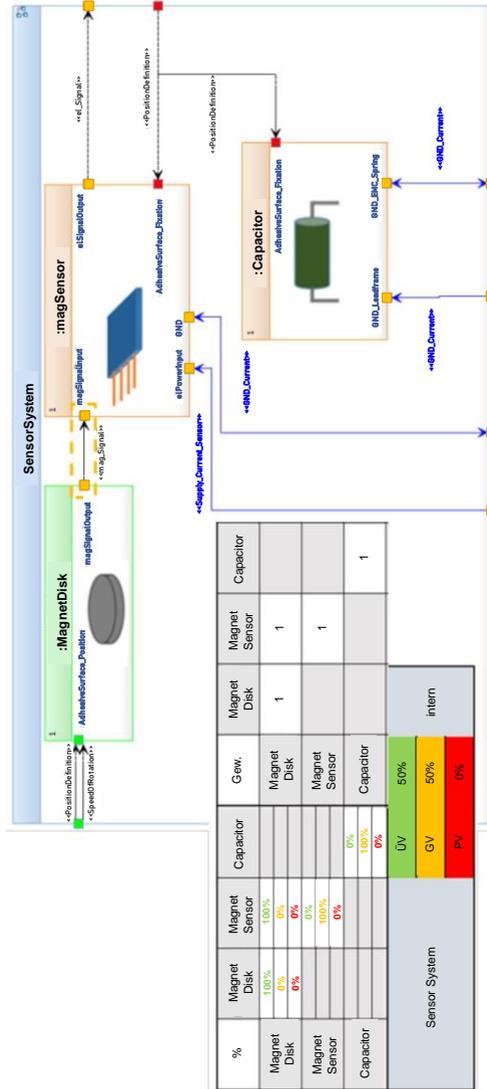


Abbildung 2.15: Identifizierung der Variationsarten im SysML-Modell und Berechnung der Variationsanteile(Wäschle et al., 2020, S. 2431).

Einen ähnlichen Ansatz verfolgen Albers et al. zur strategischen Produktgenerationsplanung. Ziel ist dabei die Auswahl von Produktmerkmalen für eine neue Produktgeneration, die hohes Differenzierungspotenzial bei angemessenem Entwicklungsaufwand versprechen. Die im Modell abgebildeten Informationen und Zusammenhänge der Eigenschaften des Vorgängerprodukts werden mit Anforderungen an die neue Produktgeneration verknüpft (Abbildung 2.16). Dabei wird zum einen geprüft, welche Anforderungen sich in welcher Art verändern, und zum anderen inwiefern das Vorgängerprodukt die neuen Anforderungen erfüllt. Auf dieser Basis können sowohl potenzielle Differenzierungsmerkmale identifiziert werden als auch die Abschätzung des Entwicklungsaufwands unterstützt werden. (Alexander Albers, Kühn & Dumitrescu, 2017)

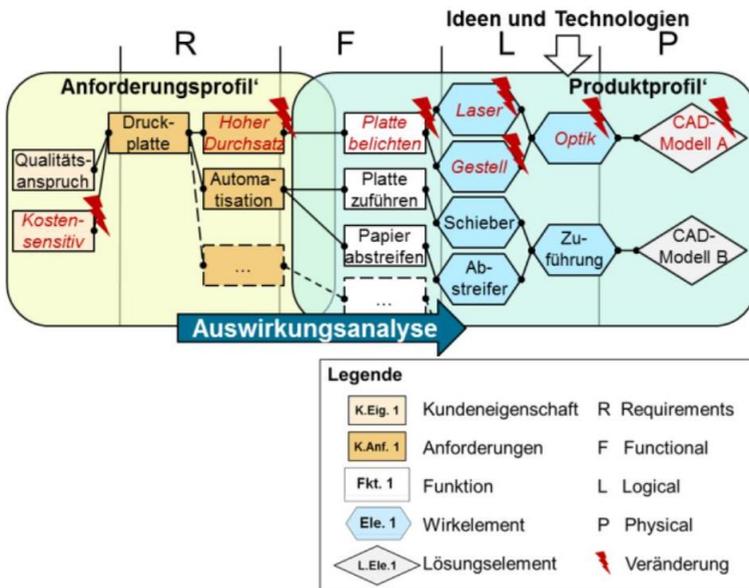


Abbildung 2.16: Verknüpfung eines neuen Anforderungsprofils mit dem Modell der vorigen Produktgeneration nach (Alexander Albers et al., 2017).

In beiden geschilderten Fällen werden die im Systemmodell enthaltenen Informationen aktiv genutzt, ggf. verändert und die Auswirkung der Veränderungen innerhalb des Modells analysiert, um Rückschlüsse auf Innovationspotenziale und Entwicklungsaufwände zu ziehen. Die Inhalte werden somit nicht nur zur Dokumentation

und Kommunikation des Systemverständnisses genutzt, sondern für das Projektmanagement bzw. die strategische Planung zukünftiger Produktgenerationen eingesetzt. Eine Verknüpfung mit weiteren Modellierungs- oder Simulationstools ist dafür nicht notwendig.

Unterstützung von funktions- und strukturbasierten Entwicklungsmethoden

Viele Entwicklungsmethoden beinhalten auch eine Beschreibung der Funktionen und oder Strukturen des zu entwickelnden Systems. Beispielsweise morphologische Kästen, QFD, etc. Im Bereich der Risiko- und Sicherheitsanalysen ist hier unter anderem die Fehlermöglichkeits- und Einflussanalyse (FMEA) zu nennen. Grundlage für die Analyse von Fehlerursachen, Fehlern und Fehlerauswirkungen ist ein Funktionsnetz und ein Strukturnetz des Systems. Die Informationen zu beiden Aspekten sind in der Systemarchitektur dokumentiert. Durch direkte Verwendung in der FMEA können Inkonsistenzen und Missverständnisse innerhalb des Teams reduziert werden. Wäschle et al. zeigen die Verwendung des SysML-Modells im Zusammenhang mit Sicherheitsanalysen beispielhaft auf. (Wäschle, Wolter, Bause, Behrendt & Albers, 2022, S. 280) In der industriellen Praxis kommen für Sicherheitsanalysen häufig spezielle Softwaretools zum Einsatz, beispielsweise APIS IQ (APIS Informationstechnologien GmbH, 2022). Eine Automatisierte Synchronisation von Funktions- und Strukturelementen zwischen dem SysML- Modell und einer solchen Software bietet hier zusätzlich die Möglichkeit zur Konsistenzhöhung und Aufwandsreduktion in der Erstellung der FMEA.

Unterstützung der Funktions-Gestalt-Synthese durch Verknüpfung mit CAD

Im Rahmen des geförderten Projekts FAS4M war das Ziel ein Brückenschlag zwischen abstrakten Funktionsmodellen und detaillierten geometrischen CAD-Modellen. (Moeser et al., 2016) SysML-Modelle sind formale, deskriptive Modelle, die Gestaltaspekte nicht abbilden können. CAD-Modelle stellen Gestaltaspekte detailliert in Form konkreter Bauteile dar. Beide Modellarten unterstützen die Synthese von Funktions-Gestalt-Zusammenhängen nur unzureichend, hierbei kommen in der Entwicklungspraxis meist Skizzen zum Einsatz. (Moeser, Albers & Klingler, 2015) Die im Projekt entstandene SysML-Erweiterung MechML ermöglicht die Einbindung und Darstellung von Skizzen über ein Softwareplugin im SysML-Modell. Die Informationen aus dem SysML-Modell inklusive der Skizzen können über ein weiteres Softwareplugin auch im CAD-Modell angezeigt werden. (Moeser et al., 2016) Abbildung 2.17 stellt die Verbindungen zwischen den Softwaretools dar.

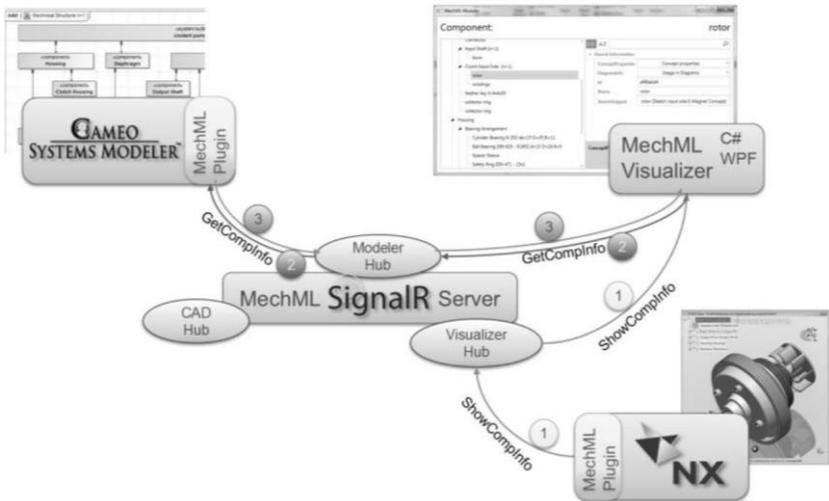


Abbildung 2.17: Aufruf und Kommunikation zwischen dem SysML-Tool, MechML Visualizer und CAD (Mooser et al., 2016).

Des Weiteren wurde eine Synchronisationsmöglichkeit der Baustruktur zwischen SysML (Blockdefinitionsdiagramm) und CAD (Modellbaum) geschaffen. So können aus dem CAD-Modell heraus die im SysML-Modell hinterlegten Informationen zu jedem Modellelement (Block) angezeigt werden, beispielsweise Anforderungen, Funktionen oder Skizzen. Wird das CAD-Modell um Komponenten erweitert, können diese wiederum als Blöcke in die Baustruktur des SysML-Modells importiert werden. (:em engineering methods AG, 2022; Mooser et al., 2016)

Unterstützung von Verifikations- und Validierungsaktivitäten durch Modellierung des Validierungssystems in SysML

Mandel et al. stellen fest, dass verständliche Ansätze zur Unterstützung von Validierungsansätzen durch MBSE fehlen. Daher wird ein Ansatz vorgeschlagen, der SysML um validierungsspezifische Stereotypen erweitert, wie in Abbildung 2.18 dargestellt. Ziel ist, neben dem Produkt- auch das Validierungssystem in SysML abzubilden, sodass Aktivitäten wie die Planung von Validierungskonfigurationen oder die Gestaltung von Prüfungsumgebungen unterstützt werden. (Mandel, Böning, Behrendt & Albers, 2021, S. 4)

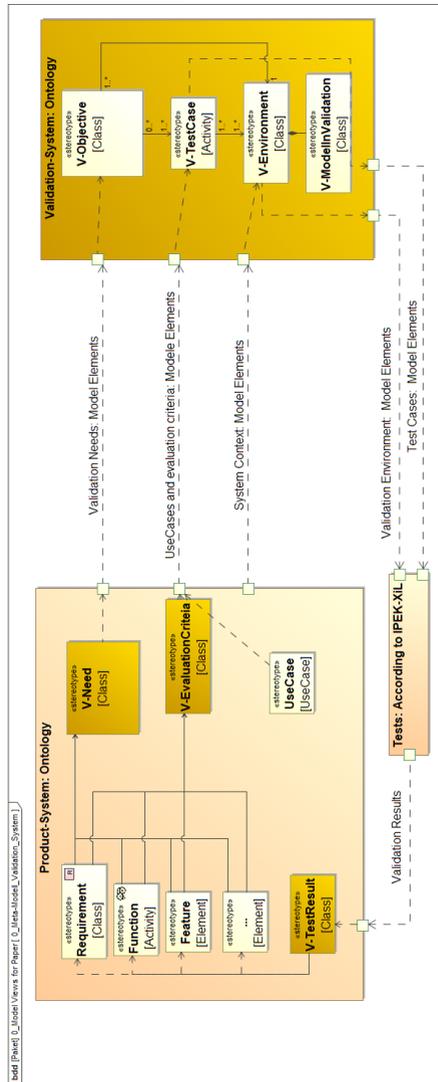


Abbildung 2.18: SysML-Stereotypen zur Abbildung von Produktsystem und Validierungssystem (Mandel, Böning et al., 2021, S. 4).

Vorteile des Ansatzes von Mandel et al. sind die integrierte Modellierung von Produkt und zugehörigem Validierungssystem sowie die Rückverfolgbarkeit zwischen modellierten Elementen. Dadurch können Informationen strukturiert gesammelt werden und die Entscheidungen zu geeigneten Validierungsumgebungen und Testfällen und die Interpretation der Ergebnisse unterstützen. (Mandel, Böning et al., 2021, S. 7)

Wissensmanagement durch Erstellung von Bibliotheken in SysML

Das SysML-Modell des Validierungssystems kann auch genutzt werden um Wissen zu den einzelnen Subsystemen und Komponenten des Systems zugänglich zu machen. Im SysML-Modell sind die möglichen Bestandteile einer Validierungsumgebung in einer logischen Hierarchie als Blöcke modelliert. Konkret verfügbare Komponenten (z. B. vorhandene Sensoren, Aktuatoren etc.) sind die unterste Ebene der Hierarchie (Abbildung 2.19). Sie beinhalten das notwendige Wissen zum Einsatz der Komponente, wie z. B.: charakteristische Eigenschaften, Schnittstellen, verfügbare zugehörige Modelle und Auswertungsalgorithmen, Erfahrungswissen. (Mandel et al., 2020, S. 3)

Mandel, Wolter et al. stellen fest, dass MBSE-Ansätze – hier eine Bibliothek für Komponenten von Validierungssystemen in SysML - die konsistente Wissensdokumentation, die Zugänglichkeit dieses Wissens und dadurch die geeignete Anwendung des Wissens unterstützt werden kann. Als Schwäche identifizieren sie, dass das im SysML-Modell gespeicherte Wissen und dessen Qualität und Zusammenhänge von der manuellen Ergänzung der Wissensbasis abhängt und diese damit einer kontinuierlichen Pflege unterliegen muss. (Mandel et al., 2020, S. 6)

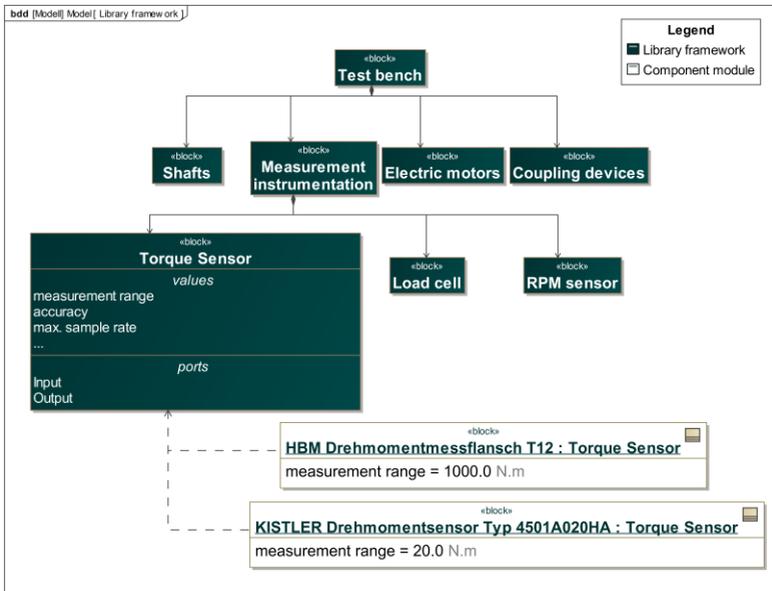


Abbildung 2.19: Rahmenwerk der Bibliothek und Verknüpfte Komponenten (Mandel et al., 2020, S. 4).

Vernetzung mit Berechnungs- und Simulationsmodellen

Auf Basis der o. g. Arbeiten untersuchen Mandel et al., wie gemischt physisch-virtuelle Modelle mit einem Systemmodell verknüpft werden können. Untersucht wird das Verhalten eines Fahrzeugs in unterschiedlichen Testfällen, das Verhalten wird durch einen Rollenprüfstand in Kombination mit einer Fahrzeugsimulation in der Software CarMaker modelliert, die Verknüpfung der CarMaker-Daten und der Daten aus SysML ist mit der Software ModelCenter realisiert. Diese stellt die Testplanungs- und Analyseumgebung dar. (Mandel, Wäsche, Lutz & Behrendt, 2021) Abbildung 2.20 zeigt schematisch das Rahmenwerk bestehend aus den unterschiedlichen Modellen und Tools.

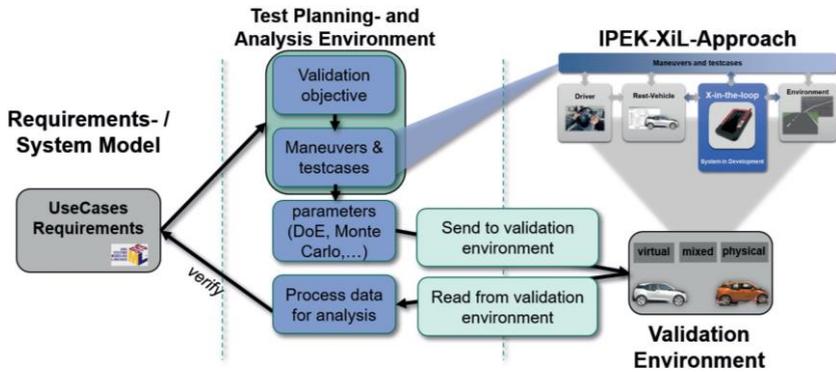


Abbildung 2.20: Schematische Darstellung des Rahmens der genutzten Modelle (Mandel, Wäschle et al., 2021).

Mandel, Wäschle et al. nutzen die im Architekturmodell rückverfolgbar modellierten Anforderungen, Anwendungsfälle und Systemzusammenhänge als Planungsbasis für Testfälle und Testumgebungen. Die Erkenntnisse aus der Validierung fließen in das Modell zurück und begründen damit nächste Entwicklungs- und Validierungsschritte. (Mandel, Wäschle et al., 2021)

Vidal und Villota verknüpfen ein SysML-Modell eines elektromechanischen Aktuators mit einer Matlab/Simulink Simulation. Das SysML-Modell strukturieren sie dabei in die Anteile Anforderungen, Struktur, Anwendungsfälle, Verhalten und Parameter. Die Verknüpfung der Modelle erfolgt über das parametrische Modell mittels eines Zusicherungsdiagramms (engl. parametric diagram) in SysML, wie Abbildung 2.21 zeigt. (Vidal & Villota, 2018, S. 150)

Ein Zusicherungsdiagramm ermöglicht die Definition komplexer Abhängigkeitsbeziehungen zwischen Parametern des SysML Modells. Die Formeln für die Abhängigkeiten (z. B. $F=m \cdot a$) werden dafür in Zusicherungsbausteinen (engl. constraint blocks) hinterlegt, die als Schnittstellen die benötigten Parameter (hier: F , m , a) besitzen. In einem Zusicherungsdiagramm können diese Bausteine dann über ihre Schnittstellen miteinander verbunden werden. (Holt & Perry, 2018, S. 97) Vidal und Villota zeigen auf, dass durch die Verknüpfung von SysML und Matlab/Simulink über Zusicherungsdiagramme die Anforderungserfüllung quantitativ analysiert werden kann. Sie kommen zum Schluss, dass für eine vollständige Rückverfolgbarkeit der Anforderungen beide Modelle über ausreichende Daten verfügen müssen. (Vidal & Villota, 2018, S. 152) Der Aufbau des MATLAB/Simulink-Modells wird im Beitrag nicht näher dargestellt, die Verwendung des Zusicherungsdiagramms lässt jedoch darauf schließen, dass die Modellbildung des Systemverhaltens in beiden Modellen übereinstimmt, sodass die im Zusicherungsdiagramm hinterlegten Zusammenhänge eins zu eins in der Simulationsumgebung übernommen werden können.

Eine andere Möglichkeit der Verknüpfung von SysML-Modellen mit Berechnungsmodellen zeigen Herzig et al. auf. Am Beispiel des SysML-Modells eines Doppelpendels im Tool MagicDraw und einem Modelica-Modell im Tool MapleSim setzen sie eine automatisierte Modelltransformation um (Abbildung 2.22). Zur Transformation zwischen Modelica und SysML gibt es einen entsprechenden OMG Standard. (Herzig, Rouquette, Forrest & Jenkins, 2013, S. 118)

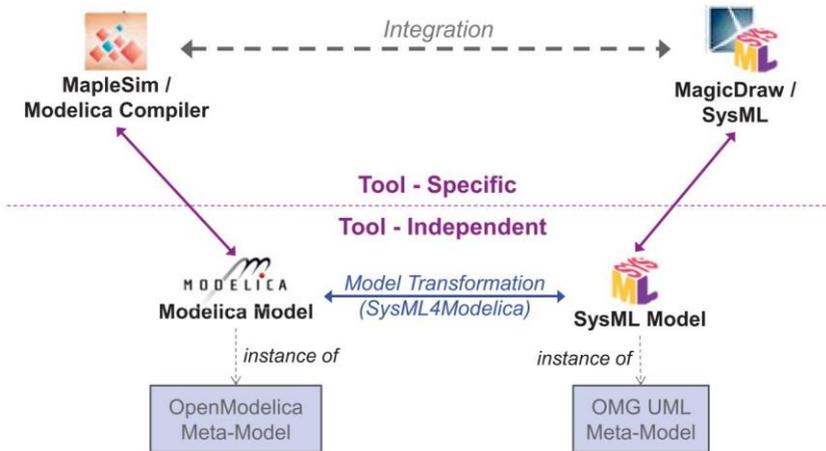


Abbildung 2.22: Ansatz zur Transformation eines SysML-Modells in ein Modelica-Modell unter Berücksichtigung Werkzeugabhängiger und –unabhängiger Randbedingungen. (Herzig et al., 2013, S. 121)

Voraussetzung dafür ist, dass die Modelle jeweils konform ihrer Sprachen aufgebaut sind. Dennoch stellen die Autoren fest, dass es neben den toolunabhängigen Modelleigenschaften weiterhin Modelleigenschaften gibt, die werkzeughängig sind. Für diese muss ein werkzeugspezifischer Integrationsprozess entwickelt werden, der eine programmierbare Schnittstelle zu beiden Werkzeugen erfordert. (Herzig et al., 2013, S. 126) Die Arbeit zeigt auf, dass aus dem SysML-Modell heraus automatisch Teile des Analysemodells generiert werden können. Vorteile sehen Herzig et al., in der Vermeidung von Inkonsistenzen zwischen den Modellen bei gleichzeitiger Reduzierung des Modellierungsaufwandes im Berechnungsmodell, die nachvollziehbare Dokumentation von Entscheidungen inklusive Begründung im SysML Modell und die mögliche Analyse und Verfeinerung von Anforderungen eines Systems über mehrere Ebenen hinweg. (Herzig et al., 2013, S. 119)

Zwischenfazit:

Die Beispiele zeigen auf, dass es bereits unterschiedliche Ansätze gibt, die in einer Systemarchitektur dokumentierten Informationen zielführend in anderen Bereichen der Produktentstehung zu nutzen. Dabei werden zwei Aspekte deutlich: Zum einen kann das SysML-Modell selbst über die Dokumentation und Kommunikation hinaus verwendet und gegebenenfalls erweitert werden um Schlussfolgerungen aus den

Informationen zu ziehen. Das zeigen die Ansätze zur Abschätzung von Entwicklungsaufwänden und zur Modellierung von Validierungssystemen sowie der Nutzung zum Wissensmanagement. Zum anderen ist die Verknüpfung und der Datenaustausch des SysML-Modells mit anderen Modellen und Werkzeugen vielversprechend. Das gilt insbesondere hinsichtlich der Zielsetzung einer durchgängigen modellbasierten Entwicklung. Die Ansätze zur Verknüpfung von SysML mit CAD, Berechnungsmodellen und Simulationstools zeigen die technische Möglichkeit hierfür. Offen bleibt in diesen Arbeiten allerdings, ob und in wie weit sich die Modellbildung des Architekturmodells und des Berechnungsmodells dabei unterscheiden dürfen.

2.5 Zusammenfassung des Stands der Forschung

Der Stand der Forschung zeigt, dass die zunehmende Produktkomplexität durch geeignete Methoden und Prozesse in der Entwicklung dieser Produkte unterstützt werden muss. Die Komplexität setzt sich dabei aus mehreren Aspekten zusammen. Die Anzahl und Vernetzung der Anforderungen an die Produkte steigen, dokumentenbasiertes Anforderungsmanagement wird dem nicht gerecht, die modellbasierte Abbildung von Zielsystemen ermöglicht die Darstellung von Zusammenhängen zwischen Zielen und Anforderungen. Ein weiterer Komplexitätstreiber ergibt sich daraus, dass neue Produkte und ihre Subsysteme auf vorangegangenen Produkt- und Systemgenerationen basieren, jedoch für veränderte Ziele, Anforderungen und / oder veränderten Randbedingungen eingesetzt werden. Übernahme- und Variationsanteile müssen also entsprechend dem Pull-Prinzip der Validierung möglichst frühzeitig vor dem Hintergrund der neuen Ziele und Anforderungen auf Gesamtsystemebene bewertet werden können. Der IPEK-XiL-Ansatz bietet ein Rahmenwerk um entwicklungsbegleitend und dem aktuellen Produktreifegrad angemessenen Validierungsaktivitäten systematisch zu planen und umzusetzen. Hierbei können für das Sul sowie für Repräsentation der umgebenden Systeme und Randbedingungen virtuelle, physische oder gemischt physisch-virtuelle Modelle zum Einsatz kommen. Nicht zuletzt erhöht sich die Produktkomplexität auch durch mehr Softwareintegration, flexiblere Funktionalität und dadurch auch mehr Subsysteme unterschiedlicher Domänen, die in ihrem Zusammenwirken die Funktionen des Gesamtsystems anforderungsgerecht erfüllen müssen.

Ein zentraler Ansatz, um mit dieser Produktkomplexität umzugehen ist modellbasiertes Systems Engineering. Die Grundidee ist dabei, ein Architekturmodell des Systems als zentralen Informationsspeicher in der Produktentwicklung zu nutzen, sodass alle weiteren Aktivitäten und verwendeten Modelle diese Informationen nutzen und idealerweise damit verknüpft sind. Das Systemarchitekturmodell ist damit

das Rückgrat der Entwicklung. In ihr ist das Systemverständnis des Entwicklungsteams in Form von Systemkontext, Anwendungsbeschreibung, Funktionsstruktur und deren Allokation auf die Produktstruktur dargestellt. Außerdem sind Ziele, Anforderungen und Randbedingungen im Architekturmodell beinhaltet und mit den Elementen verknüpft, auf die sie sich beziehen. Eine systematische Modellerstellung mit Bezug auf Referenzsysteme ermöglicht die generationsübergreifende Nutzung des Modells und bietet dadurch Potenzial, das Verhältnis von Modellerstellungsaufwand zu Modellnutzen zu verbessern.

Forschungsaktivitäten zeigen unterschiedliche Möglichkeiten zu Nutzung von Systemarchitekturen über Verständnisdokumentation und Kommunikation hinaus auf. Hier wird auch deutlich, dass es realisierbare Ansätze für die werkzeugseitige Anbindung des Architekturmodells an andere Modelle gibt. Die Synchronisation von Modellinhalten und Parametern ist technisch möglich. Dennoch werden diese Möglichkeiten in der Industrie kaum genutzt. Offen bleibt in den Forschungsaktivitäten die methodische Unterstützung der Modellverknüpfung. Es liegt daher der Schluss nahe, dass für Entwicklungsteams in der Praxis nicht klar ist, wie die inhaltliche Verknüpfung der Modelle geeignet umgesetzt werden kann.

Die vorgestellten Studien zeigen, dass Unternehmen die Relevanz und die Potenziale von modellbasiertem Systems Engineering erkannt haben und im Begriff sind, entsprechende Ansätze einzuführen. Insbesondere in der Automobilindustrie wird die Erstellung und Nutzung von Systemarchitekturen auch durch prozessuale Vorgaben wie ASPICE oder ISO 26262 gefordert. Dennoch wird in den Studien auch deutlich, dass die Potenziale des modellbasierten Systems Engineering in der Praxis aktuell nicht ausgeschöpft werden. Neben geringer Erfahrung im Erstellen und Lesen von Systemarchitekturmodellen in SysML ist insbesondere die mangelnde Nutzung des Systemarchitekturmodells in den weiteren Aktivitäten der Produktentstehung ein Grund für die geringe Potenzialausschöpfung. Der Mehrwert eines Architekturmodells ist für die Entwickelnden heute häufig nicht sichtbar, es ist unklar, wie das darin explizierte Wissen genutzt werden kann, beispielsweise für den Aufbau anderer Systemmodelle (z. B. Simulationsmodelle) oder in etablierten funktionsbasierten Methoden wie FMEA. Folge des unklaren Mehrwerts ist eine begrenzte Akzeptanz von MBSE in der Praxis.

3 Zielsetzung und Forschungsvorgehen

Auf Basis des aufgezeigten Stands der Forschung wird im Folgenden der Forschungsbedarf und die Forschungslücke umrissen. Darauf aufbauend wird die Zielsetzung dieser Arbeit beschrieben. Die Forschungshypothese wird durch Forschungsfragen operationalisiert. Abschließend gibt dieses Kapitel einen Überblick über das Forschungsvorgehen.

3.1 Forschungsbedarf

Der Stand der Forschung zeigt deutlich, dass die Umsetzung von modellbasiertem Systems Engineering in der Praxis hinter ihren forschungsseitig aufgezeigten Potenzialen zurückbleibt. Insbesondere fehlen Durchgängigkeit und Konsistenzsicherung unterschiedlicher Modelle des zu entwickelnden Systems. Um komplexe Systeme mit hoher Qualität effizient zu entwickeln, ist die konsistente Erweiterungen der Wissensbasis durch modellbasierte Analyse- und Syntheseschritte von großer Bedeutung. Modelle des zu entwickelnden Systems mit unterschiedlichen Verkürzungen und Zwecken werden bislang weitgehend unabhängig voneinander erstellt und weiterverwendet. Die Beziehung zwischen den Artefakten der unterschiedlichen Modelle wird nicht systematisch berücksichtigt. Eine klare Beziehung der Modellartefakte ist die Basis für eine Durchgängigkeit von Modellen und in Folge auch für einen zielführenden Einsatz automatischer Modelltransformationen. (Pohl et al., 2012, S. 34) Es fehlen jedoch Ansätze um die Informationen aus einem Architekturmodell mit anderen etablierten Modellen zu vernetzen (Dumitrescu et al., 2021) Daraus resultieren ein geringer wahrgenommener Mehrwert von Systemarchitekturmodellen und dementsprechend geringe Akzeptanz und Umsetzung von modellbasiertem Systems Engineering.

Als zentrales Element der Durchgängigkeit müssen die Modellartefakte der Systemarchitektur nicht nur als Eingangsgrößen für andere Modelle dienen, sondern die Erkenntnisse aus der Anwendung dieser Modelle muss wiederum in der Systemarchitektur verankert und für die Mitglieder des Entwicklungsteams zugänglich und inhaltlich nachvollziehbar sein. **Es bedarf also methodischer Unterstützung zur Schaffung von klaren und bidirektionalen Beziehungen zwischen den Artefakten der Systemarchitektur und den Artefakten anderer in der Produktentstehung verwendeter Modelle.**

Albers betont im Vorwort zu Ebel, dass das Zielsystem und die darin enthaltenen Ziele, Anforderungen und Randbedingungen von zentraler Bedeutung für die Definition und Gestaltung eines Produkts sind. (Ebel, 2015) Durch Validierungsaktivitäten wird das Zielsystem weiterentwickelt. Erst in der Validierung entsteht durch Abgleich von aktuellem Systemzustand und Zielzustand Wissen, d.h. es werden Teilziele, Anforderungen und Randbedingungen für das weitere Vorgehen abgeleitet. (Albert Albers et al., 2010, S. 5) Ein Ansatz zur Zielsystemerweiterung und -konkretisierung auf unterschiedlichen Detaillierungsebenen sind virtuelle, modellbasierte Validierungsansätze in Form von Berechnungen und Simulationen. Entsprechend der geforderten Durchgängigkeit müssen die Elemente des konkretisierten Zielsystems (z. B. Anforderungen) wiederum mit Artefakten der Systemarchitektur verknüpft werden. **Für die zielführende Entwicklung von Anforderungen bedarf es der Konsistenz und Durchgängigkeit der zur Zielsystemerweiterung und -konkretisierung eingesetzten Modelle mit den in der Systemarchitektur explizierten Zusammenhängen.**

Der in dieser Arbeit adressierte Forschungslücke wird auf Basis der identifizierten Bedarfe konkretisiert: **Es fehlen methodische Ansätze zur bidirektionalen Verknüpfung von Systemarchitektur und Berechnungs- oder Simulationsmodellen, die zur modellbasierten Erweiterung und Konkretisierung des Zielsystems auf unterschiedlichen Detailebenen des SID eingesetzt werden.**

3.2 Ziel und Forschungshypothese

Entsprechend der identifizierten Forschungslücke ist das Ziel dieser Arbeit, einen Ansatz zur methodischen Unterstützung der Verknüpfung zwischen Artefakten eines Systemmodells in SysML und eines Berechnungsmodells zur Zielsystemkonkretisierung in Matlab zu entwickeln um dadurch Akzeptanz und Nutzung von Systemarchitekturen zu unterstützen.

Forschungshypothese

Durch einen Ansatz zur systematischen Nutzung der Systemarchitektur für die Modellbildung eines Berechnungsmodells und die Dokumentation der daraus abgeleiteten Entscheidungen und Anforderungen können Faktoren verbessert werden, die bislang zu geringer Akzeptanz und Nutzung der Systemarchitektur führen.

Um diese Hypothese zu untersuchen, werden die darin enthaltenen Aspekte durch mehrere Forschungsfragen adressiert, die im Folgenden aufgezeigt werden.

Wie bereits in Kapitel 2.4 aufgezeigt, sind Zweck und Vorteile von Systemarchitekturmodellen in der Praxis zwar bekannt, die tatsächliche Nutzung und effektive Einbindung in die Produktentwicklung ist jedoch nicht vollständig etabliert. Um eine bessere Nutzung von Systemarchitekturen zu ermöglichen, ist es wichtig, die **Akzeptanzfaktoren** für die aktive Nutzung einer Systemarchitektur in der Praxis zu erkennen. Daher beschäftigt sich Kapitel 4 im Rahmen einer Interviewstudie mit der folgenden Fragestellung:

- **Forschungsfrage 1:** Durch welche Faktoren wird die Akzeptanz und Nutzung der Systemarchitektur in der Praxis behindert? (siehe Kapitel 4.4)

Um eine Systemarchitektur gezielt und systematisch als **Grundlage zur Erstellung anderer Systemmodelle – z. B. Berechnungsmodelle** – nutzen zu können, muss der Zusammenhang zwischen Struktur und Elementen beider Modelle definiert sein. Dabei ist auch dem Umstand Rechnung zu tragen, dass das zu entwickelnde System gemäß dem Ansatz der Produktgenerationsentwicklung auf Elementen des Referenzsystems und Variationen dieser beruht (s. Kapitel 2.1.3). Demzufolge müssen beide Modelle die Berücksichtigung von Variationsanteilen ermöglichen. Im Rahmen der vorliegenden Arbeit dient ein Berechnungsmodell als Beispiel für ein weiteres Systemmodell neben der Systemarchitektur. Vor diesem Hintergrund liegen den Kapiteln 5.2.1 und 5.2.2 die folgenden Fragestellungen zu Grunde:

- **Forschungsfrage 2:** Wie kann das Architekturmodell strukturiert werden um eine inhaltliche Anbindung an das Berechnungsmodell und die Berücksichtigung von Variationsanteilen zu unterstützen? (siehe Kapitel 5.2.1)
- **Forschungsfrage 3:** Welche Elemente des Architekturmodells werden im Berechnungsmodell abgebildet und auf welche Weise werden sie abgebildet? (siehe Kapitel 5.2.2)

Eine **aktive und systematische Nutzung** von Methoden und Ansätzen – so auch Erstellung und Nutzung einer Systemarchitektur – wird erleichtert, wenn der Beitrag zu den unterschiedlichen Aktivitäten der Produktentwicklung bewusst ist. Basis für die Beschreibung allgemeiner Aktivitäten in der Produktentstehung ist das erweiterte ZHO-Modell, das in Kapitel 2.1.1 vorgestellt und in Kapitel 2.3.1 um die Aspekte der Kreation und Validierung ergänzt wird. Kapitel 5.2.3 beleuchtet daher die folgende Fragestellung:

- **Forschungsfrage 4:** Wie lassen sich Erstellung und Nutzung einer Systemarchitektur durch das erweiterte ZHO-Modell beschreiben? (siehe Kapitel 5.2.3)

Das Treffen von Entscheidungen und die Ableitung von Anforderungen basieren auf dem in **Validierungsaktivitäten** neu gewonnenen Wissen. Berechnungsmodelle dienen in diesem Verständnis einer virtuellen, modellbasierten Validierung, ihre Erstellung und Nutzung sollte somit nicht nur mit dem in einem Architekturmodell abgebildeten Systemverständnis konsistent sein, sondern gleichzeitig auch die Anwendung methodischer Ansätze zur systematischen Validierung ermöglichen. Das IPEK-XiL-Framework ist ein Ansatz, der die Strukturierung von Validierungsaktivitäten unter Berücksichtigung der Kundenanforderungen und Anwendungsfälle, der umgebenden Systeme und des Restsystems unterstützt. In Kapitel 5.2.4 wird daher auf Basis der folgenden Frage der Zusammenhang zwischen Architekturmodell und IPEK-XiL-Framework analysiert:

- **Forschungsfrage 5:** Wie kann das IPEK-XiL-Framework genutzt werden um Zusammenhänge zwischen Architekturmodell und Berechnungsmodell zu beschreiben? (siehe Kapitel 5.2.4)

Um ein Zielsystem ergänzen bzw. verfeinern zu können, müssen die in virtuellen, modellbasierten Aktivitäten erzeugten Ergebnisse vor dem Hintergrund der Erwartungen der Endkunden bewertet werden (vgl. Kapitel 2.3.2). Die subjektive Wahrnehmung von Endkunden kann dabei z. B. durch Erkenntnisse aus Studien, Umfragen und vorangegangenen Untersuchungen antizipiert und in Form objektivierter Größen (bspw. akzeptable Reichweite von E-Fahrzeugen in km) zur Bewertung der Ergebnisse herangezogen werden. Zentral für die Bewertung und damit auch für die Ableitung von Entscheidungen zu Zielsystemänderungen und Ableitung von Anforderungen ist der Vergleich unterschiedlicher Lösungsalternativen. Daraus leitet sich die zentrale Frage für Kapitel 5.2.5 ab.

- **Forschungsfrage 6:** Wie können alternative Lösungskonzepte im Berechnungsmodell abgebildet, analysiert und vergleichend bewertet werden? (siehe Kapitel 5.2.5)

Ein Architekturmodell bietet entsprechend der Ausführungen in Kapitel 2.4.1 das Potenzial, den Zusammenhang zwischen dem im Laufe der Produktentstehung anwachsendem Systemwissen und den Systemeigenschaften herzustellen und so als zentrale Wissensquelle zu dienen. Um dieses Potenzial nutzen zu können, müssen dokumentierte Informationen und getroffene Entscheidungen mit dem Modell vernetzt und über das Modell zugreifbar gemacht werden. Dabei ist nicht nur die Verlinkung von Informationen wichtig, sondern auch die inhaltliche Nachvollziehbarkeit der Zusammenhänge. Kapitel 5.2.6 widmet sich daher der folgenden Fragestellung.

- **Forschungsfrage 7:** Wie können die Ergebnisse des Berechnungsmodells und die daraus abgeleiteten Entscheidungen und resultierenden Anforderungen in der Systemarchitektur rückverfolgbar und inhaltlich nachverfolgbar dokumentiert werden? (siehe Kapitel 5.2.6)

Die Beantwortung der aufgezeigten Fragen bildet die Grundlage zur Entwicklung eines methodischen Ansatzes, der entsprechend der Forschungshypothese die systematische Nutzung einer Systemarchitektur zur Erstellung von Berechnungsmodellen unterstützt, den Mehrwert sichtbar macht und dadurch zur Akzeptanz und Potenzialausschöpfung von modellbasiertem Systems Engineering beiträgt. Die erwartete Situation nach Einführung des Ansatzes ist, dass es dem Entwicklungsteam vereinfacht wird, die Konsistenz und damit auch den Zusammenhang zwischen verschiedenen Ebenen des Berechnungsmodells und der daraus abgeleiteten Anforderungen mit den Inhalten der Systemarchitektur zu verbessern - auch vor dem Hintergrund, dass es unterschiedliche Systementwürfe unter Kombination verschiedener Produkt- und Entwicklungsgenerationen der beteiligten Subsysteme geben kann.

3.3 Forschungsvorgehen

Die Struktur des Forschungsvorgehens orientiert sich am Forschungsdesign nach (Marxen & Albers, 2012, S. 1014). Abbildung 3.1 stellt die einzelnen Aspekte des Forschungsdesigns in Verbindung mit dem in dieser Arbeit gewählten Vorgehen und dem Ergebnis dar. Im Rahmen der Beobachtung erfolgt die Ermittlung des Stands der Forschung mittels Literaturrecherche in Kapitel 2 dieser Arbeit. Ergebnis ist die Zielsetzung der Arbeit auf Basis des identifizierten Forschungsbedarfs, siehe Kapitel 3. Des Weiteren umfasst die Beobachtung eine semistrukturierte Interviewstudie mit Entwickelnden aus der Industrie, deren Vorgehen und Ergebnis in Kapitel 4 dargestellt ist. Ziel der Studie ist es, Faktoren zu identifizieren, die die Akzeptanz und Nutzung von MBSE in der Praxis behindern. In Kapitel 5 wird auf Basis der Erkenntnisse aus Literatur und Interviewstudie ein Ansatz erarbeitet, mit dem Ziel, die Verwendung des Architekturmodells zur Erstellung eines Berechnungsmodells methodisch so zu unterstützen, dass die identifizierten Hinderungsfaktoren verbessert werden. Der entwickelte Ansatz wird in Kapitel 6 am Beispiel der Anforderungsermittlung für Subsysteme eines elektrischen Fahrzeugantriebsystems experimentell angewendet. Die experimentelle Anwendung im industriellen Umfeld ist aufgrund der Randbedingungen der vorliegenden Arbeit nicht möglich, daher erfolgt die Anwendung des Ansatzes durch die Verfasserin. Auf Basis der experimentellen Anwendung zeigt Kapitel 7 vor dem Hintergrund der identifizierten Hinderungsfaktoren die erwarteten Potenziale und Risiken bei einer Implementierung des Ansatzes auf.

