

Systemorientierung in der modellbasierten modularen E/E-Architecturentwicklung

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
STEFAN RAUE
aus Schwedt/Oder

Tübingen
2018

Tag der mündlichen Qualifikation: 08.02.2019

Dekan:	Prof. Dr. Wolfgang Rosenstiel
1. Berichterstatter:	Prof. Dr. Wolfgang Rosenstiel
2. Berichterstatter:	Prof. Dr. Stefan Schmerler

Danksagung

Diese Arbeit entstand mit der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Tübingen und der Abteilung E/E-Architektur und Standardisierung der Daimler AG.

Ich möchte mich bei allen herzlich bedanken, die zum Gelingen dieser Dissertation beigetragen haben.

Mein besonderer Dank gilt meinem Doktorvater Prof. Dr. Wolfgang Rosenstiel für die hervorragende wissenschaftliche Betreuung und den Freiraum in der Lösungsfindung. Seine Unterstützung hat stetig die Entwicklung der Ansätze voran gebracht und wesentlich zum Erfolg beigetragen.

Ebenso möchte ich Herrn Prof. Dr. Stefan Schmerler für die wertvollen Anregungen und die Übernahme des Zweitgutachtens danken. Seine Bereitschaft, diese Forschungstätigkeit in seiner Abteilung zu fördern, ermöglichte diese Arbeit.

Mein weiterer Dank gilt meinen Kolleginnen und Kollegen der Daimler AG. Insbesondere danke ich Dr. Bernd Hedenetz und Markus Conrath für die operative Einbindung, den Gestaltungsfreiraum, die Mittel, meine Konzepte zu operationalisieren und für das Korrekturlesen der Manuskripte. Yves Duhr und Alexander Lepple danke ich für Diskussionen und das Korrekturlesen und Thorsten Stecker für die praxisnahen und hilfreichen Impulse. Für die wertvollen Beiträge bedanke ich mich bei den Studenten Mina Assad, Laura Habib, Steffen Haid, Daniel Seidenspinner und Liu Zheng, die mich bei der Umsetzung meiner Ansätze unterstützten.

Für die spontane Hilfestellung bei den Details zum E/E-Architekturmodellierungswerkzeug danke ich den Experten Hussein Kobeisi und Steffen Naß der Vector Informatik GmbH.

Abschließend bedanke ich mich bei meinem engsten Umfeld für den Rückhalt in jeder Phase der Arbeit. Mein ganz besonderer Dank gilt meiner Frau, die mit ihrer stetigen Motivation, ihrem Verständnis und ihrem Einsatz eine große Stütze für mich war und ohne die ein Abschluss dieser Arbeit nicht möglich gewesen wäre. Meinen Eltern danke ich für ihre stetige moralische Unterstützung und das Korrekturlesen. Meinem Sohn Eythan und meiner Tochter Maya, die über meine normale Arbeit hinaus noch in der letzten Phase dieser Arbeit oft auf mich verzichten mussten, danke ich für ihr Verständnis.

Kurzfassung

In modernen Kraftfahrzeugen wird heute für fast jede neue Funktionalität verteilte Software eingesetzt. Sie kommt auf E/E-Komponenten zur Ausführung und ist damit Teil von verteilten Elektrik/Elektronik (E/E)-Systemen, die eine Kunden- oder eine technische Funktionalität erbringen. Durch die Zunahme der verteilten Software-basierten Fahrzeugfunktionen ist heute die Kontrolle der Komplexität im Fahrzeug eine Kernherausforderung [54, 126].

Gleichzeitig müssen Fahrzeughersteller mit immer schnelleren Innovationszyklen umgehen [24, 54]. Es werden Plattform- und Modulstrategie-Ansätze mit dem Ziel eingesetzt, möglichst gemeinsame Komponenten über viele Fahrzeuge hinweg zu verwenden [168], der Komplexität zu begegnen und Entwicklungs-Zeit und -Kosten zu reduzieren [46].

In diesem Spannungsfeld ist insbesondere die E/E-Architektur des Fahrzeugs betroffen. Sie verbindet alle E/E-Komponenten der E/E-Systeme und ermöglicht deren Interaktion. E/E-Architekturen werden modelliert mit dem Ziel E/E-Architekturen zu entwerfen, zu bewerten, zu dokumentieren und abzusichern. Die E/E-Architekturmodellierung wird eingesetzt um eine möglichst optimale Lösung zu finden [69, 159], wobei sich die Entwurfskriterien oft diametral gegenüberstehen [171]. Bei der Auslegung der E/E-Architektur ist die Absicherung der Interaktion der E/E-Komponenten, die über Software und Hardware erfolgt, ausschlaggebend. Die E/E-Komponenten sind einerseits in E/E-Systemen und andererseits in E/E-Modulen der Modulstrategien organisiert. E/E-Module werden durch die Modulstrategien in der mechanischen Fahrzeugentwicklung als Teil eines vereinfachten, schnelleren Produktionsprozess definiert und weiterentwickelt. Die Evolution (Weiterentwicklung) der E/E-Module wird in den Modulstrategien ausschließlich Hardware-orientiert, d. h. ohne Betrachtung der Software und Auswirkung auf diese durchgeführt. Für die E/E-Architektur wird die Interaktion der Software heute lediglich nach Best Engineering Practice bewertet und wird somit ohne durchgängigen Prozess, Werkzeugunterstützung und Methodik durchgeführt. Damit sind sobald die Software mit einbezogen wird die o. g. Ziele, die mit der E/E-Architekturmodellierung verfolgt werden, heute nicht umsetzbar.

Diese Arbeit hat das Ziel, die Herausforderungen für die zukünftige E/E-Architekturauslegung in diesem Spannungsfeld zu identifizieren und ein Lösungskonzept anzubieten. Dafür wird ein neuer systemischer Ansatz verfolgt. Es wird für die E/E-Architekturmodellierung untersucht, wie E/E-Systeme und E/E-Module einbezogen und die Evolution von E/E-Systemen einschließlich der E/E-Module durchgeführt werden kann.

Als Lösungskonzept wird mit dieser Arbeit das E/E-System Produktlinien Engineering vorgestellt. Das Konzept wird im industriellen Umfeld ausgearbeitet und löst den heutigen modulatorientierten Ansatz ab. Dabei

wird der heutige Ansatz um Funktionen und E/E-Systeme erweitert und die Abhängigkeiten zwischen den Funktionen, E/E-Modulen und E/E-Systemen erfasst. Dadurch wird eine effiziente und modellübergreifende Wiederverwendung von E/E-Systemen ermöglicht. Zudem wird auf Methoden der Software-Entwicklung zurückgegriffen um die Evolution von E/E-Systemen zu realisieren: Zweigeteilt wird hierbei über eine neue Modellierungsebene eine Bewertung einer Evolution im E/E-Architekturmodell vorgenommen und mit einem neuen Modell für die Funktionen eine Umsetzung der Evolution im E/E-Architekturmodell ermöglicht. Für das E/E-System Produktlinien Engineering werden Anwendungsfälle erarbeitet und prototypisch im Werkzeug für die E/E-Architekturmodellierung PREEvision implementiert. Mit einer praktischen Fallstudie werden die Anwendungsfälle evaluiert.

Mit den Konzepten dieser Arbeit werden erstmals die Ziele für die E/E-Architekturmodellierung nicht nur für die Hardware erreicht sondern eine vollständige Betrachtung eines E/E-Systems einschließlich der funktionalen Anteile (Software) realisiert. Ein weiterer Beitrag dieser Arbeit ist eine Steigerung der Modellierungseffizienz für fast alle Anwendungsfälle.

Insgesamt wird das E/E-System Produktlinien Engineering in einem industriellen Kontext im Geschäftsbereich Mercedes-Benz Cars der Daimler AG erarbeitet und evaluiert. Es werden konkrete Fragestellungen und Probleme aus der Praxis der E/E-Architecturentwicklung für Kraftfahrzeuge behandelt und E/E-Systeme und Szenarien aus der E/E-Architektur-Domäne Chassis/Fahrerassistenz herangezogen. Produktiv eingesetzt werden einige Konzepte des E/E-System Produktlinien Engineerings im E/E-Plattform-Entwicklungsprojekt STAR3, das die nächste Generation der Mercedes-Benz S-Klasse und nachfolgender Kraftfahrzeuge bei Mercedes-Benz Cars umfasst.

Inhaltsverzeichnis

Liste der Abbildungen	xi
Liste der Tabellen	xvii
Liste der Definitionen	xix
Liste der Anforderungen und Designentscheidungen	xxi
Liste der Anwendungsfälle	xxiii
Liste der Abkürzungen	xxv
1 Einleitung und Motivation	1
1.1 Bedeutung der Elektrik/Elektronik in der Entwicklung von Fahrzeugen	2
1.2 Zunehmende Komplexität von E/E-Architekturen und modellbasierter E/E-Architekturentwurf	3
1.3 Grenzen des heutigen Vorgehens	7
1.4 Zielsetzung der Arbeit	9
1.5 Gliederung der Arbeit	12
2 Grundlagen der Arbeit und Definitionen	15
2.1 Fahrzeugentwicklungsprozess	15
2.1.1 Allgemeiner Fahrzeugentwicklungsprozess in Kooperation mit Zulieferern	15

2.1.2	Entwicklung von E/E- und Software-Komponenten	17
2.2	E/E-Architekturen in der Fahrzeugentwicklung	22
2.2.1	Einflussfaktoren und Beschreibung von E/E-Architekturen	22
2.2.2	Für die E/E-Architekturauslegung relevante Vernetzungstechnologien	24
2.2.3	Abgrenzung weiterer E/E-Architekturbegriffe	26
2.2.4	E/E-Architekturen bei Mercedes-Benz Cars	27
2.3	Methoden zur Komplexitätsreduktion und Variantenerzeugung .	28
2.3.1	Baukastenprinzip	28
2.3.2	Plattformbauweise	29
2.3.3	Modularisierung	30
2.3.4	Mercedes-Benz Modulkonzept	31
2.4	Zusammenfassung	33
3	Stand der Technik	35
3.1	E/E-Architkturentwurf	35
3.1.1	E/E-Architkturentwurf in Relation zum Fahrzeugentwicklungsprozess	36
3.1.2	Kriterien bei der Auslegung einer E/E-Architektur in der Automobilindustrie	39
3.1.3	Ansätze zum Entwurf einer E/E-Architektur in der Automobilindustrie	40
3.2	E/E-Architekturmodellierung	42
3.2.1	Ziele der E/E-Architekturmodellierung	43
3.2.2	Ebenenmodell	44
3.2.3	Modellbasierte Bewertung und Absicherung von E/E-Architekturen bei Mercedes-Benz Cars	46
3.2.4	Variantenmanagement von E/E-Architekturen	49
3.2.5	Werkzeug für die E/E-Architekturmodellierung	51
3.3	Evolution in der Softwareentwicklung	52
3.3.1	Aspect weaving	52

3.3.2	Ontology-based modelling	53
3.3.3	Feature Oriented Refactoring (FOR)	53
3.4	Wiederverwendung beim Produktlinien Engineering	56
3.4.1	Produktlinien in der Entwicklung von Software	57
3.4.2	Modellierung von Merkmalen mittels Feature-Oriented Domain Analysis (FODA)	59
3.4.3	Ansätze für Produktlinien in der modellbasierten E/E- Architekturmodellierung	61
3.5	Zusammenfassung	65
4	Konzeption eines E/E-System Produktlinien Engineerings	67
4.1	Bewertung des Standes der Technik und Ableitung von Anfor- derungen	67
4.1.1	Evolution von E/E-Systemen	69
4.1.2	Modellstruktur und Variantenmanagement für E/E- Systeme	73
4.1.3	Wiederverwendung von E/E-Systemen	76
4.1.4	Potentiale bei der Einführung von E/E-Systemen in die E/E-Architekturmodellierung	78
4.2	Analyse der E/E-Architekturmodellierung und Ableitung wei- terer Anforderungen	78
4.2.1	Analyse der Funktionsnetzmodellierung	79
4.2.2	Analyse der Evolution von E/E-Systemen	87
4.3	Analyse von Lösungsansätzen und Ableitung von Designent- scheidungen für ein Konzept	91
4.3.1	Bewertung der Evolution von E/E-Systemen durch Ab- straktion	91
4.3.2	Umsetzung der Evolution von E/E-Systemen durch Transfer von Methoden der Software-Evolution	97
4.3.3	Neugestaltung der Modellstruktur und des Variantenma- nagements für E/E-Systeme	98

4.3.4	Wiederverwendung von E/E-Systemen durch Transfer und Anpassung des Produktlinien Engineering Prozesses	101
4.4	E/E-System Produktlinien Engineering Konzept	104
4.4.1	Entwurf der Konzeptionellen Funktionsarchitektur- Ebene zur Bewertung der Evolution von E/E-Systemen	105
4.4.2	Entwurf des Feature Oriented Architecture Refactorings (FOAR) zur Umsetzung der Evolution von E/E-Systemen	108
4.4.3	Entwurf eines E/E-Systemmodells für Modellstruktur und Variantenmanagement von E/E-Systemen	111
4.4.4	Entwurf des Teilkonzepts zur Wiederverwendung von E/E-Systemen	114
4.4.5	Anwendungsfälle für das E/E-System Produktlinien En- gineering	117
4.5	Zusammenfassung	124
5	Prototypische Umsetzung des E/E-System Produktlinien En- gineerings	127
5.1	Evolution von E/E-Systemen	127
5.1.1	Operation Erstellung	129
5.1.2	Operation Entfernung	132
5.1.3	Operation Verschiebung	133
5.1.4	Operation Spaltung	134
5.1.5	Operation Fusion	136
5.2	Modellstruktur und Variantenmanagement für E/E-Systeme	141
5.2.1	Modell der Konzeptionellen Funktionsarchitektur-Ebene	141
5.2.2	E/E-Systemmodell	142
5.2.3	Erstellung von E/E-Architekturmodell-Varianten	142
5.2.4	Propagation	144
5.3	Wiederverwendung von E/E-Systemen	145
5.3.1	Dynamische Erstellung von E/E-System-Modellcontainern	145
5.3.2	Integration von E/E-Systemen	146

5.3.3	Austausch von E/E-Systemen	147
5.4	Umsetzung im Werkzeug für die E/E-Architekturmodellierung .	149
5.4.1	Integration der neuen Modellstruktur	150
5.4.2	Werkzeugunterstützung für die Anwendungsfälle	156
5.5	Zusammenfassung	166
6	Evaluierung	171
6.1	Fallbeispiel der Arbeit	171
6.1.1	E/E-Architektur-Domäne Chassis/Fahrerassistenz	171
6.1.2	Radarsysteme	173
6.1.3	Untersuchung der Konzepte am Fallbeispiel	175
6.2	Qualitative Evaluierung	177
6.2.1	Realisierung der Ziele der E/E-Architekturmodellierung .	179
6.2.2	Automatisierungsstufe der allgemeinen Anforderungen .	180
6.2.3	Nachverfolgbarkeit	182
6.2.4	Robustheit	183
6.2.5	Modellqualität	184
6.3	Quantitative Evaluierung mittels einer Fallstudie	185
6.3.1	Evolution von E/E-Systemen	187
6.3.2	Variantenkonfiguration im Variantenmanagement	192
6.3.3	Wiederverwendung von E/E-Systemen	195
6.3.4	Evaluierung der Automatisierung	200
6.4	Zusammenfassung und Diskussion	204
7	Zusammenfassung und Ausblick	209
A	Modularisierung	215
A.1	Effekte der Modularisierung	215
A.2	Klassen von Produktarchitekturen	216
	Literatur	219

Liste der Abbildungen

1.1	Entwicklung der Anzahl von ECUs, Bussystemen, Signalen und Bandbreite	4
1.2	Gegenüberstellung der heutigen E/E-Architekturmodellierung und dem E/E-System Produktlinien Engineering	11
2.1	Produktlebenszyklus und Entwicklung	17
2.2	Deutsche und japanische Zuliefererketten im Vergleich	18
2.3	AUTOSAR ECU Software-Architektur	20
2.4	Modellbasierte Funktionsentwicklung	21
2.5	Das V-Modell für die Kooperation zwischen OEM und Zulieferer	22
2.6	Schematische Darstellung der vernetzten E/E-Komponenten verortet in der Topologie des Fahrzeugs	28
2.7	Ordnungsebenen für Module	33
3.1	E/E-Architekturentwicklung im Fahrzeugentwicklungsprozess . .	37
3.2	Beispiele für den Einsatz der E/E-Architekturmodellierung . . .	43
3.3	E/E-Architektur-Ebenen der EEA-ADL	47
3.4	Referenzprozess für die Entwicklung von Software-Produktlinien	58
3.5	Modulorientierter Produktlinienansatz zur Wiederverwendung von E/E-Modulen in der E/E-Architekturentwicklung	63
4.1	Überblick zur Konzeption des E/E-System Produktlinien Engineerings (Kapitel 4)	68
4.2	Heutige modulorientierte E/E-Architekturmodellierung	69

4.3	Gegenüberstellung der Vorgehen zur Modellierung der Skalierung von Funktionen	82
4.4	Heterogene Modellierung der Skalierung am Beispiel der Radar-Fahrerassistenzsysteme	84
4.5	Fortlaufender Einfluss von Innovationen und Modulstrategie . . .	86
4.6	Abhängigkeiten zwischen den E/E-Modulen durch Software	89
4.7	Beziehung der E/E-Systeme zu Funktionen und E/E-Modulen .	100
4.8	Erweiterung des Produktlinienansatzes der E/E-Architekturentwicklung	102
4.9	Konzeptionelle Funktionsarchitektur-Ebene und Relationsmodell	107
4.10	Funktionen-Modell	108
4.11	E/E-Systemmodell und Konfigurationsmodell	115
4.12	Konzept zur Wiederverwendung von E/E-Systemen	118
4.13	Anwendungsfälle für die Ebenenrelation (Anwendungsfälle 4.1, 4.2)	119
4.14	Anwendungsfälle für die Evolution von E/E-Systemen (Anwendungsfälle 4.3, 4.4, 4.5, 4.6, 4.7)	120
4.15	Erweiterung der heutigen E/E-Architekturmodellierung zum E/E-System Produktlinien Engineering	125
5.1	Aktivitätsdiagramm für das allgemeine Vorgehen bei den Anwendungsfällen zur Evolution (Anwendungsfälle 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7)	128
5.2	Aktivitätsdiagramm für die Erstellung von Funktionen (Anwendungsfall 4.3)	130
5.3	Aktivitätsdiagramm für die Erstellung von Ebenenrelationen (Anwendungsfall 4.1)	131
5.4	Aktivitätsdiagramm für die Erstellung von E/E-Systemen (Anwendungsfall 4.3)	132
5.5	Aktivitätsdiagramm für die Entfernung von Ebenenrelationen (Anwendungsfall 4.1)	133

5.6	Aktivitätsdiagramm für die Verschiebung von E/E-Systemen (Anwendungsfall 4.5)	135
5.7	Aktivitätsdiagramm für die Spaltung von Ebenenrelationen (Anwendungsfall 4.2)	136
5.8	Aktivitätsdiagramm für die Spaltung von Funktionen (Anwendungsfall 4.6)	137
5.9	Aktivitätsdiagramm für die Spaltung von E/E-Systemen (Anwendungsfall 4.6)	138
5.10	Aktivitätsdiagramm für die Fusion von Ebenenrelationen (Anwendungsfall 4.2)	139
5.11	Aktivitätsdiagramm für die Fusion von Funktionen (Anwendungsfall 4.7)	140
5.12	Aktivitätsdiagramm für die Fusion von E/E-Systemen (Anwendungsfall 4.7)	140
5.13	Aktivitätsdiagramm für die Erstellung einer E/E-Architekturmodell-Variante (Anwendungsfall 4.8)	143
5.14	Aktivitätsdiagramm für die Erstellung eines E/E-System-Modellcontainers als Grundlage für die Wiederverwendung eines E/E-Systems (Anwendungsfall 4.9)	146
5.15	Aktivitätsdiagramm für die Integration eines E/E-System-Modellcontainers zur Wiederverwendung des E/E-Systems (Anwendungsfall 4.9)	148
5.16	Aktivitätsdiagramm für den Austausch eines wiederverwendeten E/E-Systems (Anwendungsfall 4.10)	149
5.17	Implementierung der Modelle des E/E-System Produktlinien Engineerings in PREEvision	153
5.18	Propagationsregeln des E/E-System Produktlinien Engineerings für die Propagation ausgehend von Merkmalen in PREEvision	154
5.19	Der Wiederverwendungsmechanismus für das E/E-System Produktlinien Engineering in PREEvision	155

5.20	Framework für das E/E-System Produktlinien Engineering in PREEvision	157
5.21	Metrik <code>CreateFeature</code> für die Erstellung von Funktionen in PREEvision (Anwendungsfall 4.3)	158
5.22	Metrik <code>CreateInterfaceRelationship</code> und GUI für die Erstellung, Entfernung und Evolution einer Ebenenrelation in PREEvision (Anwendungsfälle 4.1, 4.2)	159
5.23	Metrik <code>New</code> für die Erstellung von E/E-Systemen in PREEvision (Anwendungsfall 4.3)	160
5.24	Metrik <code>Move</code> für die Verschiebung von Funktionen eines E/E-Systems in PREEvision (Anwendungsfall 4.5)	161
5.25	Metrik <code>SplitFeature</code> für die Abspaltung von Funktionsumfängen in eine separate Funktion in PREEvision (Anwendungsfall 4.6)	162
5.26	Metrik <code>Split</code> für die Abspaltung eines E/E-Systems in PREEvision (Anwendungsfall 4.6)	163
5.27	Metrik <code>MergeFeatures</code> für die Fusion von Funktionen in PREEvision (Anwendungsfall 4.7)	163
5.28	Metrik <code>Merge</code> für die Fusion von E/E-Systemen in PREEvision (Anwendungsfall 4.7)	164
5.29	Konfiguration und Absicherung von E/E-Architekturmodellvarianten in PREEvision (Anwendungsfall 4.8)	165
5.30	Metrik <code>CreateReuseUnit</code> für die Erstellung einer <code>ReuseUnit</code> für gegebenes E/E-System in PREEvision (Anwendungsfälle 4.9, 4.10)	166
5.31	Metrik <code>IntegrateReuseUnit</code> für die Integration einer <code>ReuseUnit</code> in eine <code>Productline</code> in PREEvision (Anwendungsfälle 4.9, 4.10)	167
5.32	Implementierung des E/E-System Produktlinien Engineerings	168
6.1	Sensorik der Fahrerassistenzsysteme	173

6.2	Die Konzeptionelle Funktionsarchitektur-Ebene als Daten- grundlage für die Evaluierung	176
6.3	Das Konfigurationsmodell als Datengrundlage für die Evaluierung	178
6.4	Qualitativer Vergleich des Automatisierungsgrads der allgemei- nen Anforderungen	180
A.1	Klassen von Produktarchitekturen	218

Tabellenverzeichnis

4.1	Bewertung zur Erfüllbarkeit der Ziele der E/E-Architekturmodellierung für die E/E-System-Evolution	71
4.2	Bewertung des Wiederverwendungsmechanismus zur Wiederverwendung von Hardware und Software	77
4.3	Vergleich und Bewertung der Vorgehensweisen zur Modellierung von Skalierung	83
4.4	Bewertung der Detaillierungsstufe für E/E-Architektur-Fragestellungen	95
4.5	Bewertung der Methoden der Evolution aus der Softwareentwicklung	98
4.6	Vergleich der Anwendungsfälle der modulatorientierten E/E-Architekturmodellierung mit neuen Wiederverwendungsmechanismen	103
6.1	Übersicht der qualitativen Evaluierung des E/E-System Produktlinien Engineerings	179
6.2	Vergleich der Modellierungsdauer für das Erstellen von neuen Funktionsumfängen (Anwendungsfall 4.3)	189
6.3	Vergleich der Modellierungsdauer für die Fusion (Anwendungsfall 4.7)	190
6.4	Vergleich der Modellierungsdauer für das Aufspalten (Anwendungsfall 4.6)	191
6.5	Vergleich der Modellierungsdauer für das Verschieben (Anwendungsfall 4.5)	192

6.6	Vergleich der Modellierungsdauer für das Entfernen (Anwendungsfall 4.4)	193
6.7	Vergleich der Modellierungsdauer für die Variantenkonfiguration (Anwendungsfall 4.8)	196
6.8	Vergleich der Modellierungsdauer für das Wiederverwenden (Anwendungsfall 4.9)	199
6.9	Vergleich der Modellierungsdauer für das Austauschen (Anwendungsfall 4.10)	201
6.10	Modellierungsdauer bei manuellem Vorgehen gegenüber Automatisierung	203

Liste der Definitionen

2.1	Definition (E/E-Komponente)	17
2.2	Definition (Software-Komponente)	19
2.3	Definition (Software-Komposition)	19
2.4	Definition (E/E-System)	19
2.5	Definition (E/E-Architektur)	22
2.6	Definition (E/E-Architektur-Konzept)	26
2.7	Definition (E/E-Architektur-Variante)	26
2.8	Definition (E/E-Architektur-Plattform)	26
2.9	Definition (E/E-Architektur-Domäne)	26
2.10	Definition (Baukasten)	28
2.11	Definition (Gleichteile)	29
2.12	Definition (Verblockung)	29
2.13	Definition (Rückverblockung)	29
2.14	Definition (Fahrzeug-Plattform)	29
2.15	Definition (Modul)	30
2.16	Definition (Modulstrategie)	31
2.17	Definition (Modulusprägung)	32
2.18	Definition (Dynamisierung)	32
2.19	Definition (Modulzyklusplan)	32
2.20	Definition (Modulheft)	32
2.21	Definition (Submodul)	32
2.22	Definition (E/E-Modul)	33

3.1	Definition (Propagation)	50
3.2	Definition (Merkmal)	54
3.3	Definition (Basis-Modul)	54
3.4	Definition (Derivat-Modul)	54
3.5	Definition (Software-Produktlinie)	57
3.6	Definition (Domain Engineering)	58
3.7	Definition (Application Engineering)	58
3.8	Definition (Variationspunkt)	60
3.9	Definition (Variante)	60
3.10	Definition (Orthogonales Variabilitätsmodell)	60
3.11	Definition (Merkmalsmodell)	61
4.1	Definition (Konzeptionelle Funktionsarchitektur-Ebene)	105
4.2	Definition (Relationsmodell)	106
4.3	Definition (Funktionen-Modell)	108
4.4	Definition (Basis-Beitrag)	108
4.5	Definition (Derivat-Beitrag)	109
4.6	Definition (+ - Operator)	109
4.7	Definition (• - Operator)	109
4.8	Definition (E/E-Systemmodell)	112
4.9	Definition (E/E-Modulmodell)	112
4.10	Definition (Konfigurationsmodell)	112
4.11	Definition (E/E-System-Modellcontainer)	114
4.12	Definition (E/E-System-spezifische Modellartefakte)	116
4.13	Definition (E/E-System-unspezifische Modellartefakte)	116

Liste der Anforderungen und Designentscheidungen

4.1	Anforderung (Evolution von E/E-Systemen)	73
4.2	Anforderung (E/E-Systeme in der E/E-Architekturmodellierung)	75
4.3	Anforderung (Variantenmanagement mit E/E-Systemen)	76
4.4	Anforderung (Wiederverwendung für E/E-Systeme)	77
4.5	Anforderung (Heterogenität des Funktionsnetzes)	80
4.6	Anforderung (Heterogenität der Skalierung)	83
4.7	Anforderung (Konkretisierung)	87
4.8	Anforderung (Unabhängigkeit vom Funktionsnetz)	90
4.9	Anforderung (Transparenz der Kopplung)	90
4.1	Designentscheidung (Abstrakte Ebene zur Bewertung der Evo- lution)	96
4.2	Designentscheidung (Verbindung zwischen abstrakter Ebene und Funktionsnetz)	96
4.3	Designentscheidung (Transfer des Feature-Oriented Refactoring)	98
4.4	Designentscheidung (Ausgestaltung des E/E-Systemmodells) . .	101
4.5	Designentscheidung (Transfer des Produktlinienengineerings) . .	104

Liste der Anwendungsfälle

4.1	Anwendungsfall (Herstellen und Entfernen der Ebenenrelation)	119
4.2	Anwendungsfall (Evolution der Ebenenrelation)	120
4.3	Anwendungsfall (Erstellen)	121
4.4	Anwendungsfall (Entfernen)	121
4.5	Anwendungsfall (Verschieben)	121
4.6	Anwendungsfall (Aufspalten)	122
4.7	Anwendungsfall (Fusion)	122
4.8	Anwendungsfall (Variantenkonfiguration mit E/E-Systemmodellen)	122
4.9	Anwendungsfall (Wiederverwendung von E/E-Systemmodellen)	123
4.10	Anwendungsfall (Austausch von E/E-Systemmodellen)	124

Liste der Abkürzungen

ALDW	Spurwechsel-Warnung, engl. <i>Active Lane Departure Warning</i>
API	Schnittstelle zur Anwendungsprogrammierung, engl. <i>Application Programming Interface</i>
BF	Bestätigungsfahrzeug
BSM	Totwinkel-Assistent, engl. <i>Blind Spot Monitoring</i>
CAN	Controller Area Network
CAN-FD	CAN Flexible Data-Rate
CASE	Rechnergestützte Softwareentwicklung, engl. <i>Computer-Aided Software Engineering</i>
CPA	Kollisionsvermeidungs-Assistent, engl. <i>Collision Prevention Assist</i>
DCU	Domänenkontroller, engl. <i>Domain Control Unit</i>
ECU	Steuergerät, engl. <i>Electronic Control Unit</i>
E/E	Elektrik/Elektronik
EEA	Elektrik/Elektronik-Architektur, engl. <i>Electrical/Electronic-Architecture</i>
EEPROM	Electrically Erasable Programmable Read-Only Memory
EF	Erprobungsfahrzeug
ESP	Elektronisches Stabilitätsprogramm
FAP	Fahrerassistenzpaket
FLH	Fahrzeuglastenheft
FOR	Merkmalsorientierte Refaktorisierung, engl. <i>Feature Oriented Refactoring</i>
FOAR	Merkmalsorientierte Architektur Refaktorisierung, engl. <i>Feature Oriented Architecture Refactoring</i>
GUI	Grafische Benutzeroberfläche, engl. <i>Graphical User Interface</i>
HWC	Hardware Komponente, engl. <i>Hardware Component</i>

KBL	Kabelbaumliste
KH	Konzeptheft
KLH	Komponentenlastenheft
K-Matrix	Kommunikations-Matrix
MDS	Mercedes-Benz Entwicklungssystem, engl. <i>Mercedes-Benz Development System</i>
MFA	Mercedes-Benz Frontantriebsarchitektur, engl. <i>Mercedes-Benz Front-Wheel Architecture</i>
MHA	Mercedes-Benz Hocharchitektur, engl. <i>Mercedes-Benz High Architecture</i>
MRA	Mercedes-Benz Heckantriebsarchitektur, engl. <i>Mercedes-Benz Rear-Wheel Architecture</i>
LIN	Local Interconnect Network
MOPF	Modellpflege
MOST	Media Oriented System Transport
OEM	Automobilhersteller, engl. <i>Original Equipment Manufacturer</i>
PKW	Personenkraftwagen
RDU	Radarauswerteeinheit, engl. <i>Radar Decision Unit</i>
RTE	AUTOSAR Laufzeitumgebung engl. <i>AUTOSAR Runtime Environment</i>
SA	Sonderausstattung
SDTR	Solo-Distronic
SLH	Systemlastenheft
SMPC	Stereo-Mehrzweckkamera, engl. <i>Stereo Multi Purpose Camera</i>
SOP	Produktionsstart, engl. <i>Start of Production</i>
STAR	Standard-E/E-Architektur
SWA	Software-Architektur
SWC	Software Komponente, engl. <i>Software Component</i>
SWComp	Software Komposition, engl. <i>Software Composition</i>
XML	Erweiterbare Auszeichnungssprache, engl. <i>Extensible Markup Language</i>

Kapitel 1

Einleitung und Motivation

Von den etwa 130 Jahren Fahrzeuggeschichte, die Carl Friedrich Benz im Jahre 1886 mit seinem Patent für den *Benz Patent-Motorwagen Nummer 1* begann, wurden nach stetiger Weiterentwicklung die letzten 40 Jahre davon auch mit Software zurückgelegt [23]. Traditionell waren die Systeme mechanisch bzw. später auch elektrisch geprägt. Wurde die Software anfangs nur lokal und für die Motorkontrolle verwendet, so wird heute für fast jede neue Funktionalität verteilte Software als Teil von Elektrik/Elektronik (E/E)-Systemen eingesetzt.

Mit der Zunahme an verteilten Software-basierten Fahrzeugfunktionen ist heute die Kontrolle der dadurch auch zunehmenden Komplexität im Fahrzeug eine Kernherausforderung [54]. Betroffen sind alle Entwicklungsphasen - von der Spezifikation der Anforderung und dem E/E-System-Design bis zum Integrieren und Testen der E/E-Systeme. Gravierender betroffen ist jedoch die sogenannte E/E-Architektur, denn sie verbindet alle E/E-Komponenten miteinander und ermöglicht deren Interaktion. Die Auslegung und Absicherung der E/E-Architektur wird heute modellbasiert, d. h. durch Modellierung der E/E-Architektur, durchgeführt.

Diese Arbeit zielt darauf ab, die Herausforderungen für eine zukünftige E/E-Architekturarbeit zu identifizieren und ein Lösungskonzept anzubieten mit dem den Herausforderungen begegnet werden kann. Sie entstand an der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen und in der Abteilung E/E-Architektur und Standardisierung der Daimler AG. Einige Einblicke und Teilkonzepte wurden in vorherigen Beiträgen vorgestellt [124, 125].

Der Rest dieses Kapitels detailliert die Bedeutung und die zunehmenden Komplexität von E/E im Fahrzeug. Danach werden die Grenzen des heutigen Vorgehens beschrieben und die Zielsetzungen dieser Arbeit formuliert bevor das Kapitel mit einer Übersicht über die Struktur der Arbeit schließt.

1.1 Bedeutung der Elektrik/Elektronik in der Entwicklung von Fahrzeugen

Seitdem neben den mechanischen Komponenten in der Fahrzeugentwicklung auch E/E eingesetzt wird, ist deren Bedeutung für OEM (Automobilhersteller, engl. *Original Equipment Manufacturer*) stetig gestiegen. Mittlerweile setzen E/E-Systeme in allen Bereichen des Fahrzeugs Funktionen mittels Software um - von der Sitzverstellung (Body&Comfort) über die Bremsregelung (Chassis/Fahrerassistenz) und die Abgasreinigung (Powertrain) bis zur Navigation (Infotainment).

E/E-Systeme sind so weit verbreitet, dass sich mechanische Innovationen mehr und mehr zu Software-Innovationen verlagern [135]. Schon heute ist Elektronik und im speziellen Software Hauptquelle für Innovationen [112] und zukünftig werden etwa 80% der Innovationen mit E/E-Systemen umgesetzt [25, 54, 96].

Die Umsetzung der Innovationen in der E/E führt dazu, dass deren Anteil an Entwicklungskosten von Oberklassefahrzeugen etwa 40% und in Hybriden sogar über 70% beträgt [44]. Aber auch in den Produktionskosten stellen in Oberklassefahrzeugen E/E-Systeme mit etwa 23% einen steigenden Kostenfaktor dar [96]. Hierbei wird Software zukünftig von dominierendem Stellenwert für die Fahrzeugkosten, weil sie E/E-Systeme billiger macht und zudem einige neue bzw. verbesserte Funktionalitäten nur mit Hilfe von Software umsetzbar sind [165].

Die Software im Fahrzeug unterscheidet sich von herkömmlicher Software. Fahrzeugsoftware von E/E-Komponenten ist mit einer Fehlerrate von z. B. etwa 1 Teil pro Million im Jahr hoch verlässlich ausgelegt und unterscheidet sich zudem von anderer Software durch Anforderungen an [112]:

- Zuverlässigkeit,
- funktionale Sicherheit,
- Echtzeit-Verhalten,
- minimale Ressourcen-Verwendung,
- Robustheit und
- mechatronische Steuerungen und Regelungen.