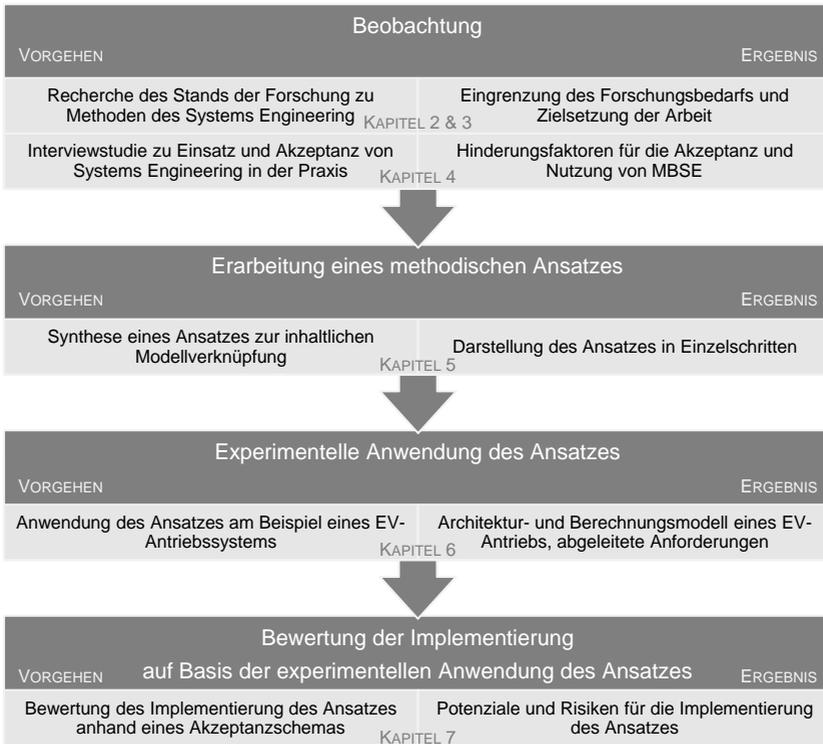


Abbildung 3.1: Darstellung des Forschungsvorgehens in Anlehnung an das Forschungsdesign nach (Marxen & Albers, 2012, S. 1014).

4 Interviewstudie zur Akzeptanz von Methoden des Systems Engineering

Wie in Kapitel 2 aufgezeigt, unterstützt Systems Engineering die ganzheitliche und durchgängige Systementwicklung. Die Anwendbarkeit und Integrierbarkeit dieser Methoden in etablierte Produktentwicklungsprozesse ist Voraussetzung für die Akzeptanz und effiziente Anwendung von Systems Engineering in der Praxis. Im Rahmen einer Interviewstudie wird daher die Akzeptanz von Systems Engineering – insbesondere MBSE – untersucht. Eine Zusammenfassung der Ergebnisse zum Zeitpunkt der Erhebung ist in (Albert Albers, Lohmeyer & Radimersky, 2012) veröffentlicht. In der vorliegenden Arbeit werden zusätzlich Beobachtungen aus der aktuellen Entwicklungspraxis in der Automobilbranche eingebracht. Am Ende dieses Kapitels werden die Ergebnisse im Rahmen der Zusammenfassung und Reflektion in den Kontext aktueller Studien und Forschungsergebnisse gesetzt. So werden bestehenden Faktoren herausgearbeitet, die die Akzeptanz und Nutzung von MBSE behindern.

4.1 Design der Interviewstudie

In Einzelinterviews werden mehrere Interviewpartner aus der industriellen Praxis bezüglich Ihrer Erfahrungen mit entwicklungsmethodischen Ansätzen im Allgemeinen und Methoden des Systems Engineering im Speziellen befragt. Um die Erkenntnisse aus den Interviews zu strukturieren und eine vergleichende Auswertung der Ergebnisse zu erleichtern werden die Interviews semi-strukturiert durchgeführt. Hierfür dient das in Abbildung 4.1 dargestellte Schema zur Akzeptanzbewertung.

	PERFORMANZ	PRÄSENTATION	PROZESS
INDIVIDUELL	wahrgenommene Leistungsfähigkeit	intuitiv handhabbar	flexibel & anpassbar
ORGANISATORISCH	Aufwand/Nutzen Verhältnis	lehr- und lernbarkeit	wiederverwendbar & erweiterbar

Abbildung 4.1: Schema zur Akzeptanzbewertung von Methoden, Werkzeugen und Ansätzen nach (Albert Albers, Lohmeyer & Radimersky, 2012).

Das Schema besteht aus zwei zentralen Akzeptanzaspekten. Es beschreibt drei Defizite von Entwicklungsmethoden und beleuchtet diese hinsichtlich zweier Dimensionen der Akzeptanz. Badke-Schaub et al. stellen fest, dass die Themen Performanz, Präsentation und Prozess übergeordnete Defizite von Methoden beschreiben. Dabei bezieht sich der Begriff Performanz auf die Leistungsfähigkeit der Methode. Defizite in der Performanz äußern sich z. B. indem die Methode für den Anwendungsfall nicht ausreichend ist oder der Nutzer die Eignung der Methode nur schwer einschätzen kann. Der Begriff Präsentation steht dafür wie gut eine Methode dargestellt bzw. vermittelt werden kann. Dies beinhaltet auch wie gut die Methode für den Nutzer handhabbar ist und welches Vorwissen zur Anwendung der Methode benötigt wird. Der Begriff Prozess zielt darauf ab, wie gut die Methode in das bestehende Entwicklungsumfeld integriert werden kann. Mängel im Bereich Prozess äußern sich beispielsweise durch schlechte Anpassbarkeit oder Erweiterbarkeit der Methode. (Badke-Schaub, Daalhuizen & Roozenburg, 2011)

Diese drei Defizite können nach Albers et. al. nochmals weiter detailliert werden in die Dimensionen individuelle Akzeptanz und organisatorische Akzeptanz. Individuelle Akzeptanz bezieht sich dabei auf die Wahrnehmung der drei Aspekte aus persönlicher Sicht eines einzelnen Mitglieds eines Entwicklungsteams. Dabei drückt sich die Performanz in der wahrgenommenen Leistungsfähigkeit aus, d.h. welche Vorteile nimmt die Person durch die Anwendung der Methode wahr. Der Aspekt der Präsentation ist aus individueller Sicht einer Person, wie intuitiv die Methode gehandhabt oder angewendet werden kann. Der Prozessuale Aspekt ist für eine einzelne Person weniger die formale Integration in bestehende Prozesse, sondern vielmehr, ob sich die Methode flexibel in den Arbeitsalltag und die persönlichen Aktivitäten integrieren und auf aktuelle Fragestellungen anpassen lässt. Der Fokus der Organisation liegt eher auf formalen Aspekten. Hierbei bedeutet Performanz, dass das Verhältnis von finanziellem und zeitlichem Aufwand zum erkennbaren Nutzen der Methode stimmen muss. Unter den Aspekt der Präsentation fällt die Frage,

wie gut die Methode durch Trainingsmaßnahmen lehrbar und erlernbar ist. Das Kriterium Prozess umfasst zum einen, wie gut sich die Methode in die bestehende Prozesslandschaft integrieren und wo nötig erweitern lässt, aber auch ob sich die Ergebnisse im Entwicklungsprozess weiterverwenden lassen. In der Kombination ergeben sich sechs Kriterien, die während der Interviews zur Bewertung der von den Interviewpartnern genannten Methoden herangezogen werden. (Albert Albers, Lohmeyer & Radimersky, 2012).

Die insgesamt zehn interviewten Personen sind in der Produktentwicklung unterschiedlicher Branchen tätig. Unter anderem Automobilbau, mobile Arbeitsmaschinen, Medizintechnik, Gerätekonstruktion, Maschinen- und Anlagenbau und Ingenieurdienstleistung. Ebenso variiert die Unternehmensgröße von großen und Mittelständischen Unternehmen bis hin zu KMU. Die Interviews haben jeweils eine Dauer von ca. 30 Minuten. Während dieser Zeit werden seitens der Interviewten zunächst Methoden, Ansätze und Werkzeuge gesammelt, die im Entwicklungsalltag eingesetzt werden, um die interdisziplinäre Zusammenarbeit zu vereinfachen und ganzheitliche Systembetrachtung zu unterstützen. Anhand des Schemas in Abbildung 4.1 werden die Nennungen von den Interviewpartnern beurteilt. Sofern im Verlauf des Gesprächs die Aspekte Systems Engineering und MBSE nicht beleuchtet werden, wird in einem zweiten Schritt gezielt nach der Bekanntheit und der Definition dieser Aspekte gefragt, ebenso nach der Einschätzung der interviewten Person bezüglich zukünftiger Bedeutung des Systems Engineering in der Praxis. Die Auswertung erfolgt im Nachgang der Interviews anhand der aufgezeichneten Interviewmitschnitte. Einschätzungen und genannte Methoden werden unter Berücksichtigung des im Interview eingesetzten Bewertungsschemas in Cluster zusammengefasst.

4.2 Ergebnisse der Interviewstudie

Im Vergleich der Interviewergebnisse wird deutlich, dass das Verständnis des Begriffs „Systems Engineering“ stark variiert. Ein Teil der Interviewpartner versteht den Begriff mehr im Sinne eines Denkansatzes oder „Mindsets“ während der Entwicklung. Dies umfasst domänenübergreifendes Denken und systemische und systematische Analyse von Zusammenhängen, Strukturelementen und Funktionen. Dadurch wird aus Sicht der Interviewten auch die systematische Beschreibung und Gestaltung der erkannten Zusammenhänge unterstützt und die Komplexität des Produkts somit besser handhabbar. Der Denkansatz umfasst jedoch nicht zwingend konkrete Methoden und Werkzeuge. Ein anderer Teil der Interviewpartner verknüpft den Begriff des Systems Engineering eng mit dem Ansatz der Systemtechnik und

der damit verbundenen formalisierten Systembeschreibung auf Basis von Funktionen mit definiertem In- und Output und Systemstrukturen. Methoden, Sprachen und Tools des MBSE stehen dabei im Fokus. Die Vision des MBSE, zukünftig nicht mehr dokumentenbasiert, sondern modellbasiert – anhand interdisziplinärer Produktmodelle entlang des Produktlebenszyklus - zu entwickeln wird von den Interviewpartnern zwar grundsätzlich als gewinnbringend eingeschätzt, die Umsetzbarkeit dieser Vision wird jedoch in Frage gestellt. Wenigen Interviewpartnern ist zum Zeitpunkt der Interviewstudie der Begriff des Systems Engineering nicht geläufig.

Abbildung 4.2 zeigt die Nennungen von Methoden, Ansätzen und Werkzeugen im Rahmen der Interviewstudie. Durch die vergleichende Auswertung der Interviews lassen sich die Nennungen grob in die dargestellten Bereiche gruppieren. Es wird deutlich, dass aus Erfahrung der Interviewpartner viele Prozess-, Projekt- und Informations-lenkende Methoden die interdisziplinäre Produktentwicklung unterstützen. Daneben werden auch Ansätze zur System- und Funktionsmodellierung genannt. Hierbei finden sich vor allem bei der Systemmodellierung sowohl domänenspezifische Modelle (z. B. CAD, MKS) als auch Modelle zur domänenunabhängigen Abbildung von Systemverhalten (z. B. Systemsimulation). Auch in der Modellierungssprache SysML wird zum Zeitpunkt der Interviews von den Befragten Potenzial zur Unterstützung domänenübergreifender Systementwicklung gesehen. Mit Fokus auf den in Kapitel 3 aufgezeigten Forschungsbedarf und die abgeleiteten Forschungsfragen wird daher im Folgenden auf die Systemmodellierung mit SysML im Sinne des MBSE eingegangen.

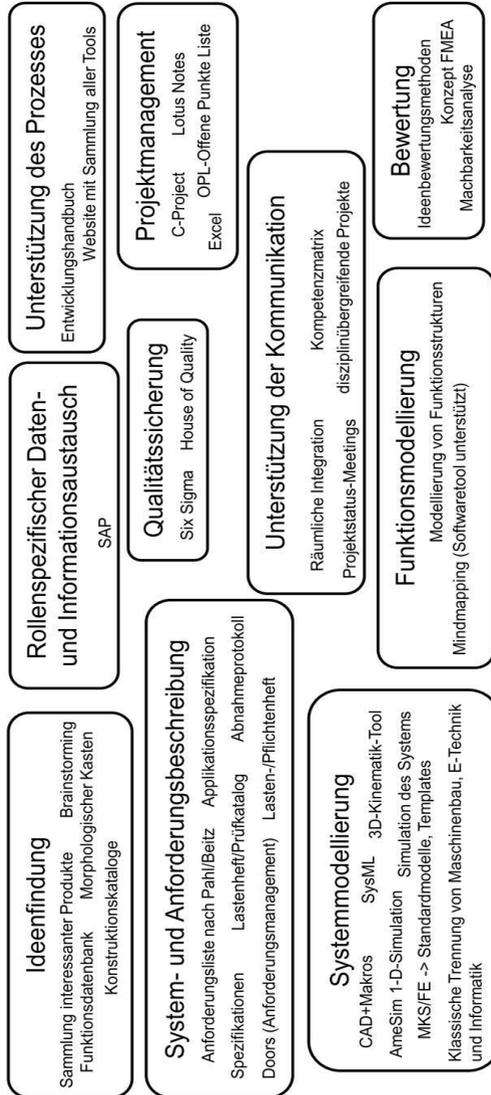


Abbildung 4.2: Nennung von Methoden, Ansätzen und Werkzeugen in den Interviews (Albert Albers, Lohmeyer & Radimersky, 2012).

Zum Zeitpunkt der Interviews wird zwar ein gemeinsames, dokumentiertes und kommunizierbares Systemverständnis als wichtig erachtet, die Leistungsfähigkeit von Architekturmodellen in der Modellierungssprache SysML wird jedoch von vielen Interviewpartnern kritisch gesehen. Die für den Maschinenbau klassische Funktionsmodellierung mit getrennten Stoff- Energie und Informationsflüssen wird als aufwendig wahrgenommen. Des Weiteren fehlt den Interviewten sowohl methodisch als auch werkzeugseitig eine geeignete Anbindung an weitere Modelle wie z. B. CAD-Modelle oder Simulationsmodelle. Der Modellierungsaufwand wird im Allgemeinen als hoch eingeschätzt, der spürbare Projektnutzen ist durch die fehlende Möglichkeit zur Berechnung und Simulation aus Sicht der meisten interviewten Personen jedoch gering. Die Handhabung der verfügbaren Tools zur Modellierung mit SysML wird als zusätzlich aufwendig und oft wenig intuitiv beurteilt. Aufgrund von Standards zur Modellierung empfinden einige der Befragten auch die Flexibilität als eingeschränkt. Die Zusammenfassung der Beurteilungen zeigt damit, dass die individuelle Akzeptanz zu diesem Zeitpunkt gering ist.

Aus organisatorischen Gesichtspunkten schätzen die Interviewpartner die Erstellung von Systemmodellen mit SysML positiver ein. Im Wandel von dokumentenbasierter Entwicklung hin zu modellbasierter Entwicklung werden insbesondere Chancen hinsichtlich einer durchgängigen, domänenübergreifenden Abbildung von Systemzusammenhängen und deren Verknüpfung mit Anforderungen und Verifikations- und Validierungsmaßnahmen gesehen. Es wird erwartet, dass dadurch kritische Punkte eines Projekts bereits frühzeitig, fachdisziplinübergreifend identifiziert und dokumentiert werden können. So werden auch frühzeitige Maßnahmen zur Risikominimierung in einem Projekt unterstützt. Sofern sich diese Erwartung bestätigt und damit der Nutzen des MBSE in den Entwicklungsprojekten spürbar ist, sehen die Interviewpartner auch den Einführungs- und Integrationsaufwand auf Seiten des Unternehmens als akzeptabel an. Allerdings besteht noch kein klares Bild, wie die inhaltliche Verknüpfung zu Verifikations- und Validierungsmaßnahmen und deren Ergebnissen erfolgen kann, sodass diese Informationen im weiteren Entwicklungsprozess zugänglich und nutzbar sind. Zum Zeitpunkt der Interviews wird in den befragten Unternehmen noch keine ganzheitliche Systemmodellierung im Sinne des MBSE eingesetzt. Vielmehr kommen sehr spezifische und ausgereifte Modelle zum Einsatz, um bspw. das Systemverhalten zu simulieren oder die Gestalt des Systems zu entwickeln und darzustellen. Die disziplinübergreifende Zusammenarbeit wird zumeist versucht durch begleitende Dokumente – z. B. Beschreibung der Systemkonfiguration, Anforderungsspezifikationen, etc. – zu unterstützen. Zum Teil werden diese Dokumente von einer zentralen Person gepflegt, um den Überblick über das Gesamtsystem und das Identifizieren potenzieller Risiken zu erleichtern. Übereinstimmend über alle Interviews hinweg wird jedoch bestätigt, dass die Produkte zunehmend komplexer werden und das Zusammenspiel von Elektronik, Software und

Mechanik kontinuierlich an Bedeutung gewinnt. Infolgedessen müssen Entwickler unterschiedlicher Disziplinen zunehmend gemeinsam Entscheidungen treffen, um ein optimales Gesamtsystem zu entwickeln. Als Basis für solche Entscheidungen muss ein gemeinsames Verständnis der Systemzusammenhänge und der disziplinübergreifenden Wechselwirkungen im System geschaffen werden.

4.3 Ergänzende Beobachtungen aus der Praxis

Ergänzend zu der in Kapitel 4.1 und Kapitel 4.2 vorgestellten Interviewstudie werden weitere Beobachtungen aus der aktuellen Entwicklungspraxis erhoben. Basis der Beobachtungen sind Gespräche mit vier Personen, die im Zeitraum 2018 bis 2023 in der interdisziplinären Entwicklung mechatronischer Produkte der Automobilindustrie tätig waren. Drei der vier Personen waren dabei in der Systementwicklung tätig, eine Person war projektleitend tätig. Fokus der Gespräche ist die Erstellung und Nutzung von Systemarchitekturmodellen im Projektalltag. Die Gesprächspartner vertreten dabei als Mitglieder des Entwicklungsteams die individuelle Sicht entsprechend Abbildung 4.1. Die Gespräche wurden anhand der folgenden drei Leitfragen geführt:

- Wen unterstützt die Systemarchitektur und wodurch? (Performanz)
- Wie gut ist ein Architekturmodell in SysML für Teammitglieder verständlich, die nicht aktiv an dessen Erstellung beteiligt sind? (Präsentation)
- Wie könnte die Systemarchitektur besser genutzt werden? (Prozess)

Als Systemarchitektur wird von den Gesprächspartnern der Aufbau des Systems und dessen funktionale Zusammenhänge verstanden. Einige Gesprächspartner beziehen in ihrem Verständnis den Begriff „Architektur“ dabei nicht auf ein einzelnes Modell, sondern verstehen die Gesamtheit aller Systemmodelle als Repräsentation der Systemarchitektur. Für andere Gesprächspartner ist das Systemarchitekturmodell ein qualitatives Modell z. B. in SysML. Übereinstimmend heben die Gesprächspartner hervor, dass es von großer Bedeutung ist, die Konsistenz der unterschiedlichen Systemmodelle (z. B. Verhaltensmodelle, Thermomodelle, qualitative Funktionsbeschreibungen, ...) sicherzustellen. Das Potenzial und der Nutzen eines qualitativen Architekturmodells wird in einem gemeinsamen und eindeutigen, explizierten Systemverständnis des Entwicklungsteams gesehen. Zum Systemverständnis gehört die Struktur des zu entwickelnden Systems, dessen Schnittstellen nach außen und die inneren Schnittstellen zwischen Subsystemen sowie die funktionalen Zusammenhänge und Verhaltensmodelle des Systems. Dieses explizierte Verständnis bildet auch die Basis für gemeinsame Diskussionen und Entwicklungsentscheidungen, und dadurch für die Konkretisierung des Zielsystems. Zur konkreten,

quantifizierten Ableitung von Anforderungen kommen im Umfeld der Befragten im Wesentlichen virtuelle, ausführbare Modelle zum Einsatz. Ein Teil der Gesprächspartner sieht es vor diesem Hintergrund als Nachteil an, dass qualitative Architekturmodelle keine Simulation oder Berechnung erlauben. Andere Befragte sehen in diesem Zusammenhang das Potenzial, die Konsistenz aller Systemmodelle durch ein qualitatives Architekturmodell zu unterstützen. Allerdings wird dieses Potenzial laut der Gespräche aktuell nicht ausgeschöpft. Hintergrund sind unter anderem der hohe Aufwand in der Modellerstellung und eine suboptimale Zugänglichkeit des Modells für diejenigen Teammitglieder, die nicht direkt an der Modellerstellung beteiligt sind. Daraus resultiert, dass auch die Nutzung des im Modell hinterlegten Wissens durch alle beteiligten Domänen und Rollen nicht gegeben ist. Als weiterer Nachteil wird genannt, dass Prämissen und Randbedingungen (z.B. für Simulationsmodelle) nicht direkt aus dem Architekturmodell exportiert werden können oder zumindest im Rahmen einer gemeinsamen Datenbasis von allen Modellen genutzt werden können. Eine weitere Feststellung ist, dass die Ergebnisse der quantitativen Systemmodelle und die darauf beruhenden Entscheidungen aktuell keine Verlinkung in das qualitative Architekturmodell finden. Das Potenzial, für das Team gemeinsam zu explizieren warum und auf Basis welcher virtuellen Validierungsergebnissen eine bestimmte Lösung gewählt wurde, wird gesehen aber nicht genutzt.

Die Einschätzungen und Aussagen der Gesprächspartner sind konsistent mit dem in Kapitel 2.4.3 dargelegten Stand der Forschung zum Einsatz von modellbasiertem Systems Engineering in der Praxis. Trotz der kleinen Interviewgruppe können die Ergebnisse daher als relevant und verallgemeinerbar bewertet werden.

4.4 Zusammenfassung und Reflektion der Ergebnisse

Zusammenfassend kann aus der Interviewstudie und den ergänzenden Praxisbeobachtungen entnommen werden, dass ein ganzheitliches Systemverständnis von allen Interviewten als notwendige Voraussetzung für den Umgang mit Komplexität in der Produktentwicklung gesehen wird. In der Vision des MBSE von der dokumentenbasierten Entwicklung hin zu einer modellbasierten Entwicklung zu kommen, sehen die Interviewten vor allem einen Vorteil in Bezug auf eine bessere Nachverfolgbarkeit und Nachvollziehbarkeit von Anforderungen, Systemzusammenhängen und Verifikations- und Validierungsmaßnahmen. Das kann die Grundlage für disziplinübergreifende Entscheidungen zur Optimierung des Gesamtsystems bieten. Allerdings ist zum Zeitpunkt der Interviews die Nutzung von SysML-Modellen unter den Interviewpartnern nicht üblich, Gründe liegen sowohl in geringer individueller als auch organisatorischer Akzeptanz. Die Erwartung an verbesserte Durchgängigkeit des Entwicklungsgeschehens durch MBSE besteht auch heute, wie Dumitrescu et. al.

in ihren Forschungen feststellen. Dennoch wird MBSE nach wie vor nicht flächendeckend eingesetzt, sondern der Leistungsstand hängt stark von Aspekten wie Unternehmensgröße und Branche ab. (Dumitrescu et al., 2021)

Sowohl individuell als auch organisatorisch treten in der Interviewstudie vor allem befürchtete Defizite in der Performanz gepaart mit Defiziten in der Präsentation in den Vordergrund. Ein zentraler Aspekt ist dabei die geringe direkte Leistungsfähigkeit bzw. der geringe direkte Nutzen eines SysML-Modells und gleichzeitig aufwendiger Modellerstellung. Im Gegensatz zu Berechnungs- oder Simulationsmodellen lassen sich keine quantitativ verwertbaren Ergebnisse aus einem SysML-Modell generieren. Auch ist keine automatisierte Transformation des Architekturmodells in Berechnungs- und Simulationsmodelle etabliert. Neuere Forschungsergebnisse beschäftigen sich mit der Anbindung unterschiedlicher Modellarten an SysML-Modelle. So zeigen beispielsweise sowohl Herzig et al. als auch Vidal und Villota, dass die werkzeugseitige Anbindung von SysML-Modellen an Berechnungsmodelle möglich ist. (Herzig et al., 2013; Vidal & Villota, 2018) Offen bleibt dabei allerdings, wie die Modelle aufgebaut sein müssen um den automatisierten Datenaustausch zu ermöglichen und welche Daten oder Artefakte ausgetauscht werden sollten. Ebenso ist unklar, wie eine geeignete Anbindung und Nutzung der Systemarchitektur an Verifikations- und Validierungsaktivitäten und deren Ergebnisse erfolgen kann. Mandel et al. entwickeln methodische Ansätze, die die Verknüpfung von Validierungssystem und Produktsystem in SysML ermöglichen und zeigen hierbei auch Modellverknüpfungen zu Simulationsmodellen. (Mandel et al., 2020; Mandel, Wäschle et al., 2021; Mandel, Böning et al., 2021) Ein weiterer Aspekt ist jedoch die Art der Rückführung der Validierungsergebnisse im Sinne des Wissensmanagements und die Dokumentation von Entwicklungsentscheidungen, die auf den Ergebnissen beruhen.

Allerdings wird der langfristige Projektnutzen des MBSE bezüglich Nachverfolgbarkeit und Nachvollziehbarkeit und damit der Vorteil zum Umgang mit Komplexität in der Produktentwicklung gesehen. In jüngerer Zeit ist zu beobachten, dass die modellbasierte Entwicklung und damit die Abbildung von Systemarchitekturen mit SysML an Bedeutung zunimmt, wie beispielsweise Gausemeier et al. und Stützel und Paul. feststellen. (Gausemeier et al., 2015; Stützel & Paul, 2021). Insbesondere in der Automobilindustrie ist dies unter anderem durch das vom VDA etablierten Prozessmodell Automotive SPICE und die Norm ISO 26262 (vgl. Kapitel 2.2.4) begründet. Beide fordern die Dokumentation der Systemarchitektur, um Konsistenz und bidirektionale Nachverfolgbarkeit zwischen Anforderungen und Systemarchitektur herzustellen. (ISO 26262-4; VDA QMC Working Group 13 / Automotive SIG, 2017)

Um individuelle und organisatorische Akzeptanz der Abbildung von Systemarchitekturen mit SysML zu fördern ist es wichtig, auch die prozessualen Aspekte der Anpassbarkeit und Wiederverwendbarkeit zu berücksichtigen. Daraus kann abgeleitet werden, dass die SysML-Modellerstellung nicht zum alleinigen Zweck der Nachverfolgbarkeit erfolgen sollte, sondern idealerweise als Grundlage für bereits etablierte Methoden der Produktentwicklung dient. Diese sind beispielweise funktions- und strukturbasierte Methoden wie FMEA, QFD etc. oder Systemsimulationen. Insbesondere das Aufzeigen der Unterstützungsmöglichkeiten von MBSE für Gesamtsystemsimulationen birgt das Potenzial, dem Kritikpunkt mangelnder Performanz von SysML-Modellen entgegenzuwirken. Des Weiteren könnte die Gesamtsystemsimulation im Gegenzug dazu beitragen, alternative Systemvarianten auf hoher Abstraktionsebene zu vergleichen und Entwicklungsentscheidung nachvollziehbar zu dokumentieren, wie es beispielsweise auch durch den ASPICE-Prozess in Base Practice 5 der Architekturerstellung (SYS.3) gefordert wird. (VDA QMC Working Group 13 / Automotive SIG, 2017)

Aus den Interviewergebnissen und der Reflektion dieser vor dem Hintergrund des Stands der Forschung können die folgenden Hinderungsfaktoren für die Nutzung von MBSE abgeleitet und damit **Forschungsfrage 1** beantwortet werden.

Faktoren, die die Akzeptanz und Nutzung von MBSE behindern:

- **Das Aufwand/Nutzen Verhältnis der Architekturmodellerstellung wird als sehr hoch empfunden:** Der Modellierungsaufwand wird im Vergleich zur Nutzbarkeit des Architekturmodells als sehr hoch wahrgenommen. „Nur“ verbesserte Dokumentation und Kommunikation ist nicht ausreichend. Es ist unklar, wie das Architekturmodell für weitere Aktivitäten genutzt werden kann und welche Struktur sich dafür eignet
- **Die Arbeit mit quantitativen Modellen wird nicht unterstützt:** Die Darstellung der Systemarchitektur in SysML wird als wenig leistungsfähig wahrgenommen, da sie keine Berechnungen oder Simulationen ermöglicht. Es ist unklar, wie das Architekturmodell sowohl methodisch als auch werkzeugtechnisch für die Erstellung von Simulations- oder Berechnungsmodellen genutzt werden kann.
- **Die inhaltliche Nachvollziehbarkeit abgeleiteter Anforderungen wird nicht unterstützt:** Es werden Chancen zur frühzeitigen Identifikation von Zielkonflikten durch durchgängige, domänenübergreifende Abbildung von Systemzusammenhängen und deren Verknüpfung mit Anforderungen gesehen. Die Nachvollziehbarkeit der Inhalte der Anforderungen ist dadurch jedoch noch nicht gegeben, diese ist jedoch für die Auflösung von Zielkonflikten von Bedeutung.

Im Folgenden wird untersucht, wie die Effekte der identifizierten Faktoren verringert werden könnten. Hierzu wird ein Ansatz abgeleitet, der aufzeigt, wie ein SysML-Modell die Erstellung und Nutzung eines Berechnungsmodells unterstützen kann. Ziel ist, den Nutzen des SysML-Modells zu erhöhen, indem sowohl die Modellbildung des Berechnungsmodells als auch die inhaltliche und Nachvollziehbarkeit der aus den Berechnungen konkretisierten Anforderungen zu erleichtern.

5 Ansatz zur Unterstützung quantitativer Modellbildung durch die Systemarchitektur

Aufbauend auf den Ergebnissen der Literaturrecherche und der Interviewstudie (Kapitel 4) – wird ein Ansatz zur Anforderungsableitung erarbeitet, der ein Systemmodell in der Modellierungssprache SysML als Basis für eine quantitative Modellierung des Systems in Form eines Berechnungsmodells in Matlab nutzt um so die abgeleiteten Entscheidungen und Anforderungen im Architekturmodell abbildbar und inhaltlich nachvollziehbar zu dokumentieren. Die folgenden Kapitel fokussieren auf die Ziele des Ansatzes und die Herleitung und Beschreibung des Ansatzes. Abschließend wird der Ansatz nochmals zusammengefasst.

5.1 Anforderungen an den Ansatz

Literaturrecherche und Interviewstudie (Kapitel 4) zeigen, dass Systems Engineering – und damit auch die modellbasierte Darstellung der Systemzusammenhänge – im Entwicklungsalltag vermehrt an Bedeutung gewinnt. Zum einen motiviert durch zunehmende Produktkomplexität, zum anderen insbesondere in der Automobilentwicklung durch Entwicklungsstandards wie ASPICE (VDA QMC Working Group 13 / Automotive SIG, 2017) und Funktionale Sicherheit (ISO 26262-4). Ebenfalls wird deutlich, dass die Akzeptanz des modellbasierten Systems Engineering sehr stark vom wahrgenommenen Nutzen abhängt. Ein Architekturmodell erzeugt für viele Entwickler zunächst wenig individuell sichtbaren Nutzen. Allerdings kann es im Sinne eines Rückgrats für die Produktentwicklung die Harmonisierung bereits etablierter Entwicklungsmethoden ermöglichen und damit eine verbesserte Konsistenz des Produkts in sich selbst. Um diese Rückgrat-Funktion erfüllen zu können, muss einem Entwicklungsteam klar sein, wie das Systemmodell in den bisherigen Methoden nutzbar ist und diese unterstützt.

Die Konkretisierung des Zielsystems ist ein wichtiger Baustein des Produktentstehungsprozesses, der die Grundlage für ein valides Endprodukt bildet. Es gibt eine Vielzahl etablierter Methoden zur Anforderungsentwicklung. Um eindeutige, quantifizierte und damit prüfbare Anforderungen an ein Produkt oder an Subsysteme des Produkts abzuleiten kommen auch Berechnungs- und Simulationsmodelle zum Einsatz. In der Regel entstehen im Laufe des Entwicklungsprozesses mehrere solcher Modelle auf unterschiedlichen Systemebenen mit unterschiedlichen Modellzwecken

(z. B. Verbrauchsberechnung des Fahrzeugs in verschiedenen Zyklen, Thermosimulation einzelner Subsysteme, FEM-Simulation von Bauteilen, ...). Häufig werden diese Modelle von unterschiedlichen Experten erstellt, damit besteht die Gefahr von Inkonsistenzen in den zu treffenden Annahmen oder den abgebildeten Randbedingungen. Eine Aufgabe des Systems Engineering ist es hier, die Konsistenz über alle Produktmodelle hinweg zu sichern. Ein Architekturmodell hat das Potenzial durch seine Rückgrat-Funktion hierbei zu unterstützen - sofern es systematisch eingesetzt wird.

Basierend auf diesem Ziel und den Ergebnissen der vorangegangenen Studie können folgende Anforderungen an den Ansatz abgeleitet werden:

- Der Ansatz soll bereits etablierte Methoden, Ansätze und Frameworks berücksichtigen, um die Nutzung der Systemarchitektur in anderen Aktivitäten der Produktentstehung zu vereinfachen.
- Der Ansatz soll die inhaltliche Verknüpfung der Systemarchitektur mit einem Berechnungsmodell zur Entwicklung von Anforderungen ermöglichen, um eine Basis für die werkzeugseitige Transformation zu schaffen.
- Der Ansatz soll die Dokumentation der Berechnungsergebnisse und darauf beruhender Entwicklungsentscheidungen ermöglichen, um die inhaltliche Nachvollziehbarkeit von Anforderungen zu unterstützen.

5.2 Herleitung und Definition des Ansatzes

In den folgenden Kapiteln wird mit Hilfe der in Kapitel 3.2 dargelegten Forschungsfragen und unter Berücksichtigung der Anforderungen in Kapitel 5.1 ein methodischer Ansatz synthetisiert, der die Inhalte der Systemarchitektur für die Erstellung eines Berechnungsmodells zur Anforderungsermittlung nutzt. Zu diesem Zweck gehen die folgenden Kapitel auf Forschungsfrage zwei bis sieben ein. Auf Basis der Antworten auf diese Forschungsfragen, werden einzelne methodische Schritte abgeleitet, die ein Entwicklungsteam dabei unterstützen die im Systemarchitekturmodell explizierten Zusammenhänge konsistent für die Modellerstellung einer virtuellen Validierungsumgebung zu nutzen und damit die Erweiterung des Zielsystems zu unterstützen (siehe Kapitel 5.3).

5.2.1 Struktur des Architekturmodells und Berücksichtigung von Variationsanteilen des Modells der SGE

Eine Dekomposition des Systems über mehrere Systemebenen hinweg dient nicht nur dem Verständnis der Systemzusammenhänge, sondern ist auch für die Ableitung und Abgrenzung von Subsystemanforderungen von Bedeutung, wie Pohl zeigt. (Pohl et al., 2012) Validierungsaktivitäten unter Nutzung des IPEK-XIL-Ansatzes bedingen ebenfalls klare Abstraktionsebenen zur Identifikation des abzubildenden Sul und der Granularität des abzubildenden Restsystems zum Einsatz. (Albert Albers & Düser, 2008) Im Rahmen des Modells der SGE wird deutlich, dass sich auch Variationen auf sehr unterschiedlichen Systemebenen abspielen können. (Albert Albers et al., 2022) Daraus lässt sich ableiten, dass eine eindeutige, klare Trennung von Abstraktionsebenen im Architekturmodell die Verwendung der beinhalteten Informationen in anderen Aktivitäten der Produktentstehung unterstützt. Albers et al. zeigen auf, dass Prinzipvariationen anhand der Veränderung von Teilelementen des Systems und oder deren Schnittstellen identifiziert werden können, Ausprägungsvariationen können durch veränderte Attribute gleichbleibender Teilsysteme und Schnittstellen identifiziert werden. (Albert Albers, Rapp et al., 2020) Teilelemente und Schnittstellen werden in SysML in Form von logischen Blöcken und deren Interfaces abgebildet, anhand deren Änderungen können Variationsarten identifiziert werden, wie auch Wäschle et al. zeigen. (Martin, 2019; Wäschle et al., 2020) Die Ausprägungen von Teilsystemen können in SysML durch Instanzen (parts) der entsprechenden Blöcke und deren Werte dargestellt werden. Eine Ausprägungsvariation kann also anhand veränderter Werte identifiziert und abgebildet werden. Demzufolge ist es sinnvoll, die Sicht auf logische Strukturen in der Systemarchitektur klar von der Sicht auf physische Eigenschaften zu trennen. Dieses Prinzip der Trennung findet sich auch in Berechnungsmodellen wieder. Die Systemelemente und deren Schnittstellen werden für ein definiertes Lösungsprinzip im Berechnungsmodell abgebildet, Ausprägungsvariationen können dann z. B. durch Parametervariationen abgebildet werden. Für ein verändertes Lösungsprinzip muss auch die Abbildung der Systemelemente und deren Schnittstellen verändert werden. Eine Trennung dieser beiden Aspekte im Architekturmodell kann also die inhaltliche Übernahme von Parametern und Modellstrukturen zwischen den beiden Modellen vereinfachen.

Das in Kapitel 2.4.2 erläuterte SPES-Modeling-Framework grenzt die Systemebenen in Form von Abstraktionsebenen klar voneinander ab. Wobei jede Ebene direkt aus der darüber liegenden Systemebene abgeleitet ist und mit dieser somit in einem eindeutigen Zusammenhang steht. Zusätzlich unterscheidet das SPES-Modeling-Framework unter anderem zwischen logischer Sicht und technischer Sicht, erlaubt also die Trennung von Lösungskonzept und Ausprägung. (Pohl et al., 2012) Das

Framework scheint als grundlegende Struktur für den zu entwickelnden Ansatz somit geeignet zu sein und wird im Folgenden verwendet. Abweichend zur originalen, englischen Darstellung wird die „Requirements Viewpoint“ in dieser Arbeit jedoch mit dem Terminus „Ziel- und Anforderungs-Sicht“ überschrieben, um dem umfassenderen Verständnis des Zielsystems (siehe Kapitel 2.2.1) Rechnung zu tragen.

Abbildung 5.1 zeigt schematisch auf, wie die Struktur des SPES-Modeling-Frameworks im zu entwickelnden Ansatz genutzt werden kann, um vorangegangene Produkt- und Entwicklungsgenerationen auf unterschiedlichen Systemebenen zu berücksichtigen.

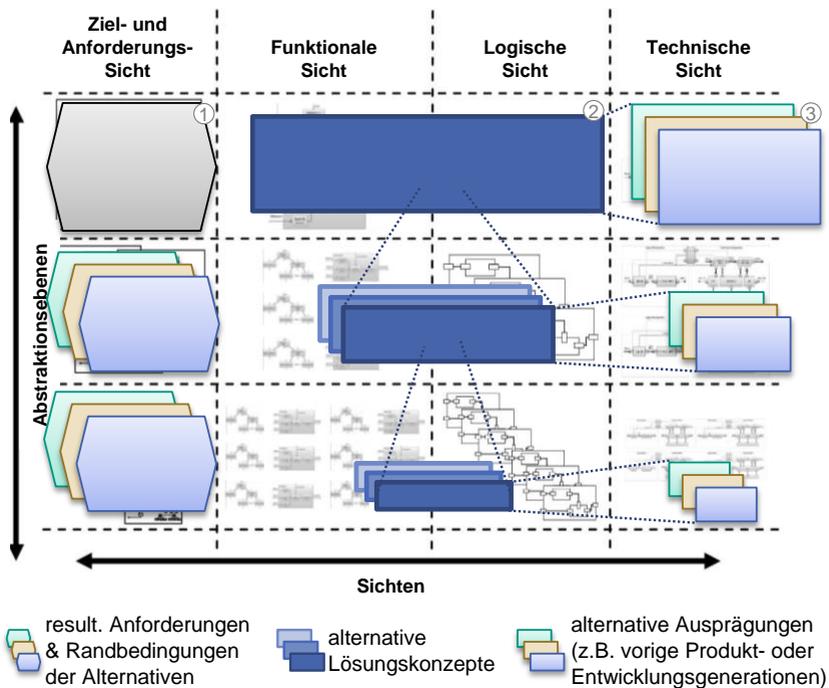


Abbildung 5.1: Elemente alternativer Systemarchitekturen eingeordnet in die Darstellung des SPES-Modeling-Framework von (Pohl et al., 2012)

In der ersten Abstraktionsebene wird in der Anforderungssicht der Kontext des zu entwickelnden Systems sowie Stakeholderanforderungen an das System abgebildet (in Abbildung 5.1 in der Anforderungs-Sicht der ersten Abstraktionsebene grau

dargestellt ①). Der Kontext beinhaltet zum Beispiel im Sinne eines System of Systems die Beschreibung umgebender Systeme mit denen das zu entwickelnde System wechselwirkt und die geforderten Schnittstellen des Systems zu diesen. Außerdem werden die Anwendungsfälle des zu entwickelnden Systems beschrieben sowie die relevanten Stakeholder. Auch bereits definierte Anforderungen sind beschrieben, die sich beispielsweise aus Kundenwünschen, aus den umgebenden Systemen, aus geltenden Gesetzen oder Umwelteinflüssen wie z. B. Umgebungstemperatur etc. ergeben. Auf Basis des Kontextes und der Anforderungen auf oberster Systemebene werden die notwendigen Systemfunktionen abgeleitet und als Startpunkt der späteren Funktionsstruktur in der Funktionssicht abgelegt. Die Logische Struktur des Systems besteht auf dieser Ebene aus einer Black-Box-Ansicht des zu entwickelnden Systems. Die Kombination aus beidem (Funktion und logische Struktur) kann als Lösungskonzept verstanden werden (in Abbildung 5.1 in der „Funktionalen Sicht“ und der „Logischen Sicht“ der ersten Abstraktionsebene dunkelblau dargestellt ②). Die technische Umsetzung dieser Black-Box entsteht üblicherweise iterativ im Laufe des Produktentstehungsprozesses durch die Aktivitäten der Produktentstehung - Systementscheidungen, Berechnungen, Detailgestaltung etc. auf den unteren Systemebenen und in den Domänen. Die Integration der Komponenten und Subsysteme führt wiederum zu technischen Lösungen auf höheren Systemebenen (siehe Abbildung 5.3). Dennoch kann bereits auf oberster Ebene berücksichtigt werden, ob es eine bereits bestehende Lösung (Element des Referenzsystems, z. B. vorangehende Produkt- oder Entwicklungsgeneration der gleichen oder ähnlichen Produkte) gibt, die die geforderte Funktionalität unter den Anforderungen und Randbedingungen erfüllt. Abhängig von den getroffenen Entwicklungsentscheidungen wird das entwickelte System schlussendlich eine von mehreren alternativen Ausprägungen haben (in Abbildung 5.1 in der „Technischen Sicht“ der ersten Abstraktionsebene mehrfarbig dargestellt ③).

Die beschriebenen Zusammenhänge setzen sich über alle Systemebenen bis zu den Domänenebenen hin fort (in Abbildung 5.1 schematisch durch die dritte Abstraktionsebene dargestellt). Daran wird deutlich, dass es zu einer hohen Anzahl alternativer Systemausprägungen geben kann und diese zu unterschiedlichen Anforderungen an die Subsysteme führen. Für die Entwicklung konsistenter und quantifizierter Anforderungen über alle Systemebenen hinweg muss also eine Möglichkeit geschaffen werden, unterschiedliche Systemausprägungen auf quantifizierter Basis zu unterscheiden und zu bewerten. Für den vorliegenden Ansatz wird das SPES-Modeling-Framework in Kombination mit dem hier aufgezeigten Verständnis von alternativen Lösungen und Ausprägungen genutzt, um die systematische und strukturierte Beschreibung und Bewertung alternativer Systemausprägungen zu unterstützen. Übernahme- und Ausprägungsvariationen vorhandener Subsysteme werden in der technischen Sicht des Architekturmodells durch Angabe

des Referenzprodukts und durch Verweis auf die entsprechende technische Spezifikation abgebildet. Prinzipvariationen spiegeln sich in der logischen und funktionalen Sicht des Architekturmodells wider. Beide Variationsarten können, nachdem sie identifiziert sind, zusätzlich durch entsprechende Farbgebung kenntlich gemacht werden und darauf aufbauend auch Variationsanteile berechnet werden. Wäschle et al. zeigen diese Option am Beispiel eines Aktuators auf, siehe Kapitel 2.4.4. (Wäschle et al., 2020)

Forschungsfrage 2 kann damit zusammengefasst wie folgt beantwortet werden: Die Strukturierung des Architekturmodells nach dem SPES-Framework kann sowohl die Anbindung an das Berechnungsmodell als auch die Berücksichtigung von Variationsanteilen in beiden Modellen unterstützen.

5.2.2 Aufbau des Berechnungsmodells unter Nutzung des Architekturmodells

Der Aufbau eines Berechnungsmodells entspricht einer virtuellen Systemintegration (siehe auch Kapitel 2.3.4 und Kapitel 5.2.3). Im Folgenden wird darauf eingegangen, wie im Rahmen des zu entwickelnden Ansatzes das Architekturmodell genutzt wird um das Berechnungsmodell zu erstellen und zu parametrieren.

In Abbildung 5.2 sind die zentralen Elemente der einzelnen Sichten des Architekturmodells in SysML dargestellt. Im zu entwickelnden Ansatz bildet das Berechnungsmodell die Elemente und Zusammenhänge ab, die in der funktionalen und logischen Sicht des Architekturmodells enthalten sind. Auch wenn Berechnungs- und Simulationsmodelle in unterschiedlichen Programmen mit unterschiedlichen Zwecken aufgebaut werden, sollte die Modellbildung anhand des im Architekturmodell verankerten Systemverständnis erfolgen. So kann das Architekturmodell in seiner Rückgrat-Funktion die Konsistenz zwischen den quantitativen Analysen unterstützen und damit die Übertragbarkeit der Ergebnisse auf das Gesamtsystem verbessern. Die Art und Weise der Modellierung hängt vom Zweck der quantitativen Analyse und vom verwendeten Tool ab. Beispielsweise werden in einer FEM-Simulation die Subsysteme durch 3D-Modelle abgebildet, während in einer Simulation mit Matlab/Simulink mathematische Blöcke die Subsysteme darstellen. Daher wird im Folgenden auf den Zusammenhang der im Berechnungsmodell zu wählenden Subsystemgrenzen, Schnittstellen, Funktionen etc. mit dem Architekturmodell eingegangen, nicht jedoch auf die konkrete Umsetzung in einem Modell. Diese erfolgt am Beispiel des elektrischen Antriebssystems in Kapitel 6.2.

im Berechnungsmodell genutzt werden, sofern sie für die Ermittlung des beobachteten Systemverhaltens nicht relevant sind. Durch die Konsistenz wird eine eindeutige Verknüpfung der aus der Simulation resultierenden Anforderungen mit dem Architekturmodell wird dadurch vereinfacht. Des Weiteren ermöglicht das im Falle einer werkzeugseitigen Anbindung die automatische Übernahme von Systemelementen und Schnittstellengrößen in das Berechnungsmodell.

Die in der funktionalen Sicht des Architekturmodells enthaltenen Systemzustände werden im Berechnungsmodell analog zum Architekturmodell benannt und in der Beschreibung des Zustandsverlaufs genutzt. Die Möglichkeiten des Zustandswechsels sollten dabei konsistent zu den Zustandsdiagrammen des Architekturmodells umgesetzt sein. Die im Architekturmodell identifizierten Operationen entsprechen den im quantitativen Modell abzubildenden Funktionen der einzelnen Subsysteme. Dabei ist die Zuordnung von Funktion und Subsystem entsprechend den Aktivitätsdiagrammen im Architekturmodell umzusetzen. Die Funktionen selbst werden je nach Zweck des Berechnungsmodells und verwendetem Tool durch die entsprechenden Berechnungen dargestellt. Dabei sollten die Ein- und Ausgangsgrößen der Berechnungen den für die Funktion relevanten Schnittstellengrößen der logischen Sicht im Architekturmodell entsprechen.

Nachdem die Modellbildung des Berechnungsmodells anhand funktionaler und logischer Sicht erfolgt ist, stellt die technische Sicht des Architekturmodells die konkrete Parametrisierung des Berechnungsmodells dar. Damit sind die in der technischen Sicht enthaltenen Werte zentrale Grundlage für die Entwicklung quantifizierter Anforderungen an die Subsysteme. Je nach Reifegrad des Systems und des Referenzsystems (z. B. vorige Produkt- oder Entwicklungsgeneration eines (Sub-) Systems) können diese Werte von verifizierten technischen Eigenschaften bis zu vagen Annahmen oder Lösungsideen aufgrund erster grober Abschätzungen reichen. Entsprechend des Analyse-Synthese-Zyklus werden sie während der Entwicklung fortlaufend konkretisiert.

Die Anforderungssicht des Architekturmodells enthält die kundenrelevanten Anwendungsfälle inklusive zugehöriger Nutzer, die als Eingangsgrößen der Simulation dienen. Also die Belastung auf der obersten im Berechnungsmodell abgebildeten Systemebene. Ebenso enthält die Anforderungssicht die Randbedingungen, unter denen das Systemverhalten analysiert werden muss. Des Weiteren sind die Kundenwünsche hier dokumentiert. Die Kundenwünsche werden zur Analyse der Ergebnisse und damit zur Bewertung der jeweils abgebildeten Systemausprägung herangezogen. Neben den Systemeigenschaften in der technischen Sicht sind somit auch die Kundenwünsche in der Anforderungssicht ein zentrales Element zur Entwicklung quantitativer Anforderungen. Hierbei gilt, dass auf rein virtueller Ebene

nur objektiviert – d.h. in diesem Fall quantifizierte - Kundenwünsche bewertet werden können.

Zusammenfassend kann zur Beantwortung der **Forschungsfrage 3** also festgestellt werden, dass die funktionale und logische Struktur des Architekturmodells zur Modellbildung des Berechnungsmodells genutzt wird. Die dort abgebildeten Subsysteme, ihre Systemgrenzen und Schnittstellen sowie ihre funktionalen Zusammenhänge werden – ggf. zweckorientiert verkürzt – im Berechnungsmodell abgebildet. Um das Systemverhalten ermitteln zu können, muss das Berechnungsmodell parametrierbar werden. Die Parametrierung erfolgt anhand der technischen Sicht des Architekturmodells, in der die Ausprägungen der modellierten Subsysteme definiert sind. Hierbei kommt eine Mischung aus realen Parametern (beispielsweise bei Einsatz vorangegangener Produkt- oder Entwicklungsgenerationen eines Subsystems) und angenommenen Parametern (z. B. aus rechnerischen Abschätzungen von Gestaltungsideen für ein Subsystem) zum Einsatz. So wird das Berechnungsmodell quantifiziert und ermöglicht die Analyse des resultierenden Systemverhaltens auf allen abgebildeten Systemebenen vor dem Hintergrund objektivierter Kundenwünsche.

5.2.3 Beschreibung der Nutzung eines Architekturmodells in der Produktentstehung

Das in Kapitel 2.1.1 vorgestellte erweiterte ZHO-Modell beschreibt den Prozess der Produktentstehung als Abfolge von Analyse- und Syntheseschritten durch die das Handlungssystem aus verfügbarer Information Ziele ableitet und diese in konkrete Objekte überführt. (Lohmeyer, 2013) Ein Architekturmodell nach der in Kapitel 5.2.1 vorgestellten Struktur verbindet Ziele und Anforderungen mit der funktionalen, logischen und technischen Beschreibung eines Produkts. Die Erstellung und aktive Nutzung eines Architekturmodells als Teil des Produktentstehungsprozesses wird im Folgenden durch das erweiterte ZHO-Modell beschrieben. Es stellt sich dabei die Frage, wo sich die Synthese- und Analyseschritte des erweiterten ZHO-Modells in der Erstellung und Nutzung eines Architekturmodells wiederfinden. Der zu entwickelnde Ansatz verfolgt das Ziel, quantitative Anforderungen auf unterschiedlichen Systemebenen mit Hilfe eines Berechnungsmodells durchgängig und konsistent zu entwickeln. Anforderungen werden dabei dadurch abgeleitet, dass das Verhalten des SiD bei unterschiedlichen Anwendungsfällen für mehrere Systemvarianten virtuell berechnet, analysiert und bewertet wird. Darauf aufbauend können Varianten-Entscheidungen getroffen und für unterschiedliche Anforderungen abgeleitet werden. Um die Systematik der Anforderungsentwicklung im Berechnungsmodell mit den Inhalten des Architekturmodells in Verbindung zu bringen, werden in den

Synthese- und Analyseschritten zusätzlich die Validierungs- und Kurationsanteile nach Albers et al. (Albert Albers et al., 2013) berücksichtigt.

Nach Albers ist Validierung die einzige Aktivität im Produktentstehungsprozess, durch die Wissen generiert wird. (Albert Albers, 2010) Die Entwicklung quantitativer Anforderungen funktioniert, indem der Zusammenhang zwischen Eigenschaften und Verhalten von (Sub-) Systemen verstanden und bewertet wird und infolgedessen quantifizierte Zielwerte und resultierende Beanspruchungen abgeleitet werden. Daraus lässt sich ableiten, dass zur Anforderungsentwicklung Validierungsaktivitäten notwendig sind. Das wird auch bei der in Kapitel 5.2.4 vorgestellten Integration von Validierung in das erweiterte ZHO-Modell deutlich (vgl. Abbildung 2.8). Dabei basiert die Konkretisierung und Erweiterung des Zielsystems auf der Synthese der Wissensbasis des Handlungssystems. Die Wissensbasis selbst wird durch die Analyse des Objektsystems erweitert. Validierung ist somit die Abfolge von Objektsystemanalyse und Zielsystemsynthese. Die Analyse des Zielsystems wiederum zur Identifizierung des Lösungsraums und die darauffolgende Synthese des Objektsystems ist nach Albers die Kreation. (Albert Albers et al., 2013)

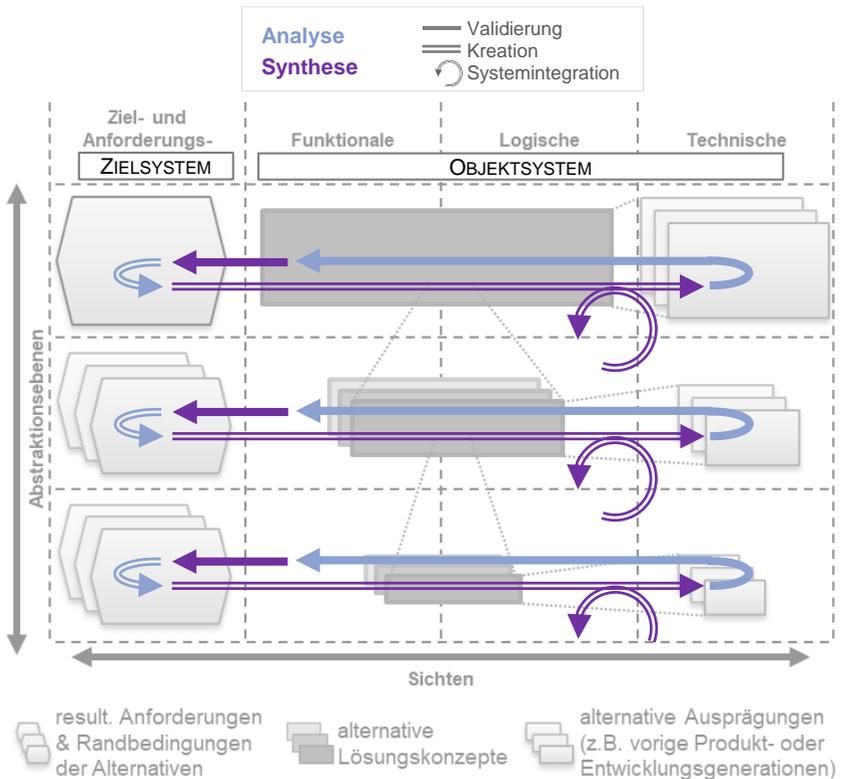


Abbildung 5.3: Beschreibung der Erstellung und Nutzung des Architekturmodells durch das erweiterte ZHO-Modell unter Berücksichtigung von Validierung und Kreation in den Analyse- und Syntheseschritten

In Abbildung 5.3 ist die Folge von Validierung und Kreation auf die gewählte Struktur des Architekturmodells übertragen. Die Anforderungssicht wird dabei als Repräsentation des Zielsystems verstanden. Wie in Kapitel 5.2.1 dargelegt, enthält der die Anforderungssicht neben den Anforderungen auch den Kontext des zu entwickelnden Systems und die Anwendungsfälle und verknüpft diese miteinander. „Ein Zielsystem beinhaltet alle expliziten Ziele eines zu entwickelnden Produktes, einschließlich derer Abhängigkeiten und Randbedingungen, innerhalb eines definierten Interessenbereichs (d.h. innerhalb eines System-of-Interest) zu einem bestimmten Zeitpunkt.“ (Lohmeyer, 2013, S. 85) Diese Definition ist durch die Ziel- und Anforderungssicht repräsentiert.

Die funktionale Sicht, die logische Sicht und die technische Sicht repräsentieren, wie – d.h. durch welche Funktionen und Elemente in welchen Ausprägungen – das zu entwickelnde System diese Ziele (zukünftig) erfüllt. *„Objektsysteme sind Artefakte, also materielle und immaterielle Ergebnisse des Handlungssystems. Der Zweck eines Objektsystems ist im korrespondierenden Zielsystem beschrieben. In der Produktentstehung muss zu jedem Objekt- ein dazugehöriges Zielsystem existieren. Objektsysteme erfüllen in der Produktentstehung drei Bedeutungen: Sie sind Ressourcen, Erkenntnisobjekte und Ergebnisse des Handlungssystems.“* (Meboldt, 2008, S. 175) Die Inhalte der funktionalen, logischen und technischen Sicht können demzufolge dem Objektsystem zugeordnet werden. Das Handlungssystem ist im Architekturmodell nicht abgebildet, sondern erstellt und nutzt das Architekturmodell durch die dargestellten Analyse- und Syntheseschritte.

Das erweiterte ZHO-Modell verdeutlicht, dass es keinen definierten „Anfang“ und auch keinen definierten „Ausstieg“ aus dem Wechsel zwischen Synthese und Analyse gibt, sondern dass Produktentwicklung vielmehr eine kontinuierliche und iterative Analyse-Synthese-Abfolge ist. Dies spiegelt sich in Abbildung 5.3 innerhalb der Systemebenen wider. In der Kreation führt die Analyse des Zielsystems zur Synthese eines Systementwurfs, der – virtuell, physisch oder gemischt-virtuell-physisch umgesetzt – im Rahmen der Validierung analysiert wird um darauf aufbauend das Zielsystem in einem Syntheseschritt weiter zu konkretisieren. Der Systementwurf erfolgt dabei jedoch nicht ausschließlich auf einer Systemebene, sondern schließt darunterliegende Systemebenen mit ein. Diese werden auch in der Umsetzung des Systems zur späteren Analyse mitberücksichtigt – verdeutlicht wird das in Abbildung 5.3 durch die Verbindung der Systemebenen mit Schritten der Systemintegration. Insbesondere bei noch geringem Reifegrad des Systems erfolgt dabei die Systemintegration oft rein virtuell in Form von Berechnungs- oder Simulationsmodellen. Die Ergebnisse werden dann im Rahmen der Validierung analysiert und vor dem Hintergrund des Zielsystems bewertet. Die in der Analyse gewonnenen Erkenntnisse werden als Anforderungen und Randbedingungen wieder in das Zielsystem synthetisiert.

Die Erzeugung analysierbarer Simulationsergebnisse ist nur durch Festlegung quantifizierter technischer Merkmale des Systems möglich. Die Auswirkung dieser Eigenschaften auf das Systemverhalten kann dann objektiv bewertet werden. Bei positiver Bewertung des Systemverhaltens kann eine zuvor getroffene Annahme als quantifizierte Anforderung in das Zielsystem überführt werden. Des Weiteren können aus dem Systemverhalten resultierende Belastungen für die Subsysteme als quantifizierte Anforderungen in das Zielsystem einfließen.

Abbildung 5.3 zeigt im Sinne der **Forschungsfrage 4**, wie sich die Elemente und Schritte des erweiterten ZHO-Modells in der Systemarchitektur selbst sowie während ihrer Erstellung und Nutzung widerspiegeln. Die Modellierung der Funktionen, Strukturen und Ausprägungen entsprechen der Kreation. Das lässt sich direkt auf die Erstellung eines Berechnungsmodells übertragen. Die Analyse und Bewertung des Zusammenwirkens der Systemelemente in verschiedenen Ausprägungen entspricht der Validierung. Diese ist nicht direkt im Architekturmodell möglich, sondern geschieht im Berechnungsmodell. Durch Dokumentation der resultierenden Anforderungen wird auch die Anforderungssicht im Architekturmodell konkretisiert, der Synthese-Anteil der Validierung kann also als Übergang hin zur Anforderungssicht beschrieben werden.

5.2.4 Zusammenspiel des Ansatzes mit der Verwendung des IPEK-XiL-Ansatzes und -Frameworks

Wie in Kapitel 5.2.3 dargelegt, sind Validierungsaktivitäten maßgeblich für die Entwicklung quantitativer Anforderungen, da sie Aufschluss über das Verhalten des beobachteten Systems (Sul – System under Investigation) im Kontext der umgebenden Systeme geben. Dies wird auch im „Pull-Prinzip der Validierung“ deutlich (Albert Albers, Matros et al., 2015). Dieses versteht die Validierung als kontinuierliche und prozessbegleitende Aktivität während des gesamten Produktentstehungsprozesses. Validierungsumgebungen sind hierbei vielfältig und reichen entsprechend dem IPEK-XiL-Ansatz von rein virtueller Darstellung des Systems in Simulationen über verschiedene gemischt physisch-virtuelle Validierungsumgebungen bis hin zu Versuchen mit dem vollständig physikalisch integrierten System in der realen Umwelt. Ausgehend von diesem Verständnis kann die Simulation eines Systems zur Ableitung von Anforderungen aus der Analyse und Bewertung des Systemverhaltens gegenüber der Kundenerwartung als Validierungsaktivität in einem rein virtuellen Validierungssystem verstanden werden. Im Folgenden wird aufgezeigt, wie im hier entwickelten Ansatz der IPEK-XiL-Ansatz und das zugehörige Framework berücksichtigt sind und wo die Schnittstellen zwischen den Ansätzen liegen. Abbildung 5.4 stellt den Übertrag der Elemente des IPEK-XiL-Ansatzes in eine nach dem SPES-Framework strukturierte Systemarchitektur dar.

Der IPEK-XiL-Ansatz setzt das zu entwickelnde System (System in Development – SiD) in der jeweiligen Validierungsumgebung in Zusammenhang mit den umgebenden Systemen (Connected Systems), die mit dem SiD in Kontakt stehen und es beeinflussen. Im Fall der Fahrzeugentwicklung beispielsweise Fahrer, Rest-Fahrzeug und Umwelt. Diese bilden sozusagen den relevanten Kontext des SiD in der

zu analysierenden Situation. Die zu analysierende Situation selbst wird durch Fahrmanöver bzw. Testfälle abgebildet, mit denen das Gesamte System (SiD und Connected Systems) während der Validierung beaufschlagt wird.

Im Verständnis der Modellierungssprache SysML können die Fahrmanöver und Testfälle als use cases des SiD interpretiert werden. Anwendungsfälle und relevanter Kontext (also Fahrer, Rest-Fahrzeug und Umwelt) des SiD werden in Abbildung 5.4 gemeinsam mit den Stakeholder-Anforderungen an das SiD in der Ziel- und Anforderungssicht verortet (grau ①). Die Anwendungsfälle werden dabei in Use-Case-Diagrammen dargestellt, während der Kontext des SiD mit Hilfe von Block-Definitions-Diagrammen und der Zusammenhang zwischen Kontext und SiD mit internen Blockdiagrammen visualisiert wird. Sind die Stakeholder-Anforderungen und Randbedingungen für das SiD in einem Anforderungsmanagementsystem dokumentiert, können sie von dort aus als SysML-Elemente in den Diagrammen verlinkt werden. Alternativ werden Anforderungselemente neu erstellt. Das SiD selbst wird durch die funktionale, logische und technische Sicht auf erster Ebene der Systemarchitektur repräsentiert (s. Abbildung 5.4 in blau ②), wobei in der funktionalen Sicht die Hauptfunktionen des SiD in Aktivitätsdiagrammen dargestellt sind. Die logische Sicht auf das SiD beinhaltet das SiD als einzelnen Block und im Sinne einer „white box“ auch ein internes Blockdiagramm, das die Subsysteme des SiD und deren Schnittstellen darstellt. Diese Subsysteme selbst sind wiederum die „black boxes“ in Ebene 2. Das interne Blockdiagramm des SiD ist sozusagen der „Weg“ von Ebene 1 zu Ebene 2. In der technischen Sicht ist die konkrete Ausprägung des SiD in Form von Blockdefinitionsdiagrammen mit Instanzen der jeweiligen Blöcke dargestellt. Die Detailtiefe der technischen Sicht hängt vom Entwicklungszeitpunkt und dem damit einhergehenden Reifegrad des SiD zusammen und kann von Annahmen zentraler technischer Eigenschaften bis hin zu detaillierten Parametern inkl. Toleranzen reichen. Im Sinne eines Rückgrats des Systemwissens, könnten Detailinformationen dabei auch über Verlinkung auf z. B. Dokumente zur Verfügung gestellt werden.

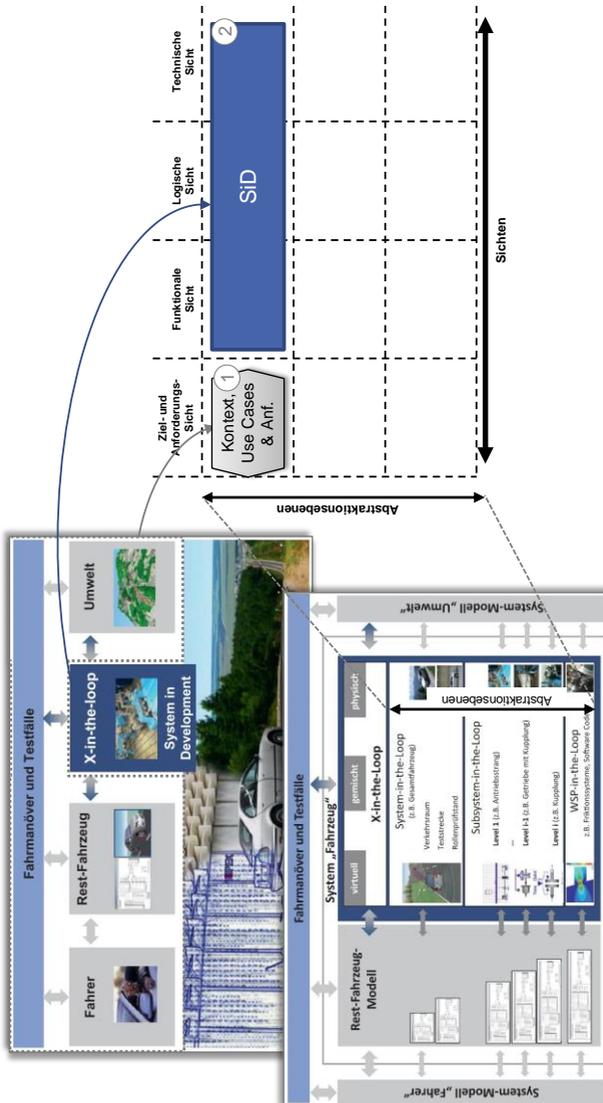


Abbildung 5.4: Übertragung der Elemente des IPEK-XiL-Ansatz und -Framework in den zu entwickelnden Ansatz.

In Abbildung 5.4 wird zudem deutlich, dass sich die Abstraktionsebenen des SPES-Frameworks und des IPEK-XiL-Framework übereinbringen lassen um das SiD in Subsysteme und diese wiederum in weitere Subsysteme bis hin zu einzelnen Komponenten oder einem Wirkflächenpaar zu gliedern. Der IPEK-XiL-Ansatz unterscheidet hierfür neben dem Begriff des SiD noch den Begriff des „System under Investigation“ – Sul als (Sub-) System, das im Fokus der aktuellen Validierungsaktivität steht (Abbildung 2.12). Insbesondere dann, wenn diese nicht der Eigenschaftsabsicherung, sondern der Gewinnung von Wissen über das Subsystem dient. „Das Sul kann demnach einen Teil des SiD darstellen oder das gesamte SiD sein.“ (Albert Albers, Behrendt et al., 2016)

Die Unterscheidung in Sul und SiD sowie das Verständnis zu deren Zusammenhang werden in der vorliegenden Arbeit genutzt, um die Auswertung der Simulationsergebnisse hinsichtlich durchgängiger und konsistenter Anforderungen in allen Systemebenen zu unterstützen. In Abbildung 5.5 ist der Zusammenhang zwischen SiD und Sul im SPES-Framework am Beispiel der zweiten Systemebene dargestellt. Das Architekturmodell besteht auf der zweiten Systemebene aus mehreren Subsystemen des SiD, die zur Erfüllung der Hauptfunktionen zusammenwirken (in Abbildung 5.5 im blauen Rahmen ③). Jedes einzelne dieser Subsysteme kann nun als Sul betrachtet werden. Dabei fungieren die weiteren Subsysteme des SiD als Restsystem, das zur Analyse des Sul abgebildet werden muss und aus dem zeitgleich Randbedingungen für das Sul resultieren.

Forschungsfrage 5 lässt sich damit wie folgt beantworten: Durch die Konsistenz der Elemente des IPEK-XiL-Ansatzes und –Frameworks mit den verschiedenen Sichten und Elementen des Architekturmodells ermöglicht der zu entwickelnde Ansatz dem Entwickler weiterhin den IPEK-XiL-Ansatz für die Validierungsaktivität Simulation zu nutzen und gleichzeitig systematisch im Architekturmodell das SiD und die daraus folgenden Sul auf tieferen Systemebenen zu identifizieren und zu beschreiben. Die Systemarchitektur kann damit zur gezielten Planung von Sul und zur Beschreibung der relevanten Schnittstellen und Wechselwirkungen genutzt werden – und somit zur Modellbildung des Berechnungsmodells.

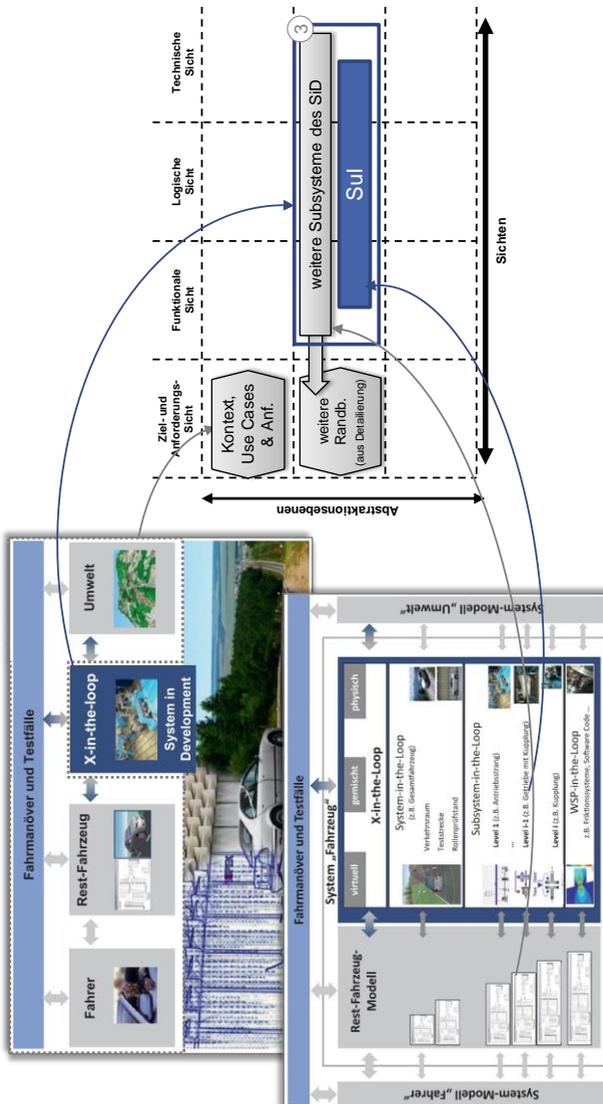


Abbildung 5.5: Zusammenhang zwischen IPEK-XiL-Framework und Ansatz in tieferen Systemebenen.

5.2.5 Berechnung und Bewertung alternativer Systemvarianten

Kapitel 5.2.1 zeigt auf, dass es in der funktionalen, der logischen und der technischen Sicht des Architekturmodells jeweils alternative Lösungen für das Gesamtsystem oder einzelne Subsysteme gibt. Alternative Lösungen für einzelne Subsysteme können zudem unterschiedlich miteinander kombiniert werden. So ergibt sich ein großer potenzieller Lösungsraum aus alternativen Systemvarianten für das zu entwickelnde Produkt. Für eine effiziente Entwicklung eines kundenwerten Produkts ist es notwendig, diese alternativen Systemvarianten bereits bei noch geringem Reifegrad des Systems hinsichtlich ihres Potenzials zur Erfüllung der Kundenanforderungen und bezüglich der Umsetzbarkeit und dem zu erwartenden Entwicklungsaufwand zu bewerten und zu vergleichen.

Alternative Systemvarianten können in zwei Dimensionen charakterisiert werden, die in Abbildung 5.6 dargestellt sind. Zum einen gibt es Alternativen bezüglich der Funktionsweise, der beteiligten Subsysteme und deren Interaktionen – hier als alternative Lösungskonzepte in der funktionalen und logischen Sicht des Architekturmodells beschrieben. Dies kann, wie in Kapitel 5.2.1 dargelegt, im Sinne des Modells der SGE als Prinzipvariation verstanden werden. (Albert Albers, Bursac & Wintergerst, 2015; Albert Albers et al., 2020) Zum anderen kann jedes einzelne Lösungskonzept, das in der funktionalen und logischen Sicht beschrieben wird, alternative technische Ausprägungen haben (z.B. geometrische Abmessungen, Übersetzungsverhältnisse, Widerstandswerte, Reglerparameter...). Bezogen auf das Referenzsystem kann eine solche Varianz im Sinne des Modells der SGE als Ausprägungsvariation verstanden werden. (Albert Albers, Bursac & Wintergerst, 2015; Albert Albers et al., 2020)

Die Abbildung der alternativen Systemvarianten im Berechnungsmodell hängt im Detail immer von der Art des Modells und dem verwendeten Tool ab. Die Auswahl dieser beiden Aspekte ist wiederum abhängig davon, welches Systemverhalten oder welche Systemeigenschaft analysiert werden soll. Dennoch bietet das Architekturmodell in SysML entsprechend Kapitel 2.4.2 die Basis, für die im Berechnungsmodell abzubildenden Systemfunktionen, Subsysteme, deren Schnittstellen und Interaktionen. Die Parametrierung des Berechnungsmodells erfolgt dann entsprechend der in der technischen Sicht des Architekturmodells dokumentierten Ausprägungsvarianten.

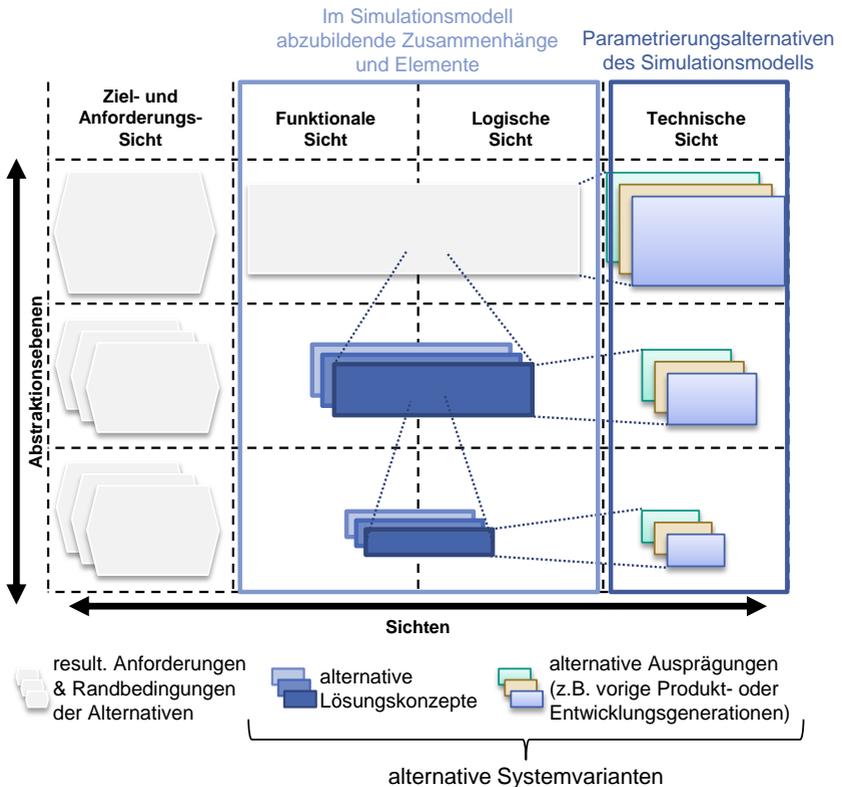


Abbildung 5.6: Dimensionen alternativer Systemvarianten im Architekturmodell.

Durch die eindeutige Wahl von Schnittstellen und Subsystemgrenzen können im Berechnungsmodell einzelne Subsysteme leicht ausgetauscht werden, sodass nicht für jede Systemvariante mit Prinzipvariation das gesamte System für die Simulation neu modelliert werden muss. So wird auch ermöglicht, die Simulation mehrerer Systemvarianten mit Prinzipvariation durch Komponentenaustausch als eine Art Parametervariation durchzuführen und ggf. zu automatisieren. Durch Wahl der gleichen Anwendungsfälle und Belastungszyklen des Systems (entsprechend der Ziel- und Anforderungssicht auf oberster Systemebene) können nun mehrere unterschiedliche Systemvarianten analysiert und verglichen werden. Die generierten Informatio-

nen sind für diese Analyse in geeigneter Form darzustellen um in der Ergebnisinterpretation unterschiedliche Aspekte beleuchten zu können. Durch die bewusste Abbildung mehrerer Systemebenen und Beaufschlagung mit mehreren realistischen Belastungszyklen kann die Analyse dabei nicht nur hinsichtlich der Erfüllung der Kundenwünsche erfolgen, sondern auch hinsichtlich resultierender Belastungsarten und –zyklen von Subsystemen. So kann neben dem voraussichtlichen Wert für den Kunden auch eine Indikation für die technische Machbarkeit und den zu erwartenden Entwicklungsaufwand getroffen werden. Dabei ist die Zuordnung der Analyseergebnisse zu den logischen Elementen des jeweiligen Architekturentwurfs durch die gleiche Struktur von Architektur- und Berechnungsmodell vereinfacht.

Die Ausführungen beantworten damit **Forschungsfrage 6**. Zusammenfassend kann die Modellierung einer Systemarchitektur mit SysML also die Strukturierung des Berechnungsmodells unterstützen, um eine bessere Austauschbarkeit einzelner Subsysteme im Berechnungsmodell zu erreichen. So können durch Parametervariation und Subsystemvariation alternative Systemvarianten einfacher und schneller berechnet und vergleichend dargestellt werden. Konsistente Subsystemgrenzen und Schnittstellen im Architekturmodell und im Berechnungsmodell vereinfachen die Analyse und Interpretation der Simulationsergebnisse hinsichtlich Anforderungen und Randbedingungen. Die Zuordnung der konkretisierten Ziele und Anforderungen zu den Subsystemen und deren Funktionen ist infolge dessen ebenfalls erleichtert. So kann beim Aufbau einer Nachverfolgbarkeitskette zwischen Anforderungen und Architektur zusätzlich auch die inhaltliche Nachvollziehbarkeit der Anforderungen und Randbedingungen erhöht werden. Erhöhte inhaltliche Nachvollziehbarkeit leistet wiederum einen Beitrag zur Konsistenz der Anforderungen untereinander und zur Richtigkeit der Anforderungen im angenommenen Kontext.

5.2.6 Dokumentation von Berechnungsergebnissen und Entscheidungen in der Systemarchitektur

Zur Nutzung des Architekturmodells als Rückgrat des Entwicklungsprozesses in Form der zentralen Informationsquelle ist es notwendig, nicht nur Informationen daraus zu extrahieren, sondern die aus den Aktivitäten entstehenden Informationen auch im Architekturmodell zu verankern. Durch die Verlinkungen innerhalb des Architekturmodells werden Zusammenhänge zwischen verschiedenen Elementen des Architekturmodells dargestellt. Das ermöglicht die Rückverfolgbarkeit von Abhängigkeitsketten über die Abstraktionsebenen und Sichten der Systemarchitektur hinweg, z. B. zwischen Anforderung und Subsystem oder auch zwischen mehreren Anforderungen. Eine reine Rückverfolgbarkeit ist jedoch für eine inhaltliche Nachvollziehbarkeit der Anforderungen nicht ausreichend.

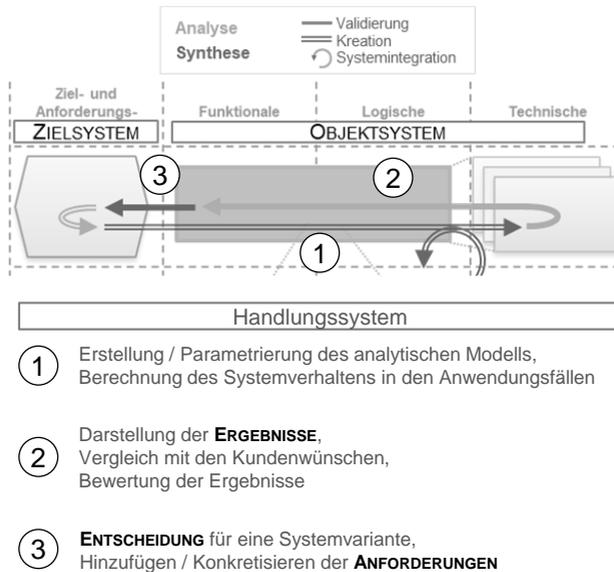


Abbildung 5.7: Schritte und Ausgangsgrößen des Handlungssystems bei der Ableitung von Anforderungen.

Auf Basis der Analysen in Kapitel 5.2.3 werden die Ausgangsgrößen des Handlungssystems bei der Entwicklung und Konkretisierung von Anforderungen detaillierter betrachtet. Abbildung 5.7 stellt die Analyse- und Synthese-Schritte von Validierung und Kreation vereinfacht auf der obersten Systemebene dar. Im Rahmen der Objektsystemanalyse ① erfolgt die initiale Erstellung des Berechnungsmodells bzw. die Parametrierung des Modells entsprechend unterschiedlichen Systemvarianten und die Berechnung des Systemverhaltens in den unterschiedlichen Anwendungsfällen. Die Ergebnisse für unterschiedliche Systemvarianten werden im Rahmen der Objektsystemanalyse ② vergleichend dargestellt und vor dem Hintergrund der Kundenwünsche bewertet. Auf Basis dieser Bewertung erfolgt die Synthese des Zielsystems ③ indem zunächst die Entscheidung für eine Systemvariante getroffen wird und darauf aufbauend die resultierenden Anforderungen dem Zielsystem hinzugefügt, oder im Zielsystem entsprechend geändert werden. Um einem Mitglied des Entwicklungsteams zu ermöglichen, die Herkunft von Anforderungen inhaltlich nachzuvollziehen, muss die getroffene Entwicklungsentscheidung und die dieser zugrundeliegenden Datenbasis zugänglich gemacht werden. SysML ermöglicht die Dokumentation von Begründungen innerhalb des Modells durch das Einfügen eines

Kommentars. Ein Kommentar ist eine textuelle Anmerkung, die an ein beliebiges Modellelement geheftet werden kann. (Weilkiens, 2014) Es bietet sich also an, dieses Modellelement für die Dokumentation der Entwicklungsentscheidung in der Systemarchitektur zu nutzen.

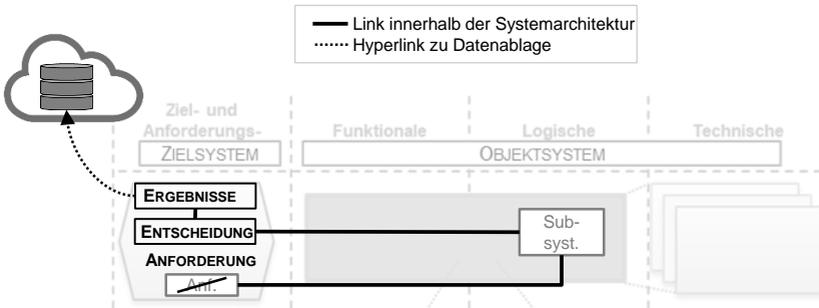


Abbildung 5.8: Verlinkungskonzept von Ergebnissen und Entscheidungen mit und innerhalb der Systemarchitektur.