Die steigende Verwendung von Software-basierten Fahrzeugfunktionen hat dazu geführt, dass Software einen signifikanten Einfluss auf die Qualität des Fahrzeugs hat [54]: Die hohen Anforderungen an die Qualität sind neben der

erwähnten Fehlerrate auf die Zufriedenstellung der Kunden und die gesetzlichen Anforderungen zurückzuführen. Es werden zunehmend Schnittstellen zum Fahrer durch Software realisiert, weshalb Benutzerfreundlichkeit und Bedienkomfort zusätzlich immer wichtigere Qualitätsanforderungen werden.

1.2 Zunehmende Komplexität von E/E-Architekturen und modellbasierter E/E-Architekturentwurf

Ein modernes Oberklasse-Fahrzeug hat heute etwa 80 ECUs (Steuergerät, engl. *Electronic Control Unit*) verbunden über ein komplexes Netzwerk einschließlich bis zu 100 Millionen Codezeilen an Software [44, 112]. Eine Abschätzung für die Zukunft kann am Verlauf einiger E/E-relevanten Größen (Abbildung 1.1) gemacht werden. Es wird erwartet, dass die Anzahl von ECUs und Bussystemen nicht signifikant steigen werden. Ursache dafür ist hauptsächlich die Herausforderung, zusätzliche E/E-Komponenten im Fahrzeug zu platzieren. Hingegen spricht die erwartete Zunahme von Signal- und Bandbreitenbedarf (Abbildung 1.1) und der rapide Anstieg des Softwareumfangs von 8MB (1990) über 60MB (2003) bis auf 1000MB (2010) [25] für eine Zunahme von Software-basierten Fahrzeugfunktionen und deren Dynamik. Für die Zunahme der Dynamik spricht auch, dass es eine zunehmende Anzahl von E/E-Systemen gibt, die gänzlich ohne zusätzliche ECUs auf bestehende E/E-Systeme aufsetzen. Beispiele hierfür sind

- der Fahrerlebnisschalter, mit dem die Parametrierung von den E/E-Systemen Lenkung, Motorsteuerung, Federung und weiteren zwischen Sportlich, Komfortabel und Energieeffizient geändert werden kann oder
- das Aktivkomfort-System, das mit definierten Programmen der E/E-Systeme Klimatisierung, Beduftung und Ionisierung, Ambientes Licht, Sitz-Massage, -Beheizung und -Belüftung ansteuert und dazu Musik und Videos abspielt.

Zudem müssen OEMs eine hohe Variantenanzahl beherrschen. Zum einen bieten sie durch verschiedene Fahrzeugklassen mit vielen Fahrzeug-Varianten (z. B. Limousine oder Cabriolet) dem Kunden ein breites Fahrzeug-Portfolio an. Zum anderen werden meist durchgängig über das gesamte Fahrzeug-Portfolio hinweg Sonderausstattungen und verschiedene Motorvarianten angeboten. Die Variantentreiber sind hauptsächlich die variierenden Kundenwünsche aber zudem auch unterschiedliche gesetzliche Anforderungen in den Absatzmärkten [168]. Insgesamt werden die Fahrzeuge so stark individualisiert, dass statistisch gesehen unter Umständen keine zwei identischen Fahrzeuge produziert werden [34]. Selbst bei einem Kleinwagen, z. B. für den VW



Abbildung 1.1: Entwicklung der Anzahl von ECUs, Bussystemen, Signalen und Bandbreite. Die einzelnen Graphen zeigen die Entwicklung für einige E/E-relevante Größen im Vergleich unterschiedlicher Fahrzeuggenerationen der Mercedes-Benz E-Klasse (W120-W212). Rechts wird die Erwartung für die zukünftige Entwicklung skizziert. ECU Steuergerät (engl. *Electronic Control Unit*). Abbildung aus [66].

Fox, beträgt die Anzahl an durch den Kunden konfigurierbaren Varianten bereits $7,53 \cdot 10^{12}$ [86]. Nicht alle dieser Ausstattungen sind logisch kombinierbar. Zudem werden durch den Vertrieb Ausstattungslinien (z. B. AVANTGARDE, EXCLUSIVE, AMG Line) und Ausstattungspakete (z. B. Spur-Paket, Park-Paket, Fahrerassistenz-Paket Plus) gebündelt angeboten. Dies führt dazu, dass in der Praxis die Anzahl an verkauften Varianten geringer ist.

Insgesamt führen diese Punkte dazu, dass die Komplexität von E/E-Systemen in hoch ausgestatteten Fahrzeugen ein hohes Niveau erreicht hat [126] und die Kontrolle eine Kernherausforderung darstellt [54], da:

- alle Entwicklungsphasen betroffen sind [54],
- die Verantwortung von OEMs sich von dem Zusammenbau von Einzelteilen hin zur Integration von Systemen verschoben hat, vor allem weil die ursprünglich unabhängigen Funktionen heute in Beziehung zueinander stehen und interagieren [23],
- die Integration von Hard- und Softwarekomponenten zu einem verlässlichen Gesamtsystem erheblichem Aufwand erfordert [126],
- unvollständige oder volatile Anforderungen, schlecht spezifizierte und verwaltete Schnittstellen, Problemfeststellung erst in den letzten

Integrations- und Test-Phasen der Entwicklung vorhandene Herausforderungen bei der Entwicklung von E/E-Systemen sind [27] und

- die Kontrolle der Komplexität noch schwieriger ist, wenn mehrere Zulieferer an der Entwicklung des E/E-Systems beteiligt sind [54].

Zudem müssen die OEMs mit der steigenden Komplexität bei gleichzeitig immer schnelleren Innovationszyklen umgehen [24]. Insbesondere im Bereich Infotainment sind die Innovationszyklen inzwischen kürzer als die derzeitigen Entwicklungszyklen für neue Fahrzeugmodelle [54].

Um der Komplexität zu begegnen, verwenden OEMs einerseits Produktions- und Fertigungsorientierte Plattform- und Modulstrategie-Ansätze, so dass viele Komponenten über viele Fahrzeuge hinweg gemeinsam sind [168]. Durch diese Ansätze streben die Top 5-6 OEMs eine jährliche Kostenersparnis von 8-10 Milliarden Euro und 20-30% Reduktion von Entwicklungs-Zeit und -Kosten an [46]. Hierbei wird erwartet, dass die Volkswagen-Gruppe im Volumen-Segment und die Daimler AG im Oberklasse-Segment den höchsten Grad an Plattform-Standardisierung und das größte Sortiment an Fahrzeugen pro Plattform haben werden [46]. Die Plattformen werden über viele Jahre hinweg eingesetzt, weshalb die Fahrzeuge mit sehr vielen Alt-Komponenten und Alt-Strukturen zurecht kommen müssen [168].

Andererseits wird Software strategisch wiederverwendet. Diese lange bekannte Idee [117, 118] fand nach systematischer Erforschung [17, 113] in der Automobilindustrie eine Adaption. So werden Software-Produktlinienansätze verwendet und sind von zentraler Bedeutung [23, 54, 155]. Sie sind die beste Option [24] und die Technologie der Wahl für die Wiederverwendung im großen Maßstab [144]. Die Wiederverwendung ist sogar ein Wettbewerbsvorteil geworden und muss [24]:

- innerhalb des Fahrzeugs (z. B. keine Dopplung von Signalen),
- über Produktfamilien (z. B. gleiche Auslegung von E/E-Systemen über Baureihen hinweg) und
- über Fahrzeuggenerationen (z. B. Trennung von stabilen und innovationsabhängigen E/E-Systemanteilen) erfolgen.

Demgegenüber steht die Tatsache, dass es in der Praxis verteilte und heterogene Daten- und Entwicklungswerkzeugumgebungen für die E/E-Belange gibt [66]: Alle mechanischen Komponenten werden längst digital zu einem kompletten Fahrzeug integriert, weil die Gewissheit besteht, dass ohne einen integrierten digitalen Entwicklungsansatz heute kein Fahrzeug mehr entwickelbar ist. Für die E/E-Belange wird die gesamtheitliche Datenintegrität hingegen über schwache Austauschmöglichkeiten zwischen den Entwicklungswerkzeugen oder manuelle Austauschprozesse hergestellt. Zukünftig sehen Experten

deshalb, dass es entweder eine durchgängige Kopplung der Entwicklungswerkzeuge oder eine gesamtheitliche Datenquelle für die E/E-Belange geben muss um die gesamten E/E-Umfänge zu integrieren und die Gesamtkomplexität zu meistern [66].

Zweifellos wird in dieser Gemengelage die E/E-Architektur von steigender Wichtigkeit und Interesse für die OEMs [168]. Die E/E-Architektur verbindet die E/E-Komponenten miteinander, indem sie fahrzeugweit die Schnittstellen, die Struktur und die Interaktion der E/E-Komponenten beschreibt. E/E-Architekturen sind somit gravierend von

- dieser Vielzahl an ECUs, Bussystemen, Signalen,
- dieser hohen Anzahl an Fahrzeugvarianten,
- dieser steigenden Komplexität der E/E-Systeme,
- den Plattform- und Modulstrategie-Ansätzen einerseits und
- der strategischen Wiederverwendung von Software andererseits

betroffen, da all diese miteinander agierenden Faktoren direkt und frühzeitig aufschlagen. Für die Auslegung der E/E-Architektur versuchen OEMs möglichst optimale Lösungen zu finden [69, 159], was insbesondere dadurch erschwert wird, dass sich viele Entwurfskriterien diametral gegenüberstehen [171] (z. B. hohe Performance versus Kostenersparnis). Dementsprechend wird die Entwicklung der E/E-Architektur für das gesamte Fahrzeug über alle Bereiche hinweg durchgeführt [50].

Der Entwurf von E/E-Architekturen wird heute im Zuge der beschriebenen Komplexität modellbasiert durchgeführt. Mit einem E/E-Architekturmodell werden E/E-Architekturkonzepte in einer frühen Entwicklungsphase entworfen, bewertet, dokumentiert und abgesichert. So kann z. B. auch die Integrierbarkeit von Innovationen bewertet werden. Hierbei hat die E/E-Architekturmodellierung noch keine lange Tradition [127] und keine Standards [120]. OEMs haben eigene Lösungen entwickelt und sie mit kommerziell verfügbaren Werkzeugen kombiniert [127, 170]. Die heutige E/E-Architekturmodellierung ist mit steigendem Modellierungsaufwand konfrontiert. Dieser lässt sich auf

- die gestiegene Komplexität der E/E-Architekturmodellierung durch die genannten Faktoren,
- die steigenden Umfänge der E/E-Architekturmodelle bei heute etwa 1,5 Million Modellartefakten,
- die steigende Anzahl an E/E-Architektur-Konzepten und -Varianten, die notwendig sind um eine gute E/E-Architektur zu entwerfen und

- die Anzahl an parallel betriebenen E/E-Architekturmodellen

zurückführen.

Im Zuge dessen sind im Allgemeinen neue Konzepte für die Methodik und Struktur der E/E-Architekturmodellierung erforderlich um diese zentral, bau-reihenübergreifend und effizient zu gestalten. Zudem sind im Speziellen einige Fragestellungen auch nicht mit der heutigen Vorgehensweise beantwortbar, wie der nachfolgende Abschnitt verdeutlicht.

1.3 Grenzen des heutigen Vorgehens

Mit der E/E-Architekturmodellierung und dem Aufwand, der damit verbunden ist, werden verschiedene Ziele verfolgt. Hauptsächlich sollen E/E-Architekturen entworfen, bewertet, dokumentiert und abgesichert werden. Dazu gehört ebenfalls der Technologieentscheid, d. h. welche Vernetzungstechnologie wo eingesetzt werden soll, die Bewertung der Integrierbarkeit von Innovationen als auch die Bewertung bei Evolution, d. h. bei Weiterentwicklung, der E/E-Systeme.

Hierbei gibt es heute keine Vorgehensweise zur gesamtheitlichen Betrachtung von E/E Systemen im E/E-Architekturentwurf. Es werden mit der E/E-Architekturmodellierung lediglich Software in Form von Funktionen und Hardware als E/E-Module der Modulstrategie getrennt als Teilaspekte von E/E-Systemen erfasst. E/E-Module werden durch Modulstrategien in der mechanischen Fahrzeugentwicklung als Teil eines vereinfachten, schnelleren Produktionsprozess definiert. Das Zusammenspiel mit Funktionen von E/E-Systemen ist methodisch nicht durch die Modulstrategie abgedeckt. Für die E/E-Integration ist jedoch eine Absicherung dieses Zusammenspiels durch die E/E-Architekturmodellierung ausschlaggebend. Derzeit ist im E/E-Architekturmodell somit nicht ersichtlich, welche komplexen Abhängigkeiten zwischen den E/E-Modulen, hervorgerufen durch Funktionen, bestehen und welche der heute 1,5 Millionen Modellartefakte zu welchem E/E-System gehört, z. B. welches E/E-System welche Signale sendet und empfängt. Erschwerend kommt hinzu, dass die Evolution der E/E-Module durch die Methoden der Modulstrategie ausschließlich Hardware-orientiert d. h. ohne Betrachtung der Funktionen und Auswirkung bei Änderungen durchgeführt wird. Der heutige Produktlinienansatz der E/E-Architekturmodellierung, bei dem E/E-Module zentral modelliert, weiterentwickelt und in den verschiedenen E/E-Architekturen wiederverwendet werden, ist modulatorientiert und somit nicht ausreichend wenn die Funktionen von E/E-Systemen mit einbezogen werden sollen. Insgesamt führt dies zu folgendem Bild:

Evolution Die Evolution eines E/E-Systems kann nicht vollständig bewertet werden.

Modellstruktur und Variantenmanagement In der Modellstruktur sind die Modellartefakte eines E/E-Systems weit verteilt, werden ausschließlich durch Expertenwissen zusammengehalten und die Variantenkonfiguration für eine E/E-Architektur ist an Experten gebunden, fehleranfällig und kompliziert.

Wiederverwendung Es existiert keine Möglichkeit einer Wiederverwendung von E/E-Systemen, wodurch Funktionen heterogen und mehrfach modelliert werden.

Hierbei sind die Defizite, die Evolution von E/E-Systemen zu bewerten besonders gravierend, da im Rahmen der Modulstrategie alle E/E-Module in einem Takt von 18 Monaten weiterentwickelt werden und Innovationen signifikante Auswirkungen haben können. Die Bewertung wird heute lediglich nach *Best Engineering Practice*, d. h. ohne durchgängigen Prozess, Werkzeugunterstützung und Methodik, durchgeführt. Bei der Bewertung der Evolution von E/E-Systemen und der Abschätzung der Auswirkung auf die E/E-Architektur müssen insbesondere die funktionalen Fragen beantwortet werden:

Funktionale Fragestellungen bei der Evolution

Integrierbarkeit Ist die Evolution integrierbar und wie wird die Evolution in die E/E-Architektur eingebunden? Kommen beispielsweise neue Schnittstellen oder gar Funktionen hinzu? Entfallen Funktionen oder werden Funktionen verschoben?

Strukturänderungen Wie ändert sich die Struktur des Funktionsnetzes? Muss z. B. die bestehende Strukturierung in Funktionsbeiträge einer Funktion aufgebrochen und geändert werden, d. h. gibt es eine Funktionszerlegung? Wie muss die bisherige Software-Komponenten-Struktur der Implementierung zerlegt werden? Welche Funktionsbeiträge gehören zu welchem E/E-System? Gibt es von der Skalierung abhängig unterschiedliche Funktionsverteilungen?

Schnittstellenumfänge Welche ECU-internen Schnittstellen werden extern? Welche Signale ziehen um und werden einem anderen Bussystem zugeordnet? Gibt es Schnittstellenänderungen? Ist die Kommunikation auf den Bussystemen darstellbar und wie sind die Auswirkungen auf die Buslast? Gibt es ausreichende Reserven?

Umsetzung der Evolution Nachdem eine E/E-System-Evolution bewertet wurde, muss diese außerhalb der E/E-Architekturmodellierung in den jeweiligen Fachbereichen umgesetzt werden. Damit ergibt sich eine neue Klasse

an Herausforderungen, insbesondere für Schnittstellenänderungen. Die Änderung von Schnittstellen stellt sich als schwierig dar, da eine hohe Anzahl an Fachbereichen involviert ist und zudem die Fachbereiche unterschiedlichen Organisationseinheiten angehören. Es lässt sich hierbei durchaus unterscheiden, ob die Schnittstellenänderung nur eine Fahrzeugdomäne, z. B. die Domäne Chassis/Fahrerassistenz, betrifft oder domänenübergreifende Schnittstellen geändert werden sollen. Letzteres ist meist mit signifikant höherem Aufwand verbunden.

Überdies ist oft unklar, welche Funktionen genau betroffen sind und wie groß der Aufwand bei einer Änderung der Funktion ist. Zudem können zusätzliche Rückwirkungen auf weitere Funktionen entstehen oder sich Beschränkungen durch Verblockungen abzeichnen. Insgesamt ergibt sich somit ein intensiver Klärungsbedarf. Dieser bringt oft einen großen Einsatz von Kapazitäten mit sich. Teilweise ist es sogar notwendig, den Zulieferer mit einzubeziehen, der dafür auch Kapazitäten bereitstellen muss.

Diese durch die Verteiltheit und Komplexität großer Unternehmensorganisationen hervorgerufenen Herausforderungen in der Umsetzung einer E/E-System-Evolution soll nicht Betrachtungsumfang dieser Arbeit sein.

1.4 Zielsetzung der Arbeit

Insgesamt sind bei Einbeziehung der Software die Ziele, die mit der E/E-Architekturmodellierung verfolgt werden, heute nicht mehr umsetzbar. Gleichzeitig steigt die Komplexität und Dynamik in der Software aber weiter an.

Ein großer Schwerpunkt hierbei ist die Evolution von E/E-Systemen, da die heutige E/E-Architekturmodellierung keine ausreichende Entscheidungsgrundlage darstellt um die Evolution von E/E-Systemen zu bewerten und abzusichern. In dieser Arbeit wird deshalb untersucht, wie die Evolution der E/E-Systeme in der E/E-Architekturmodellierung durchgeführt werden kann (Zielsetzung 1.1).

Zielsetzung 1.1 (Bewertung und Umsetzung der Evolution von E/E-Systemen in der E/E-Architekturmodellierung): Die Evolution von E/E-Systemen soll in der E/E-Architekturmodellierung vollständig bewertet und umgesetzt werden.

Um die Varianz und Komplexität in der E/E-Architecturentwicklung zukünftig zu beherrschen, ist eine vollständige Betrachtung aller Aspekte eines E/E-Systems notwendig. Ein E/E-System umfasst alle notwendigen Artefakte zur Erbringung einer Kunden- oder technischen Funktionalität. In dieser Arbeit wird deshalb untersucht, wie E/E-Systeme in die E/E-Architekturmodellierung einbezogen werden können (Zielsetzung 1.2).

Zielsetzung 1.2 (Einbindung von E/E-Systemen in die E/E-Architekturmodellierung): E/E-Systeme sollen in die Methoden, Modelle, Prozesse und Werkzeuge der E/E-Architekturmodellierung einbezogen werden um die funktionale Sicht und die Hardware-Sicht (E/E-Module) zusammenzuführen und eine vollständige Betrachtung aller Aspekte eines E/E-Systems ermöglichen.

Als Lösungskonzept wird das E/E-System Produktlinien Engineering im industriellen Umfeld ausgearbeitet (Abbildung 1.2). Produktlinien wurden in der Softwaretechnik zur Beherrschung von Variabilität und Komplexität eingeführt. Hierbei unterteilt man in [17]

- Domain Engineering zur Definition und Realisierung der Gemeinsamkeiten und der Variabilität der Software-Produktlinie und in
- Application Engineering zur Erstellung der Applikationen der Software-Produktlinie unter Wiederverwendung von Artefakten aus dem Domain Engineering und Ausnutzen der Variabilität der Software-Produktlinie.

Es wurde gezeigt, dass ein Produktlinienansatz auch für die E/E-Architekturmodellierung machbar ist [81, 82, 83, 84]. Dementsprechend wird das bestehende modulatorientierte Produktlinien Engineering um Funktionen und E/E-Systeme erweitert. Mit der Erweiterung des Produktlinienansatzes für die E/E-Architekturmodellierung werden Abhängigkeiten zwischen den Funktionen, E/E-Modulen und E/E-Systemen erfasst und eine zentrale und gekoppelte Wiederverwendung von E/E-Systemen ermöglicht. Damit erfolgt die Modellierung von E/E-Systemen modellübergreifend und mit einer effizienten Wiederverwendung. Für die Evolution von E/E-Systemen wird zusätzlich auf weitere Methoden der Software-Entwicklung zurückgegriffen und hinsichtlich ihrer Übertragbarkeit auf die E/E-Architekturmodellierung untersucht. Als Kern für die Evolution von E/E-Systemen erlaubt das E/E-System Produktlinien Engineering mit

- einer neuen Modellierungsebene, der neuen Konzeptionellen Funktionsarchitektur-Ebene, im E/E-Architekturmodell eine Bewertung einer Evolution und mit
- einem neuen Modell zur Erfassung der Funktionen, dem Feature Oriented Architecture Refactoring, eine zentrale Umsetzung der Evolution.

Für das E/E-System Produktlinien Engineering werden Anwendungsfälle erarbeitet und im Werkzeug für die E/E-Architekturmodellierung PREEvision prototypisch implementiert.

Mit dem E/E-System Produktlinien Engineering als Lösungskonzept zur Bewertung und Umsetzung der Evolution von E/E-Systemen (Zielsetzung 1.1) und zur Einbindung von E/E-Systemen (Zielsetzung 1.2) sollen für die zukünftige E/E-Architekturmodellierung folgende Mehrwerte erschaffen werden:

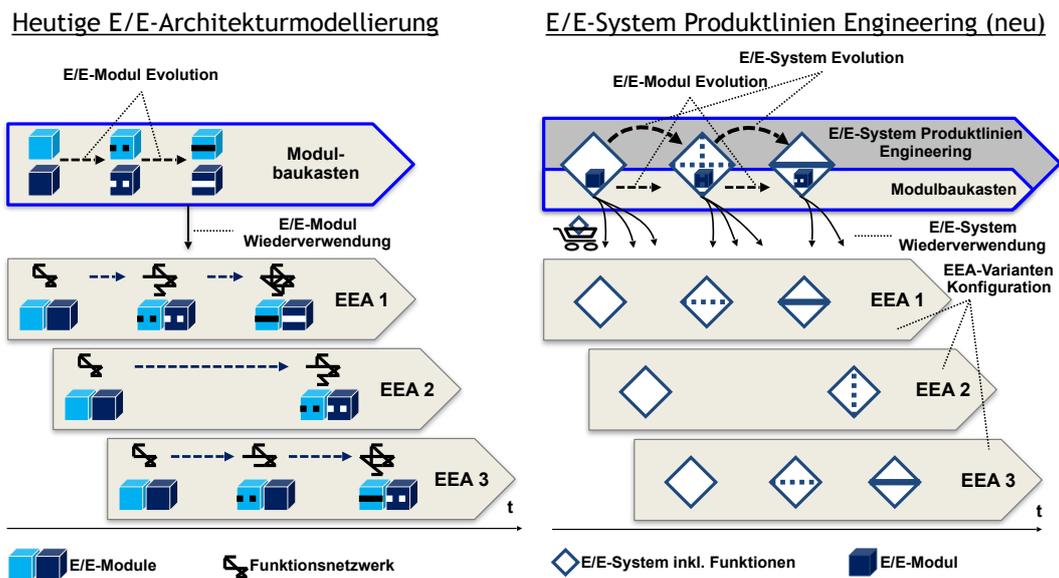


Abbildung 1.2: Gegenüberstellung der heutigen E/E-Architekturmodellierung und dem E/E-System Produktlinien Engineering. Die heutige E/E-Architekturmodellierung wird um eine Vorgehensweise zur Bewertung und Umsetzung der Evolution von E/E-Systemen (Zielsetzung 1.1) und eine Einbindung von E/E-Systemen (Zielsetzung 1.2) erweitert. Das Funktionsnetzwerk ist das Modell mit dem die verteilten Funktionen und alle ihre Funktionsbeiträge (Anteile der Funktion auf den E/E-Komponenten) beschrieben werden. Vgl. [125]. EEA E/E-Architektur.

- Aufzeigen einer Vorgehensweise um die Ziele der E/E-Architekturmodellierung zu realisieren, was wegen steigender Komplexität und Varianz für die E/E-Architekturmodellierung unverzichtbar ist,
- Reduktion des Modellierungsaufwands und der Fehleranfälligkeit, was wegen der Kosten bzw. Kapazitäten für die produktiv eingesetzte Modellierung von Wichtigkeit ist,
- Steigerung der Modellqualität, was wesentlich ist, da sie entscheidet, für welche Fragestellungen die E/E-Architekturmodellierung anstatt manueller Verwaltung eingesetzt werden kann.

Das E/E-System Produktlinien Engineering wird in einem industriellen Kontext erarbeitet und evaluiert. Dazu werden im Geschäftsbereich Mercedes-Benz Cars der Daimler AG konkrete Fragestellungen und Probleme aus der Praxis der E/E-Architecturentwicklung für Kraftfahrzeuge behandelt und E/E-Systeme und Szenarien aus der E/E-Architektur-Domäne Chassis/Fahrerassistenz herangezogen. Die Arbeit findet im Rahmen der E/E-Architektur-Plattform-Entwicklungsprojekte ES2 und STAR3 statt, die die nächste Generation der Mercedes-Benz S-Klasse und nachfolgender Kraftfahrzeuge bei Mercedes-Benz Cars umfassen. Da diese Arbeit unternehmensspezifische Prozesse für die Fahrzeug- und E/E-Architecturentwicklung berücksichtigt, sind bei einem Einsatz durch andere OEMs mitunter Anpassungen erforderlich. Aufgrund dieses industriellen Kontexts war es bei der Erarbeitung der Fragestellungen eine wichtige Richtlinie, ein Lösungskonzept zu finden, das in der Praxis anwendbar ist. Inwiefern die Konzepte dieser Arbeit bei der E/E-Architecturentwicklung nützlich sind und die erwarteten Mehrwerte erschaffen, wird anhand einer Fallstudie untersucht. In dieser wird ein Field-Application-Engineer des Modellierungswerkzeugherstellers als Modellierungsexperte mit Szenarien gemäß den erarbeiteten Anwendungsfällen konfrontiert und die Modellierung ausgewertet.

1.5 Gliederung der Arbeit

Die Arbeit setzt sich aus sieben Kapiteln zusammen, wobei dieses Kapitel das Umfeld und die Zielsetzungen dieser Arbeit definiert. Im nachfolgenden Kapitel 2 werden grundlegende Begriffe und Grundlagen der Fahrzeugentwicklung beschrieben. Schwerpunkte bilden hierbei der Fahrzeugentwicklungsprozess, E/E-Architekturen und Methoden zur Komplexitätsreduktion und Variantenerzeugung. In Kapitel 3 wird der Stand der Technik einschließlich des E/E-Architecturentwurfs, der E/E-Architecturentwicklung, der Evolution bei

der Softwareentwicklung und der Wiederverwendung bei der Produktlinienentwicklung vorgestellt.

In Kapitel 4 wird das E/E-System Produktlinien Engineering konzeptioniert. Aus der Bewertung des Stand der Technik und der Analyse der E/E-Architekturmodellierung werden dazu Anforderungen und Lösungsansätze beschrieben, Designentscheidungen getroffen und schließlich die verschiedenen Konzepte und Anwendungsfälle ausgearbeitet. In Kapitel 5 werden die Konzepte dieser Arbeit im E/E-Architekturmodell und im Werkzeug für die E/E-Architekturmodellierung in einem Framework umgesetzt.

In Kapitel 6 wird das E/E-System Produktlinien Engineering evaluiert. Dafür wird ein Vergleich mit der heutigen E/E-Architekturmodellierung und eine Fallstudie durchgeführt. Abschließend wird die Arbeit in Kapitel 7 zusammengefasst und ein Ausblick gegeben.

Kapitel 2

Grundlagen der Arbeit und Definitionen

Im nachfolgenden Kapitel werden die für diese Arbeit notwendigen Grundlagen für die Entwicklung von Fahrzeugen beschrieben. Zudem werden grundlegende Begriffe definiert.

2.1 Fahrzeugentwicklungsprozess

Dieser Abschnitt beschreibt den Fahrzeugentwicklungsprozess, AUTOSAR und das V-Modell als Grundlagen in der Fahrzeugentwicklung.

2.1.1 Allgemeiner Fahrzeugentwicklungsprozess in Kooperation mit Zulieferern

Fahrzeuge sind Produkte, die sich durch einen langen Produktlebenszyklus auszeichnen [138, S. 20f]. Nach der etwa fünfjährigen Entwicklung wird ein Fahrzeugmodell für etwa sieben Jahre produziert. Der Zeitpunkt, mit dem die Serienproduktion beginnt, wird als *Job no. 1* oder auch als SOP (Produktionsstart, engl. *Start of Production*) bezeichnet. An diese Produktionsphase schließt sich eine 10-15-jährige Betriebs- und Servicephase an.

Wie die Entwicklung konkret ausgestaltet ist, unterscheidet sich zwischen den OEM [147, S. 7]. So ist beispielsweise der Standardentwicklungsprozess für Produkte der Daimler AG das sogenannte MDS (Mercedes-Benz Entwicklungssystem, engl. *Mercedes-Benz Development System*) [42, 41, 93]. Das MDS legt Meilensteine für die Entwicklung fest, an denen definierte Entwicklungsziele und Absicherungen für die Freigabe durch das Management vorliegen müssen. Damit dient das MDS zur Feststellung und Kontrolle des Reifegrades

in der Entwicklung. Sollte es zu einer Verzögerung in einem Entwicklungsbereich kommen, so wird dies transparent und das Management kann reagieren.

Allgemein ist die Entwicklung von mehreren Phasen gekennzeichnet [93, 147, S. 7ff]:

- In einer frühen Strategiephase wird eine Zielgruppe für das Fahrzeug definiert und anhand dieser Anforderungen für umzusetzende Innovationen, kundenerlebbare Funktionen, Design, Qualität und Kosten festgelegt (Abbildung 2.1). Gleichzeitig gibt es Einflüsse durch die Gesetzgebung, Normen, Standardisierungen und den Wettbewerb. Die strategischen Anforderungen werden im sogenannten Konzeptheft dokumentiert, womit die Strategiephase beendet wird.
- In der sich anschließenden Konzeptphase werden anhand dieses Konzeptheftes die besten technischen Konzepte und Technologien für die Umsetzung ermittelt und entwickelt. Die Spezifikation für die Umsetzung erfolgt in Form eines sogenannten Fahrzeuglastenhefts.
- Abschließend folgt in der Serienentwicklungsphase die Entwicklung von Fertigungswerkzeugen und die Planung von Arbeitsabläufen für die Produktion. Parallel erfolgt eine Integration in Erprobungsfahrzeuge und eine spätere Validierung mit Bestätigungsfahrzeugen¹. Für eine letzte Optimierung der Qualität und der Produktionsprozesse wird vor der Serienproduktion eine seriennahe Produktion mit den entwickelten Fertigungswerkzeugen durchgeführt. Diese wird als Nullserie bezeichnet.

Während der laufenden Produktionsphase ist es üblich, weitere Optimierungen und auch Innovationen zu integrieren. An sogenannten Änderungsjahren erfolgen kleinere Modifizierungen. Kundenwahrnehmbare Änderungen wie z. B. Veränderungen am Exterieur, Interieur, Navigations- oder Fahrerassistenzsystemen, Motoren oder Getriebe erfolgen hingegen zur sogenannten Modellpflege (MOPF). Wegen der Sichtbarkeit von designrelevanten Änderungen wird die MOPF auch als Facelift bezeichnet.

Die eigentliche Entwicklung verläuft insbesondere für E/E-Umfänge in starker Kooperation zwischen dem OEM und den Zulieferern [147, S. 3f]. Zur Klassifizierung der Zulieferer lässt sich die Zuliefererkette in mehrere Ränge (engl. *tier*) unterteilen. Zulieferer, die den OEM direkt mit Teilen beliefern, werden als Tier-1-Zulieferer bezeichnet. Diese werden wiederum von Tier-2-Zulieferern beliefert, z. B. von Halbleiterherstellern. Bei diesem Beispiel werden Chemikalien für die Herstellung von Halbleitern benötigt - diese werden hier von Tier-3-Zulieferern geliefert. Die Zuliefererketten unterscheiden sich zwischen

¹Die Erprobung und Validierung von Fahrzeugen in der Entwicklung führt zu einer sehr hohen Gesamtleistung eines Fahrzeugmodells. Beispielsweise wurde die Mercedes-Benz E-Klasse insgesamt mit über 30 Millionen Kilometern erprobt [93]

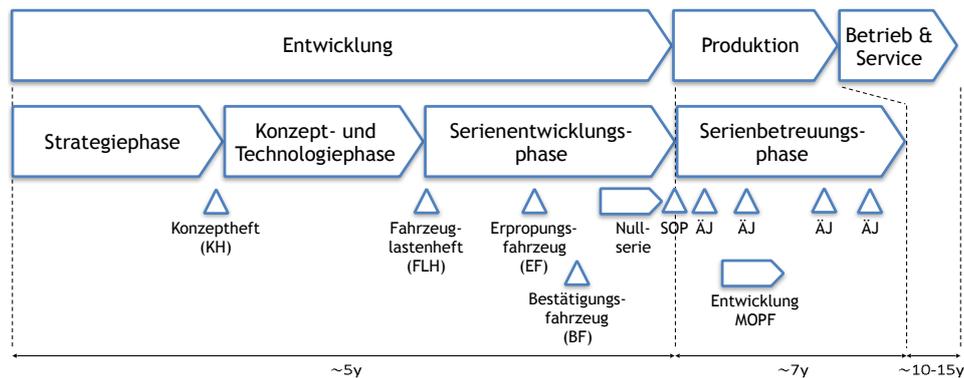


Abbildung 2.1: Produktlebenszyklus und Entwicklung eines Fahrzeugs. Drei Phasen kennzeichnen den Produktlebenszyklus eines Fahrzeug. Entwicklung wird sowohl in der Entwicklungsphase betrieben als auch in der laufenden Produktion. ÄJ Änderungsjahr; MOPF Modellpflege; SOP Start of Production (Produktionsstart)

den OEM [145]: Während deutsche OEM mit vielen Tier-1-Zulieferern kooperieren, bilden japanische OEM zu anderen OEM getrennte Kooperations- und Geschäftsbeziehungen mit ihren Zulieferern aus (jap. als *Keiretsu* bezeichnet, Abbildung 2.2).

Als Kooperationsmodell zwischen den deutschen OEM und den Tier-1-Zulieferern ist das V-Modell (Abschnitt 2.1.2) weit verbreitet [147, S. 10]. In der Regel werden die Anforderungen durch die OEM spezifiziert, die Entwicklung durch die Zulieferer durchgeführt und die Integration und die Abnahme wieder vom OEM vorgenommen. Die Schnittstelle, an der Verantwortlichkeiten übergehen, werden meist abhängig vom Projekt festgelegt [138, S. 129f].

2.1.2 Entwicklung von E/E- und Software-Komponenten

Das Kooperationsmodell zwischen Zulieferer und OEM dient zur Entwicklung von Komponenten. Es werden viele unterschiedliche Komponenten² entwickelt und zu einem Fahrzeug integriert. Komponenten können sowohl physischer Art (Hardware) als auch immaterieller Art (z.B. Software) sein.

Hardwareseitig stehen im Fokus dieser Arbeit E/E-Komponenten:

Definition 2.1 (E/E-Komponente): Eine E/E-Komponente ist eine elektrische oder elektronische Komponente, die an der Umsetzung von kundenerleb- baren oder technischen Funktionen beteiligt ist.