Da die Ergebnisse des Berechnungsmodells oft Informationen zu Subsystemen auf unterschiedlichen Abstraktionsebenen enthalten können und gleichzeitig mehrere Systemvarianten verglichen werden, ist nicht immer eine eins-zu-eins-Beziehung zwischen Ergebnisdaten und Entscheidung gegeben. Aus einem Datensatz können durch Analyse und Interpretation der Ergebnisse hinsichtlich unterschiedlicher Aspekte mehrere Entscheidungen zu unterschiedlichen Subsystemen abgeleitet werden. Das legt die Trennung der Repräsentation von Ergebnissen und Entscheidungen in der Systemarchitektur nahe. Die Ergebnisdaten selbst sowie ihre grafische Aufbereitung liegen üblicherweise auf Datenbanken ab. Für eine inhaltliche Nachvollziehbarkeit sollten sie jedoch aus dem Systemmodell heraus zugänglich sein. Diese Überlegungen führen zu dem in Abbildung 5.8 dargestellten Verlinkungskonzept für Ergebnisse und Entscheidungen. Subsysteme sind mit den für sie gültigen Anforderungen verlinkt. Entscheidungen bezüglich der Systemvariante wirken sich direkt auf die verwendeten Subsysteme aus. Die Entscheidungen werden somit auf die Subsysteme verlinkt. Basis für die Entscheidungen sind die Ergebnisse, diese werden ebenfalls als Kommentar-Elemente in das Architekturmodell eingefügt und mit allen daraus abgeleiteten Entscheidungen verknüpft. Die Ergebnis-Elemente beinhalten Hyperlinks, die direkt auf die Ergebnisdaten und deren grafische Darstellung führen. So wird eine Verlinkungskette zwischen den Elementen Ergebnisse, Entscheidungen, Subsysteme und Anforderungen geschaffen. Wie bereits erwähnt beinhalten die Ergebnisse oft Informationen zu mehreren Systemebenen und auch

die Entscheidungen können sich auf Subsysteme mehrerer Systemebenen auswirken. Vor diesem Hintergrund und mit Blick auf die Definition eines Zielsystems (siehe Kapitel 2.2.1 und Kapitel 5.2.3) werden die Ergebnis- und Entscheidungselemente in der Anforderungssicht auf der obersten Ebene integriert.

Damit ist **Forschungsfrage 7** beantwortet. Ergebnisse und Entscheidungen werden in Form des SysML-Elements Kommentar in die Systemarchitektur integriert. Die Ergebniselemente machen die Daten des Berechnungsmodells über Hyperlinks aus der Systemarchitektur heraus zugänglich. Alle Entscheidungen, die auf Basis der jeweiligen Ergebnisse getroffen wurden, werden mit diesen Ergebnissen und den betroffenen Subsystemen in der logischen Sicht verlinkt. Die neuen oder veränderten Anforderungen sind wiederum mit diesen Subsystemen verlinkt.

5.3 Zusammenfassung des Ansatzes

In Kapitel 5.2 werden die einzelnen Aspekte eines Ansatzes zur Entwicklung konsistenter und quantitativer Anforderungen durch Nutzung eines Architekturmodells beleuchtet. Um die organisatorische und individuelle Akzeptanz eines solchen Ansatzes zu berücksichtigen, werden dabei auch die Zusammenhänge mit etablierten Entwicklungsmethodischen Ansätzen und mit quantitativen Modellen berücksichtigt. Im Folgenden sollen nun die daraus abgeleiteten konkreten Schritte zur Entwicklung konsistenter, nachvollziehbarer und quantitativer Anforderungen kurz beschrieben werden. Abbildung 5.9 gibt einen Überblick über die Schritte und deren Inhalte. Zugunsten der Verständlichkeit werden die Schritte in der vorliegenden Arbeit sequenziell beschrieben. Auch ist eine System- und Kontextanalyse als Startpunkt eines Prozesses geeignet und wird von vielen entwicklungsmethodischen Ansätzen empfohlen. (Pohl et al., 2012; Weilkiens, 2014) Dennoch können die aufgezeigten schritte fraktal, iterativ und in einer an die aktuelle Projektsituation angepassten Reihenfolge durchgeführt werden.

Die Analyse des Systems und des Systemkontextes führt zur Beschreibung von Funktionen auf oberster Systemebene und von Schnittstellen und Wechselwirkungen des Systems mit umgebenden Systemen und Nutzern. Aus diesen Erkenntnissen werden sowohl typische Anwendungsfälle für das System abgeleitet als auch Kundenerwartungen und –wünsche an das System. Dies bildet die Grundlage für die zu simulierenden Systembelastungen und für die Bewertung der alternativen Systemvarianten hinsichtlich ihres Kundenwerts. Zur Durchführung der Analyse bietet sich die Anwendung etablierter Methoden der Anforderungsermittlung wie z. B. Kundenbefragungen, Wettbewerbsanalyse, Feldbeobachtungen, etc. an, Kapitel 2.2.3 adressiert übliche Methoden zur Anforderungsermittlung.

Die Modellierung der Systemarchitektur folgt entsprechend Kapitel 5.2.1 dem SPES-Ansatz, um die Systemebenen eindeutig voneinander abzugrenzen. Unter Verwendung der in Abbildung 5.2 gezeigten SysML-Elemente und –Diagramme wird sowohl das Systemverhalten auf den verschiedenen Abstraktionsebenen als auch die logische Struktur innerhalb dieser Ebenen beschrieben. Dabei werden wo möglich die Referenzsysteme (vorangegangene Produkt oder Entwicklungsgenerationen) der logischen Elemente identifiziert. In dem Zuge können auch die jeweiligen Variationsarten entsprechend des Modells der SGE identifiziert werden um dadurch eine Indikation für den zu erwartenden Entwicklungsaufwand und das Entwicklungsrisiko der jeweiligen Architekturvariante zu erhalten. (Wäschle et al., 2020) Die Schnittstellen zwischen den logischen Elementen jeder Systemebene werden durch die Beschreibung der Stoff-, Energie- oder Materialflüsse charakterisiert.

Die virtuelle Systemintegration im Sinne des IPEK-XiL-Ansatzes erfolgt bei der Erstellung und Parametrierung des quantitativen Modells. Analog zur funktionalen und logischen Sicht des Architekturmodells wird dabei das SiD inklusive der Sul auf tieferen Systemebenen modelliert (vgl. Kapitel 5.2.4). Alternative Systemelemente (z. B. auf Basis unterschiedlicher Produkt- oder Entwicklungsgenerationen) werden so modelliert, dass sie im quantitativen Modell einfach ausgetauscht werden können (vgl. Kapitel 5.2.2 & 5.2.5). Dies ermöglicht eine effiziente Durchführung der späteren Simulation, da die Systemelemente ähnlich wie einzelne Parameter durchvariiert werden können.

Die Berechnung der alternativen Systemvarianten erfolgt für die in der System- und Kontextanalyse identifizierten Anwendungsfälle. Sie werden in Form entsprechender Belastungsarten und –zyklen als Eingangsgröße für die Berechnung genutzt. Ergebnis ist durch die Variation der Systemelemente und Parametrierungen sowie die Abbildung mehrerer Anwendungsfälle eine umfangreiche Datenmenge. Für diese müssen Darstellungsformen gefunden werden, die die Analyse unterschiedlicher Aspekte erlauben. Dazu gehört die Erfüllung der Kundenwünsche, der Erfüllungsgrad ist maßgeblich für die spätere Architekturentscheidung. Neben den Kundenwünschen sind aber auch technische Machbarkeit einer Systemvariante und resultierende Belastungen für die Subsysteme der Systemvarianten sinnvolle Bewertungskriterien und sollten daher ausgewertet und dargestellt werden.

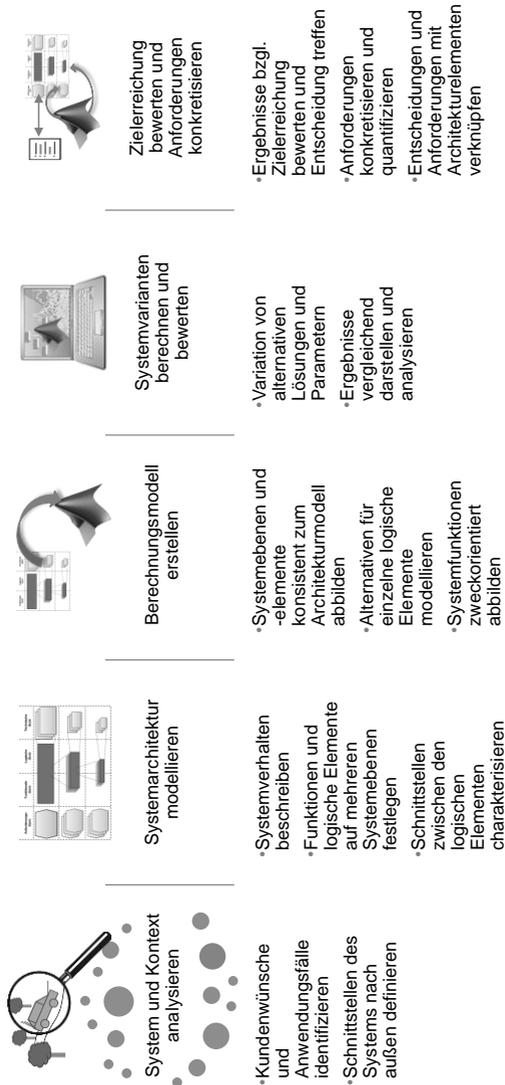


Abbildung 5.9: Ansatz zur Entwicklung konsistenter, nachvollziehbarer und quantitativer Anforderungen durch inhaltliche Verknüpfung von Architektur- und Berechnungsmodell.

Durch Vergleich und Bewertung aller in der Simulation gewonnenen Erkenntnisse werden Entwicklungsentscheidungen getroffen. Die Kriterien hierfür müssen projektangepasst gewählt und gewichtet werden. Ist eine Entscheidung getroffen, können entsprechend der in Kapitel 5.2.3 beschriebenen Systematik aus den Simulationsergebnissen auf den unterschiedlichen Systemebenen Anforderungen und Randbedingungen für die Subsysteme identifiziert werden. Diese werden in das verwendete Anforderungsmanagementsystem bzw. in die Anforderungssicht des Architekturmodells aufgenommen. Dann werden sie mit den entsprechenden Elementen des Architekturmodells verknüpft, um Nachvollziehbarkeit und Nachverfolgbarkeit herzustellen. Durch die konsistente Modellierung von Architekturmodell und quantitativem Modell ist die Zuordnung der Anforderungen zu den Abstraktionsebenen und Elementen eindeutig und die Anforderungen mehrerer Subsysteme sind inhaltlich konsistent, da sie auf Basis derselben Berechnung ermittelt wurden (vgl. Kapitel 5.2.5).

6 Experimentelle Anwendung des Ansatzes: Fahrzeugsimulation mit Parametervariation

Der im vorangehenden Kapitel beschriebene Ansatz ist auf Basis theoretischer Überlegungen zu bekannten entwicklungsmethodischen Ansätzen der Architekturmodellierung, Validierung und Produktgenerationsentwicklung erarbeitet. Im Folgenden wird der Ansatz an einem konkreten Beispiel experimentell angewendet mit dem Ziel, die Stärken und Schwächen des Ansatzes zu identifizieren. Die Randbedingungen der vorliegenden Arbeit lassen dabei keine Anwendung durch Dritte im industriellen Umfeld zu. Um dennoch zu erwartende Potenziale und Risiken des Ansatzes hinsichtlich der identifizierten Hinderungsfaktoren abzuschätzen (siehe Kapitel 7), erfolgt die experimentelle Anwendung des Ansatzes durch die Verfasserin der Arbeit.

Abbildung 6.1 zeigt Strukturierung der folgenden Unterkapitel. Nach kurzer Einführung des Beispiels in Kapitel 6.1 wird in Kapitel 6.2 der Fokus zunächst auf die Systemanalyse und die damit einhergehende Verknüpfung von Architekturmodell und quantitativem Modell gelegt. Darauf aufbauend werden in Kapitel 6.3 die Ergebnisse des quantitativen Modells analysiert und gemeinsam mit dem Architekturmodell genutzt, um alternative Systemarchitekturen zu vergleichen und aus dem Vergleich Anforderungen an die Subsysteme abzuleiten. Abschließend werden Vor- und Nachteile des Ansatzes in Kapitel 7 mit Hilfe der in Kapitel 4.1 vorgestellten Bewertungsstruktur analysiert.

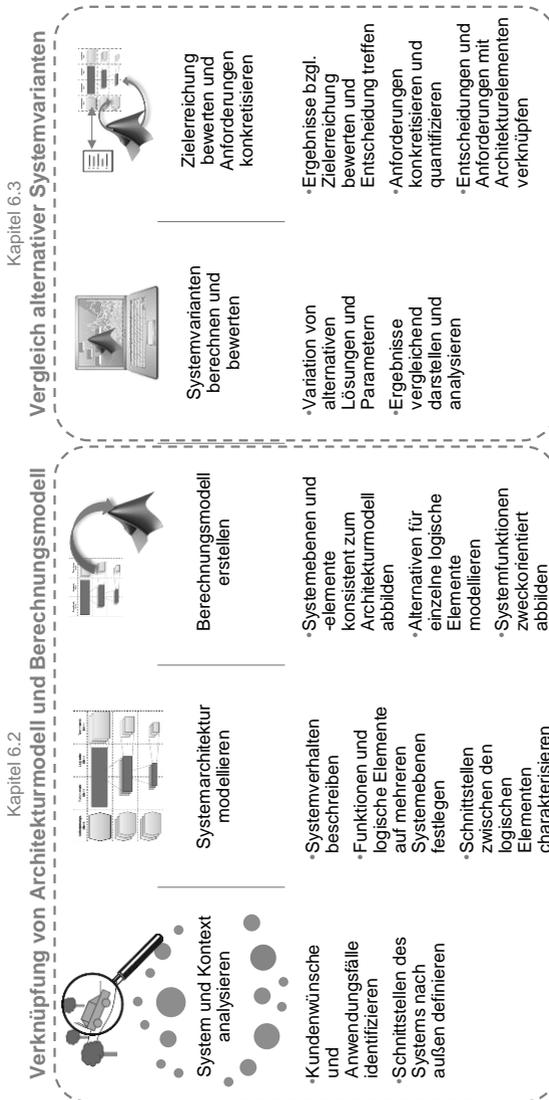


Abbildung 6.1: Strukturierung der experimentellen Anwendung des Ansatzes anhand des Beispiels elektrifizierter Fahrzeugantrieb.

6.1 Beispiel für die experimentelle Anwendung des Ansatzes

Als Beispiel dient die Entwicklung eines Antriebssystems für ein elektrifiziertes Fahrzeug (EV), wie in Abbildung 6.2 schematisch dargestellt.

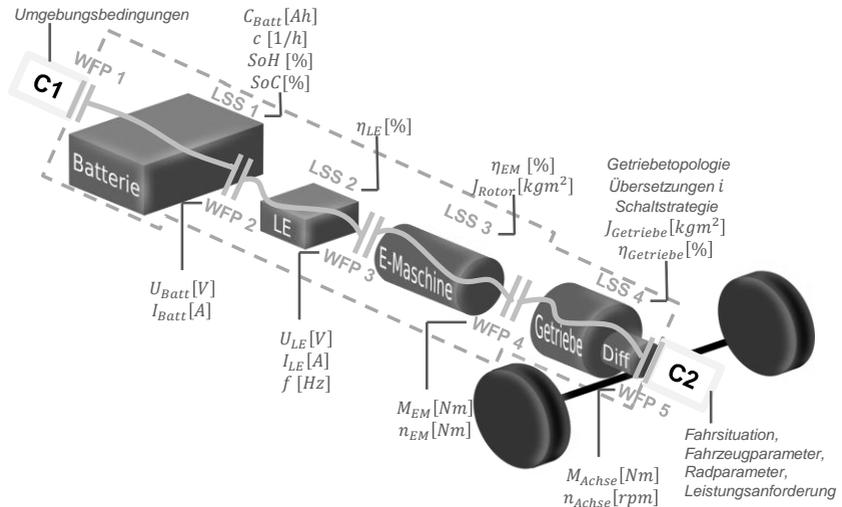


Abbildung 6.2: Wirknetz eines elektrischen Antriebssystems dargestellt im C&C²-Ansatz nach (Albert Albers, Radimersky & Brezger, 2015)

Das gesamte Antriebssystem umfasst die Subsysteme Batterie, Leistungselektronik (LE), Traktionsmaschine (E-Maschine / EM) und Getriebe inkl. Differenzial (Diff). Abbildung 6.2 basiert auf einer Beschreibung des Antriebssystems mit Hilfe des C&C²-Ansatzes nach (Albert Albers, Radimersky & Brezger, 2015). So wird zum einen die Systemgrenze über die beiden Connectoren C1 und C2 beschrieben, zum anderen werden die technischen Schnittstellen durch die an den Wirkflächenpaaren (WFP) übertragenen Größen und die zentralen Eigenschaften der Systeme durch die Leitsützstrukturen (LSS) beschrieben. In Abbildung 6.2 nicht dargestellt sind die Antriebssteuerung, die je nach aktueller Fahrsituation den Soll-Zustand der Antriebsstrangelemente definiert, und das elektrische Bordnetz, das u.a. die Antriebssteuerung mit Energie versorgt. Des Weiteren können auf tieferen Detailierungsebenen weitere Subsysteme identifiziert werden, z. B. innerhalb des Getriebes wo im Rahmen dieser Arbeit für schaltbare Getriebevarianten die Schaltelemente und die Schaltaktuatoren herausgegriffen werden.

6.2 Verknüpfung von qualitativer und quantitativer Modellbildung

Für das gewählte Beispiel des elektrifizierten Antriebssystems wird nun gemäß dem entwickelten Ansatz sowohl ein Architekturmodell als auch ein Berechnungsmodell aufgebaut. Das Architekturmodell wird in der verbreiteten Modellierungssprache SysML mit Hilfe des Softwaretools Rhapsody erstellt. Für die quantitative Modellierung gibt es mehrere Möglichkeiten. In diesem Beispiel wird eine einfache, analytische Modellierung im Softwaretool Matlab umgesetzt. Die Zusammenhänge zwischen den Modellen werden im Folgenden grafisch mittels Screenshots beider Modelle dargestellt, wobei aus dem Berechnungsmodell zur besseren Verständlichkeit nicht der Quellcode selbst, sondern die entsprechende Kommentierung des Quellcodes gezeigt wird. Zur einfacheren Zuordnung der Ebenen und Sichten des Architekturmodells befindet sich außerdem jeweils oben rechts in den Grafiken eine Abbildung des SPES-Frameworks in der die jeweilige Ebene und Sicht hervorgehoben ist.

6.2.1 Kontextanalyse – Abbildung des Kontextes in Architektur- und Berechnungsmodell

Für die Analyse des Systemkontextes wird zunächst die Grenze des zu entwickelnden Systems (SiD) definiert. Im herangezogenen Beispiel ist das Antriebssystem eines elektrischen Fahrzeugs entsprechend Abbildung 6.2 das SiD.

Im Architekturmodell werden das SiD und die Systeme, mit denen es in Wechselwirkung steht, zunächst in einem „Blockdefinitionsdiagramm“ identifiziert. Vorteil ist, dass dieser Schritt eine reine Identifizierung ermöglicht, auch wenn die Schnittstellen der Systeme noch nicht genauer analysiert sind. Der Kontext eines SiD umfasst meist mehrere Systemebenen, so ist beispielsweise das Antriebssystem im Fahrzeug eingebaut und das Fahrzeug steht wiederum in Wechselwirkung mit weiteren Systemen wie Fahrer und Umwelt. Die Anwendung der Systembeschreibung entsprechend dem IPEK-XiL-Ansatz ermöglicht zum einen eine vereinfachte Beschreibung des Kontextes indem alle relevanten Fahrzeugeigenschaften als Block „Restfahrzeug“ abgebildet werden können, zum anderen ist diese Strukturierung die Grundlage für die Konsistenz zwischen Architekturmodell und Berechnungsmodell. Abbildung 6.3 zeigt das Blockdefinitionsdiagramm aus der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells für den EV-Antrieb.

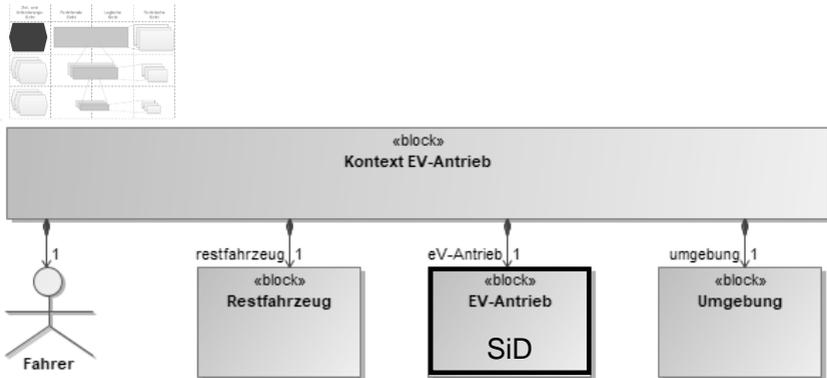


Abbildung 6.3: Kontextelemente des EV-Antriebs im Blockdefinitionsdiagramm der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells.

Das Anwendungsfalldiagramm wird genutzt, um die Anwendungsszenarien des Antriebs darzustellen. Abbildung 6.4 zeigt am Beispiel des Anwendungsfalls „Insassen und Gepäck entsprechend Fahrerwunsch von Startpunkt zu Endpunkt der Fahrt bewegen“, dass dieser durch unterschiedliche Fahrsituationen und Manöver konkretisiert werden kann. Der Kundenwunsch wird also in Form dieser Fahrsituationen expliziert und kann – sofern Daten vorhanden sind – auch im Berechnungsmodell abgebildet werden (siehe auch Abbildung 6.8 und Abbildung 6.10).

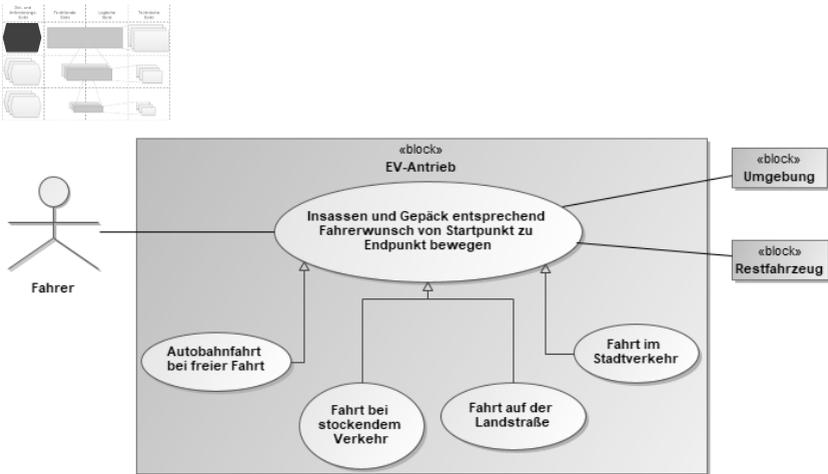


Abbildung 6.4: Anwendungsfalldiagramm für die Hauptanwendung „Insassen und Gepäck bewegen“ des EV-Antriebs in der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells.

Der funktionale Ablauf des identifizierten Anwendungsfalls „Insassen und Gepäck [...] bewegen“ wird im Architekturmodell durch ein Aktivitätsdiagramm genauer beschreiben (Abbildung 6.5). Dabei werden für alle am Anwendungsfall beteiligten sozio-technischen Systeme die jeweiligen Funktionen identifiziert, mit denen sie zur Erfüllung des Anwendungsfalls beitragen. So wird der funktionale Beitrag des EV-Antriebs zum Anwendungsfall als „Leistung an Achse bereitstellen“ konkretisiert.

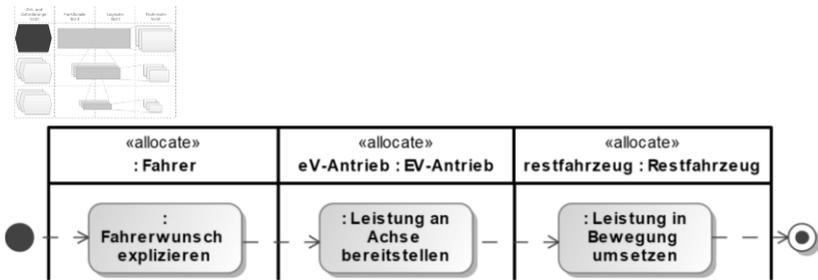


Abbildung 6.5: Funktionaler Ablauf des Anwendungsfalls „Insassen und Gepäck bewegen im Aktivitätsdiagramm der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells.

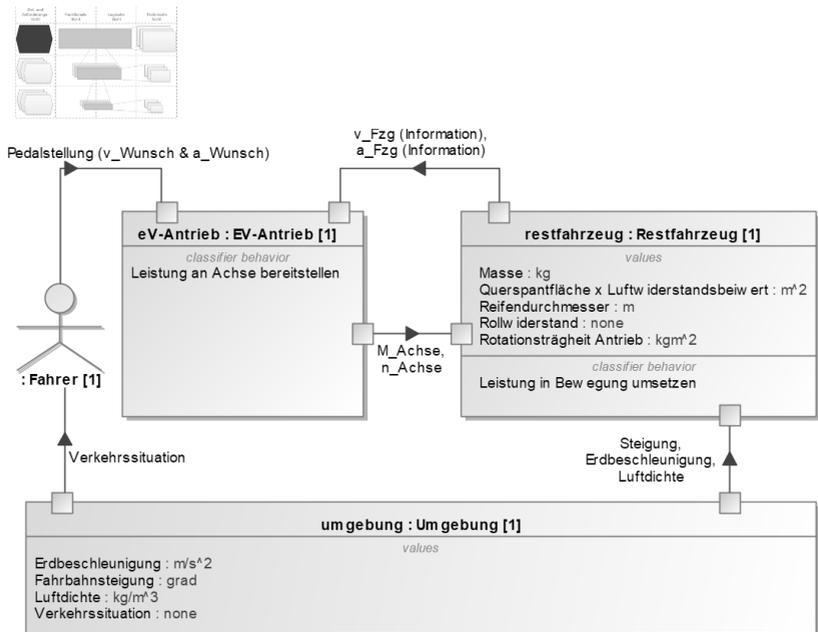


Abbildung 6.6: Logische Zusammenhänge und Schnittstellen des SiD „EV-Antrieb“ mit den umgebenden Systemen im internen Blockdiagramm der Ziel- und Anforderungssicht auf Ebene 1 des Architekturmodells

Neben dem funktionalen Ablauf können durch die Analyse des Anwendungsfalls auch die logischen Zusammenhänge und Schnittstellen zwischen dem Antriebssystem und den umgebenden Systemen beschrieben werden. Zur Darstellung wird ein internes Blockdiagramm des Fahrzeugs genutzt, siehe Abbildung 6.6.

Das in den vorigen Absätzen beschriebene Architekturmodell ist die Basis für die folgende Abbildung derselben Zusammenhänge im Berechnungsmodell. Der Fahrerwunsch in unterschiedlichen Fahrsituationen wird im Berechnungsmodell durch einen Fahrzyklus in Form von Geschwindigkeit über Zeit dargestellt. Zusätzlich ermöglicht die Modellierung auch die Angabe einer Fahrbahnsteigung. Im einfachsten Fall als konstante Steigung, die für den gesamten Zyklus wirksam ist, es ist jedoch auch die Vorgabe eines Steigungsvektors möglich. Die Quelle der Geschwindigkeitsprofile sind in diesem Beispiel neben Ausschnitten aus dem Zyklus des „Worldwide harmonized Light vehicles Test Procedure (WLTP)“ auch real aufgezeichnete Fahrten aus der Arbeit von Heinrich (Heinrich, 2016), die die entsprechenden Fahrsituationen widerspiegeln.

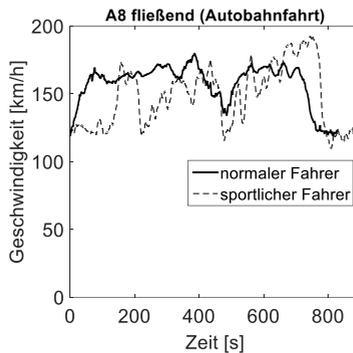
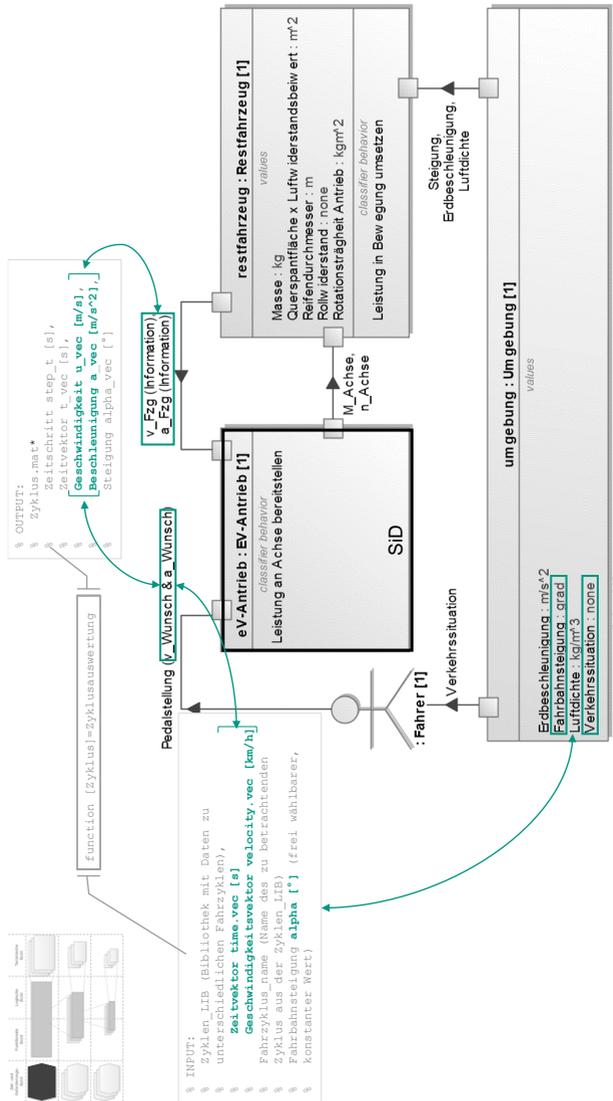


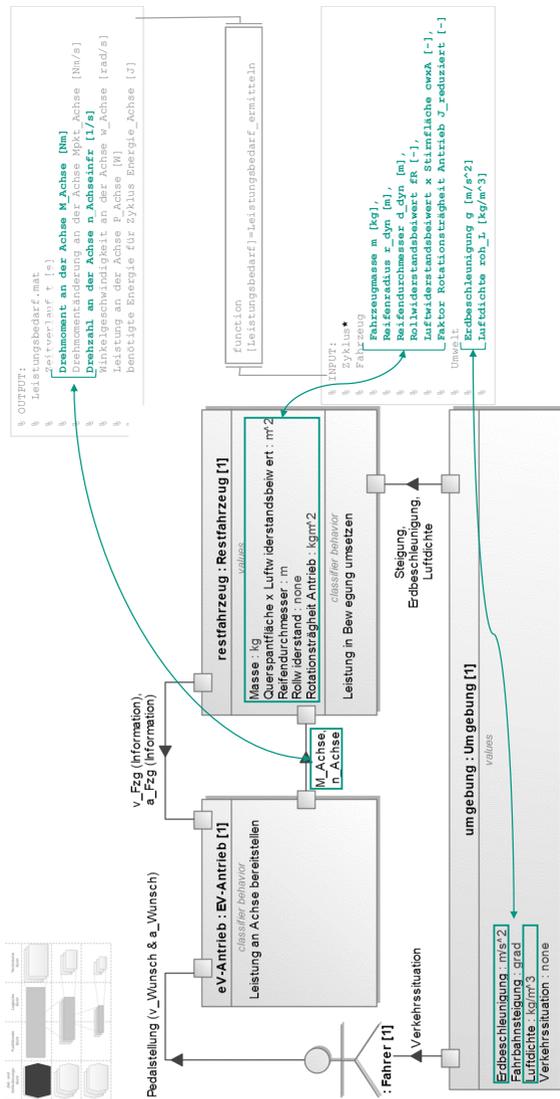
Abbildung 6.7: Geschwindigkeitsprofile für einen sportlichen und einen normalen Fahrer bei Fahrsituation „Autobahnfahrt“ auf Basis (Heinrich, 2016).

Heinrich zeigt, dass sich unterschiedliche Fahrertypen an der Art ihrer Fahrweise kategorisieren lassen. (Heinrich, 2016) Durch Nutzung der Fahrzyklusdaten und entsprechenden Fahrerklassifizierungen kann somit im Berechnungsmodell nicht nur ein Fahrzenario wie z. B. „Stop and Go“ oder „fließender Verkehr“ auf der Autobahn abgebildet werden, sondern zeitgleich kann auch das Fahrerverhalten berücksichtigt werden. Abbildung 6.7 zeigt am Beispiel einer Fahrsituation mit fließendem Verkehr die unterschiedlichen Geschwindigkeitsprofile für einen sportlichen und einen normalen Fahrer.



* Das Ergebnis „Zyklus“ der Funktion „zyklusauswertung“ ist eine Eingangsgröße der Funktion „Leistungsbedarf_ermitteln“

Abbildung 6.8: Vergleich der Modellierung der umgebenden Systeme und deren Zusammenhänge mit dem SiD „EV-Antrieb“ im Architekturmodell und im Berechnungsmodell, Fokus: Fahrerwunsch



*Das Ergebnis „Zyklus“ der Funktion „Zyklausauswertung“ ist eine Eingangsgröße der Funktion „Leistungsbedarf_ermitteln“

Abbildung 6.9: Vergleich der Modellierung der umgebenden Systeme und deren Zusammenhänge mit dem SiD „EV-Antrieb“ im Architekturmodell und im Berechnungsmodell, Fokus: Leistungsbedarf an der Achse

Abbildung 6.8 und Abbildung 6.9 zeigen, wie die im Architekturmodell identifizierten Zusammenhänge zwischen SiD und den umgebenden Systemen quantitativ im Berechnungsmodell umgesetzt werden. Um möglichst einfach unterschiedliche Manöver und Fahrertypen abbilden zu können wird der Fahrerwunsch wie bereits beschrieben in Form eines Geschwindigkeit-Zeitverlaufs abgebildet.

Zu den Zielen des EV-Antriebs gehört es, alle beschriebenen Fahrmanöver bei minimalem Energiebedarf zu ermöglichen. Die verwendeten Verläufe werden als Teil des Zielsystems zusätzlich auch in Anforderungselementen des Architekturmodells verankert (Abbildung 6.10).

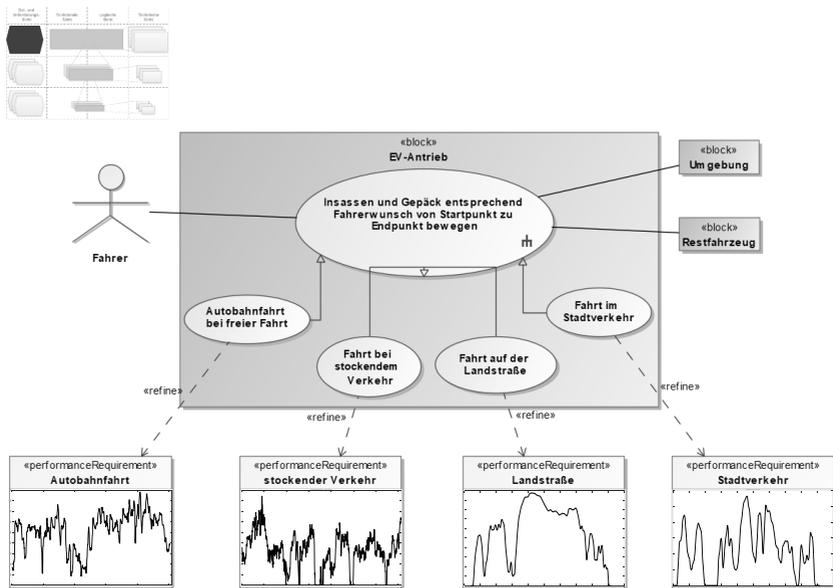


Abbildung 6.10: Repräsentation der Anwendungsfälle des SiD „EV-Antrieb“ als Anforderungen im Architekturmodell, die die im Berechnungsmodell verwendeten Zyklen beinhalten.

Um für das Berechnungsmodell (oder auch weitere Modelle und Analysen) eine konsistente Datenbasis zu erreichen, können die Verläufe nicht nur grafisch dargestellt werden, sondern auch als Link zur entsprechenden Datenreihe in den Anforderungen hinterlegt werden. Die Datenreihen können dann auf einer für alle Teammitglieder zugänglichen Datenbank abgelegt werden.

Die Modellbildung des Berechnungsmodells beruht also auf der verkürzenden Prämisse, dass der Fahrerwunsch vollständig erfüllt wird: Tatsächliche Fahrzeugbeschleunigung und Fahrzeuggeschwindigkeit entsprechen exakt dem Beschleunigungswunsch und Geschwindigkeitswunsch des Fahrers. Die Funktion „Zyklusauswertung“ des Berechnungsmodells ermittelt aus den vorhandenen Fahrdaten die Fahrzeugbeschleunigung in m/s^2 und Fahrzeuggeschwindigkeit in m/s , außerdem wird der Zeitvektor in Sekundenschritten definiert und optional kann ein Steigungsvektor oder eine konstante Steigung angegeben werden. Diese Daten werden als Ergebnis „Zyklus“ ausgegeben. Die mechanische Leistung, die der EV-Antrieb in Form von „M_Achse“ und „n_Achse“ bereitstellen muss ergibt sich aus dem Zyklus in Kombination mit Daten des Restfahrzeugs und der Umwelt. Die Funktion „Leistungsbedarf_ermitteln“ des Berechnungsmodells ermittelt entsprechend die Leistungsdaten, die über die angetriebene Achse an die Räder bereitgestellt werden müssen. Durch die beiden beschriebenen Funktionen sind also der Kontext des EV-Antriebs und dessen Zusammenhang vollständig quantitativ abgebildet.

6.2.2 Systemarchitekturmodell und Berechnungsmodell für das System in Development erstellen

Um Ziele und Anforderungen an das SiD und dessen Subsysteme konkretisieren zu können, wird das Verständnis über den Aufbau und die Funktionsweise des SiD in der logischen und funktionalen Sicht des Architekturmodells abgebildet. Hierfür wird das logische Element „EV-Antrieb“ und seine Funktion „Leistung bereitstellen“ im Architekturmodell weiter verfeinert. Ebenso wie im vorangegangenen Kapitel wird parallel dazu auch aufgezeigt, wie das Berechnungsmodell entsprechend dem Architekturmodell erstellt wird. Die Modellierung beider Modelle gliedert sich in drei Ebenen wobei Ebene 1 das Antriebssystem und die Identifizierung der Ebene 2 Subsysteme und deren Funktionen beinhaltet. Auf Ebene 2 werden die identifizierten Subsysteme weiter detailliert und so Ebene 3 Subsysteme identifiziert. Im vorliegenden Beispiel wird dieser Schritt auf Ebene 2 für das Subsystem Getriebe durchgeführt. Auch hierbei wird auf die Erstellung des Berechnungsmodells eingegangen.

6.2.2.1 Ebene 1: Antriebssystem des batterieelektrischen Fahrzeugs

Abbildung 6.11 zeigt das Blockdefinitionsdiagramm, das in der logischen Sicht auf Ebene 1 darstellt, welche logischen Subsysteme das SiD „EV-Antrieb“ bilden. Es besteht aus

- dem Hochvolt-Energiespeicher „HV_Batterie“,

- der Leistungselektronik „LE“ zur Wandlung der von der HV_Batterie zur Verfügung gestellten elektrischen Leistung in Phasenströme und –spannungen der Traktionsmaschine
- der Traktionsmaschine „EM“ zur Wandlung der elektrischen Energie in mechanische Energie am Getriebeeingang,
- dem „Getriebe“ als Leistungswandler zur Anpassung der von der Traktionsmaschine bereitgestellten Leistung an den Leistungsbedarf des Rades,
- der „Antriebssteuerung“ als Steuerungselement zur Abstimmung des Zusammenspiels zwischen allen Subsystemen,
- und dem Bordnetz „LV_Bordnetz“ zur Energieversorgung von Steuerung und weiteren Subsystemen auf tieferen Detaillierungsebenen.

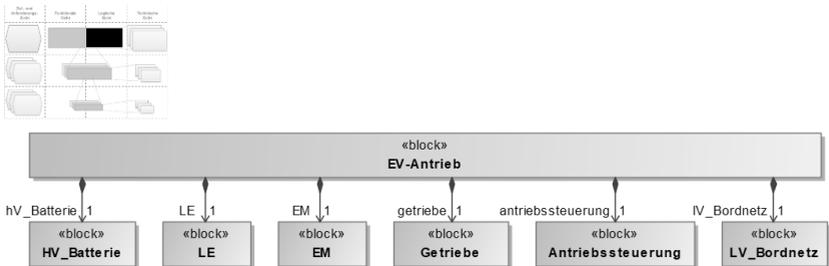


Abbildung 6.11: Blockdefinitionsdiagramm des SiD „EV-Antrieb“ in der logischen Sicht auf Ebene 1 des Architekturmodells.

Alle aufgezeigten logischen Elemente wirken zusammen um die in Abbildung 6.5 dargestellte Hauptfunktion des EV-Antriebs “Leistung an der Achse bereitstellen” umzusetzen. Hierfür erfüllt jedes der Subsysteme eine oder mehrere Funktionen. Im Architekturmodell werden die Teilfunktionen sowie deren Zuordnung zu den genannten Subsystemen in der funktionalen Sicht der Ebene 1 in einem Aktivitätsdiagramm entsprechend Abbildung 6.12 dargestellt.

Welche physikalischen Größen und Informationen zur Erfüllung der Teilfunktionen zwischen den Systemen ausgetauscht werden ist im internen Blockdiagramm des EV-Antriebs dargestellt (Abbildung 6.13). In dieser Darstellung sind auch die charakterisierenden Eigenschaften der einzelnen Subsysteme dargestellt, die zur Erfüllung der Hauptfunktion elementar sind und somit in der Entwicklung quantitativ festgelegt werden müssen. Außerdem sind in dieser Ansicht auch die durch das jeweilige Subsystem umzusetzenden Funktionen dargestellt.

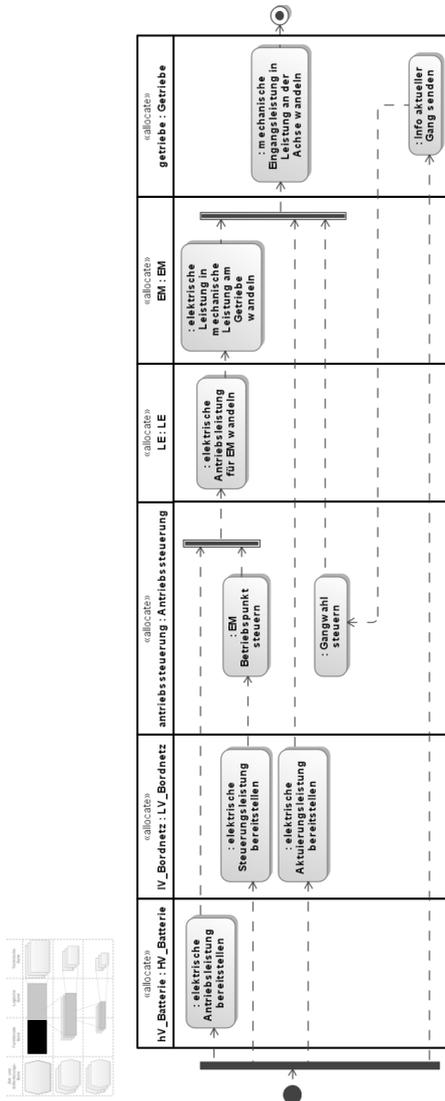


Abbildung 6.12: Aktivitätsdiagramm zur Detaillierung der Funktion „Leistung an der Achse bereitstellen“ in der funktionalen Sicht auf Ebene 1 des Architekturmodells.

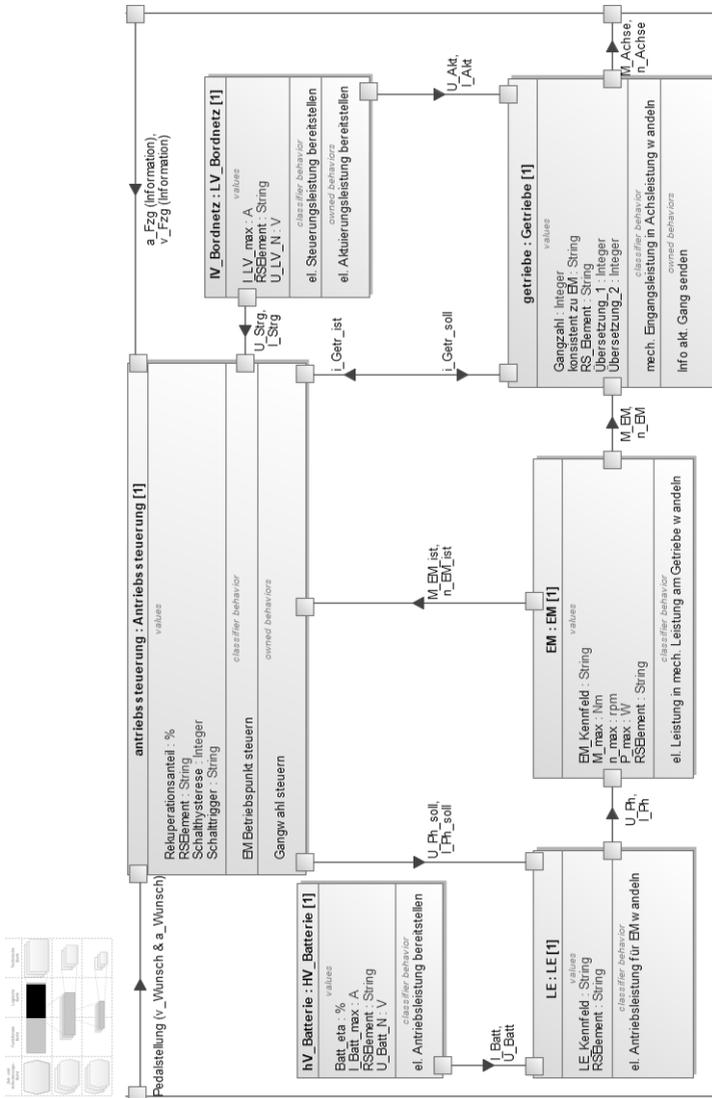


Abbildung 6.13: Darstellung der Zusammenhänge zwischen den Subsystemen des EV-Antriebs in einem internen Blockdiagramm der logischen Sicht auf Ebene 1 des Architekturmodells.

Der physikalische Leistungsfluss in Form von elektrischer und mechanischer Leistung geht von der HV_Batterie (U_{Batt} , I_{Batt}) aus in die Leistungselektronik (LE), wo die elektrische Leistung in die Eingangsleistung der Traktionsmaschine (EM) gewandelt wird (U_{Ph} , I_{Ph}). Die Traktionsmaschine (EM) wandelt diese Eingangsleistung in mechanische Leistung (M_{EM} , n_{EM}). Das Getriebe passt die mechanischen Leistungsgrößen dem Bedarf an der Achse an (M_{Achse} , n_{Achse}), im Falle eines schaltbaren Getriebes ist dafür Aktuierungsleistung notwendig, die vom LV_Bordnetz gestellt wird (U_{Akt} , I_{Akt}). Die Antriebssteuerung empfängt Informationen zum aktuellen Drehmoment und zur Drehzahl der Traktionsmaschine ($M_{\text{EM_ist}}$, $n_{\text{EM_ist}}$) sowie zur aktuellen Getriebeübersetzung ($i_{\text{Getriebe_ist}}$) und sendet die entsprechenden Sollgrößen ($U_{\text{Ph_soll}}$, $I_{\text{Ph_soll}}$, $i_{\text{Getriebe_soll}}$). Die Ansteuerung wird ebenfalls über das LV_Bordnetz versorgt (U_{Strg} , I_{Strg}).

Im Folgenden wird dargelegt, wie die im Architekturmodell abgebildeten Zusammenhänge im Berechnungsmodell umgesetzt werden. Aus der quantitativen Modellierung des Kontextes ist für einen vorgegebenen Fahrzyklus der Leistungsbedarf an der Achse verfügbar (siehe Abbildung 6.8). Bei vorgegebenen Übersetzungen des Getriebes lässt sich daraus für jede mögliche Übersetzung das notwendige Drehmoment und die Drehzahl der Traktionsmaschine ermitteln. In diesem Beispiel erfolgt diese Rückwärtsrechnung für den gesamten Zyklus und alle verfügbaren Übersetzungen in einem Schritt (Abbildung 6.14). Die Modellbildung zwischen Architekturmodell und Berechnungsmodell weicht hierbei darin voneinander ab, dass im Architekturmodell die Schnittstellengrößen „ M_{EM} “ und „ n_{EM} “ entsprechend dem Funktionsverständnis als Eingangsgrößen in das Getriebe dargestellt sind und die Schnittstellengrößen „ M_{Achse} “ und „ n_{Achse} “ sind die entsprechenden Ausgangsgrößen.

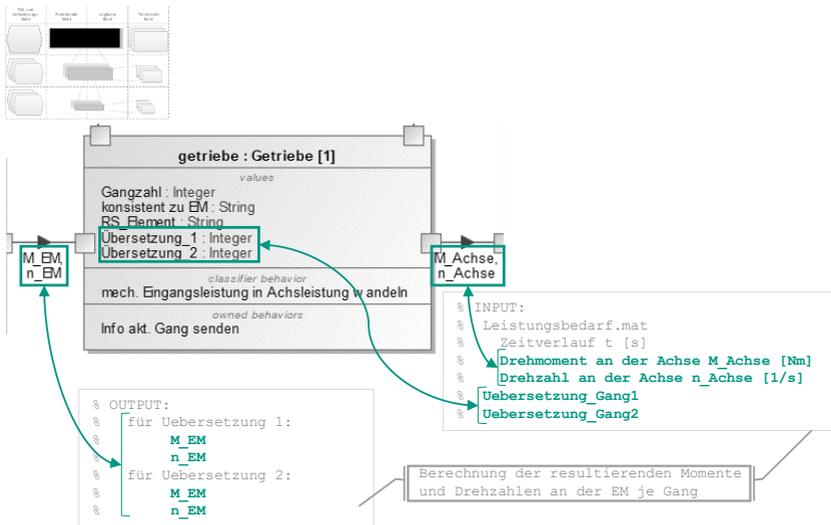


Abbildung 6.14: Berechnung des im aktuellen Fahrzustand notwendigen Drehmoments und der Drehzahl der E-Maschine für jede mögliche Getriebeübersetzung

Da das Berechnungsmodell auf dem Prinzip der Rückwärtsrechnung von einem vorgegebenen Geschwindigkeitsverlauf am Rad – und somit auch an der Achse - ausgehend beruht, sind die Eingangsgrößen und Ausgangsgrößen der Berechnungsfunktion genau umgekehrt. Dennoch sind die benötigten Schnittstellen, Parameter und resultierenden Größen zwischen Architekturmodell und Berechnungsmodell eindeutig zuordenbar. Aufgrund des unterschiedlichen Modellzwecks besitzt das Berechnungsmodell andere Verkürzungsmerkmale und pragmatische Merkmale als das Architekturmodell, obwohl beide Modelle dasselbe Original – das zu entwickelnde System – abbilden. Hier wird deutlich, dass die Durchgängigkeit von modellbasierter Entwicklung erfordert, dass die jeweiligen Modelle zweckorientiert aufgebaut werden können und trotzdem konsistent mit dem Architekturmodell sind.

Anhand Abbildung 6.14 wird der Zusammenhang zwischen Architekturmodell und Berechnungsmodell für die Funktion „Getriebeeingangsleistung in Achsleistung wandeln“, die Art der Berechnung und die verwendeten Subsysteme, Eingangsgrößen und Ausgangsgrößen erläutert. Diese Aspekte decken Zusammenhänge ab, die auf Ebene 1 in der funktionalen und logischen Sicht des Architekturmodells expliziert sind. Die konkrete Parametrierung des Modells und damit die Berechnung alternativer Systemvarianten basiert auf den in der technischen Sicht des Modells

dargelegten Informationen. Ein Beispiel für die technische Sicht auf das Getriebe gibt Abbildung 6.15.

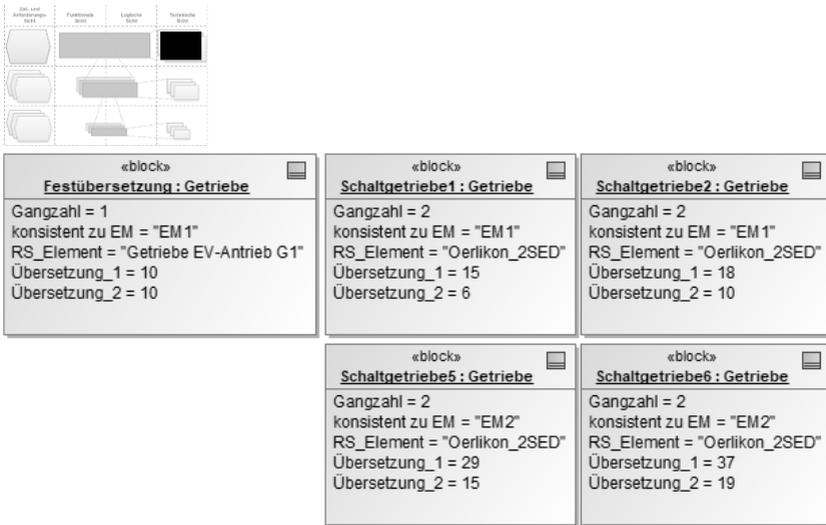


Abbildung 6.15: Beispiel für die Abbildung unterschiedlicher Getriebevarianten in der technischen Sicht auf Ebene 1 des Architekturmodells.

Leistungselektronik und Traktionsmaschine werden im hier gewählten Berechnungsmodell verkürzt durch ihre Wirkungsgradkennfelder auf Basis der Drehmomente und Drehzahlen an der Ausgangswelle der Traktionsmaschine sowie durch deren Drehmoment-Drehzahl-Kennlinie abgebildet. Relevante Eigenschaften der Antriebssteuerung sind Rekuperationsanteil und Schaltstrategie.

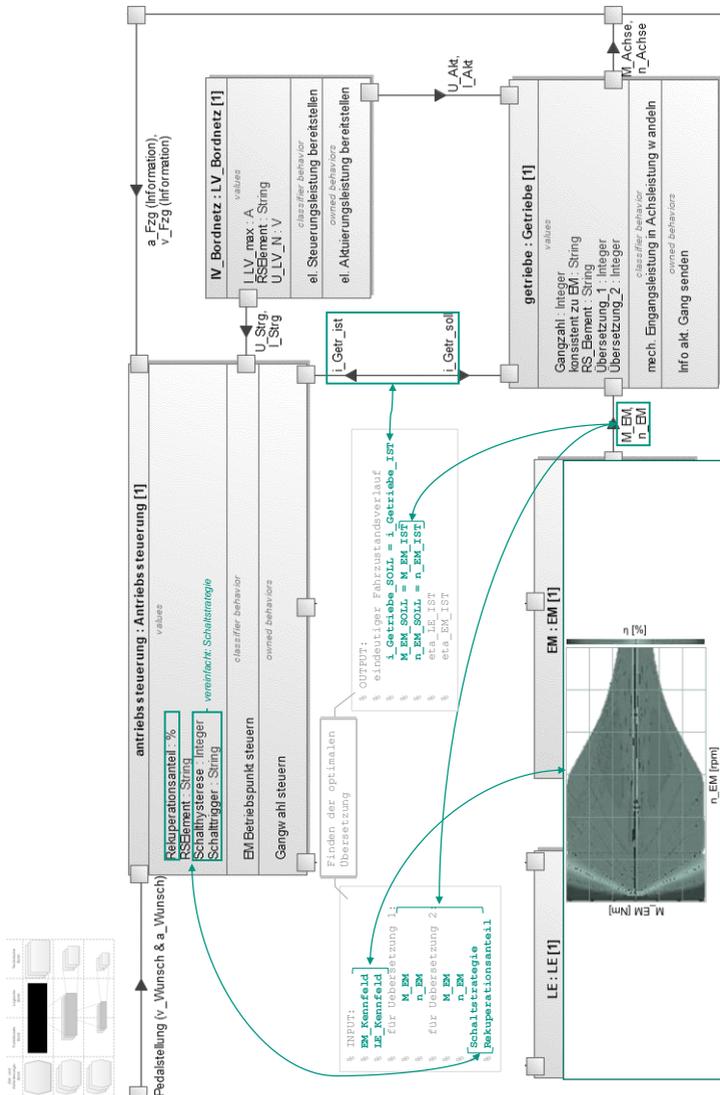


Abbildung 6.16: Ermitteln der optimalen Übersetzung für den gegebenen Fahrzustand.

Die theoretisch notwendigen Drehmomente und Drehzahlen der Traktionsmaschine werden entsprechend dem angegebenen Rekuperationsanteil modifiziert und dann mit dem Kennfeld verglichen. So können für jeden Zeitpunkt des Fahrzyklus diejenigen Betriebspunkte – und damit auch zugehörige Getriebeübersetzungen – ausgeschlossen werden, die außerhalb des Kennfelds liegen. Aus den verbleibenden Betriebspunkten wird für jeden Zeitpunkt im Zyklus derjenige gewählt, der der vorgegebenen Schaltstrategie entspricht. Beispielsweise der Betriebspunkt mit dem maximalen kombinierten Wirkungsgrad von Leistungselektronik und Traktionsmaschine. Auf diese Weise wird ein eindeutiger Fahrzustandsverlauf – d.h. Zeitverlauf von M_{EM} , n_{EM} und $i_{Getriebe}$ – ermittelt. Abbildung 6.16 stellt dar, wie die im Architekturmodell identifizierten Zusammenhänge quantitativ modelliert sind.

Anhand des eindeutigen Fahrzustandsverlaufs kann unter Vorgabe einer nominalen Batteriespannung sowohl die von der Batterie bereitzustellende elektrische Leistung als auch der resultierende Verlauf des Batteriestroms während des Fahrzyklus berechnet werden. Abbildung 6.17 stellt auch hierfür den Vergleich zwischen quantitativem und qualitativem Modell dar. Zur Vereinfachung werden in der vorliegenden quantitativen Modellierung Spannungseffekte in der Batterie, wie beispielsweise Sinken der Spannung bei sinkendem Ladezustand, nicht abgebildet.

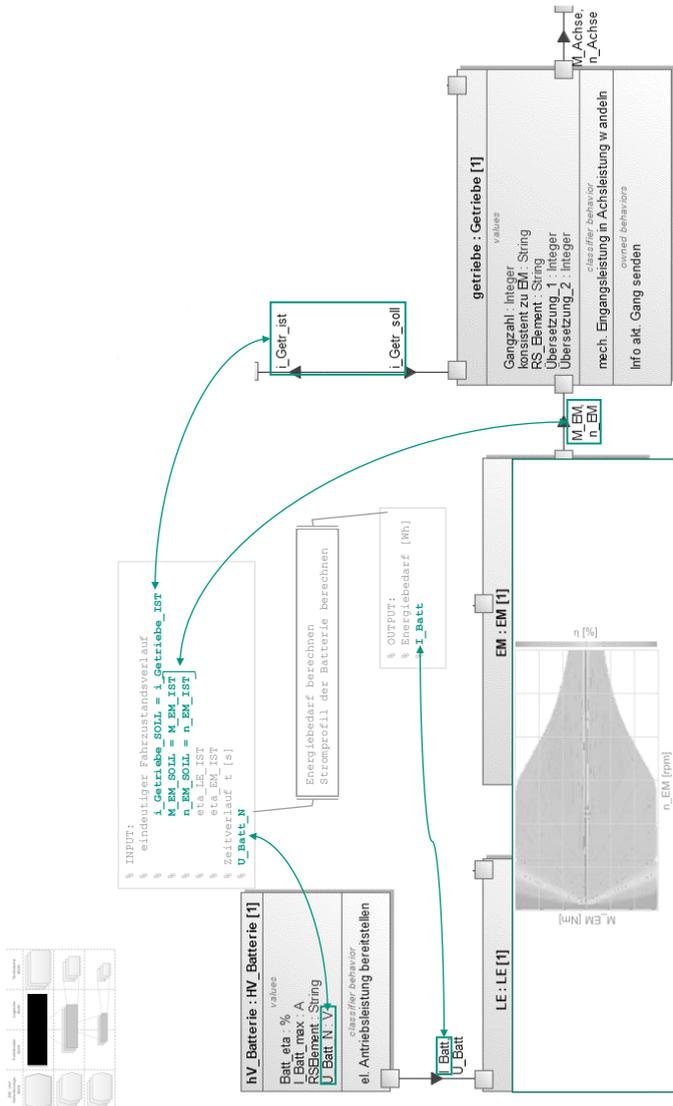


Abbildung 6.17: Berechnung des Energiebedarfs und des resultierenden Stromprofils an der Batterie.

Die Systembetrachtung auf Ebene 1 ermöglicht in diesem Beispiel den Vergleich unterschiedlicher Traktionsmaschinen-Getriebe-Kombinationen hinsichtlich Kriterien wie "Erfüllbarkeit der vorgegebenen Fahrzyklen", "Energieeffizienz in unterschiedlichen Fahrzyklen", etc. Auf dieser Basis können Anforderungen an die Eigenschaften der zu entwickelnden Teilsysteme Traktionsmaschine und Getriebe abgeleitet werden. Entsprechend der gewählten Betrachtungsebene bleiben jedoch Effekte wie z. B. Energieverbrauch für Schaltaktuatoren oder Kupplungsverluste dabei unberücksichtigt. Diese hängen stark vom Getriebekonzept und den gewählten Aktuatoren ab. Um detailliertere Konzeptentscheidungen zu treffen und Anforderungen an Teilsysteme des Getriebes ableiten zu können, muss die Betrachtung somit detailliert werden. Im Folgenden wird daher die qualitative und quantitative Modellierung des Getriebes beschrieben, um auch hier die Vor- und Nachteile der gezielten Abstimmung beider Modelle aufeinander zu bewerten.

6.2.2.2 Ebene 2: Getriebe

Die Abbildung des Getriebes im Architekturmodell hat zum Ziel, dass sie für unterschiedliche Getriebevarianten gleichermaßen anwendbar ist. Daher erfolgt in diesem Beispiel die Unterteilung in logische Elemente wie in Abbildung 6.18 dargestellt: Ein Getriebe besteht neben einer Vorübersetzung am Getriebeeingang und dem Differenzial am Getriebeausgang, aus beliebig vielen Gängen in Form von Radsätzen, die je nach Getriebekonzept durch beliebig viele Schaltelementen, zugehörige Schaltmechanismen und Schaltaktuatoren betätigt werden. Im Falle eines nicht schaltbaren Getriebes sind keine Schaltelemente vorhanden und entsprechend auch keine Schaltmechanismen oder -aktuatoren. Die Anzahl dieser Teilsysteme in einem Getriebe kann somit auch Null sein.

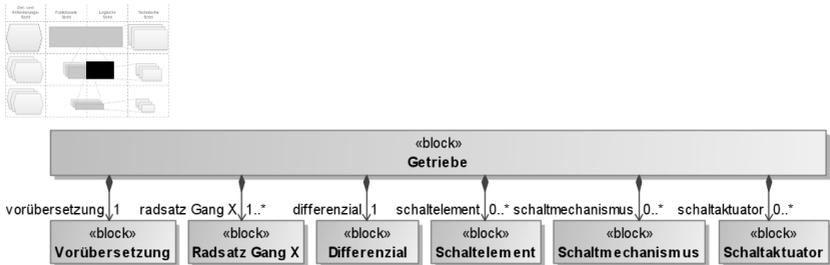


Abbildung 6.18: Subsysteme des Level-2-Systems „Getriebe“, dargestellt im Blockdefinitionsdiagramm der logischen Sicht auf Ebene 2 des Architekturmodells.

Die Funktion “Getriebeeingangsleistung in Leistung an der Achse wandeln” wird durch die logischen Elemente des Getriebes wie in Abbildung 6.19 dargestellt umgesetzt. Drehmoment und Drehzahl der Traktionsmaschine werden durch die Gesamtübersetzung aus Vorübersetzung, im Leistungsfluss befindlichen Radsatz und Differenzial in Drehmoment und Drehzahl an der Achse übersetzt. Im Falle eines Gangwechsels stellt der Schaltaktuator die Aktuierungsleistung bereit, die über den Schaltmechanismus an das entsprechende Schaltelement weitergegeben wird, so dass der gewünschte Radsatz in den Leistungsfluss eingebunden wird. Der zuvor im Leistungsfluss befindliche Radsatz wird über eine analoge Funktionskette aus dem Leistungsfluss entfernt. Abhängig vom Getriebekonzept erfolgt das Entfernen und Einbinden der Radsätze parallel (zugkraftunterbrechungsfreie Schaltung) oder nacheinander (Schaltung mit Zugkraftunterbrechung).

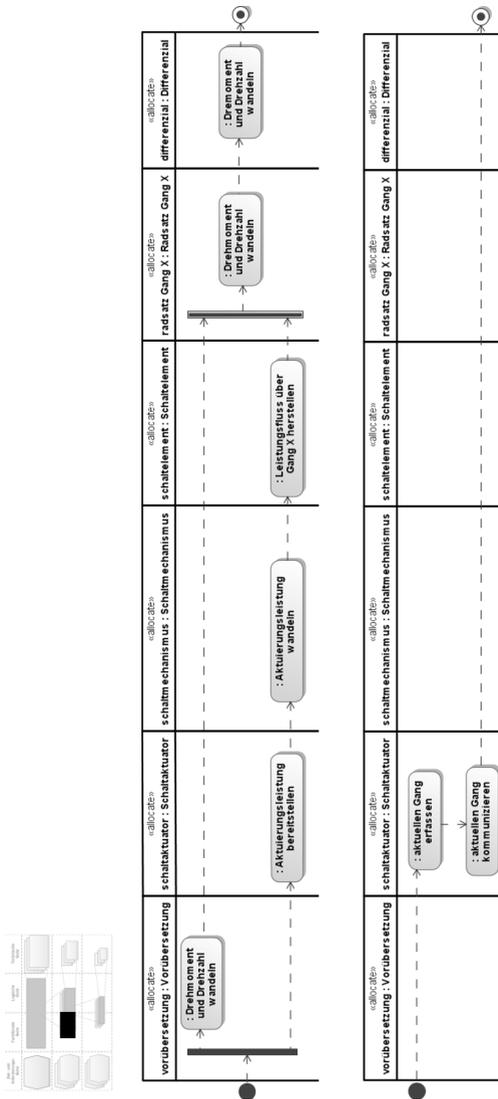


Abbildung 6.19: Detaillierung der Funktionen „Getriebeeingangsleistung in Leistung an der Achse wandeln“ und „Info aktueller Gang senden“ durch Darstellung in einem Aktivitätsdiagramm in der funktionalen Sicht auf Ebene 2 des Architekturmodells.

Entsprechend der Modellierung in Ebene 1 hat das Getriebe neben der Funktion "Getriebeeingangsleistung in Leistung an der Achse wandeln" auch die Funktion "Info aktueller Gang senden" (siehe Abbildung 6.12). In diesem Beispiel wird diese Funktion innerhalb des Getriebes vollständig durch den Schaltaktuator erfüllt. Dieser interpretiert aus seiner aktuellen Position den aktuellen Gang und kommuniziert die Information direkt an die Antriebssteuerung, weitere Subsysteme sind nicht beteiligt. Abbildung 6.19 stellt auch diese Funktion des Schaltactuators im Kontext der anderen Subsysteme des Getriebes dar. Für die Ermittlung der Reichweite auf Gesamtsystemebene spielt diese Funktion jedoch keine Rolle, daher wird sie im Berechnungsmodell nicht berücksichtigt.

Im internen Blockdiagramm des Getriebes (Abbildung 6.20) sind die Flüsse der physikalischen Leistungsgrößen und die Informationsflüsse dargestellt, die zur Erfüllung der Getriebefunktionen zwischen den Subsystemen des Getriebes wirken. Diese Größen sowie die charakterisierenden Eigenschaften der Subsysteme sind zur Funktionsabbildung auch im Berechnungsmodell abzubilden. Im Folgenden wird daher –analog zur Darstellung auf Ebene 1- dieses Diagramm genutzt, um die Modellierung im quantitativen Modell mit der Modellierung im qualitativen Modell zu vergleichen.

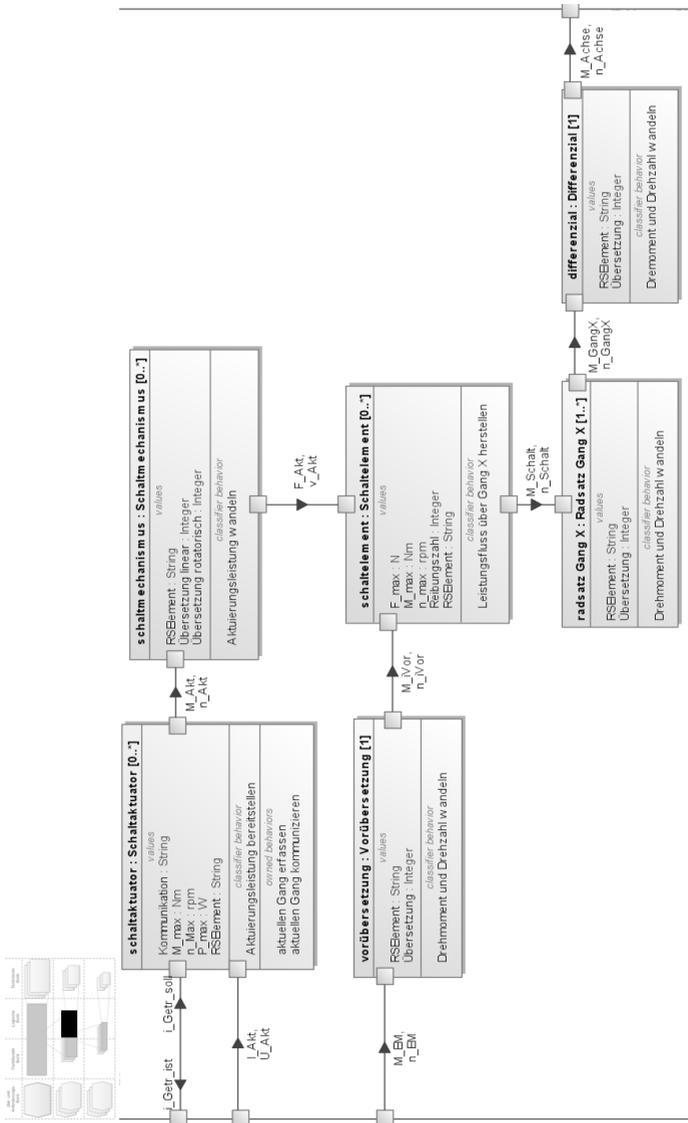


Abbildung 6.20: Zusammenhänge zwischen den Subsystemen des Getriebes dargestellt in einem internen Blockdiagramm in der logischen Sicht auf Ebene 2 des Architekturmodells.

Während auf Ebene 1 die funktionalen Eigenschaften des Getriebes auf die Anzahl der Gänge und die Gesamtübersetzungen zwischen Getriebeeingang und Getriebeausgang reduziert werden, ermöglicht die Detaillierung auf Ebene 2 auch die Abbildung von zusätzlichen Subsystemen des Getriebes wie Schaltelemente und Schaltaktuatoren. Im Berechnungsmodell, das den Vergleich unterschiedlicher Antriebssystemvarianten hinsichtlich der Energieeffizienz zum Ziel hat, bedeutet diese Detaillierung, dass zusätzliche Energieverluste durch diese Systeme berücksichtigt werden können. In diesem Beispiel ist in Anlehnung an Fracchia ein Getriebekonzept gewählt, das den Gangwechsel mit Hilfe einer Reibkupplung und einem sperrbaren Freilauf realisiert (siehe Abbildung 6.21 oben rechts). (Fracchia, 2017)

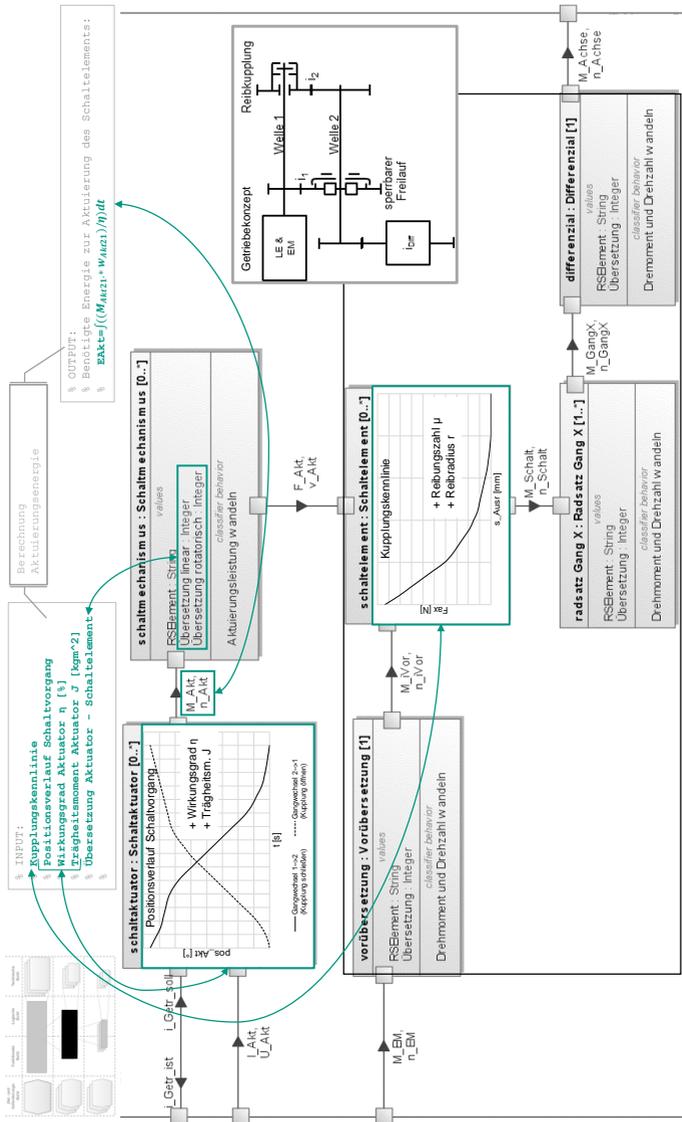


Abbildung 6.21: Berechnung der notwendigen Energie zur Aktuierung des Schaltelements je Gangwechsel.

Abbildung 6.21 stellt den Zusammenhang zwischen Architekturmodell und Berechnungsmodell hinsichtlich der benötigten Energie für einen Schaltaktuator dar. Im Berechnungsmodell wird die Betätigung der Reibkupplung durch einen elektromechanischen, rotatorischen Aktuator angenommen, dessen Drehmoment und Drehzahl über eine Mechanik in Axialkraft und -weg an der Reibkupplung übersetzt wird. Der gewünschte Positionsverlauf der Aktuatorwelle zum Öffnen und Schließen der Kupplung wird im Berechnungsmodell vorgegeben, ebenso das an der Aktuatorwelle wirksame Trägheitsmoment der bewegten Massen und der durchschnittliche Wirkungsgrad. Die Kupplung wird durch ihre Kupplungskennlinie abgebildet, die durch das maximal zu übertragende Moment, Reibradius der Kupplung und durchschnittliche Reibungszahl bestimmt ist. Aus diesen vorgegebenen Daten lassen sich der Momentenverlauf sowie der Beschleunigungsverlauf der Aktuatorwelle bestimmen und damit auch die benötigte Energie für den jeweiligen Schaltvorgang.

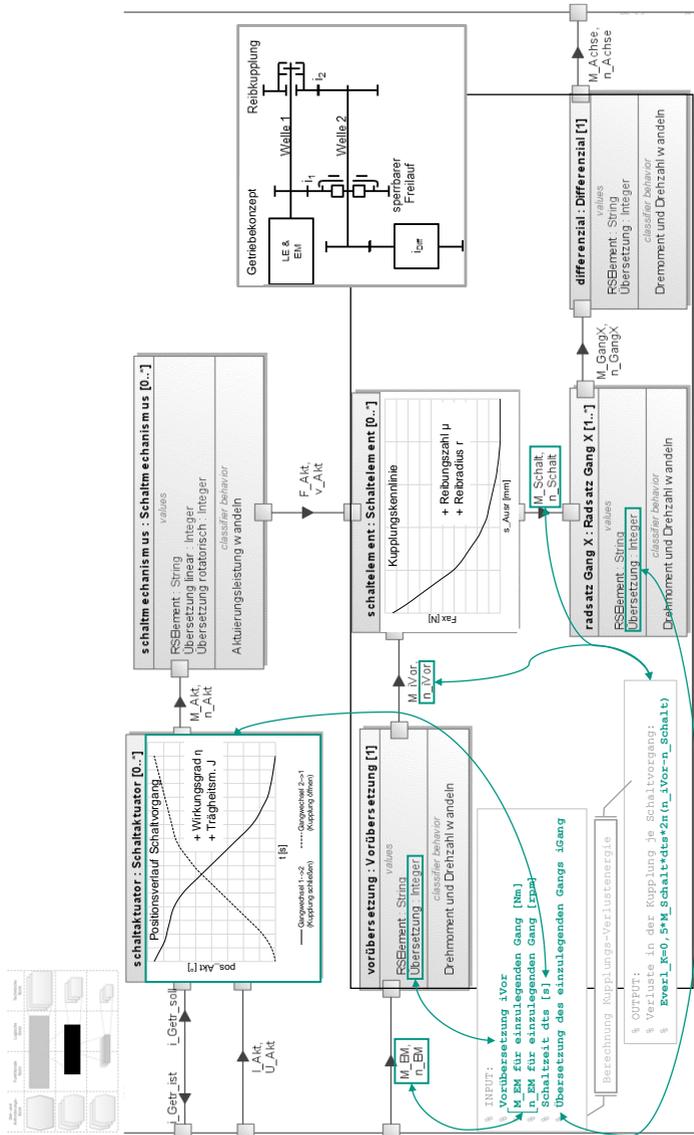


Abbildung 6.22: Berechnung der Verluste im Schaltelement je Gangwechsel.

Zusätzlich zur Energie, die die Aktuierung benötigt, entsteht während der Schaltung auch Verlustenergie in der Reibkupplung, die in Wärme umgewandelt wird. Abbildung 6.22 erläutert die quantitative Modellierung anhand des qualitativen Modells. Der Positions-Zeit-Verlauf des Aktuators gibt die Schaltzeit vor. Also die Dauer, in der sich die Kupplung im Schlupf befindet und Energie durch Reibung in Wärme umgewandelt wird. Im Berechnungsmodell werden zur Vereinfachung folgende Prämissen angenommen:

- Das Kupplungsmoment bleibt über den gesamten Schaltvorgang konstant und entspricht dem zu übertragenden Moment nach Gangwechsel.
- Die Schaltung erfolgt ohne Zugkraftunterbrechung, d.h. das Drehmoment an Welle 2 soll vor, nach und während dem Gangwechsel gleich sein, das Drehmoment der Traktionsmaschine wird entsprechend angepasst.
- Kein Geschwindigkeitssprung des Fahrzeugs, d.h. die Drehzahl an Welle 2 (Sekundärseite Kupplung) soll vor, nach und während dem Gangwechsel gleich sein, die Drehzahl der Traktionsmaschine wird entsprechend angepasst.

Mit den genannten Prämissen kann durch Vorgabe der Schaltzeit, des Drehmoments und der Drehzahl der Traktionsmaschine nach erfolgter Schaltung sowie den wirksamen Übersetzungen die Verlustenergie der Kupplung für den jeweiligen Schaltvorgang berechnet werden.

Die Gesamtenergiebilanz des Antriebs im betrachteten Fahrzyklus kann mit der Modellierung auf Ebene 2 um die zur Schaltung benötigten Energie ergänzt werden. Das ermöglicht wiederum eine Verfeinerung der abgebildeten Schaltstrategie durch die Berücksichtigung des Umstands, dass die Schaltungen nicht energieneutral sind.

6.2.3 Zusammenfassung und Schlussfolgerung zur Nutzung des Architekturmodells als Basis für Erstellung und Nutzung eines Berechnungsmodells

Die Modellierung der Systemzusammenhänge im Architekturmodell entsprechend des SPES-Frameworks ermöglicht die klare Abgrenzung der Systemebenen und die Beschreibung der logischen Elemente und deren funktionalen Zusammenhang innerhalb der Systemebene. Das bietet die Grundlage, auch das Berechnungsmodell entsprechend zu strukturieren. Das im Rahmen der Architekturerstellung aufgebaute und explizierte Systemverständnis kann dadurch einfacher auf das Berechnungsmodell übertragen werden. Aus dem Architekturmodell werden diejenigen logischen Elemente der jeweils betrachteten Abstraktionsebene übernommen, die für

die Abbildung des zu beobachtenden Systemverhaltens notwendig sind. Auch die Schnittstellen dieser Elemente werden übernommen, soweit sie für die Berechnung benötigt werden. In der vorliegenden Arbeit erfolgt die Übernahme manuell, was durch den gleichen strukturellen Aufbau unterstützt wird. Bei Entwicklung einer entsprechenden Schnittstelle zwischen den Modellierungstools wäre aber auch die automatische Übernahme von Subsystembezeichnungen und zugehörigen Ein- und Ausgangsgrößen sowie die automatische Erstellung der Ebenen-Struktur denkbar. Die Nutzbarkeit der Informationen aus dem Architekturmodell für die Erstellung des Berechnungsmodells wird erleichtert. Das kann die individuelle Akzeptanz unterstützen, da der Modellierungsaufwand für die Systemarchitektur bereits als Vorarbeit für die Erstellung anderer Modelle dient.

Die in der Systemarchitektur in Form von internen Blockdiagrammen dargestellten Wechselwirkungen zwischen den Subsystemen werden nicht direkt in das Berechnungsmodell übernommen. Die Ausführungen in Kapitel 6.2.2.1 anhand von Abbildung 6.14 machen deutlich, dass eine Übernahme aller Zusammenhänge aus der Systemarchitektur aufgrund des unterschiedlichen Modellzwecks nicht zielführend wäre. Um eine durchgängige, modellbasierte Entwicklung zu gewährleisten, müssen die einzelnen Modelle zweckorientiert aufgebaut werden und nur die jeweils relevanten Daten aus der Systemarchitektur übernommen werden. Sind die zu übernehmenden Daten identifiziert, besteht das Potenzial diese auch automatisiert, z. B. durch Parametersynchronisation oder Modelltransformation aus der Systemarchitektur in das jeweilige Modell zu übernehmen. Wird die werkzeugseitige Anbindung umgesetzt verringert sich der Modellbildungsaufwand zusätzlich.

Die gleiche Strukturierung beider Modelle dient jedoch nicht nur der vereinfachten Übernahme von Inhalten aus dem Architekturmodell. Auch die Bewertung und Interpretation der Ergebnisse wird unterstützt. Durch die Quantifizierung suggerieren Simulationsergebnisse häufig eine höhere Genauigkeit der Ergebnisse als es tatsächlich der Fall ist. Durch die gleiche Strukturierung der Systemarchitektur und des Berechnungsmodells, kann der Detaillierungsgrad und die Verkürzungsmerkmale der einzelnen Zusammenhänge im Berechnungsmodell veranschaulicht werden. Des Weiteren können Anforderungen eindeutig den Elementen der Systemarchitektur zugeordnet werden (siehe Abbildung 6.10 und nachfolgendes Kapitel 6.3.2). Im Falle von Anforderungs- oder Eigenschaftsänderungen ist ein einfacher Übertrag auf das Berechnungsmodell möglich. Annahmen und Prämissen zu Ausprägungen der Subsysteme, die gegebenenfalls im Zuge der Erstellung des Berechnungsmodells getroffen werden, können ebenfalls als Varianten in der technischen Sicht des Architekturmodells festgehalten und damit nachhaltig dokumentiert werden. Es zeigt sich damit das Potenzial, dass durch geeignete Strukturierung der Modelle auch der

Dokumentationsaufwand im Berechnungsmodell, der üblicherweise anfällt, um die Modellbildung nachvollziehbar zu machen, reduziert werden kann.

Des Weiteren kann das Berechnungsmodell durch die Strukturierung einfacher gezielt modular erstellt und beschrieben werden, sodass einzelne Zusammenhänge entsprechend des Architekturmodells unter Beibehaltung der Schnittstellen "ersetzt" werden kann. Dies ist die Basis für den Vergleich unterschiedlicher Systemvarianten und damit für die nachvollziehbare Ableitung von Anforderungen. Die Modellbildung eines Subsystems im Berechnungsmodell um eine Ebene zu detaillieren bedeutet, statische Vorgaben im gewählten Fokuspunkt aufzugeben und durch dynamische, d.h. im Zeitverlauf mitberechnete, Größen des betreffenden Subsystems zu ersetzen. Diese beruhen auf ihrer Ebene wiederum auf statischen Vorgaben, die bei einer weiteren Detaillierung ebenfalls durch dynamische Größen abgebildet werden könnten. Die statischen Größen können durch Parametrierung des quantitativen Modells variiert werden, um im Sinne eines gewünschten Gesamtsystemverhaltens geeignete Werte zu identifizieren und resultierende Anforderungen an die umgebenden Systeme abzuleiten. Die Variation der Parameter ermöglicht den Vergleich unterschiedlicher Systemeigenschaften und -varianten. Dadurch kann auch der Einfluss einzelner Parameter auf das Gesamtsystemverhalten analysiert werden. Hierauf wird in Kapitel 6.3 näher eingegangen.

Bei manueller Übernahme der Informationen in das Berechnungsmodell ist zu erwarten, dass fortschreitende Detaillierung über Modellebenen hinweg zu zusätzlichem Modellpflegeaufwand führt, um die Konsistenz zu gewährleisten. Dies kann sich sowohl aus individueller als auch aus organisatorischer Sicht auf die Performance der Methode auswirken. Bei zu hohem Pflegeaufwand leidet die individuell wahrgenommene Leistungsfähigkeit des Ansatzes, durch erhöhten Aufwand leidet auch das Aufwand-Nutzen-Verhältnis, das für die organisatorische Akzeptanz relevant ist. Dem kann auf beiden Ebenen nur entgegengewirkt werden, wenn der Nutzen des Ansatzes überwiegt oder die Möglichkeiten zur automatisierten Modelltransformation besser genutzt werden können.