²von lat. *componens*, das Zusammensetzende

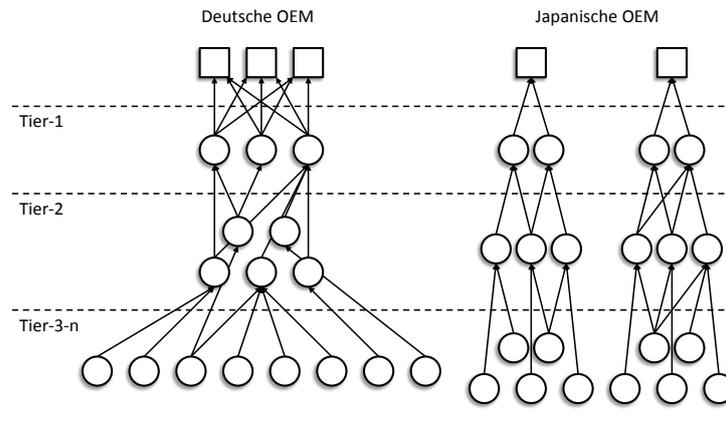


Abbildung 2.2: Deutsche und japanische Zuliefererketten im Vergleich. Während deutsche OEM grundsätzlich mit allen Zulieferern kooperieren, bilden japanische OEM voneinander getrennte Geschäftsbeziehungen mit ihren Zulieferern aus. Abbildung in Anlehnung an [145]. □ OEM, ○ Zulieferer.

Diese E/E-Komponenten sind Steuergeräte (ECU, engl. *Electronic Control Unit*), Sensoren oder Aktuatoren. Im Allgemeinen erfassen Sensoren das Fahrzeugumfeld unter Messen von physikalischen Größen (z.B. Geschwindigkeit) bzw. elektromagnetischen Größen (z.B. Radiowellen oder GPS-Signal). Die Messwerte werden zu elektrischen Signalen verarbeitet und von ECUs weiterverarbeitet um die Ansteuerung der Aktuatoren zu regeln. Der Aktuator setzt die von ECUs gesendeten elektrischen Signale in mechanische Bewegungen bzw. physikalische Größen um. Im Fahrzeug lassen sich die ECUs in mehrere Ausprägungen unterteilen. Während einerseits Sensor- bzw. Aktuator-ECUs direkt beim Sensor bzw. Aktuator verortet sind und diese ansteuern, gibt es andererseits lastfreie und sensorfreie Logik-ECUs.

AUTOSAR-Standardisierung in der Entwicklung

Auf allen ECUs wird Software zur Ausführung gebracht. Die Software-Architektur der ECUs wurde mit AUTOSAR (AUTomotive Open System Architecture) standardisiert [6]. AUTOSAR trennt die ECU-Basis-Software mit ihren hardwarenahen, ECU-spezifischen Software Modulen von der Anwendungssoftware. Diese Trennung wird durch die AUTOSAR Runtime Environment (RTE, dt. *AUTOSAR Laufzeitumgebung*) vorgenommen. Die RTE abstrahiert die Kommunikation und bietet unabhängig von der ECU-Hardware für inter- und intra-ECU-Kommunikation ein einheitliches Interface (Abbil-

dung 2.3). Über die RTE erfolgt somit die Kommunikation zwischen Software-Komponenten, die wie folgt definiert sind:

Definition 2.2 (Software-Komponente): Eine Software-Komponente (engl. *Software Component*, SWC) kapselt einen Teil einer Anwendung und ist somit an der Umsetzung von kundenerlebbaren oder technischen Funktionen beteiligt. Sie hat definierte Interfaces und ist mit anderen Software-Komponenten über diese verbunden. Eine Software-Komponente kann atomar sein, d. h. sie kann nicht auf mehrere ECUs aufgeteilt werden. Hierbei wird in Application-SWC, die einen Algorithmus implementiert, und Sensor-/Actuator-SWC, die einen Sensor bzw. Aktuator beschreibt, unterschieden.

Definition 2.3 (Software-Komposition): Eine Software Komposition (engl. *Software Composition*, SWComp) dient zur Strukturierung und verpackt zusammengehörende Application- und Sensor/Actuator-SWCs. Hierbei ist die SWComp nicht atomar und kann somit auf mehrere ECUs aufgeteilt werden.

Wie der auf der AUTOSAR-Infrastruktur zur Ausführung gebrachte Code einer SWC entsteht, wird durch AUTOSAR nicht vorgeschrieben. Der Code kann demnach sowohl geschrieben als auch modellbasiert entwickelt und automatisch generiert werden (Abbildung 2.4) [28, 35, 139]. Neben der Software-Architektur wird durch AUTOSAR eine Methodik für die Entwicklung von SWC und dazugehörige Datenaustauschformate standardisiert [6]. Durch diese Methodik und die standardisierten Datenaustauschformate wird die Zusammenarbeit mit den Zulieferern erleichtert.

Trotz der Standardisierung durch AUTOSAR gibt es AUTOSAR-Dialekte, die OEM-spezifisch sind und u. a. aufgrund von Rückverblockungsanforderungen existieren. Gegenstand aktueller Forschungen ist die virtuelle Integration modellbasierter Fahrzeugfunktionen unter AUTOSAR um die Qualität der Funktionsmodelle in der frühen Phase zu erhöhen [109, 110].

V-Modell in der Entwicklung

Im Zuge der Entwicklung kann der Zulieferer mit der Entwicklung von E/E-Komponenten, SWCs oder eines E/E-Systems (Definition 2.4) beauftragt werden.

Definition 2.4 (E/E-System): Ein E/E-System umfasst eine Menge von SWCs und E/E-Komponenten, die zusammen eine oder mehrere kundenerlebbare oder technische Funktionen umsetzen. Ein E/E-System hat dabei definierte Schnittstellen zu anderen E/E-Systemen.

E/E-Systeme werden meist evolutionär weiterentwickelt [9], wobei mit folgenden Herausforderung umgegangen werden muss [27]:

- unvollständige oder volatile Anforderungen,

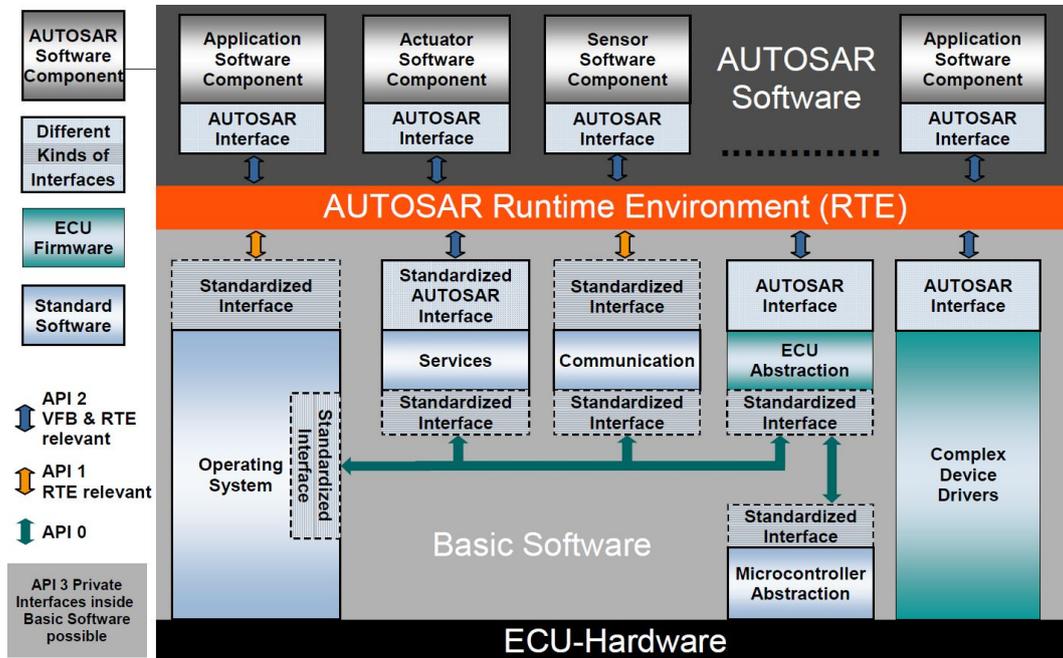


Abbildung 2.3: AUTOSAR ECU Software-Architektur. Die RTE (AUTOSAR Runtime Environment) abstrahiert die Basis-Software von der Anwendungssoftware. Die standardisierten Anteile der Basis-Software sind [6]: *Operating System* - die Softwareanteile eines AUTOSAR-Anforderungen erfüllendes Betriebssystem; *Services* - die System-Services wie z.B. Diagnose-Protokolle oder Speicher-Management; *Communication* - das Kommunikations-Framework mit u. a. dem I/O- und dem Netzwerk-Management, *Microcontroller Abstraction* - die Ebene mit Hardware-spezifischen Treibern zum Management der Mikrocontroller-Peripherie und der Bereitstellung der Mikrocontroller unabhängigen Werte. Dies schließt somit z. B. die digitale Ein- und Ausgabe, Analog-Digital-Wandler, EEPROM, Flash oder Laufzeitüberwachung (Watchdog Timer) mit ein. Hingegen sind die ECU-spezifischen Anteile der Basis-Software [6]: *ECU Abstraction* - die Abstraktionsebene für den Zugriff auf ECU inklusive Ein-/Ausgabe, Kommunikation oder Speicher abstrahiert davon, ob durch den Mikrocontroller oder Peripherie realisiert. Ferner *Complex Device Driver* - die Treiber für ressourcenkritische Anwendungen, die den Umgang mit komplexen Sensoren und Aktuatoren durch direkten Zugriff auf den Mikrocontroller erlauben. Abbildung aus [6].

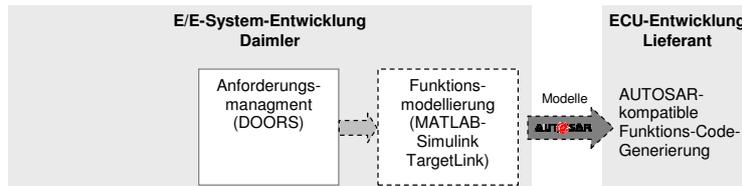


Abbildung 2.4: Modellbasierte Funktionsentwicklung. Die Anforderungen an Funktionen werden im Tool DOORS beschrieben und hinterlegt. Die Anforderungen umsetzenden Funktionen werden in MATLAB Simulink als Modell entwickelt und können über das standardisierte AUTOSAR-Datenaustauschformat an den Zulieferer übergeben werden. Dieser kann anschließend den ausführbaren Code generieren. Abbildung aus [139].

- schlecht spezifizierte und verwaltete Schnittstellen,
- Integrationstests finden Probleme zu einem späten Zeitpunkt in der Entwicklung,
- die Entwicklung und Analyse findet in getrennten Bereichen statt,
- ungenügende bereichsübergreifende Expertise und
- Koordination und Tracking über verschiedene Zulieferketten.

Ein weit verbreitetes Modell für die Zusammenarbeit zwischen OEM und Zulieferern bei der Entwicklung ist das V-Modell [21]. Der linke Ast des V betrifft die Spezifikation der zu entwickelnden Komponenten (Abbildung 2.5). Der OEM analysiert die Anforderungen und spezifiziert die Funktionen und Komponenten in System- und Komponentenlastenheften. Diese Spezifikationen werden an den Zulieferer übergeben, der die Funktion beziehungsweise die Komponente entwirft und implementiert. Während der Spezifikation werden außerdem Testfälle erstellt, mit denen die Funktionen und Komponenten getestet werden können. Das Ergebnis dieser Tests kann wiederum Lücken in der Spezifikation aufdecken. Diese werden dann geschlossen und die Implementierung entsprechend angepasst. Durchgeführt werden diese Tests auf der rechten Seite des V. Den Tests für die einzelnen Komponenten und Funktionen schließen sich Tests für die Integration mit anderen Komponenten und Funktionen an. Abschließend werden Tests in Fahrzeugen und Prototypen durchgeführt.

Die Übergabepunkte und Entwicklungsartefakte werden projektabhängig definiert. So kann der OEM beispielsweise wettbewerbsdifferenzierende Funktionen selbst entwickeln und dem Zulieferer für die Integration übergeben.

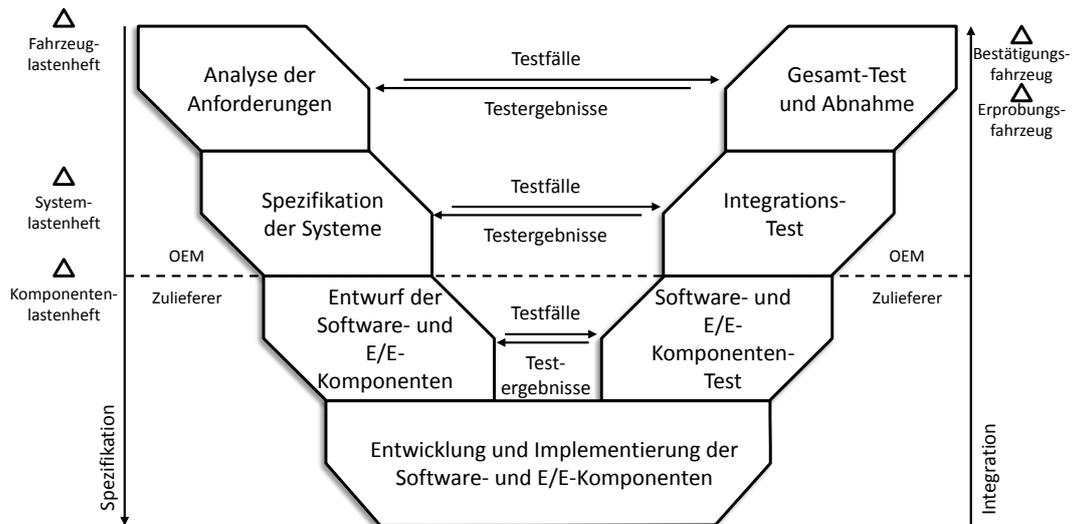


Abbildung 2.5: Das V-Modell für die Kooperation zwischen OEM und Zulieferer. Der linke Ast beschreibt die Spezifikation durch System- und Komponentenlastenhefte während der rechte Ast die Integration der zu entwickelnden Komponenten umfasst und Erprobungs- und Bestätigungsfahrzeuge hervorbringt. In der Spezifikation werden auch Tests für die Integration beschrieben, die wiederum Fehler in der Spezifikation aufdecken können. Projektabhängig werden die Übergabepunkte und Entwicklungsartefakte definiert.

2.2 E/E-Architekturen in der Fahrzeugentwicklung

Im nachfolgenden Abschnitt werden E/E-Architekturen im Fahrzeugkontext genauer beschrieben, relevante Vernetzungstechnologien aufgezeigt und E/E-Architekturen bei Mercedes-Benz Cars erörtert.

2.2.1 Einflussfaktoren und Beschreibung von E/E-Architekturen

Die Grundlage für eine Entwicklung von E/E-Systemen gemäß dem V-Modell ist die E/E-Architektur (Definition 2.5).

Definition 2.5 (E/E-Architektur): Eine E/E-Architektur beschreibt die Schnittstellen, die Struktur und die Interaktion von E/E-Komponenten.

E/E-Architekturen stehen im Spannungsfeld vieler, zum Teil gegensätzlicher Einflussfaktoren, wie z. B.:

- Innovationen (insbesondere im Premium-Segment) und verteilte Funktionen,

- hoher Varianz durch Aufbauvarianten, länderabhängiger Sonderausstattungen (SAs) und Gesetze,
- hoher Anzahl beteiligter Fachbereiche, ECUs, Funktionsentwickler, Signale, usw.,
- vorheriger E/E-Architekturen und E/E-Module (Abschnitt 2.3.4),
- unterschiedlicher Technologien.

Aufgrund dieser Einflussfaktoren ist eine Beschreibung einer E/E-Architektur nicht trivial. Zur Beschreibung der E/E-Architektur hat sich eine in Ebenen zerlegte Betrachtungsweise etabliert. Sie umfasst [18, 67, 147, S. 15ff]:

Logische E/E-Architektur Diese Ebene beschreibt den Funktionsumfang der Funktions- bzw. Softwarearchitektur.

Dafür werden Sensor-, Funktions- und Aktuatorblöcke und Verbindungen zwischen diesen verwendet. Die Blöcke werden hierbei durch ihre Schnittstellen charakterisiert.

Die Schnittstelle wird durch Art (Eingang oder Ausgang), Datentyp der Werte und Datenaufkommen beschrieben. Sensorblöcke beschreiben eine Datenquelle und haben überwiegend Ausgänge während Aktuatorblöcke eine Datensenke beschreiben und überwiegend Eingänge besitzen. Die Funktionsblöcke beschreiben die Verarbeitung von Eingangsdaten und die Ausgabe von Ergebnissen.

Zudem wird die Art der Kommunikation, d. h. Client-Server oder Sender-Receiver, beschrieben.

Technische E/E-Architektur Diese Ebene beschreibt die E/E-Komponenten, ihre Verbindungen und die Einbettung in das Fahrzeug, weshalb sie zusätzlich wie folgt unterteilt wird:

Vernetzungsarchitektur Beschrieben wird die Verbindung der E/E-Komponenten über dedizierte/proprietäre Verbindungen oder über Bussysteme.

Leistungsversorgung Auf dieser Ebene wird die Versorgung der E/E-Komponenten mit elektrischer Leistung beschrieben. Die Leistung aus Batterie oder Generator wird über Leistungsverteiler oder Sicherungs-Relais-Boxen an die E/E-Komponenten über Leistungsverbindungen verteilt. Die E/E-Komponenten haben einen Eingang für die Leistungsversorgung und einen Ausgang für die Masseanbindung. Sensoren oder Aktuatoren können auch direkt von den ECUs versorgt werden.

Leitungssatz Alle logischen Verbindungen werden als physikalische, elektrische Verbindungen beschrieben. Die elektrische Verbindung beschreibt die Anzahl der Pins und Leitungen für die logische Verbindung. Für die elektrische Verbindung muss ein Leitungstyp (Einzelleitung, mehradrige Kabel, verdrehte Zweidrahtleitung (mit/ohne Kunststoffmantel), geschirmte Leitung (z. B. Koax) oder geschirmte verdrehte Mehrdrahtleitung) definiert werden. Zudem werden Trennstellen (z. B. beim Übergang von Baugruppen) und Ausbindungen (bei Unterteilung eines Kabelstrangs in zwei Kabelstränge) und die Zuordnung von Pins auf Stecker definiert.

Topologie Diese Ebene definiert die Bauräume für E/E-Komponenten und die Segmente für die Leitungsführung. Die Bauräume werden durch Größe, Ort, und Eigenschaften wie z. B. Temperaturbereiche und Feuchtigkeit beschrieben. Segmente verknüpfen die Bauräume, Trennstellen und Ausbindungen und sind durch Länge und maximaler Bündeldurchmesser beschrieben.

Die Elemente der Vernetzungsarchitektur, Leistungsversorgung und Leitungssatz werden zusammenfassend auch als Bordnetz bezeichnet.

2.2.2 Für die E/E-Architekturauslegung relevante Vernetzungstechnologien

Ein Bussystem beschreibt die Art der Verknüpfung der Teilnehmer (Topologie des Bussystems), das Kommunikationsprotokoll und die Spezifikation der physikalischen Realisierung [128, S. 2]. Im Automobilbereich werden die folgenden Vernetzungstechnologien eingesetzt [18, 67, 128, 147, 178]:

LIN (Local Interconnect Network) Ein serielles, nach Master-Slave-Prinzip arbeitendes, linear vernetztes Bussystem mit einer maximalen Übertragungsgeschwindigkeit von 20 kBit/s [77].

Es wird als kostengünstige Technologie für die Vernetzung einfacher ECUs, Sensoren (z. B. Temperatursensor), Aktuatoren (z. B. Beleuchtungselemente) mit geringen Datenraten eingesetzt.

CAN (Controller Area Network) Ein serielles, prioritätsbasiert Nachrichten verschickendes, linear/Stern vernetztes Bussystem mit einer maximalen Übertragungsgeschwindigkeit von 1 MBit/s [74].

Es ist das am häufigsten eingesetzte Bussystem zur Vernetzung von ECUs mit mittleren Datenraten (z. B. in der Body-Domäne) oder mit hohen Datenraten für Echtzeitanwendungen (z. B. in der Domäne Powertrain).

CAN-FD (CAN Flexible Data-Rate) Eine abwärtskompatible Erweiterung des CAN-Bussystems um die Bandbreite zu steigern. Hierbei wird die Nutzdatenlänge der CAN-Botschaften von maximal 8 Byte auf 64 Byte erhöht. Damit können die Botschaften Nutzdaten wie beim CAN-Bussystem (0 bis 8 Byte) oder 12, 16, 20, 24, 32, 48 oder 64 Byte enthalten. Zudem kann optional der Bittakt für das Nutzdatenfeld und die CRC-Prüfsumme erhöht werden. Für diese Bitratenumschaltung werden Zielwerte um 4 Mbit/s angestrebt.

FlexRay Ein auf zwei Kanälen, fehlertolerantes, zeitgesteuertes, TDMA (Time Division Multiple Access)-basiert Nachrichten verschickendes, linear/Stern vernetztes Bussystem mit einer maximalen Übertragungsgeschwindigkeit von 10 MBit/s pro Kanal [76].

Es ist aus den Anforderungen an ein Bussystem für sicherheitskritische X-by-wire-Anwendungen entstanden [15, 64], die eine sichere und deterministische Kommunikation mit hohen Datenraten erfordern, d. h. die Anwendung liegt im Bereich von Fahrwerk und Antrieb.

MOST (Media Oriented System Transport) Ein serielles, Echtzeit Audio/Videofähiges, mit einer Ringtopologie aufgebautes Bussystem mit einer maximalen Übertragungsgeschwindigkeit von 150 MBit/s [29].

Es ist für Multimediaanwendungen mit sehr hohen Datenraten konzipiert und kommt dementsprechend in der Domäne Infotainment zur Anwendung.

Automotive Ethernet Ein Bussystem für schnellen Flash- und Diagnosezugang [73, 75] und für Gesamtfahrzeugvernetzung als Ergänzung zu den Standardbussystemen [63, 123, 129]. Als Übertragungsgeschwindigkeit werden 100 MBit/s und für die Weiterentwicklung 1 GBit/s anvisiert.

Beim Aufbau eines Bussystem und zur Verbindung verschiedener Bussysteme kommen weitere E/E-Komponenten zum Einsatz [18, 67, 147, S. 21]:

Sternkoppler, Repeater, Hub Zur Verstärkung elektrischer Signale bzw. Speichern eines Bits vor Ausgabe auf ein weiteres Bussegment.

Bridge, Switch, Router Zur Weitergabe von Paketen abgänglich von ihrer Adresse an entsprechenden Ausgangsport (relevant für Ethernet als Fahrzeugbus).

Gateway Zur Protokollumsetzung (d. h. beliebiges Zerlegen von Paketen und anschließendem Zusammensetzen und Verschicken) zwischen Bussystemen. Hierbei beschreibt die Routinetabelle, wie die Botschaften weitergeleitet werden sollen.

2.2.3 Abgrenzung weiterer E/E-Architekturbegriffe

Insgesamt werden also durch den Begriff E/E-Architektur fahrzeugweit die Zusammenhänge von Funktionen, E/E-Komponenten, Leitungssatz und Topologie beschrieben. Bei der Entwicklung von E/E-Architekturen kommen verschiedene weitere Begriffe zum Einsatz:

In der Entwurfsphase werden E/E-Architektur-Konzepte (Definition 2.6) diskutiert.

Definition 2.6 (E/E-Architektur-Konzept): Ein E/E-Architektur-Konzept beschreibt einen Entwurf für die Auslegung einer E/E-Architektur bzw. E/E-Architektur-Plattform während der Entwicklung.

Eine E/E-Architektur-Variante (Definition 2.7) spiegelt hierbei die E/E-Architektur einer speziellen Fahrzeugvariante wider, d. h. eine E/E-Architektur wie sie nach Konfiguration der SAs durch den Kunden verbaut werden würde.

Definition 2.7 (E/E-Architektur-Variante): Eine E/E-Architektur-Variante beschreibt ein spezifisches Produkt und wird aus einer Kombination von alternativen Ausstattungen und alternativen technischen Konzepten erstellt.

Um Skaleneffekte zu erzielen und möglichst gleiche Konzepte (Verringerung der Komplexität in der Entwicklung) einzusetzen, werden E/E-Architektur-Plattformen (Definition 2.8) entwickelt. Sie beschreiben viele ähnliche E/E-Architekturen und werden ggf. auch als E/E-Architekturfamilie bezeichnet.

Definition 2.8 (E/E-Architektur-Plattform): Eine E/E-Architektur-Plattform beschreibt die Gemeinsamkeiten der E/E-Architekturen vieler Baureihen und Derivate mit gleichem Anforderungsprofil. Von der E/E-Architektur-Plattform werden die E/E-Architekturen für die einzelnen Baureihen und Derivate abgeleitet.

Eine E/E-Architektur ist im Allgemeinen in verschiedene E/E-Architektur-Domänen (Definition 2.9) in Anlehnung an die Fahrzeugdomänen aufgeteilt.

Definition 2.9 (E/E-Architektur-Domäne): Eine E/E-Architektur-Domäne beschreibt die E/E-Architektur-Anteile für eine Fahrzeugdomäne, deren Funktionen eine gemeinsame Charakteristik aufweisen und sich oft in der Organisationsstruktur abbilden.

Die verschiedenen E/E-Architektur-Domänen sind hierbei:

E/E-Architektur-Domäne Body&Comfort umfasst die E/E-Architektur für Komfort-, Innenlicht- und Außenlichtfunktionen,

E/E-Architektur-Domäne Chassis/Fahrerassistenz umfasst Fahrwerks-, Fahrerassistenzfunktionen,

E/E-Architektur-Domäne Powertrain umfasst Antriebs- und Abgasreinigungsfunktionen,

E/E-Architektur-Domäne Infotainment umfasst Unterhaltungs- und Informationsfunktionen.

Aufgrund der Charakteristik der Funktionen in den einzelnen E/E-Architektur-Domänen [112], ergeben sich im Allgemeinen unterschiedliche Anforderungen an die Vernetzungstechnologien hinsichtlich Sicherheit, Latenz, Echtzeitfähigkeit und Datenaufkommen [107, S. 149]. So gibt es z. B. in der E/E-Architektur-Domäne Chassis/Fahrerassistenz sehr hohe Anforderungen hinsichtlich Sicherheit, Latenz und Echtzeitfähigkeit und hohe Anforderungen hinsichtlich der Datenmengen.

2.2.4 E/E-Architekturen bei Mercedes-Benz Cars

Für Mercedes-Benz PKW und Vans kommen E/E-Architektur-Plattformen, genannt Standard-E/E-Architekturen (STAR), zum Einsatz. Die Anzahl dieser E/E-Architektur-Plattformen wurde von sieben auf zwei reduziert [129]. Die Generationen STAR-1 und STAR-2 decken sowohl gängige Standardisierungen (z. B. CAN, LIN, Flexray, AUTOSAR) als auch die E/E-Module der Modulstrategie (Abschnitt 2.3.4) ab [129].

Eine E/E-Architektur für ein Oberklasse-Fahrzeug (dargestellt in Abbildung 2.6) umfasst bis zu 80 ECUs, die mit etwa 6000 Signalen miteinander kommunizieren und einen Leitungssatz von etwa 3km Länge und 40 kg Gewicht [67].

Für das Premium-Segment ist zu beobachten, dass die Netzwerkkommunikation stärker wächst als die Anzahl der ECUs und dass zukünftige Vernetzungskonzepte somit performante Bussysteme benötigen [129]. Die Ursache dafür liegt im begrenzten Bauraum bei einer steigender Anzahl von Funktionen, die einen entsprechendem Kommunikationsbedarf aufweisen.

Für die nächste Generation von Oberklasse-Fahrzeugen kann mit über 100 ECUs, die mit etwa 20.000 Signalen kommunizieren, gerechnet werden. Haupttreiber für den Anstieg sind Innovationen. Insbesondere in der E/E-Architektur-Domäne Chassis/Fahrerassistenz ist die nächste Generation ein Kandidat für den ersten Serieneinsatz von hochautomatisierten Fahrfunktionen³.

³ „Wir sind gerade dabei, erste Konzepte in Richtung hochautomatisiertes Fahren, die im Rahmen der Bertha-Benz-Gedächtnis-Tour bei der Vorentwicklung entstanden sind, in die Serienentwicklung zu übernehmen. Aus heutiger Sicht ist die nächste S-Klasse sicherlich ein Kandidat für einen ersten Serieneinsatz.“, Zitat des Daimler AG Leiter des Centers Fahrerassistenzsysteme und Aktive Sicherheit Michael E. Hafner im Interview mit Elektronik automotive (in „Wir übernehmen erste Konzepte zum hochautomatisierten Fahren“, Elektronik automotive, 22.09.2015).

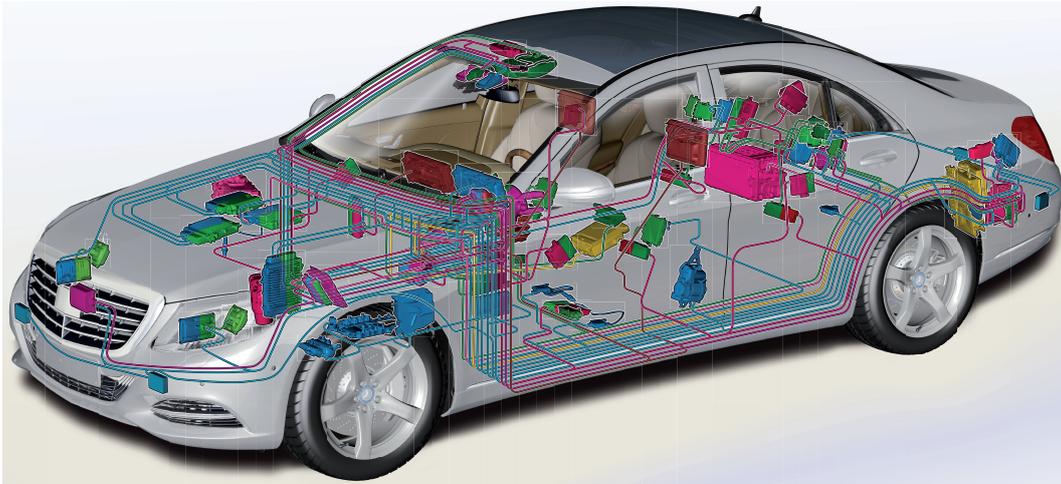


Abbildung 2.6: Schematische Darstellung der vernetzten E/E-Komponenten verortet in der Topologie des Fahrzeugs. Dargestellt ist die Mercedes-Benz S-Klasse (BR222).

2.3 Methoden zur Komplexitätsreduktion und Variantenerzeugung

Aufgrund der Komplexität, die mit der Entwicklung von Fahrzeugen einhergeht, gibt es in der Automobilindustrie Methoden um diese zu beherrschen und gleichzeitig die Kosten zu reduzieren. Im nachfolgenden Abschnitt werden drei Ansätze kurz vorgestellt. Dies beinhaltet auch die der bei Mercedes-Benz Cars verwendete Modulstrategie (Abschnitt 2.3.4).

2.3.1 Baukastenprinzip

In dieser Arbeit wird der Baukastenbegriff (Definition 2.10) mit Bezug auf die Funktionen, die das Produkt erfüllen wird, verwendet. Damit werden Funktionsbausteine zu einer Gesamtfunktion kombiniert was die Erarbeitung einer Funktionsstruktur erforderlich macht. Dies ist eine Möglichkeit zur Losgrößen-erhöhung von Gleichteilen (Definition 2.11) [116, S. 663]. Hier würde statt eines großen Spektrums der Gesamtfunktion ein kleines Spektrum vorherrschen und somit statt einer funktionsorientierten Gliederung des Produkts in Funktionsbausteine eine fertigungsorientierte Gliederung in Fertigungsbausteinen im Vordergrund stehen.

Die Verwendung von Gleichteilen wird auch als Verblockung (Definition 2.12) bezeichnet. Wird eine neue Lösung in bestehende Baureihen eingeführt, spricht man auch von Rückverblockung (Definition 2.13).

Definition 2.10 (Baukasten): Ein Baukasten umfasst Maschinen, Baugruppen und Einzelteile, die als Bausteine mit oft unterschiedlichen Lösungen durch Kombinationen verschiedene Gesamtfunktionen erfüllen [116, S. 662].

Definition 2.11 (Gleichteile): Ein Gleichteil ist ein Teil, das ohne Anpassung in mehreren Produkten verwendet wird [116, S. 663]

Definition 2.12 (Verblockung): Verblockung beschreibt die Verwendung von Gleichteilen.

Definition 2.13 (Rückverblockung): Rückverblockung beschreibt das Ersetzen einer existierenden Lösung von Komponenten oder Gleichteilen mit einer neuen Lösung aus weiterentwickelten Gleichteilen.

2.3.2 Plattformbauweise

Im Gegensatz zum Baukastenprinzip wird mit einer Fahrzeug-Plattform (Definition 2.14) eine Produktvariante nicht grundsätzlich durch Konfiguration der Bausteine erzeugt [116, S. 686]. Charakteristisch ist vielmehr, dass die ausführungsneutrale Produktplattform um Produktgestaltungselemente ergänzt wird, damit gerade die Produktverwandtschaft nach außen hin nicht wahrnehmbar ist [116, S. 686].

Fahrzeug-Plattformen in der Automobilindustrie werden über viele Jahre weiterentwickelt, weshalb die Fahrzeuge mit einem großen Maß an Legacy bzgl. der Komponenten als auch der Struktur auskommen müssen [168].

Definition 2.14 (Fahrzeug-Plattform): Eine Fahrzeug-Plattform beschreibt gleiche Teile und Strukturen, die den größten gemeinsamen Nenner einer Produktfamilie darstellen, unter funktionalen Gesichtspunkten festgelegt werden und im Produkt um produktspezifische Anbauten (Produktgestaltungselemente) ergänzt werden [116, S. 686].

Für Mercedes-Benz PKW gibt es die Fahrzeug-Plattformen MFA (Mercedes-Benz Frontantriebsarchitektur, engl. **Mercedes-Benz Front-Wheel Architecture**) für Fahrzeuge im Kompaktwagensegment, MRA (Mercedes-Benz Heckantriebsarchitektur, engl. **Mercedes-Benz Rear-Wheel Architecture**) für Fahrzeuge mit Heckantrieb und MHA (Mercedes-Benz Hocharchitektur, engl. **Mercedes-Benz High Architecture**) für Geländewagen und SUVs. E/E-Architektur-Plattformen (Abschnitt 2.2) können in Fahrzeug-Plattformen übergreifend eingesetzt werden. So wird z.B. die STAR-2-E/E-Architektur-Plattform in MRA und MHA eingesetzt.

2.3.3 Modularisierung

Module werden in der Literatur und der gelebten Praxis in den verschiedenen Unternehmen unterschiedlich verstanden und definiert [43, 95, 72, S. 129]. Hierbei gibt es deutliche Unterschiede zwischen europäischen und japanischen OEMs [149]. Ziele einer Modularisierung sind die Verringerung der Komplexität in der Entwicklung [52] und die wirtschaftliche Effizienz einerseits sowie die Erzeugung einer großen Variantenvielfalt für den Kunden andererseits [151, 72, S. 2]. Allgemein prägt ein Modul mehrere Eigenschaften:

Zusammengesetztheit Die erste Eigenschaft von Modulen beleuchtet den zusammengesetzten Charakter von Modulen. Ein Modul wird somit als eine physische Verbindung von Einzelteilen zu einem Gesamtteil definiert [72, S. 130]. Module werden auch beschrieben als Subsysteme bestehend aus Subkomponenten [53]. Alternativ werden Module auch definiert als Einheiten, wie z. B. Baugruppen oder Komponenten, die aus Subelementen mit relativ stark ausgeprägten Beziehungen zwischen diesen Subelementen bestehen [52].

Unabhängigkeit Die zweite Eigenschaft beschreibt die Wechselwirkung der Module. So existieren nach der Modularisierung nur schwache und wenige Beziehungen zwischen den Modulen, wodurch diese verhältnismäßig unabhängig sind [52, 53].

Abgegrenztheit durch definierte Schnittstellen Die dritte Eigenschaft beleuchtet die Organisation der Schnittstellen zwischen den Modulen. Das Erreichen der Unabhängigkeit wird als sehr komplexe Problematik eingeschätzt [72, S. 137]. Demnach wird das Management der Schnittstellen zwischen den Modulen die Hauptaufgabe der OEMs sein, denn einerseits müssen Module aufeinander abgestimmt sein und miteinander kommunizieren können sowie andererseits eine hohe Kompatibilität zu anderen Modulen aufweisen. Das sorgfältige Definieren der Schnittstellen ist ein Fokus im System-Design [160].