6.3 Vergleich alternativer Systemvarianten

Kapitel 6.2.2 zeigt am Beispiel EV-Antrieb, inwieweit ein Architekturmodell genutzt werden kann, um die quantitative Modellbildung und das Verständnis der modellierten Systemzusammenhänge zu unterstützen. Der Fokus liegt dabei auf der Abbildung der grundsätzlichen Zusammenhänge, nicht jedoch auf der konkreten Parametrierung des Berechnungsmodells. Wie in Kapitel 6.2.3 herausgestellt, ist der Aufwand zur Modellpflege und Konsistenzabsicherung nachteilig, wenn sich beide

Modelle in unabhängigen Softwaretools befinden. Wird diesem Aufwand ein zusätzlicher Nutzen gegenübergestellt, kann dieser Mangel in der Performanz ausgeglichen werden. Ein Zusatznutzen kann in der Darstellung und Interpretation der Ergebnisse des quantitativen Modells liegen. Im Folgenden wird daher am gleichen Beispiel untersucht, wie alternative Systemvarianten - in Form von alternativen Parametrierungen des quantitativen Modells - miteinander verglichen werden können.

6.3.1 Definition alternativer Systemvarianten durch Parametrierung

Das quantitative Modell wird in diesem Beispiel genutzt, um unterschiedliche Antriebssystemvarianten hinsichtlich ihrer Energieeffizienz in vorgegebenen Fahrzyklen und miteinander zu vergleichen. Des Weiteren sollen auch die Belastungen der Subsysteme in den Antriebssystemen analysiert werden, um daraus Anforderungen an die Subsysteme abzuleiten.

6.3.1.1 Angenommenes Entwicklungsszenario für das Beispiel

Für das vorliegende Beispiel wird folgendes Entwicklungsszenario angenommen:

Um besonders hohe Beschleunigungswerte zu ermöglichen und damit den Aspekt "Fahrspaß" in den Vordergrund zu stellen, ist die Entscheidung für ein zweigängiges Getriebe des E-Fahrzeugs gefallen. Das Getriebekonzept für ein schaltbares Getriebe mit zwei Gängen ist bereits entwickelt und in einer vorangehenden Fahrzeuggeneration im Einsatz. Daher soll das Getriebekonzept auch in der neuen Generation zum Einsatz kommen. Veränderungen der Übersetzungen sowie die Umsetzung von Maßnahmen zur Effizienzoptimierung des Getriebes sind im Rahmen des aktuellen Projekts als Gestaltvariation des Getriebekonzepts möglich.

Ebenfalls aus Vorgängergenerationen sind zwei Traktionsmaschinen verfügbar. Beide sind im Serieneinsatz und für den jeweiligen Einsatz validiert. Aus wirtschaftlichen Gründen ist eine der beiden Traktionsmaschinen für das aktuelle Projekt ohne weitere Veränderungen als Übernahmevariation zu wählen.

Als Referenz für den Faktor "Reichweite" sollen die mehrgängigen Varianten in der Entwicklung mit einer Antriebssystemvariante mit nicht schaltbarem Getriebe – also fester Gangübersetzung - verglichen werden. Ziel ist es, mit der zu entwickelnden mehrgängigen Variante trotz der zu erwartenden Effizienzverluste im Getriebe eine mindestens vergleichbare Reichweite zu erzielen.

Für das zu entwickelnde Produkt wurde im Vorfeld ein Nutzungsprofil bestehend aus mehreren Fahrzyklen identifiziert, diese sind in Tabelle 6.1 angegeben.

Tabelle 6.1: Nutzungsprofil des Fahrzeugs charakterisiert durch Anteil von Fahrzyklen an Betriebsdauer des Fahrzeugs

Anwendungsfall (Fahrzyklus)	Betrieb	Stun-
Stadtverkehr	60 %	4800 h
Landstraße	15 %	1200 h
Stockender Verkehr	5 %	400 h
Autobahnfahrt	20 %	1600 h

6.3.1.2 Gewählte Parametrierungen

Die alternativen Antriebssystemvarianten lassen sich in ihrer Funktionsweise und dem grundlegenden Aufbau alle durch das in Kapitel 6.2 vorgestellte Architekturmodell abbilden und sind wie dort gezeigt im Berechnungsmodell modelliert. Die Unterscheidung der Varianten liegt somit in der konkreten technischen Umsetzung. Diese kann im Architekturmodell in der "technischen Sicht" abgebildet werden, im Berechnungsmodell entspricht die technische Umsetzung der Parametrierung des Modells.

Um die unterschiedlichen Fahrzyklen des Nutzungsprofils im Berechnungsmodell abzubilden, werden diese wie in Kapitel 6.2.1 beschrieben durch Geschwindigkeits-Zeit-Verläufe abgebildet (siehe auch Abbildung 6.7 und Abbildung 6.10). Die verwendeten Daten für stockenden Verkehr basieren auf real erfassten Fahrdaten, ebenso wie die Daten für die Autobahnfahrt. Beide Datensätze wurden im Rahmen der Arbeiten von Heinrich erfasst. (Heinrich, 2016) Für Fahrten in der Stadt und auf der Landstraße werden Abschnitte des Zyklus des WLTP für Klasse drei Fahrzeuge genutzt, da die zur Verfügung stehenden Realdaten diese Situationen nicht in ausreichender Dauer abbilden. Stadtverkehr wird durch den ersten Abschnitt des WLTP-Zyklus ($t=0s$ bis $460s$) repräsentiert, die Fahrt auf der Landstraße durch den dritten Abschnitt des WLTP ($t=1020s$ bis $1460s$). Für den Vergleich der Reichweite unterschiedlicher Antriebssystemvarianten wird eine Batteriekapazität von 60 kWh angenommen.

Die Fahrbahnsteigung wird für alle Geschwindigkeitsprofile in diesem Beispiel als vernachlässigbar angenommen und damit nicht variiert, sondern im quantitativen Modell konstant als 0 % vorgegeben. Ebenso wird das Restfahrzeug nicht variiert, sondern durch die notwendigen charakteristischen Größen abgebildet. Es wird von

einem Mittelklassefahrzeug ausgegangen, dementsprechend werden für das Restfahrzeug die folgenden Werte definiert: Fahrzeugmasse, dynamischer Reifendurchmesser, Rollwiderstandsbeiwert, Luftwiderstandsbeiwert, Stirnfläche.

Die Antriebssystemvarianten werden jeweils für alle Fahrzyklen quantitativ analysiert. Hierbei werden auf den Ebenen eins die in Tabelle 6.2 aufgelisteten und in Abbildung 6.23 hervorgehobenen Parameter variiert, auf Ebene zwei werden die in Tabelle 6.3 aufgezeigten Parameter variiert.

Tabelle 6.2: Variierte Parameter im Berechnungsmodell auf Ebene 1

Subsystem	Parameter	Wert 1	Wert 2	Wert 3	Wert 4	Wert 5
Antriebssteuerung	Rekuperationsanteil	80 %	10 %	–	–	–
	Trigger für Schaltung	<i>best</i>	<i>worst</i>	–	–	–
LE_EM	Kennfeld	<i>EM 1</i>	<i>EM 2</i>	–	–	–
Getriebe	Übersetzungen EM 1	10	[15, 6]	[18, 10]	[15, 10, 6]	[38, 18, 10]
	Übersetzungen EM 2	[29, 15]	[37, 19]	[38, 25, 16]	[37, 25, 19]	–
	Wirkungsgrad	94%	93 %	92 %	91 %	90 %

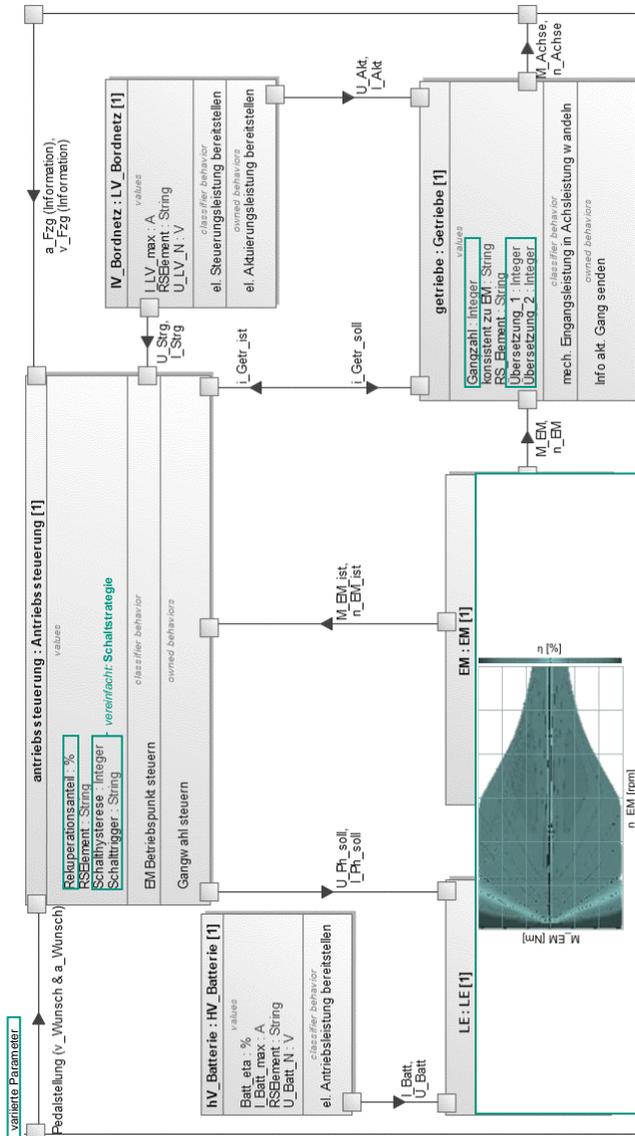


Abbildung 6.23: Variierte Parametrierung auf Ebene eins des Antriebssystems, dargestellt im qualitativen Systemmodell.

Auf Ebene eins werden die logischen Elemente Antriebssteuerung, Leistungselektronik und Traktionsmaschine sowie Getriebe variiert. Tabelle 6.2 stellt die bei der Variation eingesetzten Werte dar.

Die Antriebssteuerung nimmt auf die Reichweite des Antriebssystems insbesondere durch Rekuperationsanteil bei Verzögerungsvorgängen und Schaltstrategie Einfluss. Es bestehen allerdings technische Grenzen des Anteils rekuperierter Energie, beispielsweise aus Fahrstabilitätsgründen oder Gründen der Batteriebelastung. Um den Einfluss des Rekuperationsanteils auf das Gesamtsystemverhalten zu analysieren, wird dieser in den Berechnungen zwischen den Werten „10%“ und „80%“ variiert. Schaltstrategien in automatisierten Getrieben verfolgen in der Regel mehrere Ziele, beispielsweise hohen Schaltkomfort, gleichmäßige Beschleunigung oder einen energieoptimalen Betrieb des Antriebssystems, und hängen in ihrer Umsetzung daher von verschiedenen Faktoren ab. (Pinner, Radimersky & Ott, 2015) Der wesentliche Einfluss der Schaltstrategie auf die hier zu bewertende Energieeffizienz des Antriebssystems ist jedoch die Beeinflussung der Betriebspunkte der Traktionsmaschine inklusive Leistungselektronik. Daher wird im Berechnungsmodell die Schaltstrategie vereinfacht durch die zwei Extreme der effizienzoptimalen Betriebspunktwahl je Zeitschritt („best case“) und der für die Effizienz schlechtesten Betriebspunktwahl je Zeitschritt („worst case“) abgebildet. Also zu jedem Zeitschritt im Zyklus erfolgt die Wahl der Getriebeübersetzung so, dass die minimal mögliche bzw. die maximal mögliche Verlustleistung des Antriebssystems für den aktuellen Fahrzustand erzeugt wird. So kann bei späteren vergleichenden Analysen auch abgeschätzt werden, welches Reichweitenpotenzial eine Antriebsvariante mit optimaler Schaltung hat und welches Risiko der Reichweitenverringerung durch schlechte Schaltung besteht. Durch den beschriebenen Ansatz sind sekundliche Schaltungen während der simulierten Fahrt möglich.

Die Traktionsmaschine inklusive der zugehörigen Leistungselektronik ist durch ihr Kennfeld charakterisiert. Dieses enthält die Informationen über das maximale Drehmoment im Stillstand, die maximale Leistung und die maximal verfügbare Drehzahl sowie den Betriebspunkt abhängigen Wirkungsgrad. Im Berechnungsmodell sind die Kennfelddaten der Leistungselektronik und der Traktionsmaschine für die beiden alternativen Varianten zu je einem Kennfeld kombiniert. Antriebsmaschinen in aktuellen E-Fahrzeugen haben häufig eine maximale Drehzahl von ca. 15.000 rpm und ein maximales Drehmoment von ca. 500 Nm. Zur Verringerung der rotatorischen Massenträgheit im Antriebssystem und damit zur Steigerung der Leistungsdichte gibt es jedoch auch den Ansatz, eine Antriebsmaschine mit höherer maximaler Drehzahl (bis 30.000 rpm) und geringerem maximalem Drehmoment (ca. 140 Nm) einzusetzen. (Reichert, Epskamp, Ott & Radimersky, 2016a) Die Alternativen für das quantitative Modell sind zwei Traktionsmaschinen vergleichbarer Leistung, da

beide dasselbe Nutzungsprofil erfüllen sollen. Die erste Alternative "EM 1" ist eine Traktionsmaschine mit hohem maximalem Drehmoment und einer Maximaldrehzahl von 16.000 rpm, sie bietet das Potenzial auch ohne schaltbares Getriebe einen großen Anteil der Nutzungsszenarien zu erfüllen und dient daher in Kombination mit einer festen Übersetzung auch als Referenzantrieb für die Reichweitenbewertung. Die zweite Alternative "EM 2" ist durch eine hohe maximale Drehzahl bis 30.000 rpm gekennzeichnet, was somit jedoch mit einem niedrigeren Maximalmoment einhergeht. Um alle Nutzungsszenarien abbilden zu können, ist dabei ein Getriebe mit mehreren Gängen notwendig.

Entsprechend der Traktionsmaschine wird auch das Getriebe variiert. Auf Ebene 1 ist das Getriebe durch die Anzahl der Gänge und deren Übersetzungsverhältnisse sowie den Gesamtwirkungsgrad beschrieben. Dabei hängen geeignete Getriebeübersetzungen stark mit der Charakteristik der Traktionsmaschine zusammen. Eine grobe Vorauslegung – z. B. anhand charakteristischer Betriebspunkte – ermöglicht die Wahl der Getriebeparameter für das quantitative Modell. Für „EM 1“ wird zwischen einem Gang ($i=10$), zwei Alternativen mit zwei Gängen ($i_1=15$, $i_2=6$ und $i_1=18$, $i_2=10$) und zwei Alternativen mit drei Gängen ($i_1=15$, $i_2=10$, $i_3=6$ und $i_1=38$, $i_2=18$, $i_3=10$) variiert. Für „EM 2“ sind aufgrund der Charakteristik mit einer einzigen Übersetzung nicht alle betrachteten Fahrsituationen möglich. Daher wird für „EM 2“ zwischen zwei Alternativen mit zwei Gängen ($i_1=29$, $i_2=15$ und $i_1=37$, $i_2=19$) und zwei Alternativen mit drei Gängen ($i_1=38$, $i_2=25$, $i_3=16$ und $i_1=37$, $i_2=25$, $i_3=19$) variiert. Die betrachteten Getriebevarianten werden somit beim quantitativen Vergleich der Systemarchitekturen nicht durch-permutiert, sondern es werden zu jeder Traktionsmaschine vier schaltbare Getriebealternativen betrachtet. Wie oben beschrieben wird zusätzlich die Kombination aus EM 1 und nicht schaltbarem Getriebe als Referenz für die Reichweite mit berechnet. In Abbildung 6.15 ist ein Ausschnitt der technischen Sicht des Getriebes auf Ebene 1 des Architekturmodells dargestellt. Hier sind unterschiedliche Parametrierungen als Getriebeausprägungen erkennbar.

Bei der Detaillierung des Getriebes auf Ebene 2 (Kapitel 6.2.2.2) wird deutlich, dass ein Schaltvorgang durch die Verluste des Aktors und des Schaltelements selbst nicht energieneutral ist. Diese zusätzlichen Verluste unterscheiden sich je nach Antriebssystem, weshalb sie im Vergleich unterschiedlicher Antriebssystemvarianten hinsichtlich Energieeffizienz berücksichtigt werden sollen. Hierfür muss das Getriebekonzept definiert werden, um auch hier durch eine groben Vorauslegung Annahmen zu den Eigenschaften von Schaltaktuatorik und Schaltelementen treffen zu können. Das in diesem Beispiel angenommene Getriebekonzept entspricht dem von Fracchia vorgestellten Konzept. (Fracchia, 2017) Hierbei kommt neben einem schaltbaren Freilauf eine Reibkupplung zum Einsatz, siehe auch Prinzipskizze

rechts oben in Abbildung 6.21. Die Aktuierung des Gangwechsels erzeugt zusätzliche Verlustenergie, die von der Dauer der Schaltung abhängt. Daher wird die Schaltdauer bei Berechnung auf Ebene 2 zwischen den Werten 200 ms, 300 ms und 400 ms variiert. In dieser Betrachtung wird dabei vereinfachend von einer konstanten, situationsunabhängigen Reibungszahl der Kupplung ausgegangen. Im Zuge der Berücksichtigung von Schaltverlusten wird die Schaltstrategie außerdem um den Aspekt der Schalthysterese erweitert. Wie oben erläutert wird in diesem Beispiel eine Zeitabhängigkeit abgebildet. Im Berechnungsmodell wird die minimal zulässige Verweildauer in einer Gangstufe über den Parameter des minimalen Schaltabstands definiert, dieser wird durch die Werte 1 und 10 variiert wobei 1 bedeutet, dass ein Gang mindestens 1s lang eingelegt sein muss, also erst im zweiten Zeitschritt nach einem Gangwechsel ein erneuter Gangwechsel erfolgen kann. Entsprechend bedeutet der Wert 10, dass jeder Gang mindestens 10 s lang eingelegt sein muss, bevor ein erneuter Gangwechsel erfolgen darf. Basis dafür ist die Schaltabfolge aus Ebene 1 entsprechend „best case“ oder „worst case“ je Zeitschritt. Im Berechnungsmodell wird für alle Zeitabschnitte, die nicht der Vorgabe entsprechen der Gangwechsel ignoriert und der zuvor eingelegte Gang verwendet. In Tabelle 6.3 sind die entsprechenden Parameter auf Ebene 2 abgebildet.

Tabelle 6.3: Variierte Parameter im Berechnungsmodell auf Ebene 2

Subsystem	Parameter	Wert 1	Wert 2	Wert 3
Antriebssteuerung	min. Schaltabstand	0 s	1 s	10 s
	Schaltdauer	0,2 s	0,3 s	0,4 s

6.3.2 Analyse verschiedener Systemvarianten und Ableitung von Subsystemanforderungen

Anhand der in Kapitel 6.3.1 vorgestellten Parameter und der resultierenden Simulationsergebnisse wird im Folgenden untersucht, wie das qualitative Systemmodell den Vergleich der Systemvarianten unterstützt.

6.3.2.1 Parametervariation, Vergleich der Varianten und Anforderungsdefinition auf Ebene 1

Die betrachteten Systemvarianten werden zunächst auf dem Detaillierungsgrad von Ebene 1 analysiert. Ziel ist es dabei, eine erste Entwicklungsentscheidung zu ermöglichen, um die Vielfalt der betrachteten Antriebsvarianten einzugrenzen. Als Kriterium dient dabei die potenzielle Reichweitensteigerung der mehrgängigen Antriebsvarianten gegenüber der Antriebsstrangvariante mit fester Übersetzung. Die Referenzwerte für eine angenommene Batteriekapazität von 60 kWh sind Tabelle 6.4 zu entnehmen. Für das in Tabelle 6.1 definierte Nutzungsprofil ergibt sich eine Reichweite von 393 km.

Tabelle 6.4: Referenz für die Reichweitenbetrachtung auf Ebene 1 mit Antriebsvariante EM 1, $i=10$, 94% Wirkungsgrad, 80% Rekuperationsanteil

Anwendungsfall (Fahrzyklus)	Reichweite
Stadtverkehr	420 km
Landstraße	405 km
Stockender Verkehr	467 km
Autobahnfahrt	285 km

Im Folgenden sind die Ergebnisse für die unterschiedlichen Antriebsvarianten im angenommenen Nutzungsprofil und in den vier darin enthaltenen Anwendungsfällen dargestellt. Das Sul ist die Kombination aus Traktionsmaschine und Getriebe. Dabei ist der Wirkungsgrad aller Getriebevarianten mit 94 % vorgegeben, um die Potenziale aufgrund der Übersetzungswahl zu analysieren. Die Variation des Wirkungsgrades zur Ableitung von zu erreichenden Zielwerten für die Getriebeeffizienz erfolgt nach Festlegung von Traktionsmaschine und Übersetzungsverhältnissen.

Abbildung 6.24 stellt die Spanne des potenziellen Reichweiteneinflusses für alle Getriebevarianten der jeweiligen Traktionsmaschine im angenommenen Nutzungsprofil dar. Durch die Berücksichtigung von „best case“ Gangwahl und „worst case“ Gangwahl in der Simulation ist auch das potenzielle Risiko der Reichweitenreduzierung durch ungeeignete Gangwahl berücksichtigt. Für die angenommene Verteilung der Betriebszyklen im Nutzungsprofil bieten die Varianten mit Traktionsmaschine EM 1 die größeren potenziellen Reichweitenverbesserungen.

□ potenzielle Reichweitenverbesserung ■ Risiko durch schlechte Schaltstrategie
 ($\eta_{\text{Getriebe}}=94\%$, Rekuperationsanteil=80%)

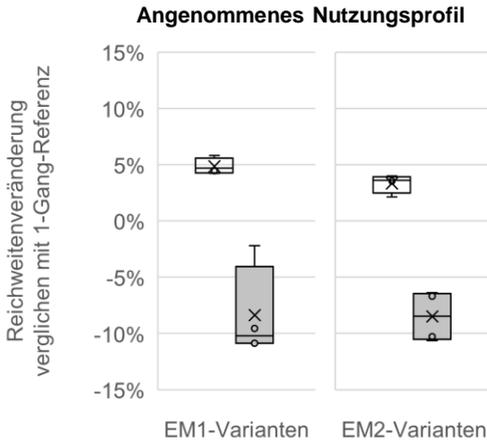


Abbildung 6.24: Reichweitenpotenziale der Antriebsvarianten auf Ebene 1 im angenommenen Nutzungsprofil

In Abbildung 6.25 ist die Spanne des potenziellen Reichweiteneinflusses für alle Antriebsvarianten in den einzelnen Anwendungsfällen dargestellt. Es wird deutlich, dass die Antriebsvarianten mit Traktionsmaschine EM 1 in drei der vier Anwendungsfälle höhere Reichweitenpotenziale im Vergleich zu einer festen Übersetzung bieten als die Antriebsvarianten mit EM 2. Nur im Anwendungsfall „stockender Verkehr“ ist das Potenzial einiger Antriebsvarianten mit EM 2 etwas höher als das Reichweitenpotenzial der EM 1-Varianten. Ebenfalls ersichtlich ist, dass alle Antriebsvarianten mit EM 1 bei schlecht gewählter Schaltstrategie auch ein höheres Risiko bergen, die Reichweite im Vergleich zu einer festen Übersetzung deutlich zu reduzieren.

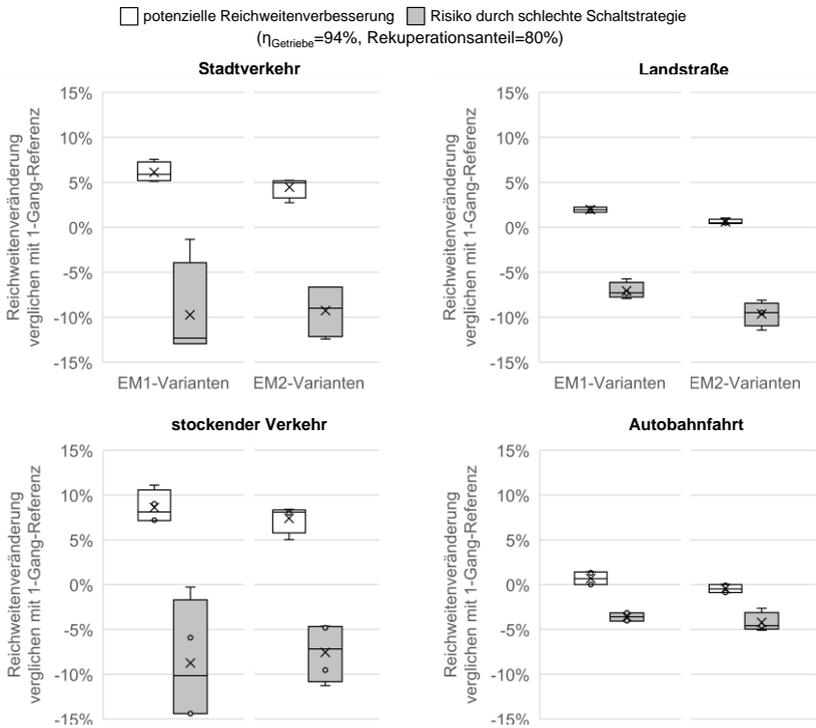


Abbildung 6.25: Reichweitenpotenziale der Antriebsvarianten auf Ebene 1 in den einzelnen Fahrzyklen

Auf Basis dieser Betrachtung wird die Entscheidung getroffen, alle Antriebsvarianten mit Traktionsmaschine EM 2 für die weitere Entwicklung auszuschließen. Im Architekturmodell wird das Analyseergebnis verlinkt und die Entscheidung mittels eines Kommentars dokumentiert. Die Entscheidung wird mit den betroffenen logischen Elementen – zunächst also der Traktionsmaschine - und dem Analyseergebnis verknüpft. Abbildung 6.26 zeigt diese Verknüpfung im Blockdefinitionsdiagramm der technischen Sicht.

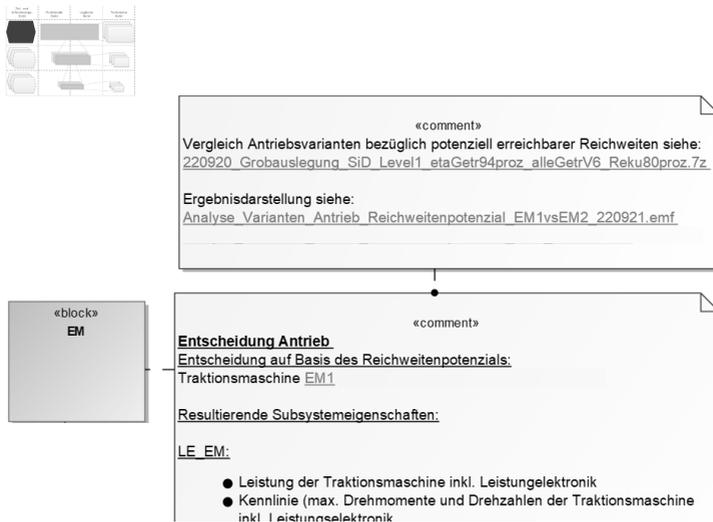


Abbildung 6.26: Verknüpfung von Analyseergebnis und Entscheidung im Architekturmodell am Beispiel Traktionsmaschine.

Sowohl für die Entscheidung als auch für die Analyseergebnisse wird der Elementtyp „Kommentar“ verwendet. Dieser ermöglicht das Einfügen von Hyperlinks auf die Ergebnisdateien des quantitativen Modells und auf die grafischen Darstellungen der Ergebnisse. So ist ein Überblick über die Simulationsergebnisse direkt aus dem qualitativen Systemmodell heraus möglich. Ebenso lassen sich im Kommentar Modellelemente verknüpfen, in diesem Beispiel die Traktionsmaschinen-Instanz „EM1“.

Im nächsten Schritt wird eine weitere Variantenreduktion vorgenommen, indem die unterschiedlichen Getriebevarianten für Traktionsmaschine EM 1 in den Anwendungsfällen miteinander verglichen werden. Das Sul ist in diesem Fall also das Getriebe.

Auch diese Betrachtung erfolgt zunächst vor dem Hintergrund des angenommenen Nutzungsprofils, siehe Abbildung 6.27. Die potenzielle Reichweitenverbesserung der Getriebevarianten liegt im Nutzungsprofil nah beieinander. Dennoch ist eine Abstufung erkennbar. Das dreigängige Getriebe mit den Übersetzungsverhältnissen [38, 18, 10] bietet das größte Potenzial, gefolgt vom zweigängigen Getriebe mit den Übersetzungen [18, 10], dem dreigängigen Getriebe mit den Übersetzungen [15, 10, 6] und dem zweigängigen Getriebe mit den Übersetzungen [15, 6].

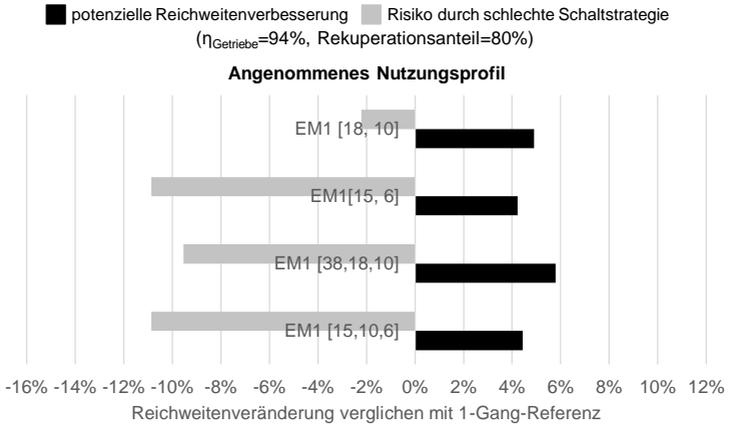


Abbildung 6.27: Vergleich der EM 1-Antriebsvarianten bezüglich Reichweitenpotenzial gegenüber 1-Gang-Referenz im angenommenen Nutzungsprofil

Abbildung 6.28 zeigt die potenziellen Reichweitenveränderungen bezüglich der einzelnen Anwendungsfälle. In den Anwendungsfällen „Stadtverkehr“ und „stockender Verkehr“ ermöglichen die Getriebevarianten [18, 10] und [38, 18, 10] die höchste erzielbare Reichweite. Die Getriebevarianten [15, 6] und [15, 10, 6] bieten dahingegen in den Anwendungsfällen „Landstraße“ und „Autobahn“ einen potenziellen Reichweitenvorteil. Die Differenz zwischen der maximalen Reichweite, die ein dreigängiges Getriebe ermöglicht und der maximalen Reichweite, die ein zweigängiges Getriebe ermöglicht ist bei gleichem Getriebewirkungsgrad 2,1% im Anwendungsfall „stockender Verkehr“ zwischen den Varianten [18, 10] und [38, 18, 10].

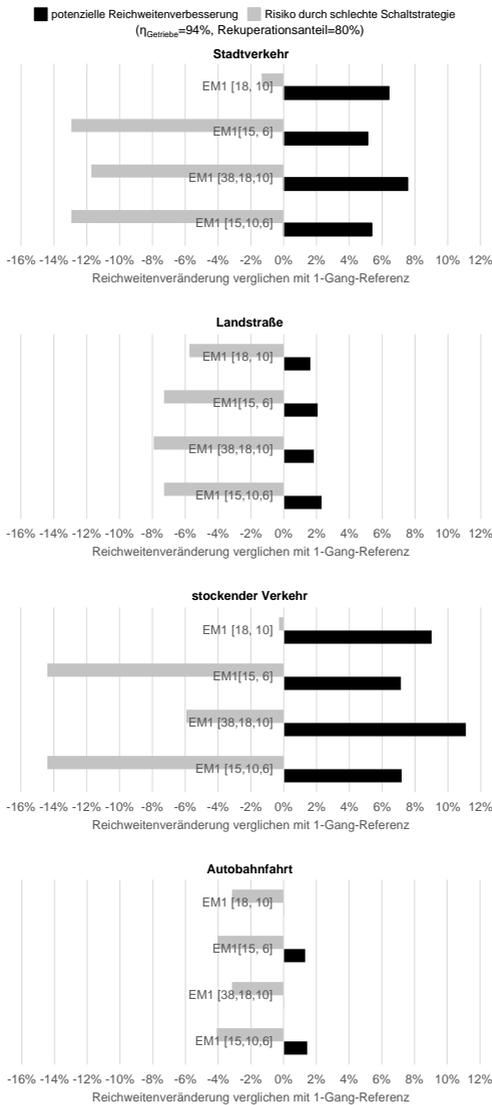


Abbildung 6.28: Vergleich der EM 1-Antriebsvarianten bezüglich Reichweitenpotenzial gegenüber 1-Gang-Referenz in den einzelnen Fahrzyklen.

Aus diesen Erkenntnissen lässt sich schlussfolgern, dass ein dreigängiges Getriebe – unter Berücksichtigung des zu erwartenden real erreichbaren Wirkungsgrades – in diesem Fall kein Vorteil gegenüber einem zweigängigen Getriebe darstellt. Im Rahmen der Varianteneingrenzung wird somit auf die zweigängigen Getriebe fokussiert, von diesen wird die Übersetzungsvariante [18, 10] gewählt. In der technischen Sicht des Architekturmodells ist diese Getriebevariante mit „Schaltgetriebe_3“ bezeichnet. Der Entscheidungskommentar wird im Reiter „Description“ entsprechend erweitert und zusätzlich mit dem logischen Element „Getriebe“ verbunden, siehe Abbildung 6.29.

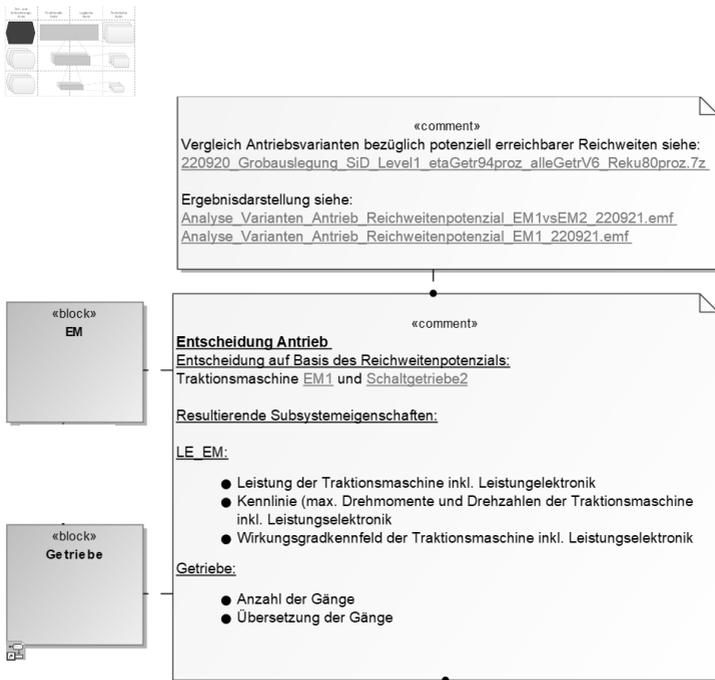


Abbildung 6.29: Ergänzung der Entscheidungsdokumentation durch Hinzufügen von beschreibendem Text und zusätzlicher Anbindung des logischen Elements „Getriebe“ an den Kommentar.

Zusätzliche Getriebestufen und Schaltelemente führen in der Regel zu einem schlechteren Wirkungsgrad eines schaltbaren Getriebes verglichen mit einem fest

übersetzenden Getriebe gleicher Leistungsklasse. Der Wirkungsgrad eines Getriebes kann in der Entwicklung durch Aspekte wie Getriebekonzept, Schmier-konzept, Wahl der Schaltelemente, etc. optimiert werden. Dies ist Bestandteil der Detailauslegung. Dennoch kann durch das Berechnungsmodell der zu erreichende Zielwert für den Getriebewirkungsgrad unter dem Aspekt minimalen Reichweiteneinflusses abgeleitet werden. Auf Detaillierungsgrad Ebene 1 wird daher nun für die gewählte Antriebsvariante der Wirkungsgrad variiert. Abbildung 6.30 zeigt die erzielbaren Reichweitenpotenziale mit verschiedenen Getriebewirkungsgraden der gewählten Antriebsvariante. Die prozentualen Anteile sind auch hier bezogen auf die Referenzreichweiten in Tabelle 6.4. In den Anwendungsfällen „Stadtverkehr“ und „stockender Verkehr“ sind bei geeigneter Schaltstrategie noch bis zu einem Getriebewirkungsgrad von 91 % bzw. 90 % positive Reichweiteneffekte möglich. Im Anwendungsfall „Landstraße“ ist ab 92 % Getriebewirkungsgrad kein Reichweitenvorteil erzielbar. Im Anwendungsfall „Autobahn“ führt das gewählte Getriebe wie bereits in Abbildung 6.30 ersichtlich nicht zu potenziell höheren Reichweiten.

Für die Entwicklung der gewählten Antriebsvariante wird für den Getriebewirkungsgrad ein Zielwert von 92% angesetzt. Dies berücksichtigt, dass die Betrachtung auf Ebene 1 noch keine fahrsituationsspezifischen Schaltverluste mit einbezieht. Schaltverluste werden entsprechend Kapitel 6.3.1.2 im Berechnungsmodell auf Ebene 2 in die Betrachtung mit einbezogen.

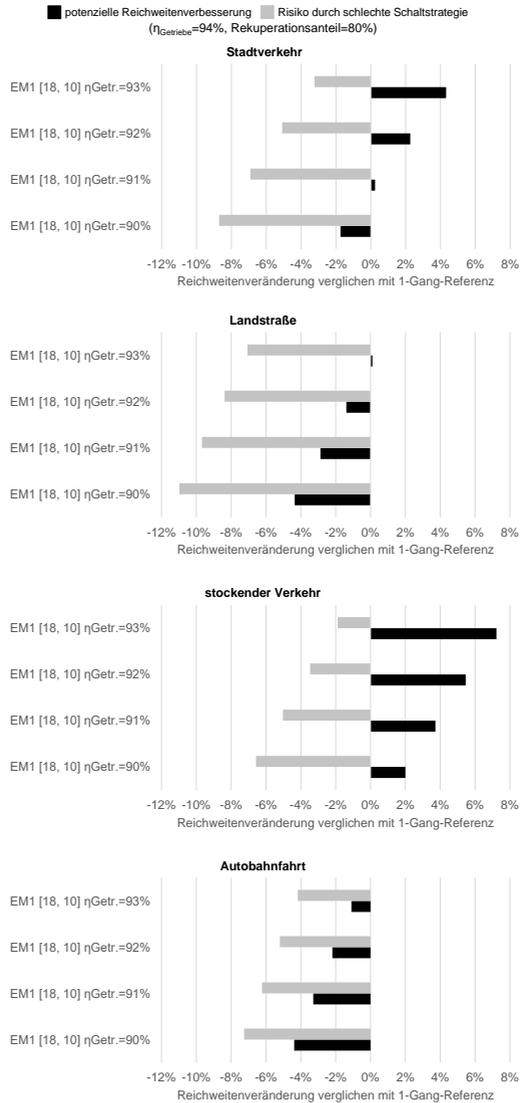


Abbildung 6.30: Reichweitenpotenziale der Antriebsvariante EM 1 [18, 10] bei unterschiedlichen Getriebewirkungsgraden in den einzelnen Fahrzyklen.

Diese weitere Konkretisierung der Entscheidung für eine Antriebsvariante wird ebenfalls durch Erweiterung des entsprechenden Kommentars im Architekturmodell dokumentiert. Der Ziel-Wirkungsgrad des Getriebes wird im Text als weitere resultierende Subsystemeigenschaft hinzugefügt, siehe Abbildung 6.31.

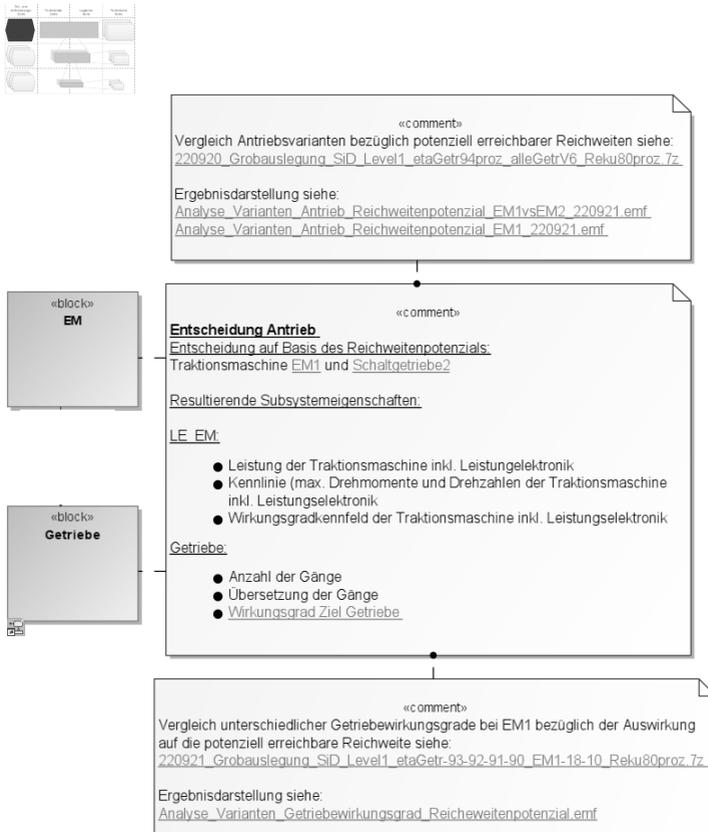


Abbildung 6.31: Ergänzung der Entscheidungsdokumentation durch Konkretisieren des Textes und zusätzlicher Anbindung der Analyseergebnisse der Wirkungsgradvariation an den Kommentar.

Die Variation des Wirkungsgrades für die gewählte Antriebsvariante im Berechnungsmodell erfolgt zwar auf Ebene eins mit der entsprechenden Modellbildung, wird aber nur für die gewählte Antriebsvariante durchgeführt. Daher kann dieser Schritt als eigenstehende Analyse mit Ergebnisdarstellung im Architekturmodell abgebildet werden. Die Verbindungen des Entscheidungs-Elements werden entsprechend um eine Verknüpfung zum Analyseergebnis der Parametervariation des Wirkungsgrads ergänzt. Die im Laufe des Entscheidungsprozesses hinzugefügten Verknüpfungen zwischen den Analyseergebnissen, der Antriebsentscheidung und dem logischen Element Getriebe sind in Abbildung 6.31 dargestellt.

Die logischen Elemente des Architekturmodells und deren Funktionen sind über einen „satisfy“-Link mit den zugehörigen Anforderungselementen verknüpft. So entsteht eine nachvollziehbare Verbindung zwischen den Ergebnissen des Berechnungsmodells und den durch die Analysen und Entscheidungen abgeleiteten Anforderungen. Abbildung 6.32 zeigt diese Verbindung für die quantitativen Ergebnisse auf Ebene 1 in einem Anforderungsdiagramm auf.

Auf Ebene 1 können durch das Berechnungsmodell unterschiedliche Systemvarianten hinsichtlich des Kriteriums Reichweite verglichen werden. Der gleichartige Aufbau von Architekturmodell und Berechnungsmodell hilft dabei, die Analyseergebnisse im Architekturmodell eindeutig zuzuordnen und die getroffenen Entscheidungen nachvollziehbar zu dokumentieren. Die eindeutige Zuordnung zu den zugrundeliegenden Daten wird durch die Verwendung von Hyperlinks zu den Ergebnissen sichergestellt.

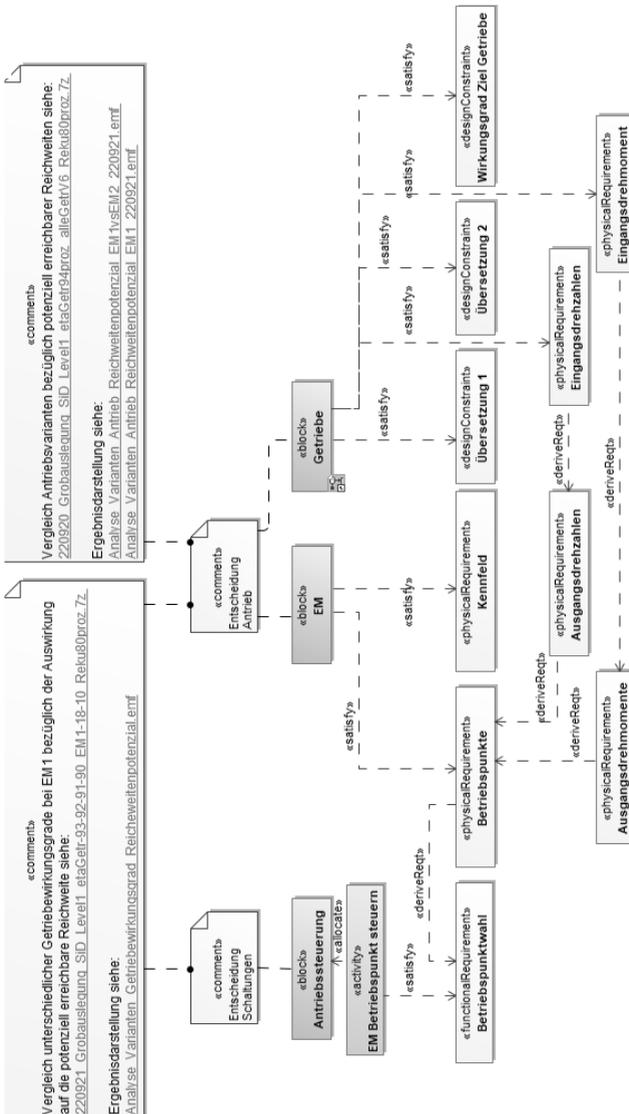


Abbildung 6.32: Verknüpfung zwischen den Ergebnissen des Berechnungsmodells und den Elementen des Systemarchitekturmodells.

6.3.2.2 Parametervariation, Vergleich der Varianten und Anforderungsdefinition auf Ebene 2

Die Antriebssystemvariante EM 1 [18, 10] wird unter Annahme eines Getriebewirkungsgrads von 92 % nun auf Ebene 2 im Berechnungsmodell analysiert. Ziel ist es dabei zum einen, die Auswirkung der Schaltverluste auf die erreichbare Reichweite abzuschätzen und zum anderen, Entscheidungen für den Ablauf der Schaltvorgänge zu treffen, sodass auf Anforderungen an die Subsysteme Schaltelement und Schaltaktuator abgeleitet werden können, diese stellen damit das Sul in dieser Analyse dar.

Wie in Kapitel 6.3.1.2 erläutert, werden auf Ebene 2 zwei weitere Parameter eingeführt: Die Dauer des Schaltvorgangs und der zeitliche Mindestabstand zwischen zwei Schaltungen. Die Einführung der Schaltungsdauer ermöglicht die Berechnung der Kupplungsverluste beim Schalten sowie der benötigten Leistung des Schaltactuators (vgl. Kapitel 6.2.2.2). Durch die Vorgabe unterschiedlicher Mindestabstände zwischen den Schaltvorgängen, kann die Auswirkung einer reduzierten Anzahl an Schaltvorgängen auf die Reichweite und auf das Belastungskollektiv der Schaltelemente und Schaltaktuatoren analysiert werden. Ziel ist die Belastung sowohl des Schaltelements als auch des Schaltactuators möglichst gering zu halten. Betrachtet werden daher die spezifische Reibleistung und der spezifische Energieeintrag in das Schaltelement sowie die erforderliche Leistung des Schaltactuators.

Zur besseren Vergleichbarkeit sind die in Abbildung 6.33, Abbildung 6.34 und Abbildung 6.35 dargestellten Ergebnisse je Anwendungsfall normiert und durch Kennzahlen bewertet. Die Bewertungsskala umfasst die Ziffern eins bis fünf wobei innerhalb jeder Kategorie niedrige Bewertungen schlechter sind als hohe Bewertungen. Eine hohe Bewertung des Verbrauchs bedeutet also z. B. einen niedrigen Verbrauch im Vergleich zu anderen Parametrierungen im gleichen Anwendungsfall.

Abbildung 6.33 gibt einen Überblick über die Auswirkung von Schaltdauern und Schaltabständen im Falle effizienzoptimaler (links) und suboptimaler (rechts) Gangwahl im angenommenen Nutzungsprofil. Deutlich ist, dass auf Ebene des Nutzungsprofils im Wesentlichen die unterschiedlichen Schaltdauern Einfluss haben. Im Folgenden werden daher zusätzlich die verschiedenen Fahrsituationen des Nutzungsprofils im Einzelnen betrachtet.

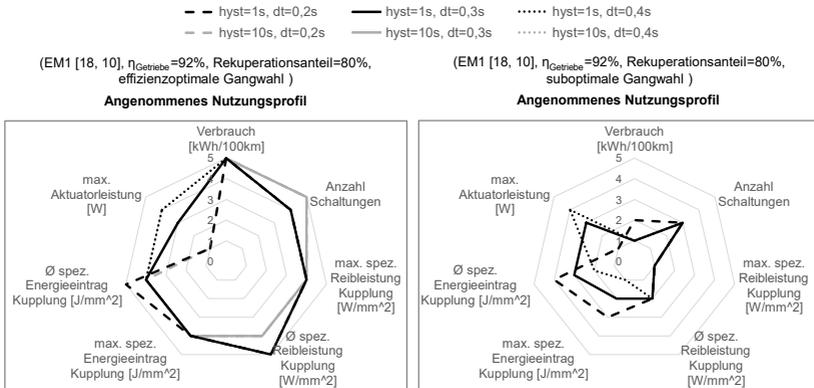


Abbildung 6.33: Vergleich der Auswirkung unterschiedlicher Schaltabstände (hyst) und Schaltdauern (dt) im angenommenen Nutzungsprofil.

In Abbildung 6.34 sind die Ergebnisse für unterschiedlichen Schaltabstände und Schaltdauern bei effizienzoptimaler Gangwahl dargestellt, in Abbildung 6.35 die entsprechenden Ergebnisse für schlechte Gangwahl hinsichtlich der Effizienz. Es ist deutlich, dass die Verbrauchswerte maßgeblich von der Gangwahl beeinflusst werden und der Reichweiteneffekt von Schaltdauer und Schaltungsabstand im Vergleich dazu gering ist.

Im Anwendungsfall „Autobahnfahrt“ wird bei effizienzoptimaler Schaltstrategie nicht geschaltet, daher liegen keine Energieeinträge bzw. Reibleistungen der Kupplung in diesem Fall vor und die Kriterien sind jeweils maximal bewertet. In den Anwendungsfällen „Stadtverkehr“, „Landstraße“ und „stockender Verkehr“ führt die suboptimale Gangwahl zu geringeren Maximalwerten des spezifischen Energieeintrags und der spezifischen Reibleistung der Kupplung. Dieser Effekt ist auf die Schaltung bei niedrigeren – und damit meist ineffizienteren – Drehmomenten der Traktionsmaschine zurückzuführen. Da die Reichweite des E-Fahrzeugs eine signifikante Rolle in der Nutzerakzeptanz spielt, wird die Annahme für minimalen Schaltabstand und Schaltdauer auf Basis der Ergebnisse bei effizienzoptimaler Gangwahl getroffen.

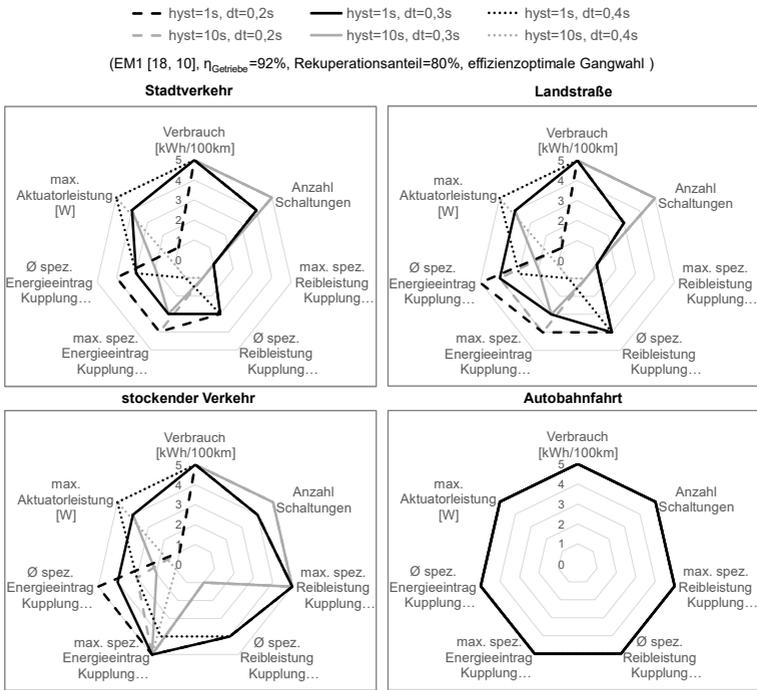


Abbildung 6.34: Vergleich der Auswirkung unterschiedlicher Schaltabstände (hyst) und Schaltungsdauern (dt) für effizienzoptimale Gangwahl in den einzelnen Fahrzyklen.

Den besten Kompromiss zwischen möglichst geringer Aktuatorleistung und möglichst geringen Reibleistungen sowie Energieeinträgen in der Kupplung stellt entsprechend der Ergebnisse ein Mindestschaltabstand von 1 s und eine Schaltungsdauer von 300 ms dar. Unter diesen Randbedingungen werden die Anforderungen an die Subsysteme des Antriebssystems abgeleitet.

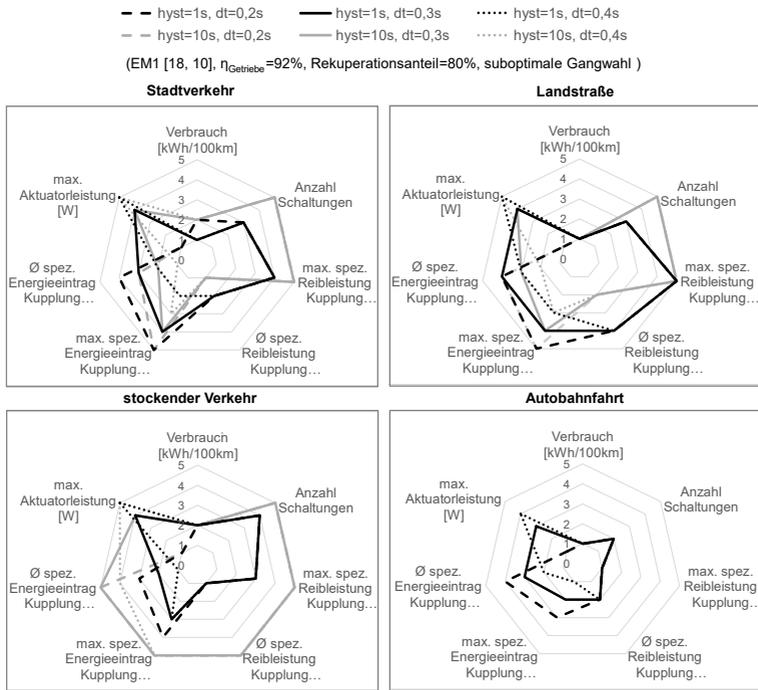


Abbildung 6.35: Vergleich der Auswirkung unterschiedlicher Schaltabstände (hyst) und Schaltungsdauern (dt) für suboptimale Gangwahl in den einzelnen Fahrzyklen.

Auf Basis der Entscheidungen auf Ebene 1 zum Antrieb inkl. des Getriebepinzips des Referenzgetriebes wird das Berechnungsmodell entsprechend Kapitel 6.2.2.2 mit einer Grobdimensionierung des Schaltelements Reibkupplung (Dimensionen, Reibwert, Kennlinie) und des Schaltmechanismus (Getriebekonzept, Übersetzungsverhältnisse) parametrisiert. Die abgeleiteten Belastungen im Rahmen der Analyse auf Ebene 2 ermöglichen eine iterative Anpassung der Größen. Demensprechend führen die Analyseergebnisse sowohl zu einer Konkretisierung der Schaltstrategie als auch zu einer Konkretisierung des Getriebekonzepts und der resultierenden Anforderungen. Abbildung 6.36 stellt einen Ausschnitt der entsprechenden Elemente des Architekturmodells und deren Verknüpfungen dar.

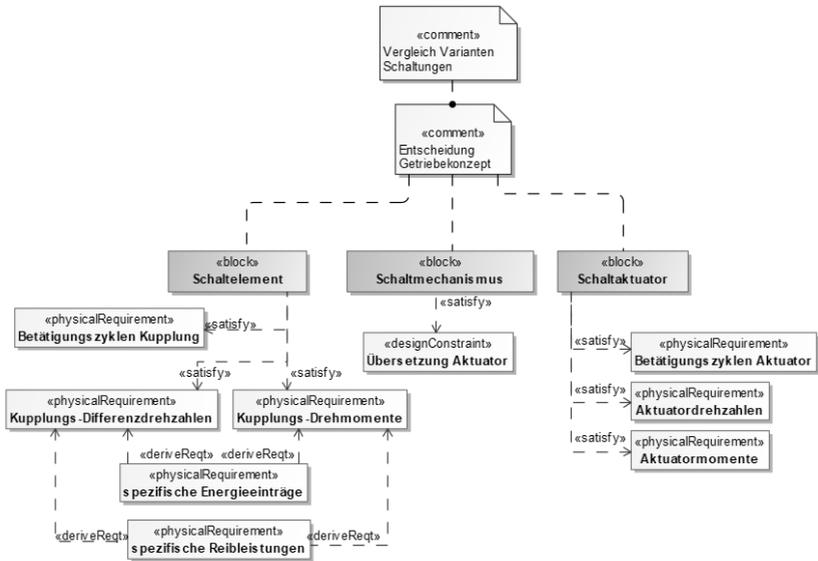


Abbildung 6.36: Verknüpfung zwischen Analyseergebnissen auf Ebene zwei, Entscheidungen und Subsystemanforderungen im Architekturmodell.

Der quantitative Variantenvergleich auf Ebene 2 zeigt, dass auch hier die Ergebnisse durch die gleichartige Struktur der beiden Systemmodelle gut im qualitativen Modell dokumentiert werden kann.

Wie in den Kapiteln 6.3.2.1 und 6.3.2.2 beschrieben, werden Anforderungen an die in der Systemarchitektur in Form von logischen Elementen abgebildeten Subsysteme abgeleitet und im Systemarchitekturmodell repräsentiert. Je nach verwendeter Entwicklungsumgebung kommt heutzutage üblicherweise ein gesondertes Anforderungsmanagement-Programm zum Einsatz, beispielsweise Doors Classic, Doors Next Generation, Polarion oder ähnliches (IBM, 2021; Siemens Product Lifecycle Management Software Inc., 2022a), siehe auch Kapitel 2.2.2. Zwischen dem Architekturmodell und dem Anforderungsmanagement-Programm können zumeist Verknüpfungen hergestellt werden, sodass die Anforderungen im Modell nicht dupliziert werden müssen, sondern direkt referenziert sind. In der vorliegenden Arbeit wird die Möglichkeit genutzt, Anforderungen direkt im Architekturmodell als gesonderte Elemente vom Typ „Anforderung“ abzubilden.

Die Anforderungselemente sind in der Anforderungssicht des Architekturmodells verortet und über „satisfy“-Beziehungen mit den entsprechenden logischen Elementen verknüpft, siehe auch Abbildung 6.36. Wie in den Modellelementen zur Dokumentation der Analyseergebnisse, werden auch in den Anforderungselementen im Reiter „Description“ Hyperlinks zu den Ergebnissen des quantitativen Systemmodells eingesetzt. So ist auch hier eindeutig nachvollziehbar, auf Basis welcher Parametrierung und Randbedingungen die jeweilige Anforderung abgeleitet ist. Die Anforderungen sind zusätzlich untereinander durch „derive Requirement“-Beziehungen verknüpft, um die Abhängigkeit voneinander abzubilden. Im Reiter „Relations“ sind alle modellinternen Verlinkungen des Anforderungselements aufgelistet. Abbildung 6.37 zeigt das Anforderungselement „Betriebspunkte“ der Traktionsmaschine und dessen Inhalte sowie die verlinkte Grafik. Bei Einsatz eines entsprechenden Anforderungsmanagementsystems ist davon auszugehen, dass Grafiken direkt im Anforderungselement verankert und dargestellt werden.

Aus den Analysen im Berechnungsmodell ergeben sich Anforderungen an Subsysteme auf allen Ebenen des Architekturmodells. Durch den Aufbau des Berechnungsmodells auf Basis des im Architekturmodell abgebildeten Funktionsverständnis lassen sich die abgeleiteten Anforderungen nachvollziehbar inklusive der Randbedingungen und Entwicklungsentscheidungen, unter denen sie gelten, dokumentieren. Die Änderung von Randbedingungen oder Entscheidungen können nicht nur im Architekturmodell, sondern auch im Berechnungsmodell einfach abgebildet und die resultierenden Anforderungen dementsprechend aktualisiert werden.

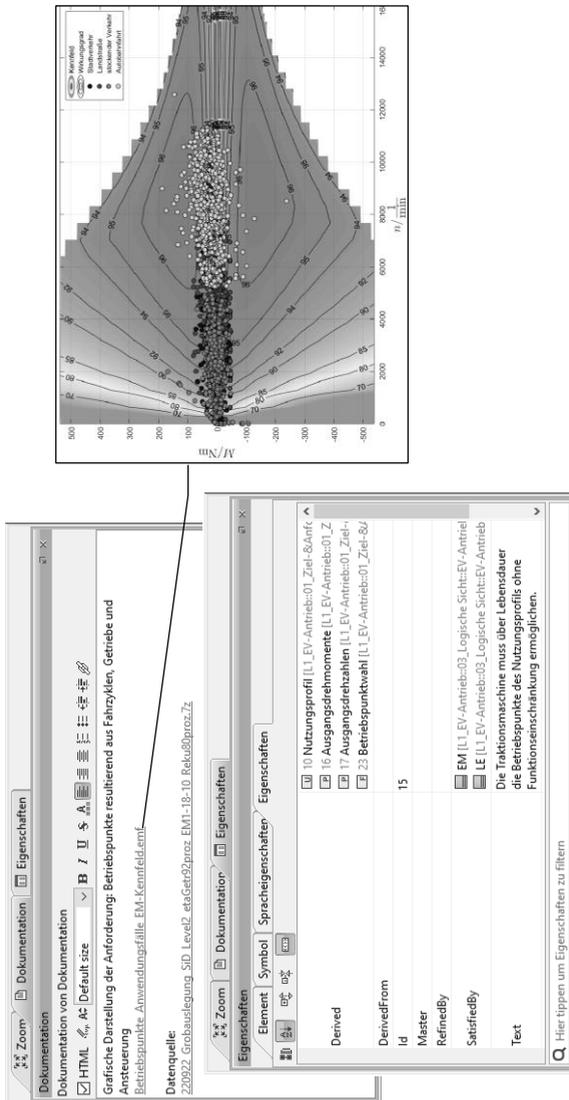


Abbildung 6.37: Modellelement der Anforderung „Betriebspunkte der Traktionsmaschine“ im Architekturmodell.

7 Bewertung der Implementierung des Ansatzes auf Basis der experimentellen Anwendung

In Kapitel 6 wird der vorgeschlagene Ansatz zur Nutzung des Architekturmodells als Basis für eine quantitative Modellierung am Beispiel ausgewählter Aspekte eines batterieelektrischen Antriebssystems durch die Verfasserin dieser Arbeit angewendet. Die in diesem Rahmen gemachten Erfahrungen und Beobachtungen werden nun zur Bewertung der Potenziale und Risiken des Ansatzes herangezogen.

Eine vollständige Evaluation durch Dritte ist aufgrund der Randbedingungen der vorliegenden Arbeit nicht möglich. Dennoch können auf Basis der Anwendung des Ansatzes durch die Verfasserin selbst im Folgenden Indikationen hinsichtlich des möglichen Einflusses auf Akzeptanz und Nutzung von Systemarchitekturmodellen abgeleitet werden. Hierfür wird wieder das in Kapitel 4.1 eingeführte Bewertungsschema entsprechend Abbildung 4.1 genutzt, das die Bewertung in die Aspekte Prozess, Präsentation und Performanz auf individueller und organisatorischer Ebene strukturiert. Außerdem werden die in Kapitel 4.4 identifizierten Faktoren berücksichtigt, die die Akzeptanz und Nutzung von MBSE behindern: Das Aufwand/Nutzen Verhältnis der Architekturmodellerstellung wird als sehr hoch empfunden. Die Arbeit mit Berechnungsmodellen wird nicht unterstützt. Die inhaltliche Nachvollziehbarkeit abgeleiteter Anforderungen wird nicht unterstützt.

7.1 Indikation des Einflusses auf Akzeptanz und Nutzung von Systemarchitekturmodellen

Der Aspekt „**Prozess**“ beinhaltet, wie flexibel und anpassbar ein Ansatz für die individuellen Teammitglieder ist. Der hier vorgestellte Ansatz zeigt dabei das Potenzial, flexibel an die aktuellen Projektbedarfe angepasst zu werden. Das betrifft z. B. die Detaillierungstiefe im Architekturmodell, die - wie in dieser Arbeit am Beispiel Getriebe in Kapitel 6.2.2 gezeigt - für einzelne Subsysteme unterschiedlich tief sein kann. Je nach Projektfokus können also einzelne Elemente genauer beschrieben und analysiert werden als andere, ohne den Gesamtzusammenhang zu verlieren. Die klare Strukturierung von Architekturmodell und Berechnungsmodell kann die einfachere manuelle Übernahme von Informationen aus dem Architekturmodell ermöglichen. Gleichzeitig bietet die Identifikation der zu übernehmenden Faktoren

eine Grundlage für eine zukünftige Automatisierung des Informations- und Datenaustausch zwischen den Modellen. Dadurch kann das Architekturmodell **die Arbeit mit Berechnungsmodellen unterstützen**. Auch die Erweiterung um zusätzliche Modelle, z. B. mit anderem Analyseziel, und Methoden wie beispielsweise FMEA ist je nach Bedarf in der aktuellen Entwicklungssituation möglich. Deutlich wird in der Anwendung jedoch auch, dass der Wechsel zwischen verschiedenen Softwaretools für die Architekturmodellierung und die Erstellung des Berechnungsmodells das Risiko eines hohen Pflegeaufwandes mit sich bringt. Insbesondere bei zunehmendem Detaillierungsgrad der Modelle steigt der Aufwand für Dokumentation und Verlinkung. Diesem Risiko kann durch eine höhere Automatisierung der Synchronisation zwischen verschiedenen Softwaretools entgegengewirkt werden. Die Grundlage dafür kann der Ansatz legen indem zu übernehmende Elemente und Parameter systematisch anhand des Architekturmodells identifiziert werden. Aus organisatorischer Sicht beinhaltet „Prozess“ die Wiederverwendbarkeit und Erweiterbarkeit. Der Ansatz bietet das Potenzial, das Architekturmodell besser in den Entwicklungsprozess zu integrieren, indem es als zentrale Informationsquelle für die Erstellung des Berechnungsmodells verwendet wird. Die daraus folgende mögliche aktivere Nutzung der Systemarchitektur kann auch zur besseren Ausnutzung der potenziellen Erweiterbarkeit des Ansatzes um die Verknüpfung zu weiteren Methoden und Prozessen führen. Ergänzungen um zusätzliche Elemente – wie hier am Beispiel der Dokumentation von Entscheidungen und Verknüpfung von Daten im Modell gezeigt - machen den Ansatz ebenfalls erweiterbar und damit anpassbar auf die jeweilige Projektsituation. Gleichzeitig kann dadurch auch potenziell die **inhaltliche Nachvollziehbarkeit der abgeleiteten Anforderungen verbessert** werden.

Unter den Aspekt „**Präsentation**“ fällt aus individueller Sicht die Frage, ob der Ansatz intuitiv handhabbar ist. Wie in Kapitel 6.2.2.1 aufgezeigt, sind zwar die Detaillierungsebenen und die darin enthaltenen Subsysteme von Architekturmodell und Berechnungsmodell vergleichbar, jedoch unterscheidet sich aufgrund des Modellzwecks die Art der Modellbildung. Während das Architekturmodell den Fahrer in seiner Funktion als regelndes Element berücksichtigt und darstellt, wird zum Zweck der Effizienzberechnung angenommen, dass der Fahrerwunsch durch das Geschwindigkeitsprofil und damit durch den Leistungsbedarf an der Achse direkt repräsentiert ist. Im Berechnungsmodell wird von dort aus auf die notwendigen Betriebspunkte des Antriebssystems bei verschiedenen Übersetzungen zurückgerechnet. Dadurch ist der Leistungsbedarf an der Achse im Berechnungsmodell eine Eingangsgröße der Berechnungsfunktion und die Betriebspunkte sind die Ausgangsgrößen. In der Architekturdarstellung für das allgemeine Funktionsverständnis sind Ein- und Ausgangsgrößen am Getriebe genau umgekehrt dargestellt. Dies lässt sich auch an Abbildung 6.14 erkennen. An dieser Stelle wird das Risiko deutlich, dass ein Entwicklungsteam intuitiv versuchen könnte, Architekturmodell

und Berechnungsmodell möglichst genau aneinander anzupassen, also z. B. auch Flussrichtungen aus dem Architekturmodell dogmatisch in andere Modelle übertragen, sodass die quantitativen Modelle in Verkürzung und Pragmatismus nicht mehr an ihren eigentlichen Zweck angepasst sind. Das würde ineffizientes Arbeiten und nicht zielführende quantitative Modelle mit sich bringen. Ist das Risiko bekannt, kann ein positiver Effekt daraus sein, dass eine bewusstere Modellierung der quantitativen Modelle bezüglich deren Einsatzzweck erfolgt und auch die Einsatzgrenzen des Modells für das Team klarer werden. Aus organisatorischer Sicht stellt sich hinsichtlich der „Präsentation“ die Frage, ob der Ansatz leicht lehr- und lernbar ist. Hierbei ist als Potenzial zu vermerken, dass der Ansatz eine Kombination bereits existierender, teilweise etablierter und z. B. aufgrund von Prozessvorgaben notwendigen Methoden beruht. Die zusätzlich zu beachtenden Punkte wie klare Strukturierung der Modelle, Einführung von Kommentaren zur Verlinkung von Daten und Entscheidungsdokumentation etc. sind eher einfach zu vermitteln. Hinsichtlich dieses Aspekts ist aus organisatorischer Sicht also wenig Risiko zu erwarten. Mehr Nachteile hinsichtlich Lehr- und Lernbarkeit haben voraussichtlich die komplexen Einzelmethoden wie z. B. Architekturmodellierung. Allerdings birgt der Ansatz das Potenzial eines höheren wahrgenommenen Nutzens der einzelnen Methoden – insbesondere der Architekturmodellierung. Das kann sich auf positiv auf Lehr- und Lernbarkeit auswirken.

„**Performanz**“ bedeutet für individuelle Teammitglieder wahrgenommene Leistungsfähigkeit. Die Ausführungen in Kapitel 6.2 zeigen, dass die Strukturierung des Berechnungsmodells anhand des Architekturmodells erfolgen kann. Unterschiedliche Detaillierungstiefen einzelner Subsysteme im quantitativen Modell können anhand der Systemarchitektur nachvollzogen, erklärt und dokumentiert werden. Der Ansatz bietet dadurch Potenzial zu einer besseren Konsistenz zwischen unterschiedlichen Entwicklungsergebnissen. Gegebene Randbedingungen – z. B. der Einsatz bereits vorhandener Subsysteme – können in der Systemarchitektur in direkter Verknüpfung zum betroffenen Subsystem dokumentiert und damit voraussichtlich im Berechnungsmodell einfacher korrekt berücksichtigt werden. Des Weiteren ermöglicht die Systemarchitektur die Darstellung von Spezifikations- und Gestaltzusammenhängen und kann das Entwicklungsteam dadurch unterstützen, die technischen Zusammenhänge zwischen den Anforderungen klarer darzustellen. Zur quantitativen Ableitung der Anforderungen kommt das Berechnungsmodell zum Einsatz. Kapitel 6.3.2 zeigt, dass Entscheidungen und Anforderungen, die auf Basis der Ergebnisse des Berechnungsmodells abgeleitet wurden, durch die Strukturierung der Modelle im Architekturmodell verankert werden können. Die gleiche Strukturierung beider Modelle bietet damit das Potenzial, Spezifikationen auch aus unterschiedlichen Quellen klar und konsistent abzuleiten und über das Architekturmodell zu verknüpfen und zugänglich zu machen. Da Entscheidungen

und Ergebnisse im Architekturmodell verlinkt sind, kann dieses zum zentralen Zugangspunkt für dieses Wissen werden. Die Nutzung als zentrale Informationsquelle im Arbeitsalltag wird dadurch für die Teammitglieder potenziell vereinfacht. **Der Aufwand für die Erstellung des Architekturmodells kann damit einer wahrgenommenen Leistungsfähigkeit gegenüberstehen, die über ein besseres Systemverständnis und bessere Kommunikation hinausgeht.** Organisatorisch bedeutet „Performanz“ das Verhältnis von Aufwand zu Nutzen. Der Aufwand für die Erstellung der einzelnen Modelle ist hoch, würde jedoch auch ohne bewusste Harmonisierung beider Modelle anfallen. Der Zusatzaufwand, um das quantitative Modell in seinen Ebenen und Modellelementen entsprechend dem Architekturmodell zu strukturieren kommt darüber hinaus noch hinzu. Durch die gleiche Strukturierung, kombiniert mit den Verlinkungen zwischen Architekturmodell und Ergebnissen des Berechnungsmodells hat der Ansatz jedoch das Potenzial, den Wissenstransfer bei wechselnden Teams oder Teammitgliedern zu unterstützen, indem die quantifizierten Argumentationen, Grafiken und Entscheidungen direkt im Architekturmodell ersichtlich und zugänglich sind. Des Weiteren kann der Ansatz helfen, die Grenzen des Gestaltungsraums durch Unterscheidung von Kontext, SiD und Sul klar zu definieren. Für den Vergleich unterschiedlicher Systemvarianten ist damit eindeutig, welche Elemente des quantitativen Modells dem Anforderungsraum zuzuordnen sind und damit nicht als Teil alternativer Systemausprägungen variiert werden dürfen.

Zusammenfassend bietet der Ansatz das Potenzial, die Systemarchitektur durch die Verknüpfung mit weiteren Methoden besser im Entwicklungsprozess zu nutzen. Das kann sich positiv auf den Faktor „unverhältnismäßiges Aufwand/Nutzen Verhältnis der Architekturmodellierung“ auswirken. Am hier aufgezeigten Beispiel der Verknüpfung mit einem Berechnungsmodell zeigt sich das Potenzial, quantitative Anforderungen innerhalb eines Systems untereinander und zu den Randbedingungen konsistent abzuleiten und nicht nur nachverfolgbar, sondern auch inhaltlich nachvollziehbar zu dokumentieren. Das kann dem entsprechenden Hinderungsfaktor direkt entgegenwirken. Ein weiteres Potenzial ist der verbesserte Wissenstransfer im Team oder zu neuen Teammitgliedern, indem das Architekturmodell als zentrale Informationsquelle – als Rückgrat - des zu entwickelnden Produktes den Einstiegspunkt für alle relevanten Daten und Zusammenhänge bilden kann. Gleichzeitig ist das Modell, aus dem die Daten erhoben werden, in seiner Struktur und den Elementen konsistent zum Architekturmodell. Das macht zum einen die Systemzusammenhänge auch im Berechnungsmodell potenziell besser nachvollziehbar, zum anderen kann es die manuelle Informationsübernahme vereinfachen und Grundlage für eine spätere Automatisierung sein. Das kann die aktive Nutzung der Systemarchitektur zur Erstellung eines Berechnungsmodells ermöglichen und adressiert damit den entsprechenden Hinderungsfaktor. Risiken des Ansatzes bestehen im

wachsenden Aufwand für die Erstellung der Verknüpfungen zwischen Architekturmodell und anderen Tools und Datensätzen sowie in einer Fehlinterpretation der Nutzung des Architekturmodells, sodass andere Modelle nicht mehr zweckorientiert gebildet werden. Dem Aufwand könnte zukünftig durch geschickte Tool-Synchronisierung entgegengewirkt werden. Die nicht zweckmäßige Modellbildung kann durch bewusste Abgrenzung zwischen den Modellen in ein Potenzial zu mehr Zweckorientierung der quantitativen Modelle gewandelt werden.

8 Zusammenfassung, Fazit und Ausblick

Abschließend werden die Ergebnisse und Erkenntnisse der Arbeit zusammengefasst und ein Ausblick auf mögliche anknüpfende Forschungsthemen gegeben.

8.1 Zusammenfassung und Fazit

Kern dieser Arbeit ist die Analyse von Faktoren, die die Umsetzung eines durchgängigen modellbasierten Systems Engineering behindern und die Ableitung eines methodischen Ansatzes, um diesen Faktoren am Beispiel Verknüpfung des Systemarchitekturmodells mit einem Berechnungsmodell entgegenzuwirken.

Der Stand der Forschung zeigt auf, dass die Relevanz eines durchgängigen, modellbasierten Systems Engineering in der Praxis erkannt wurde und erste Schritte zur Umsetzung stattfinden – nicht zuletzt getrieben durch prozessuale Anforderungen in der Automobilindustrie. Die Systemarchitekturen werden jedoch über Verständnisdokumentation und Verknüpfung von Anforderungen zu Architekturelementen hinaus nicht verwendet.

Eine Interviewstudie dient dem besseren Verständnis der **Faktoren, die eine umfassendere Nutzung der Systemarchitektur behindern** (→ Forschungsfrage 1). Es werden drei zentrale Faktoren identifiziert:

- Der Aufwand der Architekturmodellerstellung steht einem kaum konkret greifbaren Nutzen gegenüber, da konkrete Ansätze zur Nutzung der Systemarchitektur über die genannten Einsatzzwecke hinaus fehlen.
- Die modellierten Zusammenhänge lassen sich nicht für quantitative Untersuchungen, z. B. mit Berechnungsmodellen nutzen.
- Anforderungen können in der Systemarchitektur zwar mit anderen Anforderungen und Architekturelementen verknüpft werden, die inhaltliche Nachvollziehbarkeit der Anforderungen wird dadurch jedoch nicht ausreichend unterstützt.

Mit dem Ziel, die identifizierten Faktoren zu reduzieren wird ein Ansatz zur Nutzung der Systemarchitektur für die Ableitung nachvollziehbarer Anforderungen entwickelt. Der Ansatz beinhaltet den Aufbau eines Berechnungsmodells anhand der

Systemarchitektur und die Verankerung der Berechnungsergebnisse in der Systemarchitektur, um die resultierenden Anforderungen inhaltlich nachvollziehbar zu dokumentieren.

Zunächst wird der Ansatz auf theoretischer Basis unter Berücksichtigung des Stands der Forschung hergeleitet. Das SPES-Framework wird als **geeignete Struktur für das Architekturmodell** identifiziert (→ Forschungsfrage 2). Die darin gegebene Strukturierung in Abstraktionsebenen und Sichten ermöglicht zum einen die Beschreibung von Variationsanteilen im Sinne des Modells der SGE und unterstützt zum anderen die klare Systembeschreibung und damit die Übertragbarkeit der Inhalte auf das Berechnungsmodell. Die **von der Systemarchitektur in das Berechnungsmodell zu übertragenden Informationen** werden identifiziert und beschrieben, auf welche Weise sie in den beiden Modellen repräsentiert sind (→ Forschungsfrage 3). Um die mögliche **Integration der Erstellung und Nutzung einer Systemarchitektur in den Entwicklungsablauf** zu analysieren, wird das Vorgehen in den Kontext des erweiterten ZHO-Modells eingeordnet (→ Forschungsfrage 4). Für das Treffen von Entwicklungsentscheidungen und die daraus resultierende Konkretisierung von Anforderungen sind aus dem erweiterten ZHO-Modell insbesondere die Validierungsschritte von Bedeutung. Daher berücksichtigt der Zusammenhang zwischen Architekturmodell und Berechnungsmodell insbesondere auch die **Systematik des IPEK-XiL Frameworks** zur Strukturierung von Validierungsumgebungen (→ Forschungsfrage 5). Es wird aufgezeigt, wie im Berechnungsmodell **systematisch alternative Lösungskonzepte analysiert und vergleichend bewertet** werden können (→ Forschungsfrage 6). Das Architekturmodell wird genutzt, um die Ergebnisse des Berechnungsmodells sowie die darauf basierenden Entscheidungen und konkretisierten Anforderungen inhaltlich nachvollziehbar zu machen. Es wird ein **Verlinkungskonzept für die Ergebnisse des Berechnungsmodells und den daraus abgeleiteten Entscheidungen mit den Anforderungen** definiert (→ Forschungsfrage 7). Basierend auf den Erkenntnissen der theoretischen Analysen wird der Ansatz in Schritte zusammengefasst und beschrieben.

Nach theoretischer Ableitung wird der Ansatz am Beispiel des Antriebssystems eines elektrifizierten Fahrzeugs durch die Verfasserin umgesetzt. Es wird eine Systemarchitektur in SysML und ein Berechnungsmodell zur Reichweitenberechnung in Matlab aufgebaut. Die Modellstruktur und die abgebildeten Elemente in Matlab sind dabei der Systemarchitektur entnommen. Das Modell wird über zwei Abstraktionsebenen des SPES-Ansatzes detailliert, sodass der tiefste Detaillierungsgrad auf der Ebene der Getriebe-Subsysteme ist. Virtuell werden in vier definierten Manövern die Reichweiten verschiedener Antriebsvarianten berechnet. Die Varianten ergeben

sich dabei aus einer Kombination unterschiedlicher Traktionsmaschinen mit mehreren schaltbaren und nicht-schaltbaren Getrieben und unterschiedlichen Prämissen für die Schaltzeitpunkte, -abstände und -dauern. Ausgewertet werden neben der Reichweite, die den objektivierten Kundenwunsch darstellt, auch die resultierenden Subsystembelastungen auf den verschiedenen Abstraktionsebenen. Die Erfüllung der Reichweitenanforderung und die zu erwarteten Subsystembelastungen werden als Grundlage für Systementscheidungen herangezogen, die in der Systemarchitektur dokumentiert werden und die zu quantifizierten Anforderungen an die Subsysteme führen.

Abschließend werden Potenziale und Risiken des Ansatzes anhand der Beobachtungen während der Anwendung abgeleitet. Es werden Potenziale zur besseren Nutzung der Systemarchitektur durch den Ansatz identifiziert und Risiken aufgezeigt, wo der Ansatz die Akzeptanz von Entwicklungsteams negativ beeinflussen könnte. Bezogen auf die Hinderungsfaktoren ergeben sich folgende Feststellungen: Der Ansatz bietet das Potenzial, den wahrgenommenen Nutzen der Systemarchitektur zu vergrößern, indem er die Systemarchitektur als zentrales Informationselement sowohl für die Modellbildung in Matlab nutzt als auch für die Zugänglichkeit der Berechnungsergebnisse und darauf basierenden Entscheidungen. Der Ansatz kann potenziell den Modellbildungsaufwand im Berechnungsmodell reduzieren, indem Struktur und Informationen aus der Systemarchitektur verwendet werden. Damit kann er auch eine Grundlage für zukünftige automatisierte Datenübertragung schaffen. Der Ansatz ermöglicht die Integration von erarbeitetem Wissen aus der Berechnung in das Systemmodell und unterstützt dadurch potenziell die inhaltliche Nachvollziehbarkeit der abgeleiteten Anforderungen.

8.2 Ausblick

Die Idee des modellbasierten Systems Engineering ist eine durchgängige Anwendung untereinander konsistenter Modelle, deren gemeinsame Informationsbasis die in der Systemarchitektur abgebildeten Zusammenhänge sind. Dokumente werden aus den Modellen ausgeleitet, beinhalten aber keine Informationen, die nicht modellbasiert gespeichert sind.

Stand heute sind noch viele Prozesse in Unternehmen stark dokumentenbasiert, Modelle werden von den jeweiligen Experten erstellt, genutzt und weiterentwickelt, aber kaum miteinander vernetzt oder auf gemeinsamer Datenbasis erstellt. Die vorliegende Arbeit ist ein Ansatz um die potenzielle Nutzung von Systemarchitekturen als Rückgrat der Entwicklung zu unterstützen. Angrenzend und darauf aufbauend sind Untersuchungen zu den folgenden Themengebieten von Interesse:

Die Anbindung von Systemarchitekturen in SysML an gängige Berechnungs- und Simulationswerkzeuge ist aktuell kaum umgesetzt obwohl technische Möglichkeiten dazu bestehen. Um die Anbindung an mehrere unterschiedliche Modelle effektiv zu gestalten kann tiefer untersucht werden, welche Voraussetzungen in der Modellbildung dafür gegeben sein müssen

Neben Modellen zur Berechnung und Simulation basieren auch viele Methoden auf der Identifikation von Funktions- und Strukturzusammenhängen des zu entwickelnden Systems. Insbesondere vor dem Hintergrund der Sicherheit mechatronischer Produkte werden umfangreiche Methoden zur Sicherheitsanalyse angewendet wie beispielsweise FMEA, FMEDA, FTA, Gefahrenanalyse und Risikobewertung. Diese Methoden werden – ähnlich wie die Modelle – bislang kaum auf gegenseitige Konsistenz geprüft oder auf einer gleichen Datenbasis aufgebaut. Auch hier stellt sich die Frage, welche Eigenschaften eine Systemarchitektur haben muss um als Grundlage für diese Methoden zu dienen und den Arbeitsaufwand – z. B. durch automatisierte Synchronisierung von Funktions- und Produktstruktur – zu reduzieren.