Somit ergibt sich folgende in dieser Arbeit verwendete Definition:

Definition 2.15 (Modul): Ein Modul ist eine aus einer Modularisierung hervorgegangene Menge von physisch verbundenen Komponenten, die relativ unabhängig von anderen Modulen ist und definierte Schnittstellen zu anderen Modulen besitzt.

Mit der Modularisierung werden verschiedene Effekte erzielt (Anhang A.1). So wird z. B. die Wiederverwendbarkeit und die Austauschbarkeit erhöht. Zudem muss bei der Modularisierung der Zusammenhang zwischen funktionalen Elementen und den Modulen betrachtet werden. Hierzu werden Produktarchitekturen klassifiziert (Anhang A.2).

2.3.4 Mercedes-Benz Modulkonzept

Die Modularisierung für Fahrzeuge von Mercedes-Benz wurde 2007 beschlossen. Die Organisation und Planung der Module wird durch die sogenannte Modulstrategie (Definition 2.16) durchgeführt.

Definition 2.16 (Modulstrategie): Die Modulstrategie ist ein kontinuierlich andauernder Standardisierungsprozess in dem ein grobes Konzept der Module beschrieben, die Schnittstellen der Module definiert, für den Verbau der Module eine Zuordnung zu betroffenen Baureihen getroffen und eine grobe Kostenplanung für das Modul vorgenommen wird.

Durch die baureihenübergreifende Wiederverwendung von Modulen (Definition 2.15) sollen auch insbesondere Kosten eingespart werden.⁴ Die Modulstrategie hat insgesamt etwa 90 Module standardisiert, die über alle Baureihen hinweg verbaut werden [161]. Die Module haben zueinander definierte Schnittstellen und können nahezu beliebig miteinander kombiniert werden und können eine Vielzahl von Gesamtlösungen darstellen [53]. Weitere Ziele der Modulstrategie sind

- eine frühzeitige Absicherung hinsichtlich der Fertigungsmaterialien,
- des Entwicklungsaufwands,
- der Qualität und
- des Gewichts.

Durch Weglassen, Hinzufügen oder Einbau von Varianten von Komponenten des Moduls, lassen sich somit verschiedene Modulausprägungen (Definition 2.17) erzeugen [53]. Die Modulausprägung dient demnach dazu, verschiedene Laststufen für das Modul oder verschiedene Ausbaustufen für den Kunden vorzusehen, z.B. eine Einstiegsvariante, eine Volumenvariante und eine High-End-Variante des Moduls.

⁴„In jedem Modul stecken signifikante Potenziale. Wir rechnen mit einer Kostenentlastung von mehr als 1,5 Milliarden Euro jährlich. Es ist unser Ziel, dass dieser Einspareffekt spätestens 2015, besser schon 2014 voll wirkt.“, Zitat des Daimler AG Vorstandsmitgliedes für Konzernforschung und Mercedes-Benz Cars Entwicklung Prof. Dr Thomas Weber im Interview mit automobilwoche.de (in „Wir wollen jährlich 1,5 Milliarden Euro sparen“, *Automobilwoche*, 21.01.2012).

Durch den „Modulbaukasten [sind] Einsparungen von 1,5 Milliarden Euro bis 2014 eingeplant.“, schreibt die *Automobilwoche* in einem Bericht über Ankündigungen von Mercedes-Produktionsvorstand Wolfgang Bernhard auf dem Investorentag anlässlich der Eröffnung eines neuen Kompaktwagenwerks in Kecskemét (Ungarn) (in „Mercedes will Milliarden Euro an Materialkosten sparen“, *Automobilwoche*, 29.03.2012).

Definition 2.17 (Modulsausprägung): Eine Modulsausprägung ist eine der möglichen verbauten, technischen Varianten eines Moduls. Die Modulsausprägung ist somit eine sinnvolle Kombination der diesem Modul zugeordneten Komponenten.

Demnach werden in der durch die Modulstrategie erzeugten Produktarchitektur die Effekte Austauschbarkeit und Erweiterbarkeit durch das Verwenden von Modulsausprägungen erreicht: Der Varianz erzeugende Charakter der Austauschbarkeit wird also auch dadurch erwirkt, dass Modulsausprägungen ausgetauscht werden können. Ferner kann die Erweiterbarkeit auch erzielt werden indem das Modul einfach um eine neue Ausprägung erweitert wird. Diese Erweiterung oder andere Weiterentwicklung bezüglich der Standardisierung der Module wird durch den Weiterentwicklungsprozess der Modulstrategie gesteuert und als Dynamisierung (Definition 2.18) bezeichnet. Wann eine Dynamisierung durchgeführt und ein Modul in einer Baureihe verbaut wird, wird im sogenannten Modulzyklusplan (Definition 2.19) beschrieben. Alle relevanten Daten und Spezifikationen der Module werden für ein Modul in einem sogenannten Modulheft (Definition 2.20) festgehalten.

Definition 2.18 (Dynamisierung): Die Dynamisierung ist die technische Weiterentwicklung von Modulen die durch neue Technologien, Innovationen und Optimierungen getrieben wird.

Definition 2.19 (Modulzyklusplan): Der Modulzyklusplan spezifiziert alle zeitlichen Themen eines Moduls und beschreibt wann welche Modulsausprägung in welcher Baureihe zu welchem Zeitpunkt zum Einsatz kommt und wann das Modul dynamisiert wird.

Definition 2.20 (Modulheft): Ein Modulheft beschreibt genau ein Modul und beinhaltet die durch die Modulstrategie abgestimmte Spezifikation des Moduls, eine Spezifikation der im Modul enthaltenen Komponenten, eine Beschreibung der Modulsausprägungen, eine Definition der Schnittstellen, eine grobe Beschreibung des Funktionsumfangs der Modulsausprägungen, eine Kostenplanung, den Modulzyklusplan und Zielgrößen für Fertigungsmaterialien, Entwicklungsaufwand, Qualität und Gewicht.

Ferner werden die Module in einer hierarchischen Baumstruktur geordnet (Abbildung 2.7). Durch diese hierarchische Einordnung lassen sich so Integrationsaufgaben wahrnehmen. So sind beispielsweise die Module der Fahrerassistenzsysteme der Modulgruppe Regelsysteme zugeordnet, in der alle Regelsysteme aufeinander abgestimmt werden können. Gleichmaßen lassen sich im Themenfeld Elektrik/Elektronik alle E/E-Belange koordinieren. Meist werden Module aufgrund des Umfangs in eine Menge von Submodulen (Definition 2.21) unterteilt. Sie werden dann jeweils über ein Modulheft beschrieben. Statt des Begriffs Submodul wird jedoch meist der Begriff Modul verwendet.

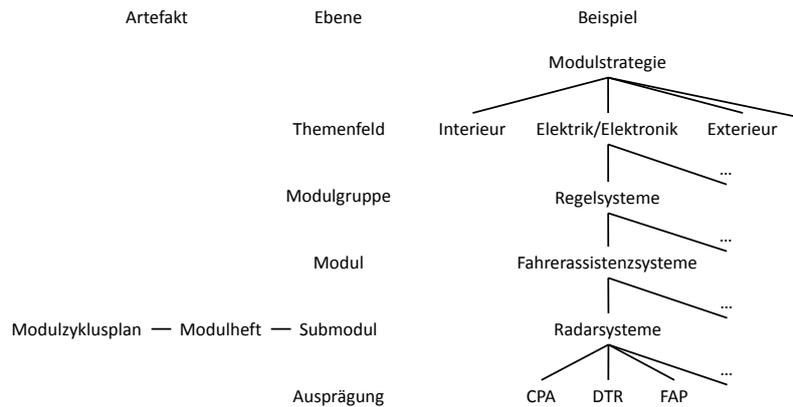


Abbildung 2.7: Ordnungsebenen für Module. Dargestellt sind die Ordnungsebenen für die Module einschließlich eines Beispiels aus dem Bereich Fahrerassistenz. Für das Submodul wird im Rahmen der Modulstrategie ein Modulzyklusplan und ein Modulheft erstellt.

Definition 2.21 (Submodul): Ein Submodul ist eine Menge von Komponenten eines Moduls.

Nicht bei allen Modulen sind unter den dazugehörigen Komponenten auch E/E-Komponenten. Sie sind für die Auslegung der E/E-Architektur oder einer E/E-Plattform nicht relevant. Deshalb beziehen sich die weiteren Abschnitte auf E/E-Module (Definition 2.22).

Definition 2.22 (E/E-Modul): Ein E/E-Modul ist ein für die Auslegung der E/E relevantes Modul, bei dem mindestens eine Komponente der dazugehörigen Komponenten E/E-Anteile hat.

In der Praxis hat die Modulstrategie aus Sicht der E/E-Architecturentwicklung verschiedene Defizite: Die Modulstrategie ist ein produktions- und fertigungsorientierter Prozess mit einem starkem Fokus auf Kosten. Zwar werden im Rahmen einer Dynamisierung eines Moduls weitere Module mit einer Hardware-Schnittstelle (z. B. bei gemeinsamem Bauraum) mitbetrachtet, jedoch sind die E/E-Umfänge (einschließlich Elektrik, Kommunikation und Schnittstellen) unterbetrachtet. Gleichzeitig hat die E/E-Architecturentwicklung keine ausreichenden Methoden um diese Defizite abzufedern (Abschnitt 4.1).

2.4 Zusammenfassung

In diesem Kapitel werden die Grundlagen für diese Arbeit beschrieben. Ein Fokus liegt auf dabei auf dem Fahrzeugentwicklungsprozess (Abschnitt 2.1), der die Meilensteine für die Entwicklung festlegt, an denen definierte Ent-

wicklungsziele vorliegen müssen, z. B. SOP (Produktionsstart) und Modellpflege (MOPF). Die Entwicklung erfolgt unter Kooperation mit Zulieferern, z. B. über das Zusammenarbeitsmodell V-Modell. In der E/E werden E/E-Komponenten (Definition 2.1) oder Software-Komponenten (Definition 2.2) durch Zulieferer und OEM entwickelt und ins Fahrzeug integriert. Hierbei ist die Software-Architektur der ECUs mit AUTOSAR standardisiert, das die ECU-Basis-Software von der Anwendungssoftware trennt.

Ein weiterer Schwerpunkt liegt auf der Beschreibung von E/E-Architekturen (Abschnitt 2.2), die im Spannungsfeld vieler, zum Teil gegensätzlicher Einflussfaktoren stehen. Da E/E-Architekturen (Definition 2.5) fahrzeugweit die Zusammenhänge von Funktionen, E/E-Komponenten, Leitungssatz und Topologie beschreiben, werden sie durch eine in Ebenen zerlegte Betrachtungsweise erfasst. Zudem werden die für die E/E-Architekturauslegung relevanten Vernetzungstechnologien LIN, CAN, Flexray, MOST und Ethernet vorgestellt und die bei der E/E-Architecturentwicklung verwendeten Begriffe E/E-Architektur-Konzept (Definition 2.6), E/E-Architektur-Variante (Definition 2.7), E/E-Architektur-Plattform (Definition 2.8) und E/E-Architektur-Domäne (Definition 2.9) definiert. Zudem wird ein Überblick über E/E-Architekturen bei Mercedes-Benz Cars gegeben, wo zwei E/E-Architektur-Plattformen im Einsatz sind.

Ein letzter Schwerpunkt liegt auf Methoden zur Komplexitätsreduktion und Variantenerzeugung (Abschnitt 2.3), in dem Grundlagen zu Baukästen, Verblockung, Fahrzeug-Plattformen und Modularisierung aufgezeigt werden. Die Beispiele für diese Arbeit basieren hierbei auf dem Modulkonzept, das bei Mercedes-Benz zum Einsatz kommt (Abschnitt 2.3.4) und Begriffe wie Modulstrategie (Definition 2.16) und E/E-Modul (Definition 2.22) verwendet.

Kapitel 3

Stand der Technik

In diesem Kapitel wird der für diese Arbeit relevante Stand der Technik beschrieben. Im einzelnen liegt der Fokus auf der E/E-Architecturentwicklung für Fahrzeuge unter Nutzung der E/E-Architekturmodellierung, der Evolution bei der Softwareentwicklung und der Wiederverwendung in der Produktlinienentwicklung.

3.1 E/E-Architecturentwurf

Im Allgemeinen gibt es firmen- und branchenspezifische Unterschiede in den Prozessen und den Rollen der E/E-Architektur [39, 60, 61, 62].

Zudem ist der Entwurf von E/E-Architekturen abhängig davon, in welchem Umfeld das Produkt eingesetzt werden soll und wie in der entsprechenden Industrie entwickelt wird. So können im Vergleich zum E/E-Architecturentwurf von Kraftfahrzeugen einige Entwurfskriterien identisch sein, wie z. B. Anforderungen an Latenz oder Wiederverwendbarkeit von Funktionen, andere jedoch deutlich abweichen, wie z. B. Zertifizierungsanforderungen, Anforderungen an Kosten oder verwendbare Technologien [31, 152].

Selbst für Kraftfahrzeuge unterscheiden sich trotz grundlegend ähnlicher Funktionalität die entwickelten E/E-Architekturen, da sich die Geschäfts- und Produktcharakteristik unterscheidet. So wird z. B. für Personenkraftwagen, Lastkraftwagen oder Baumaschinen [12] bzw. für Personenkraftwagen in speziellen Märkten [91] ein unterschiedlicher Fokus in der Entwicklung der E/E-Architekturen gelegt. Der Schwerpunkt der nachfolgenden Abschnitte liegt auf dem Entwurf von E/E-Architekturen für Personenkraftwagen.

Der E/E-Architecturentwurf ist eine integrative Entwicklungstätigkeit. Somit müssen viele unterschiedlichste Interessen, die von den verschiedenen Teilen einer großen Organisation eines OEMs stammen, systematisch aufgenommen, konzentriert und eingeflochten werden [147, 168].

3.1.1 E/E-Architekturentwurf in Relation zum Fahrzeugentwicklungsprozess

In der Literatur wird aufgezeigt, dass E/E-Architekturprozesse zum Teil wenig formalisiert und schlecht dokumentiert sind [11, 61, 98, 127, 169]. In der Praxis wird das nicht ganz so kritisch bewertet, da gerade im Rahmen der E/E-Architekturentwicklung viele Untersuchungen und Absicherungen parallel laufen, während die Entscheidungen zu E/E-Architektur-Konzepten erst am Ende der E/E-Architekturentwicklungsphasen getroffen werden.

Beim Entwurf von E/E-Architekturen unterscheidet man in eine

- E/E-Architekturentwicklung für Fahrzeug-Plattformen (Definition 2.14) und
- E/E-Architekturentwicklung für Fahrzeug-Baureihen.

Hierbei wird die E/E-Architekturentwicklung einer Fahrzeug-Plattform abgeschlossen, bevor die Strategie-Phase der Lead-Fahrzeug-Baureihe beginnt (Abbildung 3.1). Mit dem Start der Strategiephase beginnt die E/E-Architekturentwicklung für das Lead-Fahrzeug, die sich bis in die Serienentwicklungsphase hineinzieht. Je nach Rollout der einzelnen Baureihen kann es hierbei auch zu Überschneidungen mit der E/E-Architekturentwicklung für Derivat-Fahrzeuge kommen. Diese werden vom Lead-Fahrzeug abgeleitet und im Allgemeinen verkürzt entwickelt. Sollen signifikante Änderungen mit Auswirkungen auf die E/E-Architektur für eine Modellpflege (MOPF) umgesetzt werden, wird ebenfalls eine E/E-Architekturentwicklungsphase durchgeführt.

Die E/E-Architekturentwicklung einer Fahrzeug-Plattform lässt sich wie folgt charakterisieren:

- Eine neue E/E-Architektur-Plattform wird selten, d.h. im Abstand mehrerer Jahre, als separate Aktivität bzw. Projekt entwickelt [8, 106]. Dementsprechend wird der gesamte Lebenszyklus der E/E-Architektur-Plattform (bis zu 30 Jahre) betrachtet. Der Lebenszyklus geht über den Produktionszeitraum hinaus. In ihm werden noch unbekannt funktionale Änderungen erwartet und müssen flexibel integrierbar sein und in ihm muss die Übernahme der E/E-Architektur-Plattform bzw. einzelner Anteile in die Baureihen im Vorfeld geplant werden [23, 59, 147, 127].
- Es werden deshalb möglichst das gesamte Bordnetz und alle Systeme betrachtet um grundsätzliche Untersuchungen durchzuführen [8, 16, 49, 69] und den Hauptteil der Designentscheidungen einschließlich der strategischen Technologieentscheide zu treffen [59, 127].
- Gerade die E/E-Architektur-Plattform-Entwicklung ist von großer Unsicherheit begleitet, da Designvoraussetzungen und viele Anforderungen

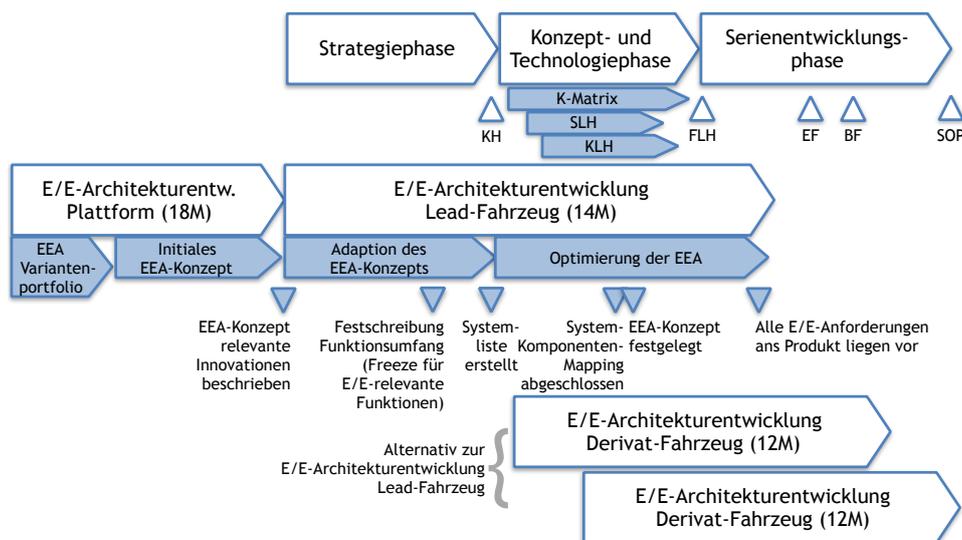


Abbildung 3.1: E/E-Architekturentwicklung im Fahrzeugentwicklungsprozess. Die Phasen in der E/E-Architekturentwicklung wurden dem Fahrzeugentwicklungsprozess (oben, Abschnitt 2.1.1) zugeordnet. Dabei wird zwischen einer Fahrzeug-Plattform- und einer Fahrzeug-Entwicklung unterschieden. Wird bei der Fahrzeug-Entwicklung von einer entwickelten E/E-Architektur abgeleitet (Derivat-Fahrzeug), verkürzt sich die Entwicklungsdauer für die E/E-Architektur. Δ Meilensteine im Fahrzeugentwicklungsprozess; ∇ Meilensteine im E/E-Architekturentwicklungsprozess; BF Bestätigungsfahrzeug; EEA E/E-Architektur; EF Erprobungs-Fahrzeug; FLH Fahrzeuglastenheft; KH Konzeptheft; KLH Komponentenlastenheft; K-Matrix Kommunikations-Matrix; SLH Systemlastenheft; SOP Start of Production (Produktionsstart)

zu Funktionen, Hardware und auch Vertrieb unbekannt, unvollständig oder noch nicht finalisiert sind [10, 38, 119]. In der Praxis sind insbesondere im Premium-Segment die Veränderung von Entwicklungszielen und Innovationen Faktoren, die zur Unsicherheit beitragen. Erstere werden z. B. aufgrund von Wettbewerbsdruck entschieden, während letztere oftmals wettbewerbsdifferenzierend sind und Vertriebsauswirkungen haben¹. Die Anmeldung von Innovationen in der Entwicklung findet kontinuierlich statt, d. h. einige sind langfristig bereits vor der E/E-Architektur-Plattform-Entwicklung bekannt, während andere kurzfristig vor der Strategiephase des Lead-Fahrzeugs integriert werden müssen. Auch der vorliegende Detaillierungsgrad und der damit verbundene Integrationsaufwand seitens der E/E-Architecturentwicklung ist heterogen: Manche Innovationen haben z. B. Ideencharakter, während bei anderen bereits Testerfahrungen vorliegen.

- Insgesamt führen diese Punkte dazu, dass in der Praxis E/E-Architektur-Plattform-Varianten untersucht werden, die zum Teil sehr divergieren. Dabei wird iterativ die E/E-Integrierbarkeit der E/E-Systeme und E/E-relevanten Innovationen einschließlich alternativer Umsetzungsmöglichkeiten untersucht. Die E/E-Architektur-Plattform-Entwicklung wird mit dem Entscheid für ein Initiales E/E-Architekturkonzept abgeschlossen. Es enthält die prägenden Umfänge E/E-Architektur-Plattform mit der Domänenstruktur, der Gatewaystruktur und das Technologieportfolio.

Die nachfolgende E/E-Architecturentwicklung der Fahrzeug-Baureihen lässt sich wie folgt charakterisieren:

- Das Initiale E/E-Architekturkonzept wird adaptiert und optimiert. Dabei wird der Funktionsumfang und die Funktionsverteilung finalisiert. Das Ergebnis, ein E/E-Architektur-Konzept für das Fahrzeug, wird in die nachfolgenden Fahrzeugprozesse, wie die E/E-Systementwicklung, E/E-Komponentenentwicklung und die K-Matrix- und Lastenheftprozesse übernommen [16, 38, 131, 132].
- Die E/E-Architecturentwicklung für Fahrzeug-Derivate wird verkürzt durchgeführt, da vom E/E-Architecturentwurf des Lead-Fahrzeugs abgeleitet werden kann. In diesem sind mitunter schon Umfänge der Derivat-Fahrzeuge berücksichtigt. Derivate-abhängige Umfänge, wie z. B. das System Verdeck einschließlich des Verdecksteuergeräts für Cabriolets, sind in der Regel in der E/E-Architektur-Plattform enthalten.

¹Diese Innovationen werden auch als USP (Unique Selling Point) bezeichnet und können durch den Vertrieb entsprechend eingesetzt werden.

3.1.2 Kriterien bei der Auslegung einer E/E-Architektur in der Automobilindustrie

Bei der Auslegung einer neuen E/E-Architektur-Plattform und der Anpassung eines E/E-Architektur-Konzepts sind verschiedenste Aspekte der E/E betroffen.

In der Praxis sind gerade im Premium-Segment Innovationen ein wichtiger Treiber bei der E/E-Architektur-Plattform-Entwicklung. Deren Integration und die Optimierung vorhandener Funktionen impliziert Veränderungen in der Funktionsverteilung und funktionale Integration.

Bei der E/E-Architektur-Plattform-Entwicklung ändert sich meist [8]:

- die Vernetzung der ECUs,
- das Technologie-Portfolio, z. B. welche Kommunikationsprotokolle verwendet werden,
- welche E/E-Komponenten in den E/E-Systemen enthalten sind,
- die Auslegung von Leitungssatz und Leistungsverteilung und
- das Package von E/E-Komponenten.

Die Auswirkung, die eine neue E/E-Architektur auf die restliche E/E-Entwicklung hat, ist somit beträchtlich. Deshalb werden bei der Auslegung und ganzheitlichen Betrachtung mehrere Kriterien herangezogen, wobei am häufigsten [8]:

- Produktkosten,
- Latenzzeiten in den Kommunikationsnetzen,
- Konfigurierbarkeit und Flexibilität für spätere Änderungen und
- Stromverbrauch im Normalbetrieb

betrachtet werden. Weitere wichtige Kriterien sind [50, 69, 91, 127, 159, 171]:

- Bauraum (begrenzend für Leitungssatz und ECUs aber auch entscheidend für Sicherheit², Umgebungsbedingungen³ und EMV⁴),

²Für den Schutz vor Angreifern (z. B. Autodieben) werden kritische Bussysteme nicht von außen leicht zugänglich (z. B. am Außenspiegel) verlegt [147]. Für den Schutz von Personen werden E/E-Komponenten mit Funktionen, die nach einem Unfall noch funktionsfähig sein sollen, in geschützten Bereichen (z. B. innerhalb der Fahrgastzelle) platziert [147].

³Die E/E-Komponenten werden für gewisse Umweltbedingungen ausgelegt, die an den Bauräumen variieren können. Typische Umweltbedingungen sind Feuchtigkeits-, Temperatur-, Schmutz- und Vibrationsbedingungen [147].

⁴Die elektromagnetische Verträglichkeit (EMV) beschreibt u. a. die Störemmission (Störverhalten einer E/E-Komponente/Leitung auf andere E/E-Komponente/Leitung) und

- Realisierbarkeit von Änderungen (Historie im Unternehmen),
- Verfügbare Ressourcen (Speicher- und Rechenressourcen der ECUs und darstellbare Buslast),
- Skalierbarkeit,
- Technische Komplexität,
- Verfügbarkeit am Markt (z. B. von Halbleitern) und
- Wiederverwendbarkeit.

Aus Sicht des OEMs ist es somit wichtig, eine möglichst optimale Lösung zu finden [65, 69, 159]. Dies ist schwierig, da sich die Kriterien zum Teil diametral gegenüberstehen [171] und Änderungen immer gesamtheitliche Wechselwirkungen haben [16]. So muss z. B. in der kostengetriebenen Automobilindustrie der Einsatz von Ressourcen optimiert aber gleichzeitig höchstmögliche Flexibilität für Änderungen vorgehalten werden [59].

Aus diesen Gründen müssen die Bewertungen E/E-Architektur-Domänenübergreifend durchgeführt werden [50]. Allerdings haben sich in der Praxis verschiedene Funktionsverteilungskonzepte [147] und E/E-Architektur-Muster herauskristallisiert, die abhängig von der technischen Komplexität und dem Grad der funktionaler Abhängigkeiten eingesetzt werden [111, 146].

3.1.3 Ansätze zum Entwurf einer E/E-Architektur in der Automobilindustrie

Die aufgezeigten Kriterien werden beim E/E-Architekturentwurf gegenübergestellt. Dafür werden in der Literatur auch neue, zum Teil theoretische oder nur Teilaspekte betrachtende, Ansätze beschrieben [7, 58, 59, 95, 101, 130] während sich in der Praxis grundsätzlich drei Ansätze bewährt haben. Zu diesen gehören der Top-Down-, der Bottom-Up- und der kombinierte Entwurf, welche nachfolgend aufgezeigt werden.

Top-Down-Entwurf Beim Top-Down-Entwurf steht eine funktionsorientierte Vorgehensweise im Vordergrund. Diese bietet sich für Neuentwicklungen an, da

- solch ein funktionaler Entwurf nicht durch existierende Lösungen für E/E-Komponenten, Leitungssatz und Bauräume eingeschränkt wird [146] und

die Störfestigkeit (Resistenz gegen Störung), die auch bei der Auslegung des Leitungssatzes mit betrachtet werden müssen (so können z. B. Leitungen in der Hutablage in die Scheibenantennen einstrahlen und so den Empfang stören) [147].

- er zu einer geringen Funktionskomplexität führt [16].

In der Praxis ist dieses Vorgehen allerdings oft ungeeignet, da E/E-Architekturen nur in den seltensten Fällen ohne gegebene Randbedingungen komplett neu entwickelt werden [131]. Gerade bei größeren OEMs sind z. B. Modulstrategien ein Faktor, der mitbedachtet werden muss (Abschnitt 2.3.4).

Im Einzelnen umfasst der Top-Down-Entwurf folgende Vorgehensweise [16, 49, 91, 127, 146]:

1. Ableitung von Funktionen ausgehend von kundenerlebbaren Anforderungen,
2. Gruppierung der Funktionen und Festlegung der E/E-Komponenten (Partitionierung),
3. Festlegung der Vernetzung zwischen den E/E-Komponenten,
4. Entwicklung des Leitungssatzes und
5. Platzierung der E/E-Komponenten und des Leitungssatzes in der Topologie (Package).

Bottom-Up-Entwurf Der Bottom-Up-Entwurf steht dem Top-Down-Entwurf gegenüber, da auf Basis von existierenden E/E-Architekturen [49, 146] eine E/E-Komponentenorientierte Vorgehensweise im Vordergrund steht. Aufgrund dieser Hardwareorientierung fokussiert der Bottom-Up-Ansatz stärker die Kostenaspekte [16].

Im Einzelnen umfasst der Bottom-Up-Entwurf folgende Vorgehensweise [16, 146]:

1. Ausgangspunkt sind bestehende E/E-Komponenten,
2. Prüfung und Einflechten von Innovationen, indem ggf. weitere E/E-Komponenten in die bestehenden Netzwerkstrukturen integriert werden,
3. Verteilung der Funktionen (Partitionierung) und
4. Überarbeitung der Vernetzung und des Leitungssatzes inkl. Platzierung in der Topologie (Package).

Kombinierter Entwurf Eine Kombination der aufgezeigten Ansätze stellt der kombinierte Entwurf dar, der den Bottom-Up-Entwurf durch eine Top-Down-Perspektive komplettiert [127]. Da der Entwurf individuell gestaltbar ist, ermöglicht er sowohl gegebene Randbedingungen, wie z. B. Modulstrategien

(Abschnitt 2.3.4), einzubeziehen als auch eine funktionsorientierte Ableitung für Innovationen vorzunehmen [131].

Im Einzelnen umfasst der kombinierte Entwurf folgende Vorgehensweise [131, 132]:

1. Top-Down-Ableitung von Funktionen ausgehend von kundenerlebba- ren Anforderungen,
2. Bottom-Up-Platzierung der für die Funktionen benötigten Sensoren und Aktuatoren in der vorgegebenen Topologie (Package)
3. Ableitung weiterer ECUs gemäß Funktionsarchitektur und Topologie und Festlegung dieser gemäß Modulstrategie,
4. parallel Vernetzung der E/E-Komponenten und Festlegung des Leitungssatzes und
5. iterativer Optimierungsprozess mit individueller Gestaltung der Fragestellungen.

Zudem können OEMs Unterstützung bei der Auslegung der E/E-Architektur von Tier-1-Zulieferern erhalten. In dem Fall werden sie entsprechend früh eingebunden und können auch entsprechende Werkzeugunterstützung anbieten, sodass zusammen erarbeitete oder vorgefertigte Lösungen lediglich durch den OEM integriert werden müssen [49, 69, 91, 120, 146, 159].

3.2 E/E-Architekturmodellierung

Um mit den aufgezeigten Ansätzen für den E/E-Architekturentwurf die einzelnen Kriterien gegenüberzustellen und werkzeuggestützt zu bewerten, setzen OEMs die E/E-Architekturmodellierung ein. Hierbei werden die E/E-Architekturen durch Zerlegung in verschiedene Ebenen beschrieben. Modelliert werden die E/E-Systeme bzgl. Struktur und Schnittstellen ohne Inhalte, Verhaltensmodellierung oder Simulation.

Allerdings hat die systematische, ebenenübergreifende Modellierung der E/E-Architektur in der frühen Entwicklungsphase noch keine lange Tradition [127]. Eine Ursache dafür ist die große Unsicherheit, da Designvoraussetzungen und viele Anforderungen unbekannt, unvollständig oder noch nicht finalisiert sind [10, 38, 119].

Dementsprechend sind zum Zeitpunkt, an dem die ersten E/E-Architekturentscheidungen getroffen werden müssen, keine sehr detaillierten Modelle für Hardware und Software erstellbar [127] und Ansätze, die eine Vollständigkeit verlangen, in der Praxis unbrauchbar [16].

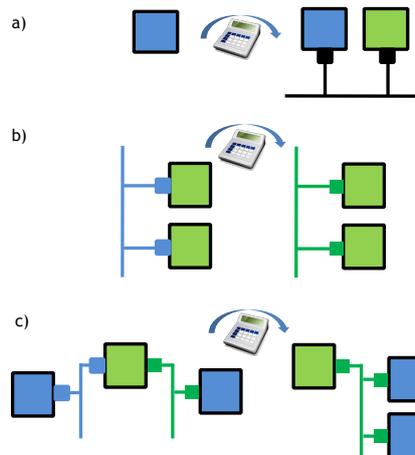


Abbildung 3.2: Beispiele für den Einsatz der E/E-Architekturmodellierung. Mit der Modellierung der E/E-Architektur werden vielfältige Ziele verfolgt. Beispielsweise wird mit der Modellierung eine Bewertung a) der Integrierbarkeit einer Innovation, b) von E/E-Architekturänderungen und c) von E/E-Architekturkonzepten zur optimalen Auslegung der E/E-Architektur vorgenommen.

Für die E/E-Architekturmodellierung gibt es keine Standards [120], weshalb die meisten OEMs eigene Lösungen entwickelt haben und sie mit kommerziell verfügbaren Werkzeugen kombinieren [127, 170]. Nachfolgend werden die Ziele, die Modellierungsebenen und Vorgehensweisen und das Modellierungswerkzeug näher beschrieben.

3.2.1 Ziele der E/E-Architekturmodellierung

Im Allgemeinen werden durch die Modellierung der E/E-Architektur im Rahmen der Entwicklung einer E/E-Architektur mehrere Ziele verfolgt, die nachfolgend näher erörtert werden [16, 131, 132].

Frühzeitiger Entwurf, Bewertung, Dokumentation, Absicherung der E/E-Architektur und Technologieentscheid In der frühen Phase der E/E-Architekturentwicklung werden unterschiedliche Konzepte für die Auslegung der E/E-Architektur modelliert und bewertet. D.h. die Auslegung der E/E-Architektur wird durch das E/E-Architekturmodell dokumentiert und kann für die virtuelle Absicherung der E/E-Architektur verwendet werden. Im Zuge dessen wird das E/E-Architekturmodell auch eingesetzt, um eine Bussystemwahl bzw. einen Technologieentscheid durchzuführen. Ursache dafür ist, dass die unterschiedlichen Konzepte für die Auslegung der E/E-Architektur unterschiedliche Bussysteme oder Technologien zur Realisierung nutzen.

Beispiel Mit Hilfe des E/E-Architekturmodells kann die Integrierbarkeit von Innovationen aus E/E-Architektursicht bewertet werden (Abbildung 3.2 a)). Dabei kann z.B. die erwartete Buslast abgeleitet und damit das Bussystems gewählt bzw. die Technologie entschieden werden.

Erfassen und Bewerten von Auswirkungen bei E/E-Architekturänderungen Änderungen an der E/E-Architektur können in ein vorhandenes E/E-Architekturmodell modelliert werden. Zum einen wird durch diese Modellierung die Änderung erfasst und transparent gemacht. Zum anderen werden die Auswirkungen deutlich und können bewertet werden.

Beispiel Wechselt eine ECU zu einem anderem Bus (Abbildung 3.2 b)), wird dieser Wechsel als Änderung im E/E-Architekturmodell modelliert und die Auswirkungen, z. B. auf die Buslast, ausgewertet.

Optimierung der E/E-Architektur in einer frühen Entwicklungsphase Durch die ganzheitliche Modellierung der E/E-Architektur wird eine Übersicht und eine ganzheitliche Bewertung ermöglicht. Es können mehrere Konzepte modelliert und verglichen werden. Diese Mechanismen werden genutzt um die E/E-Architektur schon in der frühen Entwicklungsphase zu optimieren.

Beispiel An welchen Bus eine ECU gehängt wird, entscheidet darüber, welche Leitungen verwendet werden. So können Leitungslängen reduziert werden, wenn ECUs an einem statt an zwei Bussystemen lokalisiert werden. (Abbildung 3.2 c)).

3.2.2 Ebenenmodell

Um die Komplexität, insbesondere der kombinatorischen Vielfalt der möglichen E/E-Architektur-Konzepte, zu erfassen, ist eine formale Beschreibung der E/E-Architektur erforderlich [16]. Für solch eine formale Beschreibung wird eine Architecture Description Language (ADL) verwendet [79].

Für das Automotive-Umfeld wurden mehrere ADLs entwickelt, z. B. AML [45], EAST-ADL [100] und AADL [40]. In der E/E-Architekturentwicklung hat sich allerdings ein Modell etabliert [147], das

- Ansätze von EAST-ADL und AUTOSAR, die auf eine Beschreibung der Fahrzeugtopologie verzichten,
- mit den Ansätzen zur Beschreibung und Austausch von Leitungssatz, wie z. B. KBL [163],

kombiniert [1]. Die zugrunde liegende Beschreibung EEA-ADL wird hierbei durch ein Metamodell beschrieben, das mehr als 200 Klassen, 500 Attribute und ebenso viele Relationen umfasst [104].

Die EEA-ADL ist konform zum Meta Object Facility (MOF)-Standard. Dieser Standard [78]

- dient zur Beschreibung von Metamodellen,
- ist selbstbeschreibend,
- wird durch die Object Management Group (OMG) erstellt und
- sieht typischerweise vier Abstraktionsebenen vor.

Hierbei abstrahiert eine Ebene die darunterliegende Ebene wie folgt:

M3-Ebene - Meta-Metamodell-Ebene Beinhaltet die Beschreibung der Struktur und Semantik zur Definition der Meta-Metadaten (z. B. MOF).

M2-Ebene - Metamodell-Ebene Beinhaltet Beschreibungen (Meta-Metadaten), die die Struktur und die Semantik der Meta-Daten beschreiben (z. B. EEA-ADL).

M1-Ebene - Modell-Ebene Beinhaltet Meta-Daten, die die Daten auf der Informations-Ebene beschreiben (z. B. Funktionsnetz-Modell).

M0-Ebene - Informations-Ebene Beinhaltet die Daten, die durch ein Modell beschrieben werden sollen (z. B. Funktionsnetz).

Die EEA-ADL führt hierarchische Modellierungsebenen ein und stellt eine Verbindung zwischen den Modellartefakten der einzelnen Ebenen über Mappings her [16, 47, 104, 131, 132, 121, 147]. Die Modellartefakte, die eine E/E-Architektur beschreiben, sind Teil einer Produktlinie (E/E-Architekturmodell). Im Einzelnen können mit den Modellierungsebenen die folgenden Teilaspekte einer E/E-Architektur beschrieben werden (Abbildung 3.3):

REQ/CF - Anforderungsebene Modellieren und Sammeln von Anforderungen als technische Spezifikationen (Requirement (REQ), z. B. Funktionale Sicherheit (FuSi)), kundenerlebbaren Funktionen (Customer feature (CF), z. B. Distronic) und Feature-Funktions-Netzwerk (Use-Cases, Kette aus auslösendem Element, Funktionalität und umsetzendem Element [103]).

LA - Logische Architektur Modellierung von Fahrzeugfunktionen durch Zerlegung in einzelne Funktionalitäten (z. B. Längsregler)

- SW - Softwarearchitektur** Modellierung der Software-Struktur von Systemen (z. B. AUTOSAR-SWC)
- I - Implementierung** Modellierung der Struktur der Implementierung (z. B. Struktur der Simulink-Modelle)
- NET - Vernetzungsebene** Modellierung der logischen Vernetzung der E/E-Komponenten über Bussysteme, Anbindungen und konventionellen Verbindungen.
- LV - Leistungsversorgungsebene** Modellierung der elektrischen Infrastruktur mit elektrischen Potentialen, dem Bordnetzkonzept und Massekonzept.
- WH - Leitungssatzebene** Modellierung der Implementierung der logischen Vernetzungen als elektrische und physikalische Verbindungen (inklusive Eigenschaften wie Länge, Gewicht und Kosten).
- TOP - Topologie** Modellierung der geometrischen Eigenschaften der Bauräume und Leitungsführungen.
- Comm - Kommunikation** Ebenenübergreifende Verwaltung der Kommunikation (z. B. Zugehörigkeit von Signalgruppen und Schnittstellen, Scheduling der Busse, Verwaltung der Gateway-Konfiguration)

Wie und ob die einzelnen Modellierungsebenen in der Praxis eingesetzt werden ist nicht standardisiert und wird von den OEMs unterschiedlich gehandhabt [120, 127, 170]. Es werden unterschiedliche Begrifflichkeiten für die Ebenen verwendet. So wird die Anforderungsebene auch Ebene für Funktionsumfang, Funktionsstruktur oder Features genannt. Die Logische Architektur-Ebene wird auch als Funktionsarchitektur und Funktionsnetz bezeichnet. Tiefere Untersuchungen zu dieser Ebene stammen aus dem Bereich der E/E-Systementwicklungsmethodik und zeigen (oft auf der Basis von UML) eine abstrahierende, von der Implementierung unabhängige Sicht auf die verteilte Software der ECUs auf [4, 22, 23, 55, 56, 57, 157, 167, 175]. Die Vernetzungsarchitektur kann auch als E/E-Komponenten-Ebene oder Technische Architektur bezeichnet. Die Topologie-Ebene wird teilweise auch als Bauraum-Ebene bezeichnet.

3.2.3 Modellbasierte Bewertung und Absicherung von E/E-Architekturen bei Mercedes-Benz Cars

Da einzelne OEMs die E/E-Architekturmodellierung unterschiedlich handhaben [120, 127, 170], fokussiert sich der nachfolgende Abschnitt auf die Verfahren der Mercedes-Benz Cars, die die Basis für diese Arbeit bilden. Der OEM

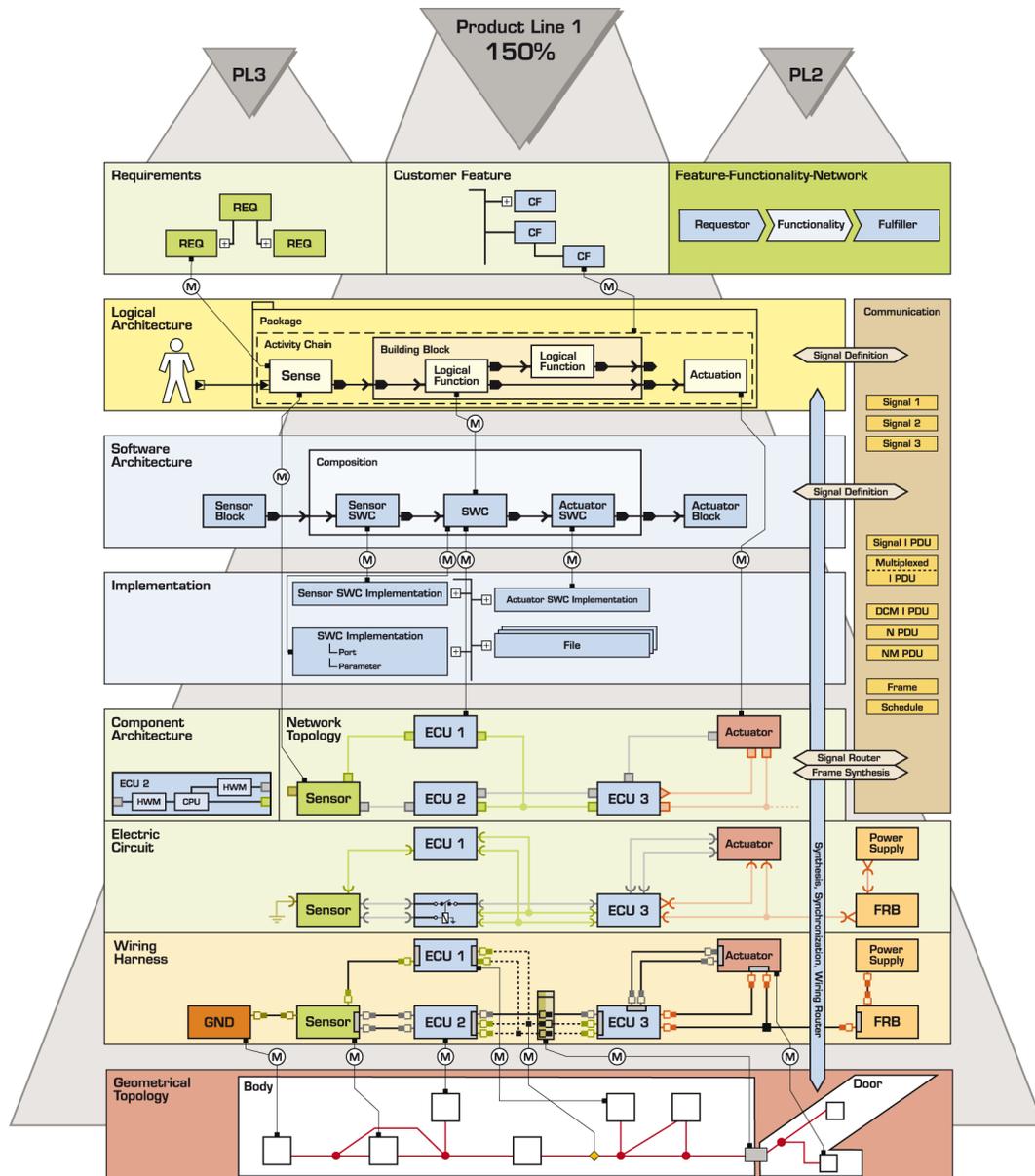


Abbildung 3.3: E/E-Architektur-Ebenen der EEA-ADL. Hierarchisch strukturierte Modellierungsebenen, die jeweils Teilaspekte der E/E-Architektur abbilden [16, 47, 104, 131, 132, 147]. Abbildung zeigt maximal mögliche Modellierung. Abbildung entnommen aus [121]. Verwendung genehmigt durch Vector Informatik GmbH. M Mapping; PL Product line (Produktlinie)

führte die EEA-ADL-basierte E/E-Architekturmodellierung im Jahr 2005 ein [16] und verwendet den kombinierten Entwurf [131, 132].

Es liegt eine heterogene Werkzeug-Landschaft vor. Diese erfordert eine Modellierungsebenen-übergreifende Verwaltung, da Konzeptuntersuchungen Modellierungsebenen-übergreifend durchgeführt werden. Dabei stehen die Modellierungsebenen in Beziehung zu anderen Werkzeugen und werden wie folgt verwendet:

REQ/CF - Anforderungsebene Nicht verwendet, da Anforderungen in DOORS spezifiziert werden.

LA - Logische Architektur Nicht verwendet, da diese Ebene nicht für AUTOSAR-Import/-Export ausgelegt ist.

SW - Softwarearchitektur → **FN - Funktionsnetz** Keine Modellierung der SW-Architektur. Statt dessen Modellierung der Funktionen in Form eines Funktionsnetzes. Hierbei beschreibt eine SW-Komposition eine verteilte Funktion und beinhaltet alle ihre Funktionsbeiträge, die die Anteile der Funktion auf den E/E-Komponenten beschreiben. Basis für die E/E-Architekturmodellierung ist das Funktionsnetz eines Serienstandes beschrieben im OEM-spezifischen Werkzeug XDIS (Zentrale Vernetzungsdatenbank) importiert über AUTOSAR-Import [139].

I - Implementierung Nicht verwendet, da modellbasierte Entwicklung direkt in Simulink erfolgt.

NET - Vernetzungsebene Modellierung der logischen Vernetzung der E/E-Komponenten über Bussysteme, Anbindungen und konventionellen Verbindungen importiert über XDIS AUTOSAR-Import ggf. ergänzt um EPDM-Anteile (OEM-spezifisches Werkzeug zur Dokumentation von E/E-Komponenten).

LV - Leistungsversorgungsebene Modellierung der elektrischen Infrastruktur von Bordnetz- und Massekonzept.

WH - Leitungssatzebene Modellierung der Implementierung der logischen Vernetzungen als elektrische und physikalische Verbindungen (inklusive Eigenschaften wie Länge, Gewicht und Kosten). Serienentwicklung in E³.cable (Schaltplan) und LDorado (Zeichnung).

TOP - Topologie Modellierung der geometrischen Eigenschaften der Bauräume und Leitungsführungen. Serienentwicklung in CATIA/Siemens NX.

Comm - Kommunikation Ebenenübergreifende Modellierung der Kommunikation importiert über AUTOSAR-Import von XDIS.

Für jede E/E-Architektur-Plattform wird ein Modell (**Productline**) erstellt mit einem Umfang im aktuellen Projekt von 1,5 Millionen Modellartefakten, die sich folgt verteilen:

- 52% auf FN-Ebene,
- 22% auf NET-, LV-, WH-Ebene,
- 8% auf Comm,
- 6% Mappings,
- 5% Variantenmanagement,
- 3% TOP,
- 2% REQ/CF (entstanden im Rahmen dieser Arbeit),
- 2% LA (entstanden im Rahmen dieser Arbeit).

Anders ausgedrückt wurden z.B. für die S/E/C-Klasse circa 1000 E/E-Komponenten, 4000 Kabel, 6000 Pins und 300 Diagramme modelliert [66].

Diese Datenmenge entsteht zum einen natürlich weil weite Teile des Fahrzeugs modelliert werden, aber auch für Innovationen werden signifikante Teile modelliert. Gerade für Innovationen werden Schnittstellen entworfen und mehrere mögliche Varianten modelliert, die im Gesamtverbund aller Funktionen bewertet und abgesichert werden. In die Bewertung fließen zum Teil auch die Ergebnisse von Metriken [48] ein. Einige wichtige Metriken sind hierbei [147]:

- Buslast,
- Leitungssatzgewicht und
- Leitungssatzkosten.

Zwar sind einige Kriterien automatisiert bewertbar, dennoch liegt der Schwerpunkt bei der E/E-Plattform-Entwicklung auf den nicht messbaren Kriterien wie z.B. technische Umsetzbarkeit oder Wiederverwendbarkeit (Abschnitt 3.1.2)

3.2.4 Variantenmanagement von E/E-Architekturen

Bei der Entwicklung der E/E-Architektur werden unterschiedliche Konzepte für die Auslegung der E/E-Architektur modelliert (Abschnitt 3.2.1) und bewertet (Abschnitt 3.2.3). Dabei unterstützt das Variantenmanagement als Werkzeugmechanismus zur Konfiguration und Verwaltung von E/E-Architektur-Varianten als auch zur Zuordnung von Konzepten zu E/E-Architektur-Varianten (Definition 2.7).

Beispiel Eine E/E-Architektur-Variante kann die Funktionen, Hardware und Vernetzung für einen Ecktyp (Serienausstattung) beschreiben. Die E/E-Architektur-Varianten, die einen Trackingtyp (z. B. durchschnittlich verkaufte Ausstattungen) oder einen Maxtyp (Vollausstattung) beschreiben, können neben weiteren Funktionen und Hardware z. B. ganz andere Bus-technologien verwenden (CAN im Ecktyp, Flexray im Trackingtyp und Automotive-Ethernet im Maxtyp).

Im E/E-Architekturmodell entspricht eine E/E-Architektur-Variante somit einer Teilmenge aller Modellartefakte des E/E-Architekturmodells. Bei der Zuordnung der Modellartefakte des E/E-Architekturmodells zu einer E/E-Architektur-Variante unterstützen Ausstattungen und technische Konzepte. Hierbei gibt es mehrere alternative Ausstattungen und alternative technische Konzepte. Einer Ausstattungs-Alternative bzw. einer technischen Konzept-Alternative werden Modellartefakte zugeordnet. Damit nicht jedes Modellartefakt einzeln den Alternativen und damit der E/E-Architektur-Variante zugeordnet werden muss, wird der Propagations-Mechanismus (Definition 3.1) verwendet:

Definition 3.1 (Propagation): Die Propagation umfasst eine Mustersuche auf dem E/E-Architekturmodell mit der Modellartefakte anhand einer definierten Suchregel gefunden werden und einer E/E-Architektur-Variante zugeordnet werden können.

Um in der heutigen E/E-Architekturmodellierung eine E/E-Architektur-Variante zu definieren, wird in der Praxis eine Vorgehensweise verwendet, die abhängig vom Fokus des E/E-Architekturmodells ist. Für die E/E-Architekturmodellierung des Leitungssatzes wird folgende Vorgehensweise angewendet um eine E/E-Architektur-Variante zu definieren:

- Zuerst werden die Hardware-Komponenten einer E/E-Architektur-Variante zugeordnet und dienen als Keime für die Propagation im Leitungssatz.
- Ausgehend von den Hardware-Komponenten werden die Anbindungen und Verbindungen zwischen den Hardware-Komponenten propagiert um sie der E/E-Architektur-Variante zuzuordnen.
- Ausgehend von den Anbindungen und Verbindungen werden danach die Schaltplanverbindungen, Schaltplanpins, Leitungen, Pins, Stecker propagiert um sie der E/E-Architektur-Variante zuzuordnen.
- Abschließend werden die Einbauorte für die Hardware-Komponenten und weitere Topologieelemente propagiert um sie der E/E-Architektur-Variante zuzuordnen.

Für die E/E-Architekturmodellierung im E/E-Architekturentwurf wird folgende Vorgehensweise angewendet um eine E/E-Architektur-Variante zu definieren:

- Zuerst werden einzelne Funktionsbeiträge von Funktionen des Funktionsnetzes einer E/E-Architektur-Variante zugeordnet.
- Die Hardware-Komponenten werden einer E/E-Architektur-Variante zugeordnet oder von Funktionsbeiträgen ausgehend werden die notwendigen Hardware-Komponenten der E/E-Architektur-Variante zugeordnet. Soll der Leitungssatz betrachtet werden, wird wie oben beschrieben propagiert.

Insgesamt kann die Definition der E/E-Architektur-Varianten in der Praxis sehr anspruchsvoll und zeitaufwendig werden: Allein bezogen auf die betrachteten Ausstattungen (die wiederum mehrere technische Konzepte haben können) werden bei der ersten Grobanalyse in der Entwicklung einer neuen E/E-Architektur meist 3 bis 5 Ausstattungsvarianten definiert. Wird die Entwicklung konkreter, werden eher 10 bis 20 Ausstattungsvarianten untersucht. Für die alle 3 Monate neu zu erstellenden Modellstände muss die Definition der E/E-Architektur-Varianten immer entsprechend angepasst werden.

3.2.5 Werkzeug für die E/E-Architekturmodellierung

Für die E/E-Architekturmodellierung gemäß EEA-ADL (Abschnitt 3.2.2) wurde im Jahr 2004 das E/E-Konzept-Tool von der Daimler AG mit der Firma Aquintos entwickelt. Aquintos, eine Ausgründung des Forschungszentrums Informatik (FZI) in Karlsruhe, war für die Implementierung verantwortlich [16]. Das Werkzeug wurde unter dem Namen PREEvision kommerzialisiert und wird seit 2009 durch die Vector Informatik GmbH [164] in Kooperation mit den Anwendern [136] weiterentwickelt. PREEvision

- integriert die verschiedenen Ebenen der E/E-Architektur [48],
- erlaubt einen gesamtheitlichen Einblick in die E/E-Architektur [48],
- ist am Markt verfügbar [120] und
- ist im Stande, den E/E-Architekturprozess widerzuspiegeln [120].

Zu den Kunden von PREEvision gehören neben der Daimler AG u. a. Audi, BMW, Bosch, Chrysler, Continental, Delphi, DENSO, Fiat, Ford, GM, Hitachi, Hyundai, Jaguar, John Deere, Land Rover, MAN, Porsche, PSA, Skoda, Toyota und Volkswagen, was es zum marktführenden Produkt für E/E-Architekturentwicklung und -management macht [177].

PREEvision ist ein CASE-Werkzeug (rechnergestützte Softwareentwicklung, computer-aided software engineering) das u. a. [2, 16, 36, 120, 131, 132, 136, 137, 177]:

- Austausch mit standardisierten Formaten, wie z.B. AUTOSAR oder KBL⁵ [163] ermöglicht,
- Editoren für graphisches Modellieren,
- umfangreiches Variantenmanagement,
- Metriken zur Bewertung der E/E-Architektur und
- ein Datenbankmodell mit der Fähigkeit der Zusammenarbeit auf einem Modell mit vielen Anwendern in unterschiedlichen, weltweit verteilten Standorten

anbietet.

3.3 Evolution in der Softwareentwicklung

Das Werkzeug für die E/E-Architekturmodellierung und die Methodik zur modellbasierten Bewertung und Absicherung von E/E-Architekturen wird im Rahmen dieser Arbeit durch Methoden aus der Softwareentwicklung erweitert um die Bewertung und Umsetzung der Evolution von E/E-Systemen (Zielsetzung 1.1) zu ermöglichen. Deshalb wird im nachfolgenden Abschnitt die Evolution in der Softwareentwicklung näher beleuchtet.

Die Evolution in der Software-Entwicklung, einschließlich deren Gesetze [97], wurden schon in den 1980er Jahren untersucht. Haupttreiber für die Evolution von Software-Systemen sind Technologieänderungen [114]. Die Evolution ist auch im Hinblick auf Software-Architektur ein weites Forschungsfeld, das in Übersichtsartikeln tiefgehend diskutiert wird [3, 19, 20, 173]. An dieser Stelle werden deshalb nur für diese Arbeit relevante Methoden aus der Software-Entwicklung nachfolgend kurz aufgeführt.

3.3.1 Aspect weaving

Die Aspektorientierte Programmierung beschreibt eine Möglichkeit um Funktionalitäten zu modularisieren, die sich im Quelltext über viele Klassen hinweg verteilen (engl. *crosscutting concern*) [92]: Solche Funktionalitäten können z.B. Security- oder Protokollierungs-Mechanismen sein und werden hier als

⁵Kabelbaumliste (KBL), XML-Dateiformat-Standard zur Beschreibung eines Kabelbaums

Aspekte (engl. *aspect*) bezeichnet. Aspekte sind somit Eigenschaften, für die die Implementierung nicht klar gekapselt werden kann. Die Aspektorientierte Programmierung unterstützt die Abstraktion und Komposition der Software-Komponenten und ihrer Aspekte, wobei das neu hinzugefügte Verhalten an den als *joint points* (engl.) bezeichneten Stellen als *advice* (engl.) bezeichnet wird und das Einpflegen des advice als *aspect weaving* bezeichnet wird [92]. Für die Aspektorientierte Programmierung ist der Programmfluss entscheidend und lässt sich deshalb unter der Verwendung von Sequenz-Diagrammen modellieren [85].

3.3.2 Ontology-based modelling

Eine Beschreibung der Software-Architektur auf formale Weise unterstützt Abschätzungen von Auswirkungen bei Software-Architekturänderungen. Basierend auf Ontologien (engl. *ontology*) kann das Wissen über die Software-Architektur formal beschrieben werden und anschließend die Software-Architekturentwicklung mit dem entstehenden Modell begleitet werden. Die Software-Architekturentscheidungen lassen sich dann visualisieren [30]. Eine formale Beschreibung der Evolution der Ontologie kann mittels Graphentransformation [150] und basierend auf Beschreibungslogik vorgenommen werden [115].

Die auf Beschreibungslogik basierende Ontologie [115] umfasst hierbei

- eine leicht erweiterbare Modellierungssprache zur Beschreibung und formalen Definition von Software-Architektur-Auslegungen,
- mehrere Operatoren um Software-Architektur-Auslegungen abzuleiten, zu kombinieren und zu vergleichen und
- eine Unterstützung zur Nachverfolgbarkeit und Durchsuchung des Modells.

3.3.3 Feature Oriented Refactoring (FOR)

Eine weitere Methode ist das Feature Oriented Refactoring (FOR) [99], die eine zu *aspect weaving* komplementäre Technologie [5] aus dem Feld der merkmalsorientierten Programmierung [13, 122] ist. Sie basiert auf der Erkenntnis, den Quelltext der Implementierung für ein Merkmal (Definition 3.2) zu separieren in einen Basis-Anteil für das eigentliche Merkmal und Anteile für die Interaktion des Merkmals mit anderen Merkmalen [122].

Interaktion umschreibt hierbei die Anpassung im Verhalten, falls mehrere Merkmale gleichzeitig vorhanden sind [26]. Ist dies der Fall, dann sind die Implementierungen der Merkmale nicht voneinander unabhängig, was als *optional*

feature problem (engl.) bezeichnet wird und die Variabilität einer Produktlinie einschränkt [89].

Das FOR wird auf bestehende Softwaresysteme angewendet. Im Gegensatz dazu gibt es Methoden um ein Software-System mit Merkmalen initial zu entwickeln, z. B. Scaling Step-Wise Refinement [14].

Definition 3.2 (Merkmal): Ein Merkmal ist eine Erweiterung der Funktionalität der Software [99].

FOR wird als ein Prozess beschrieben, der eine Software in Merkmale zerlegt, so dass ein Merkmal eine Erweiterung in der Funktionalität der Software beschreibt. Die Anwendung von FOR gewinnt eine merkmalsbasierte Beschreibung der bestehenden Software zurück [99], die

- die Abhängigkeiten zwischen den einzelnen Merkmalen identifiziert,
- die Beziehung zwischen Merkmalen und der implementierenden Softwareanteile beschreibt und
- die Software anschließend merkmalsorientiert durch das Hinzufügen und Entfernen von Merkmalen sukzessiv weiterzuentwickeln erlaubt.

FOR liegt eine Algebra zugrunde [99], deren Kern an dieser Stelle kurz umrissen wird. Zentral ist, dass einem Basisprogramm sequenziell Funktionen hinzugefügt werden um die Merkmale mit den entsprechenden Funktionalitäten umzusetzen. Das entstehende Programm P mit der Basis B und den Funktionen F und G für die entsprechenden Merkmale lässt sich wie folgt beschreiben:

$$P = G(F(B)) \quad (3.1)$$

Hierbei dürfen die Merkmale Methoden anderer Merkmale nur erweitern und nicht etwa Definitionen von Variablen oder Methoden überschreiben. Benötigt und erweitert ein Merkmal G Methoden eines anderen Merkmals F , so wird die Wiederverwendbarkeit des Merkmals G beschränkt (gemäß *optional feature problem*).

Zur Lösung wird mit FOR die Software strukturiert [99]: Das Basis-Programm B besteht aus genau einem Basis-Modul b (Definition 3.3). Ein Merkmal F besteht aus einem ein Basis-Modul f und einem Derivat-Modul (Definition 3.4). Das Derivat-Modul $\delta b/\delta F$ enthält die Software-Anteile, die durch das Vorhandensein von Merkmal F dem Basis-Modul b hinzugefügt werden müssen.

Definition 3.3 (Basis-Modul): Ein Basis-Modul enthält die Funktionen, die das Merkmal dem Programm hinzufügen soll.

Definition 3.4 (Derivat-Modul): Ein Derivat-Modul bildet die Interaktion der Merkmale ab und enthält diejenigen Software-Anteile, die durch das Vorhandensein des Merkmals einem Basis-Modul hinzugefügt werden sollen.

Das entstehende Programm mit der Basis B und den Funktionen für das Merkmal F lässt sich wie folgt beschreiben:

$$[F(B)] = f + \delta b / \delta F \bullet b \quad (3.2)$$

Hierbei sind $[\]$ die Modul-Ausdrucksweise, $+$ die disjunkte Vereinigungsmenge und \bullet das Einpflegen der Änderung am Basis-Modul [99]. Verallgemeinert gilt:

$$[F(x)] = f + \delta x / \delta F \bullet x \quad (3.3)$$

wobei x auch ein Programm mit entsprechenden Merkmalen sein kann. Um die Struktur mit mehreren Merkmalen zu beschreiben, werden noch weitere Regeln notwendig [99]: Das Einpflegen der Änderung an einem Modul entspricht dem Einpflegen der Änderung an seinen Einzelteilen (Gleichung 3.4). Das Einpflegen der Änderung an einem Basis-Modul hat keine Auswirkung auf ein anderes Basis-Modul (Gleichung 3.5). Ein Derivat ist ein Operator auf Modulen (Gleichung 3.6a), der beschreibt wie ein Merkmal ein Modul verändert. Die Änderung verteilt sich auf sowohl auf die Einzelteile (Gleichung 3.6b) als auch auf das Einpflegen einer Änderung (Gleichung 3.6c).

$$\delta b / \delta F \bullet b = \delta b / \delta F \bullet (b_1 + b_2) = \delta b / \delta F \bullet b_1 + \delta b / \delta F \bullet b_2 \quad (3.4)$$

$$\delta b / \delta F \bullet c = c \quad (3.5)$$

$$\delta / \delta F (b) = \delta b / \delta F \quad (3.6a)$$

$$\delta / \delta F (a + b) = \delta a / \delta F + \delta b / \delta F \quad (3.6b)$$

$$\delta / \delta F (a \bullet b) = \delta a / \delta F \bullet \delta b / \delta F \quad (3.6c)$$

Damit kann die Struktur der Programme B (Gleichung 3.7), $F(B)$ (Gleichung 3.8) und $G(F(B))$ (Gleichung 3.9) wie folgt beschrieben werden [99]. Für das letzte Programm werden Derivat-Module höherer Ordnung nötig: Das Derivat-Modul $\delta^2 b / \delta G \delta F$ bildet die Interaktion der Merkmale G und F ab und enthält diejenigen Software-Anteile, die dem Derivat-Modul $\delta b / \delta F$ hinzugefügt werden müssen. Insgesamt besteht also das Programm $G(F(B))$ aus

- den neuen Funktionen für das Merkmal G , die im Basis-Modul bereitgestellt werden (g),
- der Änderung durch das Merkmal G , die im Basis-Modul f eingepflegt werden ($\delta f / \delta G \bullet f$) und
- der Änderung durch die Merkmale G und F , die im Basis-Modul b eingepflegt werden ($\delta^2 b / \delta G \delta F \bullet \delta b / \delta G \bullet \delta b / \delta F \bullet b$).

Ist keine Änderung an bestehenden Basis-Modulen durch das Hinzufügen eines Merkmals notwendig, so enthält das entsprechende Derivat-Modul keine Software-Anteile.

$$[B] = b \quad (3.7)$$

$$[F(B)] = [F](B) \stackrel{(3.3)}{=} f + \delta[B]/\delta F \bullet [B] \stackrel{(3.7)}{=} f + \delta b/\delta F \bullet b \quad (3.8)$$

$$\begin{aligned} [G(F(B))] &= [G](F(B)) \stackrel{(3.3)}{=} g + \delta[F(B)]/\delta G \bullet [F(B)] \\ &\stackrel{(3.8)}{=} g + \delta(f + \delta b/\delta F \bullet b)/\delta G \bullet (f + \delta b/\delta F \bullet b) \\ &\stackrel{(3.6b),(3.6c)}{=} g + \delta f/\delta G \bullet \delta^2 b/\delta G \delta F \bullet \delta b/\delta G \bullet (f \\ &\quad + \delta b/\delta F \bullet b) \\ &\stackrel{(3.4)}{=} g + \delta f/\delta G \bullet \delta^2 b/\delta G \delta F \bullet \delta b/\delta G \bullet f \\ &\quad + \delta f/\delta G \bullet \delta^2 b/\delta G \delta F \bullet \delta b/\delta G \bullet \delta b/\delta F \bullet b \\ &\stackrel{(3.5)}{=} g + \delta f/\delta G \bullet f + \delta^2 b/\delta G \delta F \bullet \delta b/\delta G \bullet \delta b/\delta F \bullet b \end{aligned} \quad (3.9)$$

Mit diesem Hintergrund wird vorgeschlagen, ein bestehendes Programm zu refaktorisieren indem [99]:

- Merkmale und deren Reihenfolge definiert,
- Variablen und Methoden hinsichtlich der Zugehörigkeit zu genau einem Basis-Modul markiert,
- Basis- und Derivat-Module initial gemäß der vorgenommenen Markierung refaktoriert,
- mittels Wissen über das Programm und der Abhängigkeiten der Merkmale die Derivat-Module refaktoriert und
- schließlich Varianten des ursprünglichen Programms mit den entsprechenden Basis- und Derivat-Modulen generiert werden.

3.4 Wiederverwendung beim Produktlinien Engineering

Neben Methoden für die Evolution sollen im Rahmen dieser Arbeit noch Methoden für die Wiederverwendung aus der Softwareentwicklung für die modellbasierte Bewertung und Absicherung von E/E-Architekturen übernommen werden um E/E-Systeme in die E/E-Architekturmodellierung einzubinden

(Zielsetzung 1.2) und ihre Wiederverwendung zu ermöglichen. Dementsprechend wird im nachfolgenden Abschnitt ein Fokus auf die Wiederverwendung gelegt.

Die Idee, Software strategisch wiederzuverwenden, ist lange bekannt [117, 118] und eine systematische Erforschung wurde durchgeführt [17, 113]. Auch für die Automobilindustrie ist die Adaption und Verwendung von Software-Produktlinienansätzen von zentraler Bedeutung [23, 54, 155]. In den nachfolgenden Abschnitten werden diese für die Arbeit wichtigen Aspekte kurz beschrieben.

3.4.1 Produktlinien in der Entwicklung von Software

Die Produktlinien in der Softwareentwicklung (Definition 3.5, ggf. auch als Produkt-Familie bezeichnet [162]) zeichnen sich dadurch aus, dass [17, 105, 113, 155]:

- eine gemeinsame Menge an Merkmalen (Definition 3.2) für die Produktlinie verwendet und verwaltet wird,
- aus diesen Merkmalen gemeinsame Artefakte (auch als *core assets* (engl.) bezeichnet) entwickelt werden (z.B. Anforderungen, Software-Architektur, Dokumentationen, Modelle, wiederverwendbare Softwarekomponenten, Spezifikationen),
- die Entwicklung einem Paradigma folgt, dass als Software Product Line Engineering bezeichnet wird, und mit dem applizierbare Software-Komponenten ausgewählt, durch Variantenmechanismen zugeschnitten und auf einer zugrundeliegenden Software-Architektur verbunden werden.

Definition 3.5 (Software-Produktlinie): Eine Software-Produktlinie ist eine Menge an Software-intensiven Systemen [113]. Sie teilen eine gemeinsame, gezielte Menge an Merkmalen, die die Anforderungen oder Aufgaben eines bestimmten Marktsegments erfüllen. Sie werden von einer Menge gemeinsamer Artefakte nach einem vorgeschriebenen Paradigma entwickelt.

Erfahrungen bei der Verwendung von Produktlinien zeigen u. a. [17, 105, 113, 154]:

- eine erhöhte Wiederverwendungsrate,
- eine verkürzte Entwicklungszeit,
- bei verringerten Kosten und
- verbesserter Qualität.

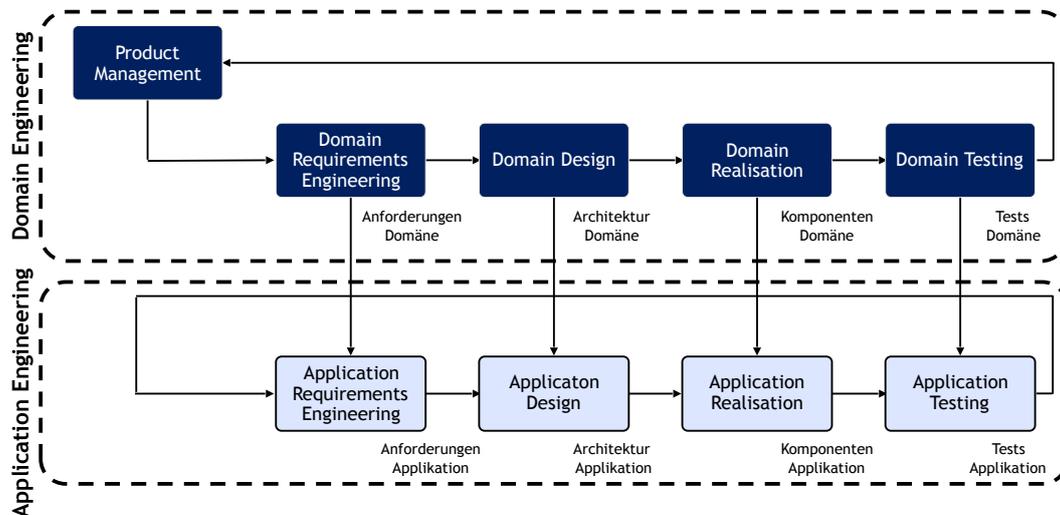


Abbildung 3.4: Referenzprozess für die Entwicklung von Software-Produktlinien. Der für diese Arbeit relevante Referenzprozess besteht aus dem Domain Engineering und dem Application Engineering. Abbildung nach [17].

Ferner wurden Referenzprozesse für die Entwicklung von Software-Produktlinien aufgezeigt [17, 113]. Der für diese Arbeit relevante Referenzprozess (Abbildung 3.4) besteht aus den Teilen Domain Engineering (Definition 3.6) und Application Engineering (Definition 3.7).