Der Stand der Forschung und die Interviewstudie im Rahmen dieser Arbeit haben wiederholt aufgezeigt, dass die Modellierung mit SysML als sehr aufwändig wahrgenommen wird. Insbesondere wenn durch Ansätze zur Verknüpfung mit Methoden und Modellen weitere formale Anforderungen an die Systemarchitektur hinzukommen, besteht das Risiko, dass der wahrgenommene Aufwand weiter steigt und nicht durch den zusätzlichen Nutzen ausgeglichen wird. Das motiviert die weitere Untersuchung von Akzeptanzaspekten und Nutzenpotenzialen, insbesondere vor dem Hintergrund, dass Entwicklungsmethoden in erster Linie dem Entwicklungsteam dienlich sein sollten, indem sie die Aktivitäten und den Arbeitsalltag vereinfachen.

Literaturverzeichnis

:Em engineering methods AG. (2022, 28. Oktober). *Tutorials: FAS4M - YouTube*. Verfügbar unter: https://www.youtube.com/playlist?list=PLII_0AK-RCGLRWiDIFHuLLco-5bLAm1wzy

Albers, A. [Albert] (2010). Five Hypotheses about Engineering Processes and their Consequences. In I. Horvath, F. Mandorli & Z. Rusák (Hrsg.), *Proceedings of the 8th International Symposium on Tools and Methods of Competitive Engineering, TMCE 2010* (Bd. 1, 343–356). Ancona, Italien: Organizing Committee of TMCE 2010 Symposium.

Albers, A. [Albert]. (2023). *Engineering neu denken und gestalten. Herausforderungen, Anwendungsszenarien und das neue Leitbild Advanced Systems Engineering* (Albers, A., Hrsg.) (acatech IMPULS). München: acatech - Deutsche Akademie der Technikwissenschaften e.V. https://doi.org/10.48669/aca_2023-7

Albers, A. [Albert], Behrendt, M., Klingler, S. & Matros, K. (2016). Verifikation und Validierung im Produktentstehungsprozess. In U. Lindemann (Hrsg.), *Handbuch Produktentwicklung* (S. 541–569). München: Hanser.

Albers, A. [Albert], Behrendt, M., Klingler, S., Reiss, N. & Bursac, N. (2017). Agile product engineering through continuous validation in PGE – Product Generation Engineering. *Design Science Journal*, 3, 1–19. <https://doi.org/10.1017/dsj.2017.5>

Albers, A. [Albert], Behrendt, M., Matros, K., Holzer, H. & Bohne, W. (2014). Development of Hybrid-Powertrains by Means of X-in-the-Loop-Approach. In *Ta- gungsband 11. Symposium Hybrid- und Elektrofahrzeuge*. Braunschweig.

Albers, A. [Albert], Behrendt, M. & Ott, S. (2010). Validation – Central Activity to Ensure Individual Mobility. In *Proceedings of FISITA 2010 World Automotive Congress*. Budapest, Ungarn.

- Albers, A. [Albert], Behrendt, M., Schroeter, J., Ott, S. & Klingler, S. (2013). X-in-the-Loop: A Framework for Supporting Central Engineering Activities and Contracting Complexity in Product Engineering Processes. In U. Lindemann, S. Venkataraman, Y. Kim, S. W. Lee, J. Clarkson & G. Cascini (Hrsg.), *Proceedings of the 19th International Conference on Engineering Design, ICED13* (S. 379–389). Seoul, Korea.
- Albers, A. [Albert] & Braun, A. (2011). A Generalized Framework to Compass and to Support Complex Product Engineering Processes. *International journal of product development*, 15(1/2/3), 6–25.
<https://doi.org/10.1504/IJPD.2011.043659>
- Albers, A. [Albert], Burkhardt, N., Meboldt, M. & Saak, M. (2005). SPALTEN problem solving methodology in the product development. In A. E. Samuel & W. Lewis (Hrsg.), *Proceedings of the 15th International Conference on Engineering Design (ICED) 2005. Engineering design and the global economy* (S. 553–554). Melbourne, Australien: Design Society.
- Albers, A. [Albert], Bursac, N. & Wintergerst, E. (2015). Produktgenerationsentwicklung – Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In H.-G. Binz, B. Bertsche, W. Bauer, O. Riedel, D. Spath & D. Roth (Hrsg.), *Stuttgarter Symposium für Produktentwicklung SSP 2015. Entwicklung smarterer Produkte für die Zukunft*. Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.
- Albers, A. [Albert] & Düser, T. (2008). Integrierte Validierungsumgebung für energieeffiziente Fahrerassistenzsysteme am Rollenprüfstand. In *IPG Technology Conference 2008*. Ettlingen.
- Albers, A. [Albert], Düser, T. & Ott, S. (2008). X-in-the-loop als integrierte Entwicklungsumgebung von komplexen Antriebsystemen. In *8. Tagung Hardware-in-the-loop-Simulation*. Kassel.
- Albers, A. [Albert], Ebel, B. & Lohmeyer, Q. (2012). Systems of Objectives in Complex Product Development. In I. Horváth, Z. Rusák, A. Albers & M. Behrendt (Hrsg.), *Tools and Methods of Competitive Engineering – TMCE 2012* (Bd. 1,2, S. 267–278). Karlsruhe.
- Albers, A. [Albert], Ebertz, J., Rapp, S., Heimicke, J., Kürten, C., Zimmermann, V. et al. (2020). *Produktgeneration 1 im Modell der PGE – Produktgenerations-*

entwicklung: Verständnis, Zusammenhänge und Auswirkungen in der Produktentwicklung. Verständnis, Zusammenhänge und Auswirkungen in der Produktentwicklung (KIT Scientific Working Papers 149). Karlsruhe.
<https://doi.org/10.5445/IR/1000127971>

- Albers, A. [Albert], Kürten, C., Rapp, S., Birk, C., Hünemeyer, S. & Kempf, C. (2022). *SGE – Systemgenerationsentwicklung. Analyse und Zusammenhänge von Entwicklungspfaden in der Produktentstehung* (KIT Scientific Working Papers 199). Verfügbar unter: <https://publikationen.bibliothek.kit.edu/1000151151>
<https://doi.org/10.5445/IR/1000151151>
- Albers, A. [Albert] & Lohmeyer, Q. (2012). Advanced Systems Engineering - Towards a Model-Based and Human-Centred Methodology. In I. Horváth, Z. Rusák, A. Albers & M. Behrendt (Hrsg.), *Tools and Methods of Competitive Engineering – TMCE 2012* (Bd. 1,2, S. 407–416). Karlsruhe.
- Albers, A. [Albert], Lohmeyer, Q. & Ebel, B. (2011). Dimensions of objectives in interdisciplinary product development projects. In *Proceedings of the 18th International Conference on Engineering Design (ICED 11)* (Bd. 2, S. 256–265). Copenhagen, Denmark.
- Albers, A. [Albert], Matros, K., Behrendt, M. & Jetzinger, H. (2015). Das Pull-Prinzip der Validierung – Ein Referenzmodell zur effizienten Integration von Validierungsaktivitäten in den Produktentstehungsprozess. *VDI Konstruktion*, (Juni 6-2015), 74–81.
- Albers, A. [Albert] & Meboldt, M. (2007). IPEMM – Integrated Product Development Process Management Model, Based on Systems Engineering and Systematic Problem Solving. In J.-C. Bocquet (Hrsg.), *Proceedings of the 16th International Conference on Engineering Design (ICED 2007)* (611-612 (exec. Summ.), full paper no. DS42_P_537). Paris, Frankreich: Design Society.
- Albers, A. [Albert] & Rapp, S. (2022). Model of SGE: System Generation Engineering as Basis for Structured Planning and Management of Development. In D. Krause & E. Heyden (Hrsg.), *Design Methodology for Future Products. Data Driven, Agile and Flexible* (S. 27–46). Cham: Springer International Publishing. Verfügbar unter: <https://link.springer.com/book/10.1007/978-3-030-78368-6>
- Albers, A. [Albert], Rapp, S., Fahl, J., Hirschter, T., Revfi, S., Schulz, M. et al. (2020). Proposing a generalized description of variations in different types of

- systems by the model of PGE - Product Generation Engineering. In *Proceedings of the Design Society: DESIGN Conference* (Bd. 1, S. 2235–2244). Cambridge: Cambridge University Press.
- Albers, A. [Albert], Rapp, S., Spadinger, M., Richter, T. [Thilo], 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. In Design Society (Hrsg.), *Proceedings of the 22nd International Conference on Engineering Design (ICED19)* (Bd. 1, S. 1693–1702). Delft, The Netherlands.
- Albers, A. [Albert], Reiß, N. [Nicolas], Bursac, N. & Breitschuh, J. (2016). 15 Years of SPALTEN Problem Solving Methodology in Product Development. In C. Boks (Ed.), *Proceedings of NordDesign 2016. August 10-12, 2016, Trondheim, Norway* (S. 411–420). Bristol, United Kingdom: The Design Society.
- Albers, A. [Albert], Reiß, N. [N.], Bursac, N. & Richter, T. [T.] (2016). The integrated Product engineering Model (iPeM) in context of the product generation engineering. In L. Wang & T. Kjellberg (Hrsg.), *26th CIRP Design Conference* (Procedia CIRP, Bd. 50, S. 100–105). Stockholm, Schweden.
- Albers, A. [Albert], Saak, M. & Burkardt, N. (2002). Gezielte Problemlösung bei der Produktentwicklung mit Hilfe der SPALTEN-Methode. In Rektor der TU Ilmenau (Hrsg.), *Tagungsunterlagen 47. Internationales Wissenschaftliches Kolloquium* (S. 83–84). Ilmenau.
- Albers, A. [Albert] & 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 (Hrsg.), *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations* (S. 151–171). London: Springer.
- Albers, A. [Alexander], Kühn, A. & Dumitrescu, R. (2017). Modellbasierte Entscheidungsunterstützung in der Produktgenerationenplanung. In S.-O. Schulze, C. Tschirner, R. Kaffenberger, S. Ackva & GfSE Gesellschaft für Systems Engineering e. V. German Chapter of INCOSE (Hrsg.), *Proceedings of Tag des Systems Engineering (TdSE) 2017*. Paderborn.
- APIS Informationstechnologien GmbH. (2022, 20. September). *APIS IQ-Software | FMEA | DRBFM | Funktionale Sicherheit*. Zugriffsdatum 28.10.2022. Verfügbar unter: <https://www.apis.de/>

- Badke-Schaub, P., Daalhuizen, J. & Roozenburg, N. (2011). Towards a Designer-Centered Methodology: Descriptive Considerations and Prescriptive Reflections. In H. Birkhofer (Hrsg.), *The Future of Design Methodology* (S. 181–197). London: Springer.
- Baumgart, I. (2016). Requirements Engineering. In U. Lindemann (Hrsg.), *Handbuch Produktentwicklung* (S. 425–453). München: Hanser.
<https://doi.org/10.3139/9783446445819.015>
- Beiser, S., Wydra, M. & Geimer, M. (2023). Simulative Untersuchung zum Energiebedarf einer schreitenden Forstmaschine. In M. Geimer & C. Kunze (Hrsg.), *Hybride und energieeffiziente Antriebe für mobile Arbeitsmaschinen: 9. Fachtagung* (S. 11–35). Karlsruhe: KIT Scientific Publishing.
- Bringmann, E. & Krämer, A. (2008). Model-Based Testing of Automotive Systems. In IEEE Computer Society (Hrsg.), *Proceedings of First International Conference on Software Testing, Verification, and Validation*. Lillehammer, Norway.
- Brinkschulte, L., Engelmann, D., Siebert, J., Iwanicki, M. & Geimer, M. (2017). MO-BiL – Eine auf mobile Arbeitsmaschinen optimierte Prüfmethode. In F. Gaeterin, M. Geimer, P. Gratzfeld & F. Henning (Hrsg.), *Hybride und energieeffiziente Antriebe für mobile Arbeitsmaschinen: 6. Fachtagung* (S. 173–194). Karlsruhe: KIT Scientific Publishing.
- Bühne, S., Halmans, G., Pohl, K., Weber, M., Kleinwechter, H. & Wierczoch, T. (2004). Defining Requirements at Different Levels of Abstraction. In *12th IEEE International Requirements Engineering Conference* (S. 346–347). Kyoto.
- Bursac, N., Rapp, S., Waldeier, L., Wagenmann, S., Albers, A. [Albert], Deiss, M. et al. (2021). Anforderungsmanagement in der Agilen Entwicklung Mechatronischer Systeme – ein Widerspruch in sich? In R. H. Steltzer & J. Krzywinski (Hrsg.), *Entwerfen Entwickeln Erleben - EEE2021*. Dresden: TUDpress.
- Charette, R. N. (2021, 7. Juni). How Software Is Eating the Car. *IEEE Spectrum* (Zugriffsdatum 27.11.2023). Verfügbar unter: <https://spectrum.ieee.org/software-eating-car>

- Cloutier, R. (2019). *Model Based Systems Engineering Survey. conducted December 2018*. Zugriffsdatum 27.10.2022. Mobile, USA: University of South Alabama. Verfügbar unter: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2019:03_2018_mbse_analysis-10.pdf
- Cooper, R. G. (1990). Stage-Gate Systems: A New Tool for Managing New Products. *Business horizons*, 33(3), 44–54. [https://doi.org/10.1016/0007-6813\(90\)90040-I](https://doi.org/10.1016/0007-6813(90)90040-I)
- Dörner, D. (1979). *Problemlösen als Informationsverarbeitung*. Stuttgart: Kohlhammer.
- Dumitrescu, R., Albers, A. [Albert], Riedel, O., Stark, R. & Gausemeier, J. (2021). *Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft. Ein Beitrag zum Advanced Systems Engineering*. Zugriffsdatum 27.10.2022 (Dumitrescu, R., Albers, A., Riedel, O., Stark, R. & Gausemeier, J., Hrsg.). Paderborn: acatech - Deutsche Akademie der Technikwissenschaften e.V. Verfügbar unter: <https://www.acatech.de/publikation/engineering-in-deutschland/>
- Düser, T. (2010). X-in-the-Loop – ein durchgängiges Validierungsframework für die Fahrzeugentwicklung am Beispiel von Antriebsstrangfunktionen und Fahrerassistenzsystemen. Dissertation. 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>
- 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>
- Fracchia, M. (2017). *Transmissions for Electric and Hybrid Vehicles*. Zugriffsdatum 30.10.2022, Vocis Ltd. Verfügbar unter: <https://www.youtube.com/watch?app=desktop&v=aexD3b0CknY>
- Gausemeier, J., Dumitrescu, R., Steffen, D., Czaja, A., Siederkehr, O. & Tschirner, C. (2015). *Systems engineering in industrial practice*. Zugriffsdatum 27.10.2022. Paderborn: Heinz Nixdorf Institut; Fraunhofer Institut für Produkti-

onstechnik IPT; Unity AG. Verfügbar unter: https://www.hni.uni-paderborn.de/fileadmin/Fachgruppen/Seniorprofessur_Gausemeier/systemsengineering/20150706_SE-Studie_enGB_Einzelseiten.pdf

Gausemeier, J., Meyer, M. & Steglich, S. (2022, Juli). *Strategie Advanced Systems Engineering. Leitinitiative zur Zukunft des Engineering- und Innovationsstandorts Deutschland*. Zugriffsdatum 27.11.2023 (Dumitrescu, R., Gausemeier, J. & Riedel, O., Hrsg.). München: acatech - Deutsche Akademie der Technikwissenschaften. Verfügbar unter: https://www.advanced-systems-engineering.de/ASE_Strategie.pdf

Heinrich, D. (2016). Modellierung des Fahrerverhaltens zur Ermittlung von Bauteilbelastungen im Fahrzeugantriebsstrang. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 92). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000054681>

Herzig, S. J., Rouquette, N. F., Forrest, S. & Jenkins, J. S. (2013). Integrating Analytical Models with Descriptive System Models: Implementation of the OMG SysML Standard for the Tool-specific Case of MapleSim and MagicDraw. *Procedia Computer Science*, 16, 118–127. <https://doi.org/10.1016/j.procs.2013.01.013>

Holt, J. & Perry, S. (2018). *SysML for Systems Engineering: A Model-Based Approach. A model-based approach* (Computing and networks, vol. 20, Third edition). Stevenage: Institution of Engineering and Technology. <https://doi.org/10.1049/PBPC020E>

Holt, J., Perry, S., Brownsword, M., Cancila, D., Hallerstede, S. & Overgaard Hansen, F. (2012). Model-Based Requirements Engineering for System of Systems. In *System of Systems Engineering (SoSE), 2012 7th International Conference on* (S. 561–566). Genova, Italy: IEEE / Institute of Electrical and Electronics Engineers Incorporated.

IBM. (2021). *Engineering Requirements DOORS Family - Überblick*. Zugriffsdatum 28.10.2022. Verfügbar unter: <https://www.ibm.com/de-de/products/requirements-management>

INCOSE. (2014). *Systems Engineering Vision 2025*. Zugriffsdatum 25.10.2022, INCOSE - International Council on Systems Engineering. Verfügbar unter:

https://www.incose.org/docs/default-source/se-vision-2025/se-vision-2025/incose-se-vision-2025.pdf?sfvrsn=602663c7_2

INCOSE. (2022, 25. Oktober). *Systems Engineering*. Zugriffdatum 25.10.2022.

Verfügbar unter: <https://www.incose.org/systems-engineering>

ISO, 26262-2 (2018-12). *Road vehicles - Functional safety - Part 2: Management of functional safety*. Berlin: Beuth Verlag GmbH.

ISO, 26262-3 (2018-12). *Road vehicles - Functional safety - Part 3: Concept phase*. Berlin: Beuth Verlag GmbH.

ISO, 26262-4 (2018-12). *Road vehicles - Functional safety - Part 4: Product development at the system level*. Berlin: Beuth Verlag GmbH.

Kano, N., Seraku, N., Takahashi, F. & ichi Tsuji, S. (1984, 15. April). Attractive Quality and Must-Be Quality. Kano, Noriaki; Seraku, Nobuhiko; Takahashi, Fumio; ichi Tsuji, Shin. *Journal of the Japanese Society for Quality Control*, 14(2), 147–156. Verfügbar unter: <http://ci.nii.ac.jp/naid/110003158895/en/>

Lewis, G. A., Morris, E., Place, P., Simanta, S. & Smith, D. B. (2009). Requirements Engineering for Systems of Systems. In *2009 3rd Annual IEEE Systems Conference* (S. 247–252). Vancouver: IEEE / Institute of Electrical and Electronics Engineers Incorporated.

Lindemann, U. (2009). *Methodische Entwicklung technischer Produkte – Methoden flexibel und situationsgerecht anwenden*. Berlin, Heidelberg: Springer Verlag. Verfügbar unter: <http://dx.doi.org/10.1007/978-3-642-01423-9>

Lohmeyer, Q. (2013). Menschzentrierte Modellierung von Produktentstehungssystemen unter besonderer Berücksichtigung der Synthese und Analyse dynamischer Zielsysteme. Dissertation. 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>

Mandel, C., Böning, J., Behrendt, M. & Albers, A. [Albert] (2021). A Model-Based Systems Engineering Approach to Support Continuous Validation in PGE - Product Generation Engineering. In *2021 International Symposium on Systems Engineering (ISSE) Proceedings* (1-8). Wien: IEEE.

- Mandel, C., Martin, A. & Albers, A. [Albert] (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. (Hrsg.), *Proceedings of the XXXIII ISPIM Innovation Conference "Innovating in a Digital World"* (1-16). Copenhagen: LUT Scientific and Expertise Publications.
- 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* (1-14). Wiesbaden.
- Mandel, C., Wolter, K., Bause, K., Behrendt, M., Hanf, M. M. & Albers, A. [Albert] (2020). Model-Based Systems Engineering methods to support the reuse of knowledge within the development of validation environments. In *2020 IEEE International Systems Conference (SysCon)* (S. 1–7). Montreal: IEEE.
- Marxen, L. & Albers, A. [Albert] (2012). Supporting Validation in the Development of Design Methods. In D. Marjanović (ed.), *DESIGN 2012. Proceedings of the 12th International Design Conference*, (DS, vol. 70, S. 1009–1018). Dubrovnik: Fac. of Mechanical Engineering and Naval Architecture.
- Meboldt, M. (2008). Mentale und formale Modellbildung in der Produktentstehung - als Beitrag zum integrierten Produktentstehungs-Modell (iPeM). Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 29). Karlsruhe: Universität Karlsruhe (TH). <https://doi.org/10.5445/IR/1000028850>
- Moeser, G., Albers, A. [Albert] & Klingler, S. (2015). Modellbasierte Funktions-Gestalt-Synthese: Unterstützung der Entwickler bei der Definition der Produktgestalt. In D. Krause, K. Paetzold & S. Wartzack (Hrsg.), *Proceedings of the 26th Design for X (DfX) Symposium (2015)* (S. 221–232). Herrsching: TuTech Verlag.
- Moeser, G., Grundel, M., Weilkens, T., Kümpel, S., Kramer, C. & Albers, A. [Albert] (2016). Modellbasierter mechanischer Konzeptentwurf: Ergebnisse des FAS4M-Projektes. In S.-O. Schulze, C. Tschirmer, R. Kaffenberger, S. Ackva & GfSE Gesellschaft für Systems Engineering e.V. (Hrsg.), *Tag des Systems Engineering* (S. 417–428). München: Carl Hanser.
- Oerding, J. (2009). Ein Beitrag zum Modellverständnis der Produktentstehung – Strukturierung von Zielsystemen mittels C&CM. Dissertation. In A. Albers

- (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 37). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000013689>
- OMG. (2019). *OMG Systems Modeling Language (OMG SysML™). Version 1.6*. Zugriffsdatum 26.10.2022, Object Management Group - OMG. SysML v1.6. Verfügbar unter: <https://www.omg.org/spec/SysML/1.6/>
- Phoenix Integration. (2020, 7. Februar). *MBE Applications | ModelCenter Software | Phoenix Integration*. Zugriffsdatum 26.10.2022. Verfügbar unter: <https://www.phoenix-int.com/application/mbe-model-based-engineering/>
- PivotPoint Technology Corp. (2018, 1. Juli). *MBE Forum: Model-Based Engineering Visual Glossary*. Zugriffsdatum 26.10.2022. Verfügbar unter: <https://model-basedengineering.com/glossary/>
- Pohl, K., Hönninger, H., Achatz, R. & Broy, M. (2012). *Model-Based Engineering of Embedded Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-34614-9>
- Pohl, K. & Rupp, C. (2021). *Basiswissen Requirements Engineering. Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level* (5., überarbeitete und aktualisierte Auflage). Heidelberg: dpunkt.verlag. Verfügbar unter: <https://se-arch.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2913478>
- Pohlandt, C., Brinkschulte, L. & Geimer, M. (2016). Entwicklungsmethoden für elektrische und hybride Antriebssysteme. *ATZoffhighway*, (05/2016).
- Ponn, J. & Lindemann, U. (2008). *Konzeptentwicklung und Gestaltung technischer Produkte*. Berlin, Heidelberg: Springer-Verlag.
- Rambo, J., Tschirner, C., Weber, H., Zuccaro, C. & Dorsch, T. (2017). *Systems Engineering. Die Klammer in der technischen Entwicklung*. Bremen: GfSE Verlag.
- Reichert, U., Epskamp, T., Ott, S. & Radimersky, A. (2016a). High speed electric drive with a three-speed gearbox. *CTI Mag – The Automotive TM, HEV & EV Drives Magazine*.

- Ropohl, G. (1975). Einleitung in die Systemtechnik. In G. Ropohl (Hrsg.), *Systemtechnik – Grundlagen und Anwendung* (S. 1–77). München Wien: Carl Hanser Verlag.
- Scherer, H., Albers, A. [Albert] & Bursac, N. (2017). Model Based Requirements Engineering for the Development of Modular Kits. In *Procedia 27th CIRP Design Conference* (Bd. 60, S. 145–150). Cranfield: Elsevier.
- Siemens Product Lifecycle Management Software Inc. (2022a). *Polarion REQUIREMENTS. Requirements Management, Requirements Gathering*. Zugriffsdatum 22.10.2022. Verfügbar unter: <https://polarion.plm.automation.siemens.com/products/polarion-requirements>
- Siemens Product Lifecycle Management Software Inc. © 2022.. *Requirements Management, Requirements Gathering, Requirements Management tools - Polarion REQUIREMENTS*. Zugriffsdatum 28.10.2022. Verfügbar unter: <https://polarion.plm.automation.siemens.com/products/polarion-requirements>
- Simpson, T. W., Bobuk, A., Slingerland, L. A., Brennan, S., Logan, D. & Reichard, K. (2012). From User Requirements to Commonality Specifications: an Integrated Approach to Product Family Design. *Research in Engineering Design*, 23(2), 141–153. <https://doi.org/10.1007/s00163-011-0119-4>
- Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Wien: Springer Verlag.
- Stützel, B. & Paul, A. (Prozesswerk GmbH, Hrsg.). (2021). *Systems Engineering in Deutschland. Die deutsche Unternehmenslandschaft im Vergleich*. Zugriffsdatum 27.10.2022. Verfügbar unter: <https://www.prozesswerk.eu/site/assets/files/1179/se-studie-2021.pdf>
- Utting, M., Pretschner, A. & Legeard, B. (2006). *A Taxonomy of Model-Based Testing* (Department of Computer Science, The University of Waikato, Hrsg.) (Workin Paper Series). Hamilton, New Zealand. Verfügbar unter: <https://pdfs.semanticscholar.org/61b3/def6e04886266bb12eb420739f66d605e332.pdf>
- VDA QMC Working Group 13 / Automotive SIG. (2017). *Automotive SPICE Process Assessment / Reference Model. Version 3.1*. Zugriffsdatum 14.10.2022. Verfügbar unter: https://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_PAM_31.pdf

- VDI/VDE-Richtlinie, 2206 (2021). *Entwicklung mechatronischer und cyber-physischer Systeme*. Berlin: Beuth Verlag GmbH.
- VDI-Richtlinie, 2221-1 (2019). *Entwicklung technischer Produkte und Systeme*. Berlin: Beuth Verlag GmbH.
- VDMA. (2002). *Leitfaden für die Anforderungsanalyse Teil 1 – Prozessbeschreibung*. Aus der Reihe *Methoden und Verfahren*. Frankfurt: VDMA Verlag.
- Vidal, E. J. & Villota, E. R. (2018). SysML as a Tool for Requirements Traceability in Mechatronic Design. In *Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering* (S. 146–152). New York, NY, USA: ACM.
- Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., Shortell, T. M. & International Council on Systems Engineering (Hrsg.). (2015). *Systems engineering handbook: a guide for system life cycle processes and activities ; INCOSE-TP-2003-002-04, 2015* (4. Aufl.). Hoboken, NJ: Wiley.
- 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. In *Proceedings of the Design Society: DESIGN Conference* (Bd. 1, S. 2425–2434). Cambridge: Cambridge University Press.
- 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 H.-G. Binz, D. Spah & D. Roth (Hrsg.), *Stuttgarter Symposium für Produktentwicklung SSP 2021* (Stuttgarter Symposium für Produktentwicklung, Bd. 6). Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.
- 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)* (S. 275–280). Rochester, NY, USA: IEEE.
- Weilkiens, T. (2014). *Systems Engineering mit SysML/UML. Anforderungen, Analyse, Architektur* (3., überarb. und aktualisierte Aufl.). Heidelberg: dpunkt.verl.

Weilkiens, T., Lamm, J. G., Roth, S. & Walker, M. (2015). *Model-Based System Architecture*. Hoboken, NJ, USA: John Wiley & Sons, Inc. Verfügbar unter: <http://doi.wiley.com/10.1002/9781119051930>

Windisch, E., Mandel, C., Rapp, S., Bursac, N. & Albers, A. [Albert] (2022). Approach for model-based requirements engineering for the planning of engineering generations in the agile development of mechatronic systems. In *Procedia CIRP Design Conference 2022* (S. 550–555). Online: Elsevier.

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).

Studentische Abschlussarbeiten, die im Kontext dieser Dissertation am IPEK – Institut für Produktentwicklung am Karlsruher Institut für Technologie (KIT) von der Autorin co-betreut wurden:

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. IPEK-thesis, Sperrvermerk*. Unveröffentlichte Bachelor-/Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.

Timucin, M. (2020). *Entwicklung eines Vorgehens zur systematischen Verknüpfung von Produkthanforderungen und Systemarchitektur unter Berücksichtigung der PGE am Beispiel ECA*. Unveröffentlichte Bachelor-/Masterarbeit. Karlsruher Institut für Technologie (KIT), Karlsruhe.

Veröffentlichungen, die unter Mitautorenschaft der Autorin dieser Forschungsarbeit entstanden sind

Albers, A. [Albert], Lohmeyer, Q. & Radimersky, A. (2012). Individuelle und organisatorische Akzeptanz von Methoden des Systems Engineering. In M. Maurer, S.-O. Schulze & GfSE (Hrsg.), *Tag des Systems Engineering (TdSE) 2012*. Paderborn: Hanser.

- Albers, A. [Albert], Radimersky, A. & Brezger, F. (2015). Funktionale Wechselwirkungen von Batteriesystemen in elektrifizierten Fahrzeugen. In H.-G. Binz, B. Bertsche, W. Bauer, O. Riedel, D. Spath & D. Roth (Hrsg.), *Stuttgarter Symposium für Produktentwicklung SSP 2015. Entwicklung smarter Produkte für die Zukunft*. Stuttgart: Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO.
- Albers, A. [Albert], Reichert, U., Bause, K., Radimersky, A. & Ott, S. (2017). *Entwicklung einer verlustoptimierten Kupplung für ein mehrgängiges Getriebe für ein Elektrofahrzeug. VDI Kupplungen und Kupplungssysteme in Antrieben*, Bonn.
- Albers, A. [Albert], Reichert, U., Ott, S. & Radimersky, A. (2017). Software supported development of a Battery Electric Vehicle powertrain considering the efficiency. In *16th International VDI Congress – Drivetrain for Vehicles 2016* (Bd. 2276, S. 321–337).
- Bause, K., Radimersky, A., Iwanicki, M. & Albers, A. [Albert] (2014). Feasibility Studies in the Product Development Process. In G. Moroni & T. Tolio (Hrsg.), *Procedia 24th CIRP Design Conference* (S. 473–478). Milano. Verfügbar unter: <https://doi.org/10.1016/j.procir.2014.03.128>
- Bause, K., Radimersky, A. & Ott, S. (2015). *Anforderungen an Getriebe in E-Fahrzeugen unter Berücksichtigung der Wechselwirkungen im Antriebssystem*. DMK - Dresdner Maschinenelemente Kolloquium, 08./09. Dezember 2015, Dresden.
- Eisele, M., Dollinger, A., Burgert, T., Müller, F., Bause, K., Radimersky, A. et al. (2016). *Entwicklung von hybriden Antriebssystemen für den Individualverkehr in der Profilvereinigung Karlsruhe* (Karlsruher Institut für Technologie (KIT), Hrsg.). Karlsruhe. <https://doi.org/10.5445/IR/1000062039>
- Eisele, M., Radimersky, A., Remmlinger, J. & Ott, S. (2015). Initiales Zielsystem eines Batteriekonfigurators zur Konzeptionierung von Batteriesystemen in mobilen Anwendungen. Design for X Symposium. In D. Krause, K. Paetzold & S. Wartzack (Hrsg.), *Proceedings of the 26th Design for X (DfX) Symposium (2015)* (S. 233–244). Herrsching: TuTech Verlag.
- Eisele, M., Schoch, S., Puls, R., Albers, A. [Albert] & Radimersky, A. (2016). Priorisierung von Aktivitäten zur Konkretisierung von Produkteigenschaften am Beispiel einer Batteriemodulentwicklung. In D. Krause, K. Paetzold & S. Wartzack

- (Hrsg.), *Proceedings of the 27th Design for X (DfX) Symposium*. Symposium, Jesteburg. TuTech Innovation.
- Gausemeier, J., Albers, A. [Albert], Radimersky, A. & Turki, T. (2012). Verbesserte Ausschöpfung vorhandener Innovationspotentiale durch Kopplung von Prozess-, Wissens- und Methodenmanagement. In *SVT 8. Symposium für Vorschau und Technologieplanung* (Bd. 306). Berlin: HNI-Verlagsschriftreihe.
- Haag, H.-C., Radimersky, A. & Meyer-Schwickerath, B. (2012). Wandlungstreiber in Wertschöpfungsnetzen erkennen. Wandlungsdruck mittels Zukunftsszenarien identifizieren und bewerten. *wt Werkstattstechnik online*, 102(09), 633–638.
- Janz, K., Ott, S., Reichert, U., Radimersky, A. & Epskamp, T. (2016). An einem Strang. *Research to Business*, (2/2016), 2–3. Verfügbar unter: https://www.kit-technology.de/fileadmin/user_upload/R2B_PDF/R2b_2_2016_neu_web.pdf
- Kaiser, J., Timke, T., Brein, D., Schmid, A., Schlote-Holubek, K., Endisch, J. et al. (2014). Sicherheit von Lithium-Ionen-Batterien als Energiespeicher in Verbindung mit Photovoltaikanlagen. Hinweise für die Feuerwehren im Einsatz. *Brandschutz/Deutsche Feuerwehr-Zeitung*, 68(8), 588–592.
- Lohmeyer, Q., Albers, A. [Albert], Breitschuh, J. & Radimersky, A. (2014). Individual and Organizational Acceptance of Systems Engineering Methods - Survey and Recommendations. In I. Horváth & Z. Rusák (eds.), *Tools and methods of competitive engineering. Proceedings of the Tenth International Symposium on Tools and Methods of Competitive Engineering - TMCE 2014, May 19 - 23, Budapest, Hungary*. Delft: Faculty of Industrial Design Engineering Delft University of Technology.
- Pinner, T., Radimersky, A. & Ott, S. (2015). Energy Optimal Gear Change Strategy for Electric Vehicles. In *14. Internationales CTI Symposium – Fahrzeuggetriebe, HEV- und EV-Antriebe*. Berlin: CTI - Car Training Institute.
- Reichert, U., Epskamp, T., Ott, S. & Radimersky, A. (2016b). High speed electric drive with a three-speed gearbox. *CTI Mag – The Automotive TM, HEV & EV Drives Magazine*, (#December 2016), 25–27. Verfügbar unter: <https://dokumen.tips/documents/the-automotive-tm-hev-ev-drives-magazine-by-cti-the-automotive-tm-hev.html?page=4>

Reichert, U., Ott, S., Radimersky, A. & Raedt, H.-W. (2016). Leichtbau in der Getriebeentwicklung durch hochbelastbare Stähle. Das Gesamtsystem im Blick. *Digital Engineering Magazin*, (06/2016), 22-25. Verfügbar unter: <https://www.digital-engineering-magazin.de/printmagazine/digital-engineering-magazin-06-2016/>

Glossar

Anforderung	Eine Anforderung ist eine durch einen Wert oder einen Wertebereich festgelegte Beschreibung eines einzelnen Produktmerkmals. Eine Anforderung kann ein Ziel nicht ersetzen, sondern lediglich konkretisierend beschreiben. (Lohmeyer, 2013)
Antriebssystem	Das Antriebssystem umfasst ergänzend zum Antriebsstrang den Energiespeicher und wird verkürzt mit Antrieb bezeichnet. Die Antriebssystemtopologie bezeichnet die hardwareseitige Auswahl, Anordnung und Dimensionierung der [Subsysteme und der] Antriebskomponenten sowie die Gestaltung der energetischen Verknüpfungen, wobei der Einbauort, die Orientierung und die Schnittstellen zum Restfahrzeug nicht definiert sind und somit zum Gestaltungsraum gehören. (Albert Albers, Behrendt, Matros, Holzer & Bohne, 2014)
Erweitertes ZHO-Triple	Das Systemtriple der Produktentwicklung beschreibt die Produktgestaltung als eine kontinuierliche Interaktion von drei Systemen. Einschließlich der Konstrukteure und der Konstruktionsaktivitäten entwickelt das Arbeitssystem gleichzeitig zwei verschiedene Systeme: das System der Ziele und das System der Objekte. Der Entwicklungsprozess bringt also neben dem Endprodukt ständig Informationen hervor, auf denen neue Ziele basieren, sowie Objekte, auf denen neue Informationen basieren. (Albert Albers et al., 2011)
Funktion	Der Begriff Funktion ist für den Produktentwickler nur im Umgang mit Modellen von technischen Systemen anwendbar. Sie ist die Umwandlung von Input zu Output, die in dem modellierten System abgebildet wird. Sie entsteht aus der Interaktion von Gestaltelementen und kann nicht direkt beeinflusst werden. (Albert Albers & Wintergerst, 2014)
IPEK-X-in-the-Loop-Ansatz	Der IPEK-X-in-the-Loop-(IPEK-XiL)-Ansatz nach Albers greift die etablierten Ansätze (Model-in-the-Loop, Software-in-the-Loop, Hardware-in-the-Loop) auf, integriert die jeweiligen Vorteile und erweitert diese konsequent um die Belange der Mechanik bzw. Mechatronik sowie der Entwickler aus unterschiedlichen Fachdisziplinen. Somit beschreibt der IPEK-XiL-Ansatz das grundlegende Verständnis in der Validierung eines Teilsystems,

	dieses in das Gesamtsystem, die Umwelt und auch evtl. weitere interagierende Systeme wie bspw. den Fahrer einzubinden. (Albert Albers, Düser & Ott, 2008)
iPeM	Das iPeM ist ein generisches Metamodell, das die relevanten Elemente zur Ableitung von situationsspezifischen Produktentwicklungsprozessmodellen umfasst. Das iPeM basiert auf dem Systemtripel der Produktentwicklung: Es beschreibt die Produktentwicklung als eine kontinuierliche Interaktion des Systems der Ziele, des Systems der Objekte und des Handlungssystems. (Albert Albers, Reiß, Bursac & Richter, 2016)
Methode	Methoden beschreiben ein zielgerichtetes Vorgehen. Sie haben einen deskriptiven Charakter und bilden das verallgemeinerte Wissen zu bestimmten Anwendungsfällen ab. Sie sollen dem Anwender als Hilfestellung zur Erreichung eines Ziels dienen. Der Ausgang der Anwendung einer Methode ist offen. (Oerding, 2009)
Modell der PGE	Als Produktgenerationsentwicklung wird die Entwicklung technischer Produkte verstanden, die sowohl durch die Anpassung von Teilsystemen (...) als Übernahmeariation (...) als auch durch eine Neuentwicklung von Teilsystemen charakterisiert ist. Darin eingeschlossen sind sowohl die Entwicklung einer neuen Produktgeneration, als auch deren Derivate bzw. Varianten. Die Anteile technischer Neuentwicklungen einzelner Funktionseinheiten können sowohl durch die Aktivität Gestaltvariation (...) als auch durch die Variation von Lösungsprinzipien – (...) als Aktivität Prinzipvariation (...) bezeichnet – erfolgen. Neue Produktgenerationen basieren immer auf Referenzprodukten, die große Bereiche der grundsätzlichen Struktur vorgeben. (Albert Albers, Bursac & Wintergerst, 2015)
Objektsystem	Objektsysteme sind Artefakte, also materielle und immaterielle Ergebnisse des Handlungssystems. Der Zweck eines Objektsystems ist im korrespondierenden Zielsystem beschrieben. In der Produktentstehung muss zu jedem Objekt- ein dazugehöriges Zielsystem existieren. Objektsysteme erfüllen in der Produktentstehung drei Bedeutungen: Sie sind Ressourcen, Erkenntnisobjekte und Ergebnisse des Handlungssystems. Am Ende einer erfolgreichen Entwicklung entspricht das zu entwickelnde Objektsystem dem Produkt. Zwischenergebnisse in der Produktentstehung sind Objektsysteme von Subzielsystemen. Bis zum fertigen Produkt dienen Objektsysteme dem Erkenntnisgewinn oder als Ressource im Handlungssystem. (Meboldt, 2008)

Referenzsystem	Das Referenzsystem für die Entwicklung einer neuen Produktgeneration ist ein System, dessen Elemente bereits existierenden oder bereits geplanten sozio-technischen Systemen und der zugehörigen Dokumentation entstammen und Grundlage und Ausgangspunkt der Entwicklung der neuen Produktgeneration sind.(Albert Albers et al., 2019)
System-in-Development	<p>Auf das physische und/oder virtuelle (Teil-)System, welches im Fokus einer spezifischen Validierungsaktivität im Sinne des IPEK-X-in-the-Loop-Ansatz ist und über welches dabei Erkenntnisse gewonnen werden sollen, lassen sich im Sinne der unterschiedlichen konkreten Validierungsaktivitäten zwei unterschiedliche Sichten ableiten.</p> <p>Zum einen und überwiegend wird das (Teil-)System beschrieben, das sich in der Entwicklung befindet bzw. welches ein Entwickler zu verantworten hat. Dieses wird demnach aus Sicht eines Entwicklers oder eines Entwicklungsteams definiert und man spricht von dem System-in-Development (SiD). Der Fokus liegt primär auf der Eigenschafts- bzw. Funktionserfüllung.</p> <p>Steht nicht die Entwicklung eines Teil-Systems, sondern die Gewinnung von Wissen über ein Teil-System zur weiteren Verwendung im Entwicklungsprozess im Fokus, ist der Begriff System-under-Investigation (Sul) zweckmäßiger. (Albert Albers, Behrendt et al., 2016)</p>
System-under-Investigation	<p>Auf das physische und/oder virtuelle (Teil-)System, welches im Fokus einer spezifischen Validierungsaktivität im Sinne des IPEK-X-in-the-Loop-Ansatz ist und über welches dabei Erkenntnisse gewonnen werden sollen, lassen sich im Sinne der unterschiedlichen konkreten Validierungsaktivitäten zwei unterschiedliche Sichten ableiten.</p> <p>Steht die Gewinnung von Wissen über ein Teil-System zur weiteren Verwendung im Entwicklungsprozess im Fokus, ist der Begriff System-under-Investigation (Sul) zweckmäßig. Dann befasst sich der Entwickler bei einer konkreten Validierungsaktivität bspw. nur mit einer Untermenge des System-in-Development oder mit der Analyse früherer Produktgenerationen und/ oder mit (Teil-)Systemen, die nicht in seiner Verantwortung liegen, oder gar mit Wettbewerbsprodukten. Die Validierungsaktivitäten zielen dann primär auf die Analyse zum Erkenntnisgewinn ab.</p> <p>Für das (Teil-) System, das sich in der Entwicklung befindet bzw. welches ein Entwickler zu verantworten hat,</p>

	wird der Begriff System-in-Development verwendet. (Albert Albers, Behrendt et al., 2016)
Zielsystem	Ein Zielsystem beinhaltet alle expliziten Ziele eines zu entwickelnden Produktes, einschließlich derer Abhängigkeiten und Randbedingungen, innerhalb eines definierten Interessenbereichs (d.h. innerhalb eines System-of-Interest) zu einem bestimmten Zeitpunkt. (Lohmeyer, 2013)