Definition 3.6 (Domain Engineering): Domain Engineering ist der Prozess des Software Product Line Engineerings, mit dem die Gemeinsamkeiten und die Variabilität der Software-Produktlinie definiert und realisiert werden [17].

Definition 3.7 (Application Engineering): Application Engineering ist der Prozess des Software Product Line Engineerings, mit dem die Applikationen der Software-Produktlinie unter Wiederverwendung von Artefakten aus dem Domain Engineering und Ausnutzen der Variabilität der Software-Produktlinie erstellt werden [17].

Hierbei umfasst das Domain Engineering im Einzelnen [17]:

Product Management Definition des Betrachtungsumfangs der Software-Produktlinie (auch als *scoping* (engl.) bezeichnet) einschließlich gemeinsamer und variabler Merkmale, rollout zukünftiger Produkte, Liste wiederverwendbarer Produkte bzw. gemeinsamer Artefakte.

Domain Requirements Engineering Spezifikation gemeinsamer und variabler Anforderungen in wiederverwendbarer textueller oder modellbasierter Form und eines Merkmalsmodells.

Domain Design Entwurf der Software-Architektur, die als Struktur für alle Anwendungen verwendet wird.

Domain Realisation Entwicklung und Implementierung der wiederverwendbaren Software-Komponenten.

Domain Testing Validierung und Verifikation wiederverwendbarer Software-Komponenten durch Testen gegen deren Spezifikation.

Das Application Engineering umfasst im Einzelnen [17]:

Application Requirements Engineering Spezifikation noch nicht im Rahmen des Domain Requirements Engineerings erfasster Anforderungen nach einer Delta-Analyse.

Application Design Instanziierung einer Software-Architektur für die Applikation durch Auswahl, Konfiguration erforderlicher Anteile der Domain Architektur und Einfügen von für die Applikation spezifischen Anpassungen.

Application Realisation Entwicklung der Applikation durch Auswahl und Konfiguration wiederverwendbarer Software-Komponenten und Implementierung von für die Applikation spezifischen Anteilen.

Application Testing Validierung und Verifikation der Applikation durch Testen gegen die Spezifikation unter Nutzung wiederverwendbarer Tests aus dem Domain Testing.

In der Automobilindustrie ist die Einführung von Software-Produktlinien

- beeinflusst von den Kundenbeziehungen zwischen OEM und Zulieferer [158],
- beeinflusst von Standardisierungen wie Standard-Software [94] und
- verbunden mit einigen Herausforderungen und Risiken [155, 166].

Für die modellbasierte Softwareentwicklung [28] werden derzeit Adaptionen von Software-Produktlinien für die Automobilindustrie untersucht [33, 71, 108, 156]. Gleiches gilt für die modellbasierte E/E-Architekturentwicklung (Abschnitt 3.4.3).

3.4.2 Modellierung von Merkmalen mittels Feature-Oriented Domain Analysis (FODA)

Die Software-Produktlinie besteht aus gemeinsamen und variablen Merkmalen aus denen die Artefakte entwickelt werden. Dementsprechend ist gerade die Definition, Beschreibung und Anpassung der Merkmale (Definition 3.2) und

damit der Variabilität im Rahmen des Product Managements ein wichtiger Beitrag [17, 162].

Dafür werden [17]:

- Variationspunkte (Definition 3.8) beschrieben,
- auf alle Artefakte (Anforderungen, Software-Architektur, Tests, usw.) angewendet,
- um Kontextinformationen angereichert (z. B. weshalb die Variation betrachtet wird),
- im Application Engineering für eine Variante (Definition 3.9) ausgewählt (auch als *binding* (engl.) bezeichnet)
- über ein orthogonales Variabilitätsmodell modelliert (Definition 3.10).

Definition 3.8 (Variationspunkt): Ein Variationspunkt ist eine Repräsentation eines variablen Gegenstands der realen Welt oder einer variablen Eigenschaft dieses Gegenstands [17].

Definition 3.9 (Variante): Eine Variante ist eine Menge von Variationspunkten [17].

Definition 3.10 (Orthogonales Variabilitätsmodell): Ein Orthogonales Variabilitätsmodell ist ein Modell, das die Variabilität einer Software Produktlinie definiert und die Variabilität, die in anderen Modellen der Software Entwicklung (z. B. Merkmalsmodelle, Use-Case-Modelle, Software-Komponenten-Modelle, Test-Modelle) definiert werden, in Verbindung setzt [17].

Eine Methode, mit der die Variabilität beschrieben werden kann, ist die Feature-Oriented Domain Analysis (FODA) [87, 88]. Hierbei wird ein Merkmalsmodell (Definition 3.11) erstellt. Es enthält hierarchisch strukturierte Merkmale (mehrere Kind-Merkmale sind einem Eltern-Merkmal zugeordnet), die im Allgemeinen die folgenden Beziehungen zueinander haben können:

Mandatory Kind-Merkmal muss ausgewählt werden,

Optional Kind-Merkmal kann ausgewählt werden,

OR mindestens ein Kind-Merkmale muss ausgewählt werden,

XOR genau ein Kind-Merkmal muss ausgewählt werden,

REQ die Auswahl des Merkmals erfordert die Auswahl eines weiteren Merkmals,

EX die Auswahl des Merkmals schließt die Auswahl eines weiteren Merkmals aus.

Definition 3.11 (Merkmalsmodell): Ein Merkmalsmodell repräsentiert die Merkmale einer Software-Produktlinie und die Beziehungen zwischen ihnen [87].

Das Merkmalsmodell kann grafisch mit Merkmalsdiagrammen dargestellt werden. Es wurde gezeigt, dass es eine wachsende Anzahl an Ansätzen mit einem Mangel an korrekten Vergleichen zwischen ihnen gibt [68].

In der Automobilindustrie kann diese visuelle Darstellung von Merkmalsmodellen hilfreich sein um die Komplexität, die mit einer große Menge an Variationspunkten einhergeht, greifbar zu machen [51, 153, 155].

3.4.3 Ansätze für Produktlinien in der modellbasierten E/E-Architekturmodellierung

Auch für die modellbasierte E/E-Architecturentwicklung wurden einige Ansätze entwickelt um Produktlinien in der Modellierung darzustellen. In diesem Abschnitt werden die für diese Arbeit relevanten Ansätze aufgezeigt.

Modulorientierter Produktlinienansatz

Die E/E-Architektur-Entwicklungsmethodik wurde um die Möglichkeit, E/E-Module der Modulstrategie (Abschnitt 2.3.4) zu modellieren, weiterentwickelt. Dafür wurde ein Prozess vorgestellt, der die E/E-Module in die E/E-Architecturentwicklung einbindet [82]. Der zugrunde liegende Produktlinienansatz wurde aus der Software-Produktlinienentwicklung abgeleitet (Abschnitt 3.4.1).

Ein Ansatz mit dem E/E-Module in die modellbasierte E/E-Architecturentwicklung [83] integriert werden, muss

- mehrere Modellierungsebenen einbinden,
- die unterschiedlichen Schnittstellen der einzelnen Modellierungsebenen bedienen,
- eine Nachverfolgbarkeit der in verschiedene E/E-Architekturen integrierten E/E-Module erlauben und
- die Schnittstellenkompatibilität überprüfen

können.

Die Einbindung von E/E-Modulen in die E/E-Architecturentwicklung hat mehrere Vorteile [84]:

- Es werden fertige E/E-Modulobjekte aus dem Modulbaukasten verwendet, was eine einmalige Modellierung der E/E-Modulobjekte, eine schnellere Modellierung und Absicherung von E/E-Architekturmodellen und eine verbesserte Modellqualität bzw. höheren Reifegrad zur Folge hat.
- Es können gleiche E/E-Modulobjekte in mehreren E/E-Architekturmodellen verwendet werden.
- Bei einer E/E-Modul-Evolution gibt es eine einfachere Verwaltung von E/E-Modellobjekten im Modulbaukasten.

Zusätzlich dazu wurde ein Ansatz vorgestellt, um mehrere E/E-Module gemeinsam aus dem Modulbaukasten wiederzuverwenden [84]. Dem Ansatz liegt ein Merkmalsmodell (Definition 3.11) zugrunde, das ein E/E-System mit mehreren E/E-Modulen assoziiert. Dadurch kann

- die Integration mehrerer E/E-Module gemeinsam einfach konfiguriert,
- die Abhängigkeiten zwischen den E/E-Systemen und E/E-Modulen verwaltet und
- die Konfiguration dokumentiert

werden.

Der Modulbaukasten (Abbildung 3.5) selbst hat die folgenden Eigenschaften [81]:

- Die E/E-Modulmodelle werden einmalig und zentral erstellt.
- Die E/E-Modulmodelle werden nur aus dem Modulbaukasten heraus zur Integration in E/E-Architekturen, zum Austausch von bereits integrierten E/E-Modulen und zur Modellierung von Innovationen wiederverwendet.
- Der Modulbaukasten ist gemäß den E/E-Modulen in der Modulstrategie strukturiert.

Insgesamt können im Modulbaukasten zwar E/E-Module modelliert, weiterentwickelt und in verschiedenen E/E-Architekturmodellen wiederverwendet werden. Aber die für die E/E-Architekturmodellierung so effiziente Wiederverwendung ist lediglich für E/E-Module verfügbar und nicht etwa für Funktionen. Diese sind in deutlich höherer Anzahl und Komplexität in der produktiven E/E-Architekturmodellierung vorhanden. Dementsprechend werfen sie in der Praxis die deutlich schwierigeren Fragestellungen auf (Abschnitt 1.3).

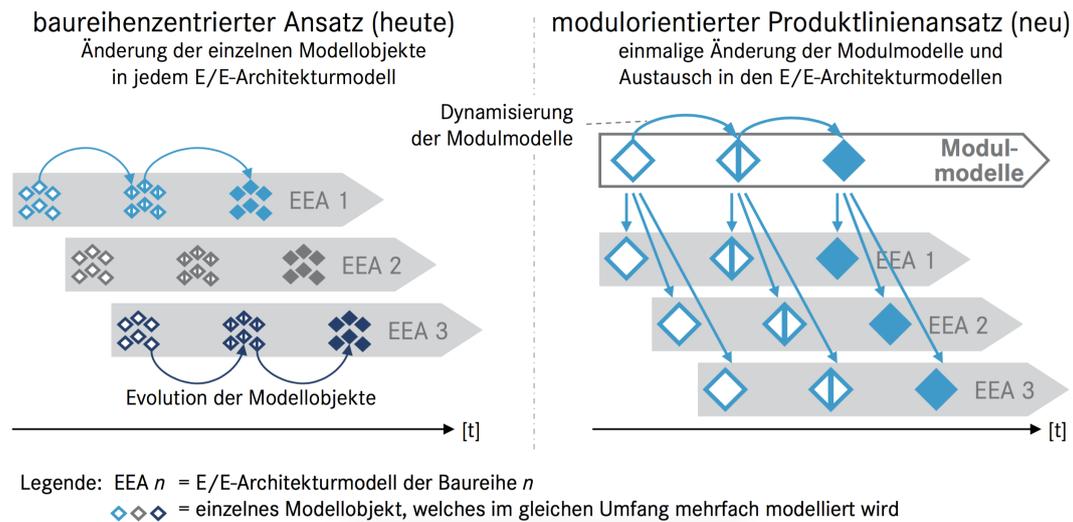


Abbildung 3.5: Modulatorientierter Produktlinienansatz zur Wiederverwendung von E/E-Modulen in der E/E-Architecturentwicklung. Statt für jedes E/E-Architekturmodell einzeln die E/E-Komponenten separat zu ändern (Baureihenzentrierter Ansatz), werden E/E-Module zentral in einem Modulbaukasten modelliert, weiterentwickelt und in den verschiedenen E/E-Architekturen wiederverwendet (Modulatorientierter Ansatz). Abbildung aus [81].

Kommerzielle Umsetzung von Wiederverwendungsmechanismen

Es gibt eine Unterstützung von Wiederverwendung durch das Werkzeug für die E/E-Architekturmodellierung PREEvision (Abschnitt 3.2.5). Sie umfasst allerdings lediglich die Wiederverwendung einzelner Modellartefakte, z.B. einer ECU, einer SWC oder einer SWComp.

Mit der Version 5.5 des Werkzeugs fanden die Modulbaukasten-Konzepte zur gemeinsamen Wiederverwendung einer Menge von Modellartefakten eine kommerzielle Umsetzung, z.B. eine ECU mit SWComp und Mapping. Das neue Wiederverwendungs-Konzept wird als **Reuse-Unit** bezeichnet und erlaubt die Wiederverwendung von Modellartefakten, die versionierbar sind. So ist beispielsweise eine SWComp versionierbar und damit können auch alle in ihr erhaltenen SWCs und Schnittstellen zwischen den SWCs (als **AssemblyConnector** bezeichnet) durch eine **ReuseUnit** wiederverwendet werden.

Die in einer **ReuseUnit** enthaltenen Modellartefakte werden durch den Benutzer in zwei Arten aufgeteilt:

- **Kern-Artefakte** ist die Menge von Modellartefakten, die die internen Artefakte und Beziehungen der **Reuse-Unit** darstellen und deren Integration in das E/E-Architekturmodell zwingend erforderlich ist (in [81] als Intra-Modulobjekte bezeichnet).

- **Rand-Artefakte** ist die Menge von Modellartefakten, die die Schnittstellen und externen Beziehungen enthalten und deren Integration in das E/E-Architekturmodell empfohlen wird (in [81] als Inter-Modulobjekte bezeichnet).

Insgesamt stellt das **ReuseUnit**-Konzept zwar eine signifikante Weiterentwicklung der durch das Werkzeug unterstützten Wiederverwendungsmechanismen dar, es ist jedoch in der Praxis nicht vollumfänglich und zudem aufwendig in der Benutzung. So führt z.B. die Verwendung von AUTOSAR-SWC in Verbindung mit **ReuseUnits** zu fehlerhaften Konsistenzchecks. Auch wurden Schnittstellen (**AssemblyConnector**) zwischen Blöcken auf der Logischen Architekturebene erst nach Anregung zu versionierbaren Modellobjekten. Erst mit dem Input aus dieser Arbeit in die Werkzeugentwicklung wurde das Wiederverwendungs-Konzept für die Verwendung von E/E-Systemen tauglich, da sie eine große Anzahl Schnittstellen umfassen.

Produktlinien in der modellbasierten E/E-Architecturentwicklung aus Zulieferer-Sicht

An dieser Stelle sei noch eine weitere bekannte modellbasierte E/E-Architektur-Entwicklungsmethode erwähnt, die auch auf Produktlinien basiert. Leitkunde für die Entwicklung der **ReuseUnits** war Continental Powertrain Systems (Regensburg). Im Fokus der Entwicklung stand die Wiederverwendung von **ReuseUnits** über mehrere Produktlinien hinweg und die Unterstützung von Baukastenansätzen.

Hierbei wird das Werkzeug für die E/E-Architekturmodellierung wie folgt eingesetzt [102]:

- **PREEvision** wird in die Entwicklungs-Toolkette eingegliedert. So wird beim Domain Engineering das Requirementsengineering in MKS RM, die Varianten- und E/E-Architekturmodellierung in **PREEvision** und das Design und die Implementierung in weiteren Design- und Implementierungstools durchgeführt. Beim Produkt Engineering wird das Requirements Engineering in MKS RM, die Produktanpassung und die Produktgenerierung in **PREEvision** und die Produktadaptierung in **PREEvision** bzw. weiteren Design- und Implementierungstools durchgeführt. Von dem Domain Engineering werden die Domainanforderungen für die Produktanforderungen wiederverwendet und die E/E-Architekturmodelle werden für die Produktarchitektur wiederverwendet.
- Somit ist **PREEvision** das zentrale Werkzeug für das Management der Produktlinien, in dem die Produktlinien aufgesetzt und die Produktvarianten angepasst und generiert werden.

- Auch ist PREEvision das zentrale Werkzeug für die E/E-Architekturaktivitäten mit der Beschreibung der Engine Systems Architecture auf der LA-Ebene, der Software Architecture auf der SW-Ebene und der Hardware Architecture auf der ECU-HW-Ebene.
- Zuletzt stellt PREEvision hier auch das Werkzeug für die Nachverfolgbarkeit zwischen Anforderungen, Features, E/E-Architektur und Implementierung dar.

Insgesamt werden bei dieser Vorgehensweise bereits bekannte Methoden angewendet und Produktlinien aus Sicht des Zulieferers beschrieben, der für Kunden die unterschiedlichsten Anpassungen vornehmen muss. Dementsprechend ist hier die Notwendigkeit der Beschreibung einer so weitreichenden Verblockung wie im Sinne von gesamten E/E-Modulen zu einem Gesamtfahrzeug natürlich nicht im Fokus.

3.5 Zusammenfassung

Der für diese Arbeit relevante Stand der Technik wird in diesem Kapitel vorgestellt. Näher beschrieben wird der E/E-Architecturentwurf (Abschnitt 3.1), der firmen- und branchenspezifisch, abhängig von der Geschäfts- und Produktcharakteristik und eine über mehrere Teile der Organisation hinweg integrative Entwicklungstätigkeit ist. Es wird in eine E/E-Architecturentwicklung für Fahrzeug-Plattformen und für Fahrzeug-Baureihen unterschieden. Beim Entwurf müssen Kriterien (Abschnitt 3.1.2), die sich zum Teil diametral gegenüberstehen, abgewogen werden um eine möglichst optimale Lösung zu finden. Dabei geht man mit einem Kombinierten Entwurf (Abschnitt 3.1.3) vor, der den Bottom-Up-Entwurf durch eine Top-Down-Perspektive komplettiert.

Um die Komplexität beim Entwurf zu beherrschen, die Kriterien gegenüberzustellen und werkzeuggestützt bewerten zu können, wird die E/E-Architekturmodellierung (Abschnitt 3.2) eingesetzt. Dabei werden Struktur und Schnittstellen ohne Inhalte, Verhaltensmodellierung oder Simulation modelliert. Für diese Modellierung gibt es keine Standards und OEMs entwickeln eigene Lösungen um die Ziele der E/E-Architekturmodellierung (Abschnitt 3.2.1) zu verfolgen. Es hat sich trotzdem etabliert, die Komplexität einer E/E-Architektur durch eine in Ebenen zerlegte Beschreibung zu erfassen (Abschnitt 3.2.2). Nach einer Modellierung im Rahmen dieser Ebenen wird die E/E-Architektur bewertet und abgesichert (Abschnitt 3.2.3). Dabei sind einige Kriterien automatisiert bewertbar aber der Schwerpunkt liegt auf den nicht messbaren Kriterien, wie z. B. technische Umsetzbarkeit oder Wiederverwendbarkeit. Gerade für eine Bewertung dieser ist ein schlagkräftiges Variantenmanagement (Abschnitt 3.2.4) von entscheidender Bedeutung, da mit diesem Varianten der E/E-Architektur erzeugt, verglichen und durch den Architekten

bewertet werden können. Ein CASE-Werkzeug (Abschnitt 3.2.5), PREEvision, ist am Markt verfügbar und unterstützt die Use-Cases.

Ein weiterer Schwerpunkt liegt auf einem weiten Forschungsfeld, der Evolution bei der Softwareentwicklung (Abschnitt 3.3). Für diese Arbeit relevante Methoden sind Aspect weaving, Ontology-based modelling und Feature Oriented Refactoring (FOR), eine zu aspect weaving komplementäre Technologie. Für letztere Methode ist Merkmal (Definition 3.2) eine entscheidende Begrifflichkeit, denn sie wird mit einem Prozess auf bestehende Softwaresysteme angewendet und gewinnt eine merkmalsbasierte Beschreibung zurück, so dass ein Merkmal eine Erweiterung in der Funktionalität der Software beschreibt. Der Methode liegt eine Algebra und eine Unterscheidung von Basis-Modulen (Definition 3.3) und Derivat-Modulen (Definition 3.4) zugrunde.

Beim letzten Schwerpunkt, der Wiederverwendung bei der Produktlinienentwicklung (Abschnitt 3.4) werden Software-Produktlinien (Definition 3.5) näher beschrieben. Zur Erstellung kommen das Domain Engineering (Definition 3.6) und das Application Engineering (Definition 3.7) zum Tragen. Um die Variabilität von einer Software-Produktlinie zu beschreiben (Abschnitt 3.4.2), wird ebenfalls merkmalsorientiert vorgegangen. Wichtige Begriffe hierbei sind Variationspunkt (Definition 3.8) und Variante (Definition 3.9). Abschließend werden Ansätze für Produktlinien in der modellbasierten E/E-Architekturmodellierung (Abschnitt 3.4.3) beleuchtet. Fokus wird dabei einerseits auf den modulatorientierten Produktlinienansatz gelegt, mit dem E/E-Module der Modulstrategie (Abschnitt 2.3.4) im Rahmen eines Modulbaukastens beschrieben und wiederverwendet werden können. Andererseits wird eine kommerzielle Umsetzung von Wiederverwendungsmechanismen mittels `ReuseUnits` diskutiert.

Kapitel 4

Konzeption eines E/E-System Produktlinien Engineerings

Im folgenden Kapitel wird das E/E-System Produktlinien Engineering konzipiert. Das Kapitel ist in vier Abschnitte aufgeteilt (Abbildung 4.1).

Zuerst wird der Stand der Technik (Kapitel 3) hinsichtlich der Evolution, Modellstruktur/Variantenmanagement und Wiederverwendung von E/E-Systemen bewertet (Abschnitt 4.1). Für identifizierte Defizite wird eine Ableitung von allgemeinen Anforderungen an eine zukünftige E/E-Architekturmodellierung durchgeführt. Zudem werden Potentiale für die Einführung von E/E-Systemen in die E/E-Architekturmodellierung aufgezeigt.

Anschließend wird die heutige E/E-Architekturmodellierung hinsichtlich dieser Anforderungen analysiert und weitere Anforderungen abgeleitet (Abschnitt 4.2). Schwerpunkte bilden hier die für diese Arbeit relevanten Analysen zur Funktionsnetzmodellierung und zu den Zusammenhängen bei der Evolution von E/E-Systemen und der Kopplung von E/E-Modulen durch Funktionen.

Im darauffolgenden Abschnitt werden mögliche Lösungsansätze analysiert und Designentscheidungen für das E/E-System Produktlinien Engineering abgeleitet (Abschnitt 4.3).

Abschließend wird das E/E-System Produktlinien Engineering einschließlich der Anwendungsfälle konzipiert (Abschnitt 4.4).

Die Beschreibung einer prototypischen Umsetzung des konzipierten E/E-System Produktlinien Engineering findet sich im nachfolgenden Kapitel 5.

4.1 Bewertung des Standes der Technik und Ableitung von Anforderungen

Die heutige E/E-Architekturmodellierung ist modulatorientiert und erlaubt somit die Erstellung, Evolution und Wiederverwendung von E/E-Modulen (Ab-

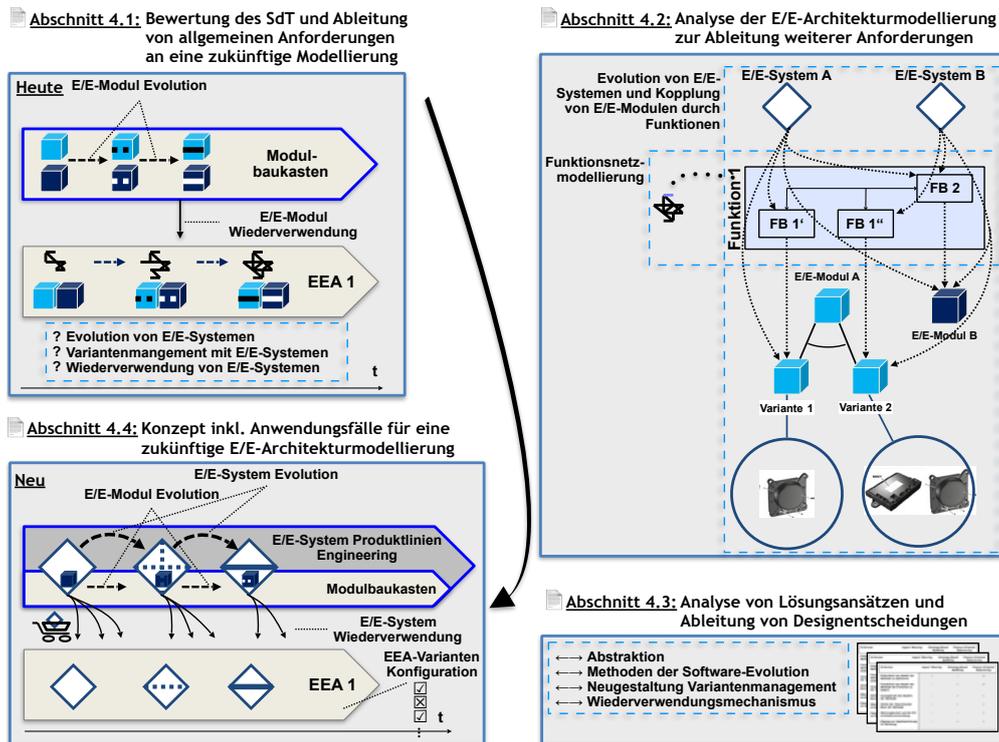


Abbildung 4.1: Überblick Konzeption des E/E-System Produktlinien Engineerings (Kapitel 4). Das Kapitel 4 ist in vier Abschnitte aufgeteilt. EEA E/E-Architektur; SdT Stand der Technik (Kapitel 3); FB Funktionsbeitrag

schnitt 3.4.3, Abbildung 4.2). Mit der Modellierung der E/E-Architektur werden verschiedene Ziele verfolgt (Abschnitt 3.2.1). Für Anwendungsfälle, die einen Fokus auf Hardware bzw. E/E-Modulen haben, sind diese Ziele in der heutigen E/E-Architekturmodellierung erreichbar.

Wenn E/E-Systeme in die E/E-Architekturmodellierung gemäß der Zielsetzung der Arbeit (Zielsetzung 1.2) eingebunden werden sollen, dann muss auch die Software im Zuge einer vollständigen Betrachtung aller Aspekte eines E/E-Systems in die E/E-Architekturmodellierung einbezogen werden. Wird die Software einbezogen, dann lassen sich die Modellierungsziele Abschnitt 3.2.1 mit den heutigen Methoden nicht mehr erfüllen.

Da weitere systemorientierte Ansätze in der E/E-Architekturmodellierung nicht bekannt sind, wird der Stand der Technik nacheinander für Evolution, Modellstruktur und Variantenmanagement und Wiederverwendung bewertet. Für die identifizierten Defizite werden gleichzeitig Anforderungen an das E/E-System Produktlinien Engineering abgeleitet.

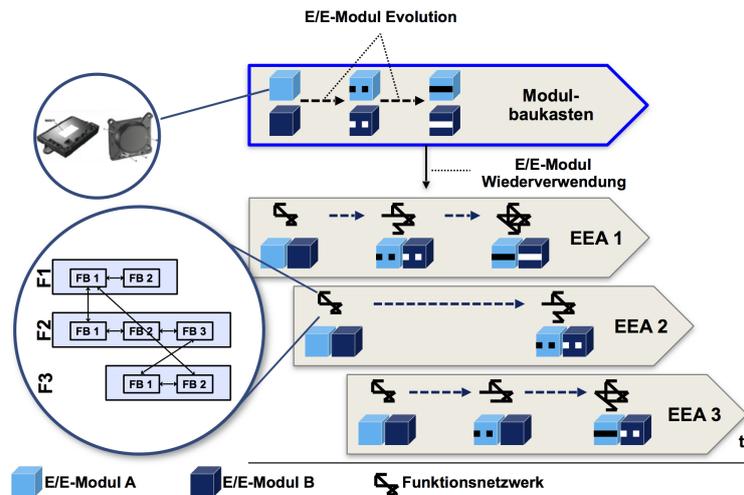


Abbildung 4.2: Heutige modulatorientierte E/E-Architekturmodellierung. Die heutigen E/E-Architekturmodellierung erlaubt die Erstellung und Evolution von E/E-Modulen im Modulbaukasten. Im Modulbaukasten modellierte E/E-Module können in verschiedenen E/E-Architekturen wiederverwendet werden. Das Funktionsnetz wird separat für die einzelnen E/E-Architekturen modelliert und gepflegt. Vgl. [125]. EEA E/E-Architektur; F Funktion; FB Funktionsbeitrag

4.1.1 Evolution von E/E-Systemen

Bei der Evolution eines E/E-Systems müssen sowohl die Hardware- als auch die Software-Anteile betrachtet werden. In der E/E-Architekturmodellierung wird die Hardware in Form von E/E-Modulen im Rahmen des Hardware-Modulbaukastens modelliert, wobei Konzepte für die Evolution der E/E-Module vorgestellt wurden (Abschnitt 3.4.3). Für die Evolution der Software-Anteile müssen in der E/E-Architekturmodellierung die Funktionen des Funktionsnetzes betrachtet werden.

Defizite im Vorgehen um E/E-System-Evolution vollständig zu bewerten Haupttreiber für die Evolution von E/E-Systemen sind Innovationen und die Optimierung von Kosten. Organisiert und geplant wird die Evolution im Rahmen der Dynamisierung in der Modulstrategie (Abschnitt 2.3.4). Bei der Bewertung einer Evolution muss auch die Software betrachtet werden und deshalb die funktionalen Fragestellungen (Abschnitt 1.3) beantwortet werden. In beiden Fällen muss die Evolution aus E/E-Architektursicht ohne Kenntnis der Implementierung bewertet werden, da diese zum Zeitpunkt der E/E-Architekturentwicklung nicht vorliegt (Abschnitt 3.1.1). Auch existiert für beide Fälle keine formalisierte Beschreibung aus E/E-Architektursicht, die z. B. einen einfachen Import aller Details in das E/E-Architekturmodell ermöglicht.

Die Festlegungen werden im Allgemeinen auf der Basis von Grobkonzepten getroffen.

Das heutige Vorgehen der Bewertung für die E/E-Architektur wird, wenn die Software mitbetrachtet wird, nach *Best Engineering Practice* durchgeführt, da aus E/E-Architektursicht

- kein durchgängiger Prozess,
- kein Werkzeug und
- keine Methodik

vorhanden ist um die Evolution zu bewerten. Die Bewertung einer Evolution wird somit ohne Systematik durchgeführt, indem die Schnittstellenbedarfe und die Auswirkungen auf das Funktionsnetz, mit der Hoffnung nichts zu übersehen, grob geschätzt werden. Diese Bewertung ist somit

- nicht effizient,
- nicht abgesichert,
- nicht nachverfolgbar und
- tendenziell unvollständig.

Ähnliche Bewertungen müssen in der Weiterentwicklung von Software getroffen werden, für die sich mehrere Methoden etabliert haben (Abschnitt 3.3). Eine Anwendung dieser Methoden in der E/E-Architekturentwicklung ist nicht bekannt.

Fehlende Nachverfolgbarkeit und fehlende Robustheit bei Änderungen Sollte eine grobe Abschätzung der Schnittstellenbedarfe in der heutigen E/E-Architekturmodellierung modelliert werden, müssen die funktionalen Anteile der Innovationen auf der Ebene des Funktionsnetzes modelliert werden. Damit ist keine klare und nachvollziehbare Trennung zwischen den ursprünglichen Inhalten der E/E-Architektur und den Neuerungen möglich. Es wird eine produktive und detaillierte Schnittstellenbeschreibung mit groben Entwürfen vermischt, deren Unterscheidbarkeit sich lediglich auf Namenskonventionen stützt.

Diese fehlende Trennung führt auch dazu, dass eine Aktualisierung des Funktionsnetz auf einen neueren Entwicklungsstand mit

- einem großen Aufwand,
- vielfältigen Nacharbeiten und

Tabelle 4.1: Bewertung zur Erfüllbarkeit der Ziele der E/E-Architekturmodellierung für die E/E-System-Evolution

Ziel der E/E-Architekturmodellierung	Bewertung für Hardware-Evolution	Bewertung für Software-Evolution
<i>Frühzeitiger Entwurf, Bewertung, Dokumentation, Absicherung der E/E-Architektur und Technologieentscheid</i>	Erfüllbar • Beispiel: Bewertung der Integrierbarkeit einer ECU	Nicht erfüllbar • Beispiel: Systematische Abbildung der Skalierung von Systemen und Funktionen
<i>Erfassen und Bewerten von Auswirkungen bei E/E-Architekturänderungen</i>	Erfüllbar • Beispiel: Wechsel einer ECU zu einem anderem Bussystem	Nicht erfüllbar • Beispiel: Abspalten von Teilumfängen bei der Evolution
<i>Optimierung der Architektur in einer frühen Entwicklungsphase</i>	Erfüllbar • Beispiel: Reduzierung der Leitungslängen durch Verortung der ECUs an einem statt 2 Bussystemen	Nicht erfüllbar • Beispiel: Funktionsverteilung zwischen E/E-Modulen zur Optimierung der Kommunikation

Die Bewertung wird getrennt für die Hardware- und Software-Evolution durchgeführt und ist mit Beispielen veranschaulicht.

- ggf. sogar einer Neumodellierung der Abschätzung der Schnittstellenbedarfe

verbunden ist. Damit ist die heutige E/E-Architekturmodellierung nicht robust gegenüber den häufig auftretenden Änderungen des Funktionsnetzes.

Defizite in der Erfüllung der Modellierungsziele bei der Evolution von E/E-Systemen Für die Ziele der E/E-Architekturmodellierung wird die Erfüllbarkeit bewertet (Tabelle 4.1). Es zeigt sich, dass in der heutigen E/E-Architekturmodellierung diese lediglich für die Evolution von Hardware erfüllbar sind.

Für die Evolution von Software, die im E/E-Architekturmodell im Form des Funktionsnetzes repräsentiert ist, wird keines der E/E-Architekturmodellierungsziele erfüllt. Ein Aspekt sind hierbei auch die vielzähligen versteckten Abhängigkeiten zwischen den Elementen des Funktionsnetzes.

Damit stellt die heutige E/E-Architekturmodellierung keine ausreichende Entscheidungsgrundlage dar, um die Evolution von E/E-Systemen zu bewerten und abzusichern. Zudem ist aus Sicht der heutigen E/E-Architekturmodellierung keine Methode bekannt, die eine Erfüllung der E/E-Architekturmodellierungsziele für die Software-Evolution erlauben würde.

Keine Übertragbarkeit der Hardware-Konzepte auf die E/E-System-Evolution Für die Evolution der Hardware gibt es, z. B. durch die Modulstrategie (Abschnitt 2.3.4), bestehende Prozesse in deren Rahmen eine Bewer-

tung durchgeführt wird. Per Definition sind die E/E-Module der Modulstrategie derart definiert, dass deren enthaltene Hardware überschneidungsfrei ist. Die Hardware ist somit nicht mehreren E/E-Modulen gleichzeitig zugeordnet. Zur Bewertung der Evolution kann

- zuerst eine Modellierung im Hardware-Modulbaukasten (Abschnitt 3.4.3) durchgeführt werden.
- Anschließend kann diese Modellierung bewertet werden.
- Mit dieser Grundlage lassen sich die E/E-Architekturbelange in die Prozesse der Modulstrategie einbringen. Im Rahmen des Modulstrategie-Prozesses wird eine Bewertungsvorlage für E/E-Architekturbelange befüllt, die anschließend ins Modulheft einfließt.

Demgegenüber steht die Evolution der Software, die als verteilte Funktion auf verschiedenen E/E-Modulen gleichzeitig zur Ausführung kommt. Somit wird bei Einbeziehung der Software in den Betrachtungsumfang bei einer Evolution die Überschneidungsfreiheit der E/E-Module aufgelöst. Sie können für die Bewertung einer Software-Evolution somit nicht mehr unabhängig voneinander betrachtet werden.

Insgesamt lassen sich durch die Verteiltheit der Software die Konzepte für die Hardware-Evolution nicht übertragen. Konzepte für eine umfassendere Modulstrategie, die die Software mit einschließt und deren Bewertung ermöglicht, sind in der Automobilindustrie nicht bekannt.

Separate E/E-System-Evolution trotz gemeinsamer E/E-Architektur-Plattform Von einer E/E-Architektur-Plattform werden die E/E-Architekturen mehrerer Baureihen abgeleitet. Die E/E-Architekturmodellierung der Baureihen wird meist in verschiedenen E/E-Architekturmodellen vorgenommen. Ursache dafür sind die unterschiedlichen Topologien für Leitungssatz und Leistungsverteilung.

Die Einführung einzelnen Baureihen, die zu einer E/E-Architektur-Plattform gehören, ist zeitlich versetzt. Zudem erfolgt die Einführung von Innovationen zeitgleich mit einer Baureihe oder zeitlich versetzt.

Obwohl Baureihen einer gemeinsamen E/E-Architektur-Plattform angehören, führt die Baureihen-individuelle E/E-Architecturentwicklung und die unterschiedlichen Einführungszeitpunkte für die Innovationen zu

- einem hohen Modellierungsaufwand und
- einer großen Anzahl an E/E-Architektur-Varianten mit mehreren E/E-Architekturmodellen.

Dies bedeutet für die Evolution des Funktionsnetzes, dass eine separate Modellierung für jedes E/E-Architekturmodell erfolgen muss (Abbildung 4.2) trotz der Zugehörigkeit zu einer E/E-Architektur-Plattform.

Ableitung der Anforderung für die Evolution von E/E-Systemen

In der heutigen E/E-Architekturmodellierung sind keine Methoden zur Bewertung und Umsetzung der Evolution von E/E-Systemen bekannt. Ziel der Arbeit ist die Bewertung und Umsetzung der Evolution von E/E-Systemen (Zielsetzung 1.1) und die Einbindung von E/E-Systemen in die Architekturmodellierung (Zielsetzung 1.2). Damit müssen die Funktionen, als Teil der E/E-Systeme, mit in den Betrachtungsumfang der E/E-Architekturmodellierung aufgenommen werden.

Da die Konzepte zur Bewertung der Hardware aufgrund der Verteiltheit der Funktionen nicht übertragbar sind, ist eine Erweiterung der Methodik für die zukünftige E/E-Architekturmodellierung notwendig.

Ohne eine Erweiterung der Methodik bleibt die Bewertung der Abhängigkeiten, die zwischen den E/E-Modulen durch die verteilten Funktionen entstehen, an Expertenwissen gebunden. Dies führt zu Ineffizienzen und Fehlern. Zusätzlich dazu wird die Evolution von E/E-Systemen aus E/E-Architektursicht nicht abgesichert durchgeführt. Somit gibt es keine Systematik bei steigender Komplexität der E/E-Architekturen und hoher Dynamik in der Änderung der Software (Abschnitt 1.2). Eine erweiterte Methodik für eine zukünftige E/E-Architekturmodellierung soll somit die Evolution von E/E-Systemen ermöglichen und automatisiert umsetzen (Anforderung 4.1). Um die Ziele der E/E-Architekturmodellierung zu realisieren, müssen:

- die Beantwortung der funktionalen Fragestellungen (Abschnitt 1.3) ermöglicht,
- die Nachverfolgbarkeit und die Robustheit verbessert und
- eine Automatisierung die Umsetzung der Evolution effizient erlaubt

werden.

Anforderung 4.1 (Evolution von E/E-Systemen): Die zukünftige E/E-Architekturmodellierung muss die Bewertung und Automatisierung der Evolution von E/E-Systemen erlauben.

4.1.2 Modellstruktur und Variantenmanagement für E/E-Systeme

Die Strukturierung im E/E-Architekturmodell und das Variantenmanagement für ein E/E-System umfasst sowohl dessen Hardware- als auch dessen

Software-Anteile. In der heutigen E/E-Architekturmodellierung wird die Hardware in Form von E/E-Modulen im Rahmen des Hardware-Modulbaukastens modelliert (Abschnitt 3.4.3). Somit sind Konzepte zur Strukturierung und zum Variantenmanagement für die Hardware verfügbar. Für Modellstruktur und Variantenmanagement der Software-Anteile müssen in der E/E-Architekturmodellierung die Funktionen des Funktionsnetz und deren Struktur betrachtet werden.

Defizite bei der vollständigen Erfassung eines E/E-Systems und der Beziehungen zwischen E/E-Systemen In der heutigen E/E-Architekturmodellierung ist kein Vorgehen bekannt, um alle Aspekte eines E/E-Systems vollständig zu erfassen. Obwohl die Funktionen des Funktionsnetzes und die Hardware der E/E-Module bereits modelliert werden, lässt sich heute nicht erschließen, welche Anteile welcher Funktionen und welche Hardware welcher E/E-Module zu welchem E/E-System zugehörig sind.

Somit lässt sich diese Beziehung heute ausschließlich außerhalb der E/E-Architekturmodellierung durch Expertenwissen aufklären. Darüber hinaus lassen sich auch die Beziehungen, die zwischen den E/E-Systemen existieren, nicht der E/E-Architekturmodellierung entnehmen.

Defizite in der Erfassung der funktionalen Kopplung der E/E-Module Die E/E-Module erfahren u. a. durch die Verteilung der Funktionen auf mehrere E/E-Module eine funktionale Kopplung. Diese beschränkt zum einen die Weiterentwicklung der E/E-Module, die wegen der funktionalen Kopplung nicht unabhängig durchgeführt werden kann. Zum anderen wird die Verblockung der E/E-Module beschränkt, da wegen der funktionalen Kopplung weitere Verblockungen anderer E/E-Module notwendig werden können.

In der heutigen E/E-Architekturmodellierung sind zwar die verteilten Funktionen und E/E-Module modelliert, aber eine Aussage zur Kopplung zwischen E/E-Modulen lässt sich nicht klar ableiten und muss durch Expertenwissen aufgefangen werden. Ferner sind keine Methoden in der E/E-Architekturmodellierung bekannt, die die Analyse der funktionalen Kopplung von E/E-Modulen erlauben.

Kein Variantenmanagement für E/E-Systeme In der heutigen E/E-Architekturmodellierung wird die Konfiguration von E/E-Architekturvarianten über die manuelle Zuordnung von Funktionsbeiträgen und anschließender automatischer Propagation der Hardware vorgenommen (Abschnitt 3.2.4). Ferner wurden Konzepte vorgestellt, wie die Funktionen des Funktionsnetzes und die E/E-Module zur Konfiguration von E/E-Architekturvarianten genutzt werden können (Abschnitt 3.4.3).

Beide Vorgehensweisen sind sehr aufwendig und benötigen das Involvieren eines Experten, der die Zuordnung der Funktionsbeiträge bzw. Funktionen vornimmt und absichert. Zusätzlich dazu ist der Hardware-Modulbaukasten heute so modelliert, dass jede einzelne E/E-Modul-Variante vollständig modelliert wird und somit Hardware, die von mehreren E/E-Modul-Varianten geteilt wird, mehrfach im E/E-Architekturmodell enthalten ist. Diese Mehrfachmodellierung von einmaliger und eindeutig identifizierter Hardware

- vergrößert das E/E-Architekturmodell unnötig,
- erzeugt Unübersichtlichkeit im E/E-Architekturmodell,
- intensiviert den Arbeitsaufwand signifikant bei der Darstellung in Diagrammen und
- verhindert die Ausführung des Leitungssatzrouters, dessen Einsatz in der Leitungssatzmodellierung essentiell ist.

Insgesamt sind somit für die heutige E/E-Architekturmodellierung keine Konzepte und Vorgehensweisen zur Konfiguration und Absicherung von E/E-Architektur-Varianten mit E/E-Systemen vorhanden.

Ableitung der Anforderung für die Modellstruktur und das Variantenmanagement von E/E-Systemen

In der heutigen E/E-Architekturmodellierung werden E/E-Systeme und damit die Kopplung zwischen den E/E-Modulen nicht vollständig erfasst. Gleichzeitig werden E/E-Systeme nicht für das Variantenmanagement verwendet. Zudem gibt es in der heutigen E/E-Architekturmodellierung keine Methodik, die sich anwenden lässt um die beschriebenen Aspekte umzusetzen. Dies macht eine Erweiterung der Modellierungsmethodik für Strukturierung und Management von E/E-Systemen erforderlich.

Ohne eine Erweiterung der Methodik bleibt die E/E-Architekturmodellierung für die Software-Umfänge ineffizient und erlaubt keine Aussagen zur Kopplung zwischen den E/E-Modulen, die insbesondere für die Bewertung von Verblockungsszenarien erforderlich sind.

Eine zukünftige E/E-Architekturmodellierung muss somit eine Erfassung aller Aspekte eines E/E-Systems erlauben (Anforderung 4.2). Dies schließt somit die Funktionen des Funktionsnetzes, die E/E-Module als auch die intransparenten Beziehungen ein. Darüber hinaus muss in der zukünftigen Methodik das Variantenmanagement um E/E-Systeme erweitert werden (Anforderung 4.3).

Anforderung 4.2 (E/E-Systeme in der E/E-Architekturmodellierung): Die Modellstruktur des E/E-Architekturmodells muss um die E/E-System-Sicht erweitert werden.

Anforderung 4.3 (Variantenmanagement mit E/E-Systemen): Das E/E-Architektur-Variantenmanagement muss mit E/E-Systemen konfiguriert und abgesichert werden, so dass E/E-Systeme in E/E-Architekturvarianten entweder enthalten oder nicht enthalten sind.

4.1.3 Wiederverwendung von E/E-Systemen

Für die Wiederverwendung von E/E-Systemen müssen sowohl die Hardware- als auch die Software-Anteile des E/E-Systems berücksichtigt werden. Die Hardware-Anteile werden durch die E/E-Module repräsentiert. Die Software wird in der E/E-Architekturmodellierung in Form des Funktionsnetzes repräsentiert, welches dementsprechend für die Wiederverwendung mit in den Betrachtungsumfang aufgenommen werden muss.

Kein Vorgehen zur Wiederverwendung von E/E-Systemen Der Haupttreiber für die Wiederverwendung von E/E-Systemen in der E/E-Architekturmodellierung ist die Modulstrategie (Abschnitt 2.3.4), die die Verblockung der E/E-Module in die einzelnen Baureihen beschreibt. Diese Verblockung muss im Rahmen der E/E-Architecturentwicklung bewertet und bei der Auslegung der E/E-Architektur berücksichtigt werden. Um die Bewertung durchzuführen, muss die Verblockung auch in der E/E-Architekturmodellierung berücksichtigt werden.

Zur Wiederverwendung der Hardware in Form von E/E-Modulen in der E/E-Architekturmodellierung wurden bereits Konzepte vorgestellt mit denen sich die Verblockung aus Hardware-Sicht bewerten lässt (Abschnitt 3.4.3). Gleiches gilt für das Austauschen eines E/E-Moduls bei einer Änderung. Allerdings unterliegen die E/E-Module einer funktionalen Kopplung durch die Software-Anteile der E/E-Systeme. Diese muss bei der Bewertung der Verblockung berücksichtigt werden. In der heutigen E/E-Architekturmodellierung werden deshalb die Software-Anteile in Form des Funktionsnetzes separat für jedes E/E-Architekturmodell erstellt.

Für die heutige E/E-Architekturmodellierung mit PREEvision wurde kürzlich eine produktive Werkzeugfunktionalität des Werkzeugherstellers vorgestellt, die auch eine Wiederverwendung von Software-Anteilen und deren Austausch bei einer Änderung erlauben soll (Abschnitt 3.4.3). Eine Bewertung dieses Wiederverwendungsmechanismus (Tabelle 4.2) zeigt, dass im Allgemeinen eine Verwendung für die Wiederverwendung von Hardware und Software mit Einschränkungen möglich ist. Wie der Wiederverwendungsmechanismus jedoch eingesetzt werden kann, um E/E-Systeme wiederzuverwenden, wurde noch nicht gezeigt und bedarf eines Konzepts. Solch ein Konzept wird es erfordern auch Teile von Funktionen (ein oder mehrere Funktionsbeiträge) wiederzuverwenden.

Tabelle 4.2: Bewertung des Wiederverwendungsmechanismus zur Wiederverwendung von Hardware und Software

Vorteile	Nachteile
<ul style="list-style-type: none"> • Regelmäßig gewartetes Produkt-Feature • Verfolgbarkeit der Wiederverwendungen in den unterschiedlichen E/E-Architekturen eines E/E-Architekturmodells • Strukturiertes Versions- und Variantenhandling 	<ul style="list-style-type: none"> • Einsatz erfordert Datenbankbetrieb • Für nicht-versionierbare Artefakte eingeschränkte Benutzung • Wiederverwendung der Software beschränkt auf gesamte Funktionen einschließlich aller Funktionsbeiträge

Nicht-versionierbare Artefakte sind E/E-Architekturmodellobjekte, für die in einem E/E-Architekturmodell nicht unterschiedliche Versionen angelegt werden können. Eine Funktion kann lediglich vollständig wiederverwendet werden, weil der Wiederverwendungsmechanismus es nur erlaubt strukturierende SWComp wiederzuverwenden.

Keine Wiederverwendung durch gemeinsame E/E-Architektur-Plattform Mehrere Baureihen können zur gleichen E/E-Architektur-Plattform gehören und können aufgrund verschiedener Topologien für Leitungssatz und Leistungsverteilung in verschiedenen E/E-Architekturmodellen modelliert sein. Diese notwendige Baureihen-individuelle E/E-Architekturentwicklung führt zu einem hohen Modellierungsaufwand in verschiedenen E/E-Architekturmodellen mit vielen E/E-Architekturvarianten. Somit muss trotz der Zugehörigkeit zur selben E/E-Architektur-Plattform die Wiederverwendung eines E/E-Systems separat für die entsprechenden E/E-Architekturmodelle modelliert werden.

Ableitung der Anforderung für die Wiederverwendung von E/E-Systemen

In der heutigen E/E-Architekturmodellierung gibt es kein Vorgehen zur Wiederverwendung von E/E-Systemen, was heute zu hohen Aufwänden in der E/E-Architekturmodellierung führt. Schwierig gestaltet sich insbesondere die Wiederverwendung der Funktionen des Funktionsnetzes und die Bewertung von Verblockungsszenarien. Somit ist eine erweiterte Methodik für eine zukünftige E/E-Architekturmodellierung die vollständige Wiederverwendung eines E/E-Systems mit allen seinen Aspekten, d. h. sowohl der Funktionen als auch der E/E-Module, notwendig (Anforderung 4.4).

Anforderung 4.4 (Wiederverwendung für E/E-Systeme): Die zukünftige E/E-Architekturmodellierung muss die Wiederverwendung von E/E-Systemen in verschiedenen E/E-Architekturen mit geringem Modellierungsaufwand erlauben.

4.1.4 Potentiale bei der Einführung von E/E-Systemen in die E/E-Architekturmodellierung

Durch die Einführung von E/E-Systemen in die E/E-Architekturmodellierung können verschiedene Potentiale für die E/E-Architekturmodellierung erschlossen werden. Nachfolgend werden diese kurz beschrieben.

Hinsichtlich der Evolution sind

- die Bewertbarkeit einer E/E-System-Evolution durch eine E/E-Architekturmodellierung,
- die Unterstützung bei der Beherrschung der Komplexität von einer hohen Anzahl von Funktionen und
- die dadurch vorhandene Basis für Design-Entscheidungen

mögliche Potentiale.

Für die Modellstruktur und das Variantenmanagement der E/E-Architekturmodellierung liegen Potentiale in

- einer Methodik für die transparente Beschreibung und vollständige Erfassung der Beziehungen zwischen E/E-Systemen, Funktionen, Funktionsbeiträgen, SWCs und E/E-Modulen,
- einer darauf aufbauenden systematischen Vorgehensweise, die den E/E-Architekten von derzeit manuell verwalteten Themen entlastet und
- zur Vereinfachungen bei der Modellierung von E/E-Systemen und Innovationen führt.

Bei der Wiederverwendung in der E/E-Architekturmodellierung liegen die Potentiale in

- einer Methodik zur Modellierung einer Wiederverwendung von E/E-Systemen gemäß des Rollouts der E/E-Systeme in die entsprechenden Baureihen-E/E-Architekturen.

Damit würden insbesondere die konzeptionellen Untersuchungen und Bewertungen in der E/E-Architekturentwicklung leistungsfähig.

4.2 Analyse der E/E-Architekturmodellierung und Ableitung weiterer Anforderungen

Nach der Ableitung von allgemeinen Anforderungen an eine zukünftige E/E-Architekturmodellierung wird nachfolgend die heutige E/E-Architekturmodellierung hinsichtlich der Ziele der Arbeit (Zielsetzung 1.1 und

Zielsetzung 1.2) genauer analysiert und weitere Anforderungen an ein Konzept abgeleitet (vgl. Abbildung 4.1).

Dieser Abschnitt beschäftigt sich hierbei lediglich mit der E/E-Architekturmodellierung an sich. Damit werden etwa Randbedingungen ausgespart, die durch den eigentlichen Modellierungsprozess entstehen. Dieser ist

- ein Expertenprozess
- ohne allgemeine Vorgaben für Modellierer oder E/E-Architekten und
- erfordert auch vom E/E-Architekten über sein E/E-Architekturwissen hinaus noch Kenntnis der E/E-Architekturmodellierung, was für den E/E-Architekten aufwendig ist.

4.2.1 Analyse der Funktionsnetzmodellierung

Da die funktionale Sicht mit in den Betrachtungsumfang der E/E-Architekturmodellierung aufgenommen werden soll (Zielsetzung 1.2), wird im nachfolgenden Abschnitt die Modellierung des Funktionsnetzes genauer analysiert und Anforderungen für die zukünftige E/E-Architekturmodellierung abgeleitet.

Heterogene Modellierung des Funktionsnetzes

Das Funktionsnetz und dessen Modellierung ist heterogen

- hinsichtlich der Tiefe, in der Funktionen beschrieben werden als auch,
- hinsichtlich der Art wie Funktionen entwickelt und geschnitten werden.

Diese Heterogenität muss beim Einführen neuer Methoden einer zukünftigen E/E-Architekturmodellierung beachtet werden.

In der Praxis kann die Beschreibung der Funktionen stark variieren und ist ggf. sogar abhängig von der Fahrzeugdomäne. In der E/E-Architecturentwicklung von Mercedes-Benz Cars z. B. kann die Beschreibung für die Fahrzeugdomäne Body&Comfort als gut detailliert bewertet werden. Dies liegt u. a. daran, dass etwa 50% der Funktionen modellbasiert entwickelt und dokumentiert werden. Zusätzlich dazu werden auch alle übrigen Funktionsschnittstellen dieser Fahrzeugdomäne dokumentiert. Allerdings werden keine Abhängigkeiten zwischen diesen Schnittstellen erfasst. Um diese zu erschließen, müssen die Funktionsmodelle oder Lastenhefte herangezogen werden. Einen Gegensatz dazu bildet die Fahrzeugdomäne Infotainment, in der die E/E-Komponenten und Funktionen oft als Black-Box vom Zulieferer stammen.

In der Praxis haben zusätzlich dazu die Fachbereiche in den Fahrzeugdomänen oft unterschiedliche Paradigmen, wie Funktionen entwickelt und geschnitten werden. Dies wird zum einen durch das Vorliegen von föderalen Entwicklungsstrukturen in der Entwicklungsorganisation begünstigt. Zum anderen hat dies fachliche Gründe. Zum Beispiel wird in der Fahrzeugdomäne Body&Comfort eine Funktion als Pfad von Ursache (z. B. Betätigen des Schalters für das Schiebedach) bis Wirkung (z. B. Schiebedach öffnet) beschrieben. In der Fahrzeugdomäne Powertrain sind z. B. die Systeme meist anhand des physikalischen Zusammenbaus und -wirkens strukturiert (z. B. Kühl-System, Öl-System, Verbrennungs-System).

Dementsprechend erfordert eine zukünftige E/E-Architekturmodellierung eine möglichst flexible Beschreibungsmöglichkeit um die funktionalen Anteile den E/E-Systemen zuzuordnen (Anforderung 4.5).

Anforderung 4.5 (Heterogenität des Funktionsnetzes): Die zukünftige E/E-Architekturmodellierung muss eine heterogene E/E-Architekturmodellierung des Funktionsnetzes unterstützen.

Heterogene Modellierung von Skalierung

Bei der E/E-Architekturmodellierung des Funktionsnetzes werden unterschiedliche Vorgehensweisen angewendet, um die Skalierung von Funktionen zu modellieren. Diese müssen bei dem Entwurf einer neuen Methodik beachtet werden.

Es zeigt sich, dass die Skalierung von Funktionen in der E/E-Architekturmodellierung

- systemvariant,
- systemisch,
- Delta und
- funktional

modelliert werden kann (Abbildung 4.3). Dabei entspricht eine Skalierungsstufe dem Funktionsumfang eines E/E-Systems.

Bei dem systemvarianten Vorgehen wird für jedes E/E-System eine Funktion im Funktionsnetz modelliert, d. h. wenn sich E/E-Systeme gegenseitig ausschließen, dann schließen sich auch die Funktionen im Funktionsnetz aus. Formal lässt sich die Menge der Schnittstellen INT_{SV_i} der Systemvariante SV_i beschreiben als

$$\begin{aligned} INT_{SV_i} &= Int(F_i), \\ \text{mit } F_i &= \{FB_1, \dots, FB_z\} \end{aligned} \tag{4.1}$$

wobei $Int(F_i)$ die Interfaces für die Funktion F_i und FB_1, \dots, FB_z die Funktionsbeiträge der Funktion F_i seien.

Bei dem systemischen Vorgehen wird für das umfangreichste E/E-System genau eine Funktion im Funktionsnetz modelliert. Für E/E-Systeme mit einem geringeren Schnittstellenumfang wird in der Funktion der Funktionsbeitrag wiederverwendet und nicht benötigte Schnittstellen ignoriert. Eine Systemvariante SV_i lässt sich somit formal beschreiben mit

$$\begin{aligned} INT_{SV_i} &= Int(FB_i) \\ &= Int(FB_z) \setminus \{int_i, \dots, int_h\} \setminus \dots \setminus \{int_a, \dots, int_1\} \end{aligned} \quad (4.2)$$

wobei INT_{SV_i} die Menge der Schnittstellen der Systemvariante SV_i , $Int(FB_z)$ die Schnittstellen für den umfangreichsten Funktionsbeitrag FB_z und $\{int_i, \dots, int_h\}$ die Menge der Schnittstellen, die für die Systemvariante SV_i zusätzlich zu denen der geringeren Systemvariante SV_h ignoriert werden sollen, seien.

Auch für das Delta-Vorgehen werden skalierende E/E-Systeme in genau einer Funktion im Funktionsnetz modelliert. Dabei wird für das Delta an Schnittstellenumfang, das eine neue Skalierungsstufe ausmacht, ein zusätzlicher Funktionsbeitrag mit gerade diesem Delta-Schnittstellenumfang definiert. Allerdings werden auch die Schnittstellen der unteren Skalierungsstufen benötigt, weshalb die entsprechenden Funktionsbeiträge erforderlich sind. Formal kann die Menge der Schnittstellen INT_{SV_i} der Systemvariante SV_i mit

$$\begin{aligned} INT_{SV_i} &= Int(FB_{1,\dots,i}), \\ \text{mit } FB_{1,\dots,i} &= \{FB_1, \dots, FB_i\} \end{aligned} \quad (4.3)$$

beschrieben werden, wobei $FB_{1,\dots,i}$ die Funktionsbeiträge für die Skalierungsstufe i und aller unteren Skalierungsstufen seien und $Int(FB_{1,\dots,i})$ die Schnittstellen dieser Funktionsbeiträge liefert.

In der funktionalen Vorgehensweise werden hingegen für die einzelnen Funktionen der E/E-Systeme jeweils Funktionen im Funktionsnetz modelliert. Für jedes E/E-System und jede Funktion wird ein Funktionsbeitrag mit dem entsprechenden Schnittstellenumfang angelegt. Dabei wird der Funktionsbeitrag mit dem größten Schnittstellenumfang wiederverwendet und nicht benötigte Schnittstellen ignoriert. Dies lässt sich vergleichbar zu Gleichung 4.2 formal beschreiben als

$$\begin{aligned} INT_{SV_i} &= INT_{F_{j_i}} \cup \dots \cup INT_{F_{j_z}}, \\ \text{mit } INT_{F_{j_i}} &= Int(FB_{j_z}) \setminus \{int_{j_i}, \dots, int_{j_h}\} \setminus \dots \setminus \{int_{j_a}, \dots, int_{j_1}\} \end{aligned} \quad (4.4)$$

wobei INT_{SV_i} die Menge der Schnittstellen der Systemvariante SV_i , $INT_{F_{j_i}}$ die Menge der Schnittstellen der Funktionsvariante F_{j_i} , $Int(FB_{j_z})$ die Schnittstellen für den umfangreichsten Funktionsbeitrag FB_{j_z} und $\{int_{j_i}, \dots, int_{j_h}\}$ die

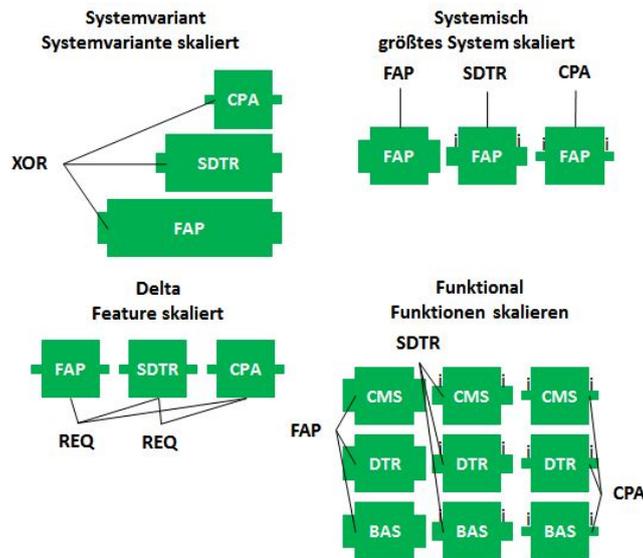


Abbildung 4.3: Gegenüberstellung der Vorgehen zur Modellierung der Skalierung von Funktionen. Im Allgemeinen werden vier Vorgehen verwendet um die Skalierung von Funktionen auf dem Funktionsnetz zu modellieren. Beispielhaft werden die Radar-Fahrerassistenzsysteme aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Die E/E-Systeme sind hierbei FAP (Fahrerassistenzpaket), SDTR (Solo-Distronic) und CPA (*Collision Prevention Assist*, engl., Kollisionsvermeidungs-Assistent). An den E/E-Systemen sind mehrere Funktionen beteiligt. Hier dargestellt sind CMS (COLLISIONMITIGATIONSYSTEM), BAS (BRAKEASSIST+Q) und DTR (DISTRONIC). Ein Funktionsbeitrag der Funktion wird dabei als Quadrat dargestellt, wobei die Dicke der seitlich herausragenden Balken den Schnittstellenumfang veranschaulichen soll, so hat z. B. FAP einen größeren Schnittstellenumfang als CPA. Sind mehrere Funktionsbeiträge jeweils in der gleichen Zeile angeordnet, so sind sie der gleichen Funktion zugeordnet. Sind Funktionsbeiträge in mehreren Zeilen angeordnet, so sind sie verschiedenen Funktionen zugeordnet. i IGNORED-PORTS (für den Funktionsbeitrag modellierte Ports, die aber ausgeschaltet sind)

Tabelle 4.3: Vergleich und Bewertung der Vorgehensweisen zur Modellierung von Skalierung

<i>Systemvariant</i> Systemvariante = Komponentenbeitrag		<i>Systemisch</i> Systemvariante = Wiederverwendung von Komponentenbeiträgen mit Ignored-Ports	
<i>Vorteile</i>	<i>Nachteile</i>	<i>Vorteile</i>	<i>Nachteile</i>
<ul style="list-style-type: none"> • Klares Erschließen der Systemvarianten • Einfaches Handling beim ZL 	<ul style="list-style-type: none"> • Intransparente Gemeinsamkeiten in der Implementierung • Keine Funktionale Sicht • Jede Systemvariante muss angelegt werden (Kombinatorik) • Hoher Aufwand in K-Matrix • Risiko von Inkonsistenzen bei Schnittstellenänderungen 	<ul style="list-style-type: none"> • Einfaches Handling beim ZL 	<ul style="list-style-type: none"> • Intransparente Gemeinsamkeiten in der Implementierung • Keine funktionale Sicht • Jede Systemvariante muss angelegt werden (Kombinatorik) • Keine Feature Sicht
<i>Delta</i> Systemvariante = $\Sigma \Delta$ -Komponentenbeitrag des Features		<i>Funktional</i> Systemvariante = Σ Wiederverwendung von Komponentenbeiträgen mit Ignored-Ports	
<i>Vorteile</i>	<i>Nachteile</i>	<i>Vorteile</i>	<i>Nachteile</i>
<ul style="list-style-type: none"> • Erzeugen von Systemvarianten ist einfach 	<ul style="list-style-type: none"> • Intransparente Gemeinsamkeiten in der Implementierung • Keine Funktionale Sicht • Erschwertes Handling beim ZL wenn SWA=Systemschnitt 	<ul style="list-style-type: none"> • Detailliertere Information da Zuordnung der Schnittstellen zu den Funktionen • Wiederverwendung von Schnittstellen verringert Risiko von Inkonsistenzen 	<ul style="list-style-type: none"> • Intransparente Gemeinsamkeiten in der Implementierung • Erschwertes Handling beim ZL • Erschwertes Handling beim ZL wenn SWA=Systemschnitt

Die Bewertung wurde aus E/E-Architektursicht vorgenommen und zeigt die Vor- und Nachteile der verschiedenen Vorgehensweisen auf. ZL Zulieferer; K-Matrix Kommunikations-Matrix; Δ Delta an Schnittstellenumfang; SWA Software-Architektur

Menge der Schnittstellen, die für die Funktionsvariante F_{j_i} zusätzlich zu denen der geringeren Funktionsvariante F_{j_h} ignoriert werden sollen, seien.

Insgesamt lässt die Modellierung abhängig vom eingesetzten Vorgehen verschiedene Rückschlüsse auf die Funktionen der E/E-Systeme zu und erzeugt sowohl in der Modellierung als auch bei der Entwicklung beim Zulieferer unterschiedliche Aufwände (Tabelle 4.3). Allerdings können mit keiner dieser Vorgehensweisen die Gemeinsamkeiten in der Implementierung transparent gemacht werden, die eine Ursache für die funktionale Kopplung von E/E-Modulen darstellt (Abschnitt 4.2.2).

In der Praxis werden die unterschiedlichen Vorgehensweisen gemischt eingesetzt (Abbildung 4.4). Somit erfordert eine zukünftige E/E-Architekturmodellierung eine flexible Beschreibungsmöglichkeit um diese Heterogenität abbilden zu können (Anforderung 4.5).

Anforderung 4.6 (Heterogenität der Skalierung): Die zukünftige E/E-Architekturmodellierung muss die heterogene Modellierung der Skalierung von Funktionen unterstützen.

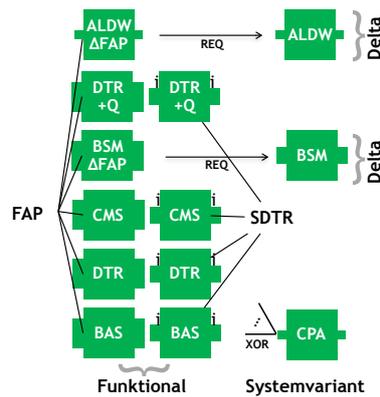


Abbildung 4.4: Heterogene Modellierung der Skalierung am Beispiel der Radar-Fahrerassistenzsysteme. In der heutigen E/E-Architekturmodellierung werden die verschiedenen Vorgehensweisen in der Praxis gemischt verwendet, wie für die Radar-Fahrerassistenzsysteme aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) veranschaulicht. Die E/E-Systeme sind hierbei FAP (Fahrerassistenzpaket), SDTR (Solo-Distronic), CPA (*Collision Prevention Assist*, engl., Kollisionsvermeidungs-Assistent), BSM (*Blind Spot Monitoring*, engl., Totwinkel-Assistent), ALDW (*Active Lane Departure Warning*, engl. Spurwechsel-Warnung). An den E/E-Systemen sind mehrere Funktionen beteiligt. Hier dargestellt sind CMS (COLLISION MITIGATION SYSTEM), BAS (BRAKE ASSIST+Q) und DTR (DISTRONIC), DTR+Q (DISTRONIC+Q), BSM und ALDW. Ein Funktionsbeitrag der Funktion wird dabei als Quadrat dargestellt, wobei die Dicke der seitlich herausragenden Balken den Schnittstellenumfang veranschaulichen soll. Δ Delta an Schnittstellenumfang; i IGNORED-PORTS (für den Funktionsbeitrag modellierte Ports, die aber ausgeschaltet sind)

Konkretisierung neuer Funktionalitäten

Neue Funktionalitäten, wie z. B. neue innovative Funktionen, werden in der Konzeptphase der E/E-Architekturentwicklung in die E/E-Architektur transferiert und integriert. Diese Konkretisierung wird in zwei aufeinander folgenden Phasen durchgeführt.

In der ersten Phase werden neue Funktionalitäten aus E/E-Architektursicht bewertet. Dabei werden

- die Verteilung der Funktionen,
- die erforderte Kommunikation auf den Bussystemen (einschließlich Timing),
- die Auswirkung auf die E/E-Architektur und
- weitere technische Aspekte

untersucht. Gleichzeitig werden auch organisatorische Aspekte betrachtet. Dazu gehört z. B. die Verteilung involvierter Personen bzw. Fachbereiche in der Entwicklungsorganisation und ihren Verantwortlichkeiten bei der Entwicklung der Funktionalität. Weitere Aspekte in dieser Phase sind die Bereitstellung von Aufwandsabschätzungen und das Treiben von Managemententscheidungen.

In der darauf folgenden zweiten Phase der Konkretisierung werden die involvierten Fachbereiche bei der Ausarbeitung und Änderung von Schnittstellen und Funktionen unterstützt. Die resultierenden Änderungen oder neue Anforderungen können dann in Vernetzungskonzepte der E/E-Architekturentwicklung und der Modulstrategie eingebracht werden.

Die beschriebene Konkretisierung wird vermehrt bei der Entwicklung einer E/E-Architektur-Plattform durchgeführt. Allerdings wird sie auch später bei der Entwicklung der Lead-Baureihe und ihrer Derivate eingesetzt. Ursache dafür ist der fortlaufende Einfluss von Innovationen und Änderungen durch die Modulstrategie. Die Innovationsentwicklung von wettbewerbsdifferenzierenden Funktionen und die Modulstrategie finden parallel zur E/E-Architekturentwicklung und damit parallel zur E/E-Architekturmodellierung statt (Abbildung 4.5). Die durch die Änderungen entstehenden Konflikte für die Integration müssen durch die E/E-Architekturentwicklung und die E/E-Architekturmodellierung aufgelöst werden. Da die E/E-Architektur-Plattform-Entwicklung bereits abgeschlossen ist, gestaltet sich die Auflösung dieser Konflikte ggf. als aufwändig und kostenintensiv. Dafür kann z. B. auf

- neue Prozessoren,
- Software-Anpassungen,

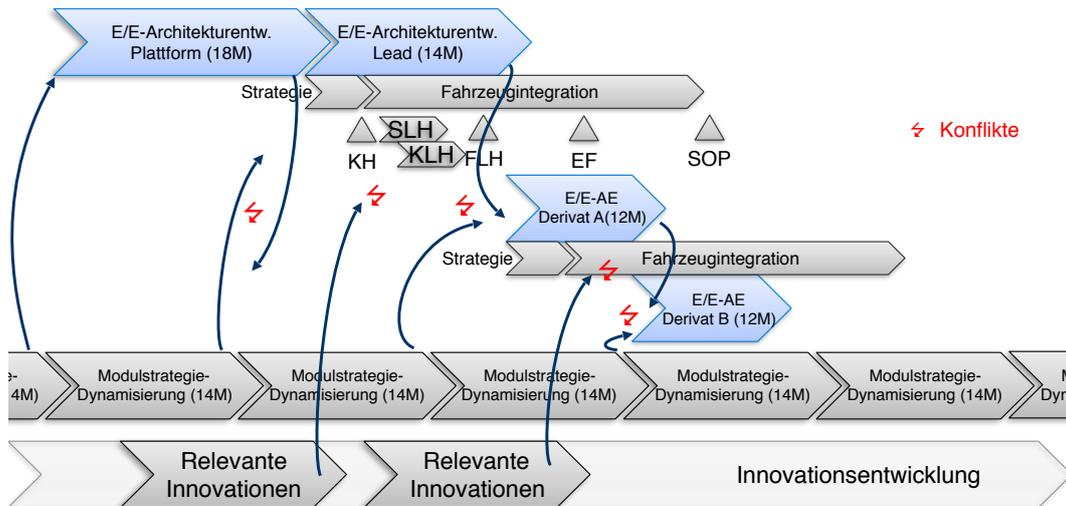


Abbildung 4.5: Fortlaufender Einfluss von Innovationen und Modulstrategie. Die Innovationsentwicklung und Modulstrategie laufen parallel zur E/E-Architekturentwicklung und E/E-Architekturmodellierung. Die E/E-Architekturentwicklung ist vor der Erstellung der System- und Komponentenlastenhefte abgeschlossen, die allerdings Innovationen beschreiben. Dadurch Entstehen Konflikte für die Integration in die E/E-Architektur. M Monate; KH Konzeptheft; SLH Systemlastenheft; KLH Komponentenlastenheft; FLH Fahrzeuglastenheft; EF Elektronik-Fahrzeug; SOP Start of Production; AE Architekturentwicklung

- neue Steuergeräte oder
- Schnittstellenänderungen

zurückgegriffen werden. Letztere sind allerdings weitaus am aufwändigsten.

Für eine zukünftige E/E-Architekturmodellierung bedeutet dies, dass eine Vorgehensweise für den Umgang mit dem fortlaufenden Einfluss von Innovationen und Modulstrategie gefunden werden muss. Da nicht alle Änderungen vorausgesehen werden können, soll dementsprechend eine zukünftige E/E-Architekturmodellierung die Konkretisierung unterstützen (Anforderung 4.7). Dazu gehört auch der Aspekt der Aktualisierung des Funktionsnetzes damit eine Konkretisierung auf einem aktuellen Datenstand durchgeführt werden kann. In der heutigen E/E-Architekturmodellierung können die hinzugefügten Innovationen allerdings nur mit hohem Aufwand abgesichert werden, weshalb sie oft neu modelliert werden müssen. Ursache ist die Neuvergabe aller Modellartefakt-IDs¹ während der Aktualisierung, sodass die gesicherten Umfänge aufwändig eingepflegt werden müssen.

¹Im E/E-Architekturmodell wird jedem Modellartefakt eine ID zugewiesen über die es eineindeutig identifizierbar ist.

Anforderung 4.7 (Konkretisierung): Die zukünftige E/E-Architekturmodellierung muss die Konkretisierung neuer Funktionalitäten unterstützen.

4.2.2 Analyse der Evolution von E/E-Systemen

Bei der Evolution von E/E-Systemen müssen neben den Anforderungen aus der Analyse der Funktionsnetzmodellierung (Abschnitt 4.2.1) noch die Charakteristik des Funktionsnetzes und die funktionale Kopplung von E/E-Modulen beachtet werden. Im nachfolgenden Abschnitt wird deshalb die Evolution von E/E-Systemen näher analysiert.

Charakteristik des Funktionsnetzes

Das Funktionsnetz ist eng verflochten. Es gibt somit viele Funktionen mit vielen Schnittstellen zu anderen Funktionen. Wird die Evolution eines einzigen E/E-Moduls untersucht, müssen gleichzeitig eine große Anzahl an Funktionen mit vielen Schnittstellen untereinander betrachtet werden.

Eine weitere Charakteristik des Funktionsnetzes sind nicht streng hierarchisch aufeinander aufbauende E/E-Systeme. Somit verwenden E/E-Systeme mit einem hohen Funktionsumfang nicht alle Anteile der Hard- und Software der E/E-Systeme mit einem kleineren Funktionsumfang.

Beispiel (vgl. Abschnitt 6.1): Werden die E/E-Systeme CPA, SDTR und FAP verglichen, so können 6 Funktionsbeiträge identifiziert werden, die sowohl von SDTR als auch von FAP verwendet werden. Von dem E/E-System CPA werden keine Funktionsbeiträge von den anderen E/E-Systemen verwendet obwohl die E/E-Systeme CPA, SDTR und FAP einen aufeinander aufbauenden Funktionsumfang haben.

Zusätzlich dazu herrscht keine Überschneidungsfreiheit zwischen den E/E-Systemen. Dementsprechend kann ein Funktionsbeitrag oder eine Hardware zu mehreren E/E-Systemen gehören.

Beispiel (vgl. Abschnitt 6.1): Der Funktionsbeitrag `BLINDSPOTMONITORING_DISPSET_V1` wird sowohl vom E/E-System BSM (Blind Spot Monitoring) als auch vom E/E-System FAP verwendet.

Weiterhin sind die einzelnen Funktionalitäten der Funktionen nicht klar ersichtlich. Welche Funktionalitäten z. B. einzelne Skalierungsstufen einer Funktion unterscheiden, lässt sich lediglich mit Expertenwissen und über den Vergleich der Schnittstellen erschließen.

Ferner entstehen durch den fortlaufenden Einfluss von Innovationen auf die E/E-Architekturentwicklung (Abschnitt 4.2.1) mehrere Funktionsnetz-Generationen für eine E/E-Architektur-Plattform.

Darüber hinaus gibt es die funktionale Kopplung von E/E-Modulen, die nachfolgend näher analysiert wird.

Klassen funktionaler Kopplung von E/E-Modulen

Die funktionale Kopplung von E/E-Modulen ist in der Verteilung der Funktionen auf mehrere E/E-Module begründet (vgl. [125]). Die Evolution und die Verblockung der E/E-Module kann somit nicht unabhängig durchgeführt werden.

Für die funktionale Kopplung der E/E-Module lassen sich drei verschiedene Klassen mit unterschiedlichen Ursachen für die Kopplung unterscheiden.

Durch verteilte Funktion Eine funktionale Kopplung kann durch eine verteilte Funktion entstehen (Abbildung 4.6 a)). Hierbei erfordert die Funktion aufeinander abgestimmte Funktionsbeiträge bzgl. Schnittstellen und Verhalten auf den entsprechenden E/E-Modulen. Soll somit ein E/E-Modul weiterentwickelt oder wiederverwendet (z. B. rückverblockt) werden, so erfordert dies ggf. eine Anpassung der an der Funktion beteiligten E/E-Module oder gar eine Rückverblockung dieser E/E-Module.

Durch Wiederverwendung von Funktionsbeiträgen Darüber hinaus kann eine funktionale Kopplung durch Wiederverwendung von Funktionsbeiträgen entstehen (Abbildung 4.6 b)). Zuzüglich zu den Abhängigkeiten durch eine verteilte Funktion beschreibt diese Klasse die Abhängigkeiten zwischen den Varianten des E/E-Moduls. Wird eine der Varianten weiterentwickelt, so müssen ggf. auch die anderen Varianten angepasst werden.

Durch Gemeinsamkeiten in der Implementierung Eine weitere Klasse funktionaler Kopplung entsteht durch Gemeinsamkeiten in der Implementierung (Abbildung 4.6 c)). Üblicherweise gibt es Funktionsbeiträge mehrerer Funktionen auf der ECU eines E/E-Moduls. Der Implementierung durch AUTOSAR Software Components wird meist eine Software-Architektur zu Grunde gelegt, bei der die Gemeinsamkeiten dieser verschiedenen Funktionsbeiträge durch die gleichen Software-Komponenten implementiert werden. So wird z. B. ein zentraler Koordinator für verschiedene Funktionen in einer SWC implementiert. Damit besteht eine m:n-Beziehung zwischen den Funktionsbeiträgen der Funktionen und den SWCs in der Implementierung. Diese gemeinsamen SWCs der verschiedenen Funktionen stellen die intransparente Gemeinsamkeit in der Implementierung dar, obwohl die Funktionen voneinander unabhängig

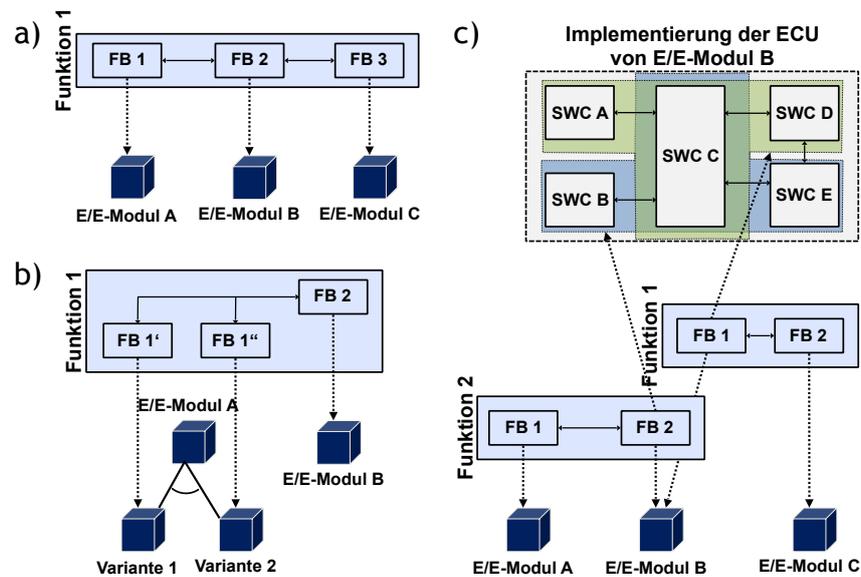


Abbildung 4.6: Abhängigkeiten zwischen den E/E-Modulen durch Software. Die drei Klassen der funktionalen Abhängigkeit von E/E-Modulen haben unterschiedliche Ursachen: a) Durch verteilte Funktionen - Funktion 1 hat die Funktionsbeiträge 1, 2, 3 auf den E/E-Modulen A, B, C. b) Durch Wiederverwendung von Funktionsbeiträgen - der Funktionsbeitrag 1 der Funktion 1 wird auf beiden Varianten des E/E-Moduls A wiederverwendet. c) Durch Gemeinsamkeiten in der Implementierung der Funktionsbeiträge: Der Funktionsbeitrag FB1 der Funktion 1 wird durch die Softwarekomponenten SWCs A, C, D implementiert (grün hinterlegt) während Funktionsbeitrag FB2 der Funktion 2 durch die SWCs B, C, E implementiert wird (blau hinterlegt). Damit teilen beide Funktionen die SWC C in ihrer Implementierung. Vgl. [124, 125]. Nachgedruckt, adaptiert, übersetzt mit Genehmigung von Springer. FB Funktionsbeitrag; SWC Softwarekomponente

erscheinen. Dies erschwert nicht nur das Verständnis sondern erfordert für die Änderung einer Funktion ggf. auch eine Anpassung der anderen Funktionen auf dem E/E-Modul. Insbesondere die Evolutionen bei denen die bisherige SWC-Struktur aufgebrochen werden muss, werden dadurch weiter kompliziert.

Die beschriebenen funktionalen Kopplungen zwischen den E/E-Modulen sind nicht in der Modulstrategie beschrieben und erschweren den Umgang mit den E/E-Modulen aus E/E-Architektursicht. Aus E/E-Architektursicht besteht derzeit keine Methodik um die Abhängigkeiten, die aus der funktionalen Kopplung resultieren, zu beschreiben.

Verzerzte Abhängigkeiten zwischen Funktionen

Für die zu den Funktionsbeiträgen zugeordneten Schnittstellen ergeben sich zwei Konsequenzen aus E/E-Architektursicht. Erstens können von den Schnitt-

stellen, die einem Funktionsbeitrag zugeordnet sind, ausgehend keine Rückschlüsse auf darüber hinaus noch zusätzlich benötigten Schnittstellen gezogen werden. Diese zusätzlich benötigten Schnittstellen können anderen Funktionsbeiträgen auf dieser ECU zugeordnet sein. Zweitens können die Schnittstellenbeschreibungen falsche Rückschlüsse über die Abhängigkeiten zwischen den Funktionen zulassen. Die zu den Funktionsbeiträgen der unterschiedlichen Funktionen zugeordneten Schnittstellen verzerren die Abhängigkeiten zwischen den Funktionen, da:

- einem Funktionsbeitrag exklusiv zugeordnete Schnittstellen eine Unabhängigkeit dieses Funktionsbeitrags zu anderen Funktionsbeiträgen (unrichtig) indizieren und
- mehreren Funktionsbeiträge gemeinsam zugeordnete Schnittstellen eine Abhängigkeit über ausschließlich diese Schnittstellen (unrichtig) indizieren.

Ableitung von Anforderungen

Insgesamt erschwert die Charakteristik des Funktionsnetzes die E/E-Architekturarbeit signifikant. Gleichzeitig erfordern E/E-Architekturansforderungen im Rahmen einer E/E-Architektur-Plattform-Entwicklung, wie z. B. Skalierbarkeit und Evolutionsfähigkeit, eine Untersuchung einer sinnvollen Funktionsrepartitionierung sowie einer Neustrukturierung ausgewählter Bereiche anhand funktionaler Abhängigkeiten. Somit ist ein Vorgehen erforderlich, das einen Umgang mit der Charakteristik des Funktionsnetzes und der Implementierungssicht erlaubt (Anforderung 4.8).

Anforderung 4.8 (Unabhängigkeit vom Funktionsnetz): Die zukünftige E/E-Architekturmodellierung muss eine E/E-Architecturentwicklung unabhängiger vom konkreten Funktionsnetz erlauben.

Zusätzlich dazu machen die funktionale Kopplung und Beschränkungen bei der Nutzung der Funktionen und Funktionsbeiträge zur Bewertung der Evolution von E/E-Modulen eine Beschreibung und eine Sicht über die Ursache der Abhängigkeiten notwendig (Anforderung 4.9).

Anforderung 4.9 (Transparenz der Kopplung): Um die Evolution und Wiederverwendung von E/E-Modulen bewerten zu können, muss die funktionale Kopplung zwischen den betroffenen E/E-Modulen aus der E/E-Architekturmodellierung ableitbar sein.

4.3 Analyse von Lösungsansätzen und Ableitung von Designentscheidungen für ein Konzept

In den vorherigen Abschnitten wurden allgemeine Anforderungen (Abschnitt 4.1) und Anforderungen aus der Analyse der E/E-Architekturmodellierung (Abschnitt 4.2) abgeleitet. Im nachfolgenden Abschnitt werden Lösungsansätze untersucht und Designentscheidungen für das Konzept abgeleitet (vgl. Abbildung 4.1).

4.3.1 Bewertung der Evolution von E/E-Systemen durch Abstraktion

Zur Bewertung der Evolution eines E/E-Moduls müssen funktionale Fragestellungen beantwortet werden (Abschnitt 1.3). Dazu gehören u. a. das Erfassen

- der benötigten Schnittstellen,
- der Änderung der Struktur des Funktionsnetzes und
- der Auswirkung auf die Buslast.

Dabei müssen die Anforderungen aus

- der Analyse der Funktionsnetzmodellierung (Abschnitt 4.2.1) und
- aus der Analyse der Evolution von E/E-Systemen (Abschnitt 4.2.2)

beachtet werden. Nachfolgend wird neben einer weiteren Herausforderung beschrieben und analysiert, wie eine Abstraktion zur Beantwortung der funktionalen Fragestellungen genutzt werden kann.

Erarbeitung einer abstrakten Sicht

Unabhängiger, umfassender und generationenübergreifend kann die E/E-Architekturarbeit werden, indem eine in der Softwaretechnik sehr häufig eingesetzte Methode angewendet wird: Die Abstraktion. Mit Abstraktion wird in der Softwareentwicklung das weniger Relevante vernachlässigt um das Wichtige in den Fokus zu bringen [80]. Wie die Anwendung der Abstraktion für die E/E-Architekturentwicklung sinnvoll einzusetzen ist, ist hierbei nicht bekannt. Damit ist offen

- welche Daten für solch eine abstrakte Beschreibung benötigt werden,

- wie diese zu einer Abstraktion verknüpft werden und
- welche Schnittgröße sinnvoll ist.

Bei der Erarbeitung der Abstraktion muss eine Sicht entstehen, die die notwendige Unschärfe für den E/E-Architekten herstellt. Dementsprechend ist eine Beschreibung der Gesamtfunktionalität sinnvoll. Dazu gehört auch, dass die Schnittstellen der Elemente dieser Sicht die Funktionalitäten besser widerspiegeln und leicht zu erfassen sind. Dementsprechend muss auch eine Durchgängigkeit und Konsistenz hergestellt werden, so dass diese neue Schnittstellenbeschreibung auf die bereits bestehenden Schnittstellen des Funktionsnetzes abbildbar und verknüpfbar sind. Gleichzeitig müssen die ggf. bestehenden Gemeinsamkeiten in der Implementierung für Funktionen, die sich die Implementierung teilen, darstellbar sein (vgl. Abschnitt 4.2.2, Abbildung 4.6). Damit kann die E/E-Architektur-relevante Überschneidung der Funktionsbeiträge des Funktionsnetzes aufgelöst und transparent gemacht werden. In Summe kann diese Sicht dazu beitragen zu entscheiden, welche Teilumfänge bei der Evolution wie betroffen sind und wie sich die Struktur des Funktionsnetzes ändert.

Um die Abstraktion zu erstellen, sind Lastenhefte ungeeignet, da die E/E-Architektur zu einem Zeitpunkt entwickelt wird, an dem die bestehenden Lastenhefte bereits implementiert sind und die neuen Lastenhefte mit einem gerade durch die E/E-Architecturentwicklung zu bewertenden Funktionshub noch nicht erstellt sind (Abschnitt 3.1.1).

Einerseits kann somit die gegenwärtige Implementierung als Datengrundlage für eine Abstraktion dienen. Um die Implementierung als Datengrundlage zu nutzen, ist ein Aufarbeitungsschritt erforderlich um die notwendige Abstraktion für den E/E-Architekten zu erzeugen. Würde diese Aufarbeitung nicht durchgeführt werden, dann könnten E/E-Architekten die Menge an Details nicht durchdringen oder müssten sehr hohen Zeitaufwand investieren.

Beispiel (vgl. Abschnitt 6.1): Würde die SWC-Architektur der Implementierung des zentralen Radarsteuergerät in das E/E-Architekturmodell ohne eine Abstraktion übernommen, so wäre der E/E-Architekt mit etwa 14 Funktionsblöcken, 4900 Provided- und Required Ports, 1700 Assembly Connectors, 1700 Interfaces und 3700 Data-Elements konfrontiert.

Andererseits kann die Befragung von Experten als Datengrundlage für die Abstraktion dienen. Dies erfordert allerdings eine ggf. intensive Diskussion mit den Experten für die entsprechenden E/E-Systeme und Funktionen. Aspekte, die hierbei im Allgemeinen beachtet werden müssen, sind u. a.

- die Identifikation der unter Umständen vielen involvierten Experten,

- deren Motivation, ihre Zeit zu investieren,
- das Motivieren und Erklären einer abstrakten Sicht und
- die Verständigung auf eine gemeinsame Denkweise und Wortwahl.

Vorteilhaft ist allerdings, dass die Arbeit mit den Experten auch relativ früh, d. h. zu einem Zeitpunkt in der die E/E-Architektur-Plattform entwickelt wird (Abschnitt 3.1.1), durchgeführt werden kann.

Ist erst eine abstrakte Sicht aufgebaut, könnte diese neben der E/E-Architekturbewertung von Evolutionen auch weitergehende Anwendungsmöglichkeiten eröffnen. Diese Abstraktion könnte z. B. auch

- zur Betrachtung und Darstellung von Wirkketten, d. h. komplexe Signalflüsse quer über viele E/E-Systeme hinweg, oder
- zur Betrachtung von Fragestellungen zur Funktionssicherheit und IT-Sicherheit (*Safety* und *Security*)²

verwendet werden. Diese weiterführenden Arten der Anwendung sind allerdings nicht im Umfang dieser Arbeit enthalten.

Variabilität bei der Erstellung einer abstrakten Sicht

Nachdem eine abstrakte Sicht zur Beschreibung der Gesamtfunktionalität eingebracht wurde, wird nachfolgend analysiert, wie der Detaillierungsgrad und Umfang dieser abstrakten Sicht näher beschrieben werden kann.

Zur Bewertung des Umfangs der abstrakten Sicht werden die E/E-System- und die E/E-Architektur-Domänengrenzen herangezogen. Bei der Evolution eines E/E-Systems können auch Nachbar-E/E-Systeme betroffen sein. Dementsprechend ist eine abstrakte Sicht nur mit einem Umfang, der mehrere E/E-Systeme überspannt, sinnvoll. Wird die E/E-Architektur-Domänengrenze als Umfang für die abstrakte Sicht betrachtet, spricht für diesen Umfang, dass der Großteil der Evolutionen innerhalb einer E/E-Architektur-Domäne stattfindet. Hierbei unterschieden sich die einzelnen E/E-Architektur-Domänen allerdings stark hinsichtlich der

- Strukturierung,
- Problemstellung,
- Arbeitsweise,

²Konzepte zur Security müssen möglichst bereits in der frühen Entwicklungsphase einfließen. Dabei kann auch der Einsatz des Werkzeugs PREEvision, in dem die E/E-Systeme modelliert werden, eine Rolle spielen [37].

- Datenbasis und
- Evolutionszyklen.

In der Praxis heißt dies z.B., dass sich die E/E-Systeme in der E/E-Architektur-Domäne wie folgt unterscheiden:

- in Body&Comfort sind sie voneinander gut abgrenzbar und zur Implementierung gut zuordenbar,
- in Chassis/Fahrerassistenz weisen sie starke Überschneidungen auf,
- in Powertrain ist eine Aggregatesicht vorherrschend und
- in Infotainment sind sie als Black-Box durch Zulieferer implementiert.

Demnach gelingt die funktionale Dekomposition der Fahrzeug-Systeme in E/E-Komponentenbeiträge und Schnittstellen der Implementierung heute nur in Body&Comfort. Insbesondere für die Fahrerassistenzsysteme existiert ein Bruch zwischen der E/E-Systemsicht und lokaler Software-Architektur (SWA). Dies macht die E/E-Architekturbetrachtungen sehr schwer (vgl. funktionale Kopplung von E/E-Modulen durch Gemeinsamkeiten in der Implementierung in Abschnitt 4.2.2). Für Fahrwerk-E/E-Systeme ist die SWA-Sicht nicht vorhanden. Jedoch werden auch Evolutionen im Rahmen der E/E-Architecturentwicklung betrachtet, die Auswirkungen über mehrere E/E-Architektur-Domänen hinweg haben. Deshalb ist es erforderlich, dass die abstrakte Sicht im Umfang variabel über mehrere E/E-Architektur-Domänen hinweg betrieben werden kann.

Hinsichtlich des Detaillierungsgrads wird für verschiedene E/E-Architekturfragestellungen im Rahmen einer E/E-Architektur-Plattform-Entwicklung klassifiziert, welche Ebene in Betracht gezogen werden muss um die Fragestellungen zu beantworten (Tabelle 4.4). Es zeigt sich, dass dies von der Fragestellung abhängt und je nach Sachlage auch detailliertere Ebenen einbezogen werden müssen. Daraus folgt wiederum, dass für die Erstellung einer abstrakten Sicht die Detaillierungstiefe variabel sein muss. Damit können je nach Fragestellung und Sachlage die abstrakte Sicht angepasst und ggf. weitere Informationen hinzugefügt werden.

Bei der Erschließung weiterer Detaillierungsstufen ist allerdings zu beachten, dass für die SWA bzw. Implementierung aus E/E-Architektursicht drei Transparenz-Klassen existieren. Abhängig von der Transparenz-Klasse ist die Erschließung von Detailinformationen unterschiedlich schwer, wie nachfolgend beschrieben ist:

Transparent Die SWCs lassen sich mit einer n:1-Abbildung zu Funktionsbeiträgen zuordnen und die Schnittstellen der SWCs lassen sich den

Tabelle 4.4: Bewertung der Detaillierungsstufe für E/E-Architektur-Fragestellungen.

	Kernaufgabe:	Kernaufgabe:	Kernaufgabe:	Kernaufgabe:	Teil- betrachtung:	Teil- betrachtung:	Teil- betrachtung:
	Modul- strategie	Neue Vernet- zungskonzepte	Neue Funktion	Neue Funktions- verteilung	Flash-Perfor- mance	Timing- Optimierung	Buslast- Optimierung
<i>E/E-Architektur- Domäne</i>	relevant	relevant	relevant	relevant	relevant	relevant	relevant
<i>E/E-Module und ECUs</i>	relevant	relevant	relevant	relevant	relevant	relevant	relevant
<i>Funktionen und FB</i>	teilweise relevant	teilweise relevant	relevant	relevant	nicht relevant	relevant	relevant
<i>SWA</i>	teilweise relevant	teilweise relevant	teilweise relevant	teilweise relevant	nicht relevant	relevant	teilweise relevant
<i>Implemen- tierung</i>	teilweise relevant	nicht relevant	nicht relevant	teilweise relevant	nicht relevant	teilweise relevant	teilweise relevant

Für verschiedene E/E-Architektur-Fragestellungen im Rahmen einer E/E-Architektur-Plattform-Entwicklung wurde bewertet, welche Detaillierungsstufe zur Beantwortung dieser Fragestellung relevant ist. Einige der Fragestellungen sind hierbei eine Kernaufgabe der E/E-Architecturentwicklung während andere teilweise betrachtet werden. FB Funktionsbeitrag; SWA Softwarearchitektur

einzelnen Funktionsbeiträgen zuordnen (z. B. E/E-Architektur-Domäne Body&Comfort),

Semi-Transparent Mit Zusatzaufwand, z. B. mit Hilfe einer Übersetzungstabelle, lässt sich eine interne Sicht generieren (z. B. E/E-Architektur-Domäne Chassis/FAS),

Black-Box Keine Information über interne SWA und Schnittstellen (z. B. Commodity ECUs).

Daraus folgt, dass die abstrakte Sicht variabel hinsichtlich des Umgangs mit diesen drei Transparenz-Klassen sein muss.

Verortung der Abstraktion

Nachfolgend wird analysiert, wie eine neue Abstraktion in die E/E-Architekturmodellierung integriert werden kann. Dazu müssen die relevanten E/E-Architekturmodellierungsprozesse untersucht werden, die das E/E-Architekturmodell verändern und eine abstrakte Sicht tangieren können.

Einer dieser relevanten E/E-Architekturmodellierungsprozesse ist die Konkretisierung (Abschnitt 4.2.1):

- In der ersten Phase der Konkretisierung müssen neue Funktionalitäten aus E/E-Architektursicht bewertet werden. Dies erfolgt zum Teil mit groben Schätzungen.

- Erst in der zweiten Phase der Konkretisierung werden diese dann so verfeinert, dass sie der Detaillierung des Funktionsnetzes entsprechen.

Die neue abstrakte Sicht kann die Informationen, die im Rahmen der ersten Phase der Konkretisierung entstehen, aufnehmen und so als Konzeptraum dienen. Dabei müssen die entstehenden abstrakten Schnittstellen bewertbar sein.

Ein weiterer E/E-Architekturmodellierungsprozess ist die Aktualisierung des Funktionsnetzes. Hier wird das bestehende Funktionsnetz durch eine neuere Version ersetzt. Das bedeutet auch, dass alle hinzugefügten Modellierungen im Funktionsnetz entfernt und nach der Aktualisierung wieder neu modelliert werden müssen. Ursache dafür ist, dass in der heutigen E/E-Architekturmodellierung kein vom Funktionsnetz separierter Konzeptraum existiert. Die abstrakte Sicht muss demnach nach der Aktualisierung des Funktionsnetzes noch Bestand haben, damit nicht alle Änderungen neu modelliert werden müssen.

Insgesamt bietet sich somit eine Verortung auf einer separaten Modellierungsebene des E/E-Architekturmodells an. Dort kann die Gesamtfunktionalität für ein besseres funktionales Verständnis dargestellt werden und gleichzeitig der Konzeptraum zur zielgerichteten Integration geschaffen werden.

Ableitung von Designentscheidungen

Die Charakteristik des Funktionsnetzes erschwert die E/E-Architekturarbeit signifikant und erfordert die Erarbeitung einer neuen Art der Beschreibung von Funktionen und Schnittstellen in Form einer abstrakten Sicht. Diese muss generationenübergreifend arbeiten, die Gesamtfunktionalität darstellen und als Konzeptraum dienen können. Die Inhalte müssen dabei variabel im Umfang, in der Detaillierungstiefe und im Umgang mit den Transparenzklassen entstehen können. Insgesamt erfordert die zukünftige E/E-Architekturmodellierung somit eine abstrakte Ebene zur Bewertung der Evolution (Designentscheidung 4.1).

Designentscheidung 4.1 (Abstrakte Ebene zur Bewertung der Evolution): Die zukünftige E/E-Architekturmodellierung muss die Integration einer variablen abstrakten Ebene erlauben, die die Gesamtfunktionalität beschreiben und als Konzeptraum dienen kann.

Die abstrakte Ebene stellt somit auch Informationen dar, die detaillierter im Funktionsnetz beschrieben sind. Um Transparenz und Durchgängigkeit im E/E-Architekturmodell sicherzustellen, ist aus diesem Grund auch ein Vorgehen nötig um die Beziehung zwischen abstrakten Schnittstellen und detaillierter Implementierung herzustellen (Designentscheidung 4.2).

Designentscheidung 4.2 (Verbindung zwischen abstrakter Ebene und Funktionsnetz): Die zukünftige E/E-Architekturmodellierung muss die Herstellung

einer transparenten und durchgängigen Beziehung zwischen abstrakter Ebene und Funktionsnetz erlauben.

4.3.2 Umsetzung der Evolution von E/E-Systemen durch Transfer von Methoden der Software-Evolution

Um die Evolution der E/E-Systeme (Anforderung 4.1) umzusetzen, werden die Methoden für die Evolution aus der Softwareentwicklung (Abschnitt 3.3) nachfolgend analysiert und hinsichtlich ihrer Übertragbarkeit auf die E/E-Architekturmodellierung bewertet. Bei einer Übertragung auf die E/E-Architekturmodellierung muss insbesondere beachtet werden, dass die Modellierung in verschiedenen Ebenen durchgeführt wird und heterogen ist (Abschnitt 4.2.1).

Die Methode Aspect Weaving (Abschnitt 3.3.1) erlaubt die Definition von *aspects* und damit die Injektion von Code an den spezifizierten Stellen, genannt *joint points*. Für die E/E-Architecturentwicklung relevante Herausforderungen, wie das Aspect Weaving mehrerer *aspects*, wurden bereits untersucht. Allerdings basiert der Ansatz auf der Nutzung von Sequenz-Diagrammen um die *joint points* zu definieren. Da keine Automatisierung zur Erkennung der *joint points* unabhängig von den genutzten Sequenz-Diagrammen existiert, ist eine Übertragung auf die E/E-Architecturentwicklung, für die keine Sequenz-Diagramme vorliegen, aufwendig.

Eine weitere Methode, das Ontology-Based Modeling (Abschnitt 3.3.2), erlaubt die Beschreibung und anschließende Durchsuchung eines Sachgebiets. Damit erlaubt das nach dieser Methode entstandene Modell, Existierendes besser zu verstehen und ist somit relevant für die E/E-Architekturmodellierung. Zudem existieren Mechanismen um das Modell zu ändern und so die Struktur zu erweitern, zu verkleinern oder zu verändern. Mit einer zunehmenden Detaillierung des Modells, die für die E/E-Architecturentwicklung notwendig ist, wird allerdings auch die Suche im Modell zunehmend aufwändiger. Insgesamt würde sich mit dem Ontology-Based Modeling die Komplexität der E/E-Architekturmodellierung deutlich erhöhen.

Eine dritte Methode, das Feature Oriented Refactoring (Abschnitt 3.3.3), definiert eine Algebra für die Beschreibung der Interaktion von verschiedenen Merkmalen. Dabei ist auch eine Methode zur Identifikation der Abhängigkeiten zwischen den einzelnen Merkmalen gegeben. Allerdings können die Merkmale nur nacheinander inkrementell hinzugefügt werden.

Tabelle 4.5: Bewertung der Methoden der Evolution aus der Softwareentwicklung

Kriterium	Aspect Weaving	Ontology-Based Modeling	Feature Oriented Refactoring
<i>Einfachheit, das Modell der Methode zu definieren</i>	+	-	++
<i>Flexibilität, das Modell der Methode bei Evolution zu ändern</i>	-	+	++
<i>Aussagekraft des Modells der Methode</i>	-	+	+
<i>Stärke der theoretischen Basis der Methode</i>	-	+	+
<i>Übertragbarkeit auf die E/E-Architekturentwicklung</i>	-	-	+
<i>Eignung zur Implementierung im Werkzeug</i>	-	+	+

Hierbei bedeuten ++ sehr gut, + gut und – mangelhaft.

Ableitung von Designentscheidungen

Insgesamt zeigt eine Bewertung der Ansätze (Tabelle 4.5), dass das Feature Oriented Refactoring als am besten für eine Umsetzung der Evolution von E/E-Systemen geeignet ist. Das Modell der Methode ist sehr einfach zu definieren, bei Änderungen sehr leicht anzupassen und aufgrund der hinterlegten Theorie aussagekräftig. Die Methode basiert auf Merkmalen, die bereits in der heutigen E/E-Architekturmodellierung modelliert werden können. Deshalb ist die Methode gut auf die E/E-Architekturentwicklung und -modellierung übertragbar und implementierbar. Da sich die für die Evolution relevanten Informationen in der abstrakten Sicht finden (Designentscheidung 4.1), muss das Feature Oriented Refactoring für diese abstrakte Sicht eingesetzt werden (Designentscheidung 4.3).

Designentscheidung 4.3 (Transfer des Feature-Oriented Refactoring): Die zukünftige E/E-Architekturmodellierung muss das Feature Oriented Refactorings für die abstrakte Sicht unterstützen.

4.3.3 Neugestaltung der Modellstruktur und des Variantenmanagements für E/E-Systeme

Für die Neugestaltung der Modellstruktur im Zuge einer Erweiterung des E/E-Architekturmodells um E/E-Systeme (Anforderung 4.2) und deren Nutzung zur Variantenkonfiguration (Anforderung 4.3), soll nachfolgend die Beziehung zwischen den E/E-Systemen, den Funktionen und den E/E-Modulen näher analysiert werden.

Generell gilt, dass ein E/E-System mehrere E/E-Module umfasst. Bei Weiterentwicklungen und Innovationen wird üblicherweise entschieden, ob der Umfang

- ein eigenes E/E-System rechtfertigt oder
- ein bestehendes E/E-System erweitert

wird. Um ein E/E-System im E/E-Architekturmodell abzugrenzen, müssen

- die zugehörige Hardware der entsprechenden E/E-Module und
- die zugehörige Software in Form von Funktionen

zugeordnet werden. Der Mechanismus, der die Zuordnung zu einem E/E-System im E/E-Architekturmodell umsetzt, muss somit mehrere Modellierungsebenen überspannen. Die Zuordnung zwischen E/E-System und E/E-Modul und Funktionen lässt sich einfach herstellen, wenn die E/E-Systeme keine Überschneidung aufweisen, d.h. wenn Funktionsbeiträge und Hardwarekomponenten ausschließlich zu einem E/E-System zuordenbar sind. Es zeigt sich allerdings, dass die Zuordnung nicht ausschließlich sein muss, weshalb es in den folgenden drei Fällen zu einer Überschneidung von E/E-Systemen kommt (Abbildung 4.7):

- a) Eine verteilte Funktion kann mehreren E/E-Systemen zugeordnet sein.
- b) E/E-Systeme können sich überschneiden, wenn wiederverwendete Funktionen und Implementierungsalternativen zu verschiedenen E/E-Systemen zugeordnet sind.
- c) E/E-Systeme können sich bei Gemeinsamkeiten in der Implementierung überschneiden.

Diese Fälle zeigen, dass die Zuordnung einer Funktion zu einem E/E-System nicht ausreichend ist, um kenntlich zu machen, welche Teile der Funktionen zu welchen E/E-Systemen gehören.

Allerdings ist es für die Nutzung der E/E-Systeme für die Variantenkonfiguration notwendig, dass die E/E-Systeme klar zueinander abgegrenzt sind. Gleichzeitig können sich E/E-Systeme gegenseitig ausschließen oder bedingen, was die Modellierung der Beziehungen zwischen den E/E-Systemen notwendig macht. Diese Modellierung der Beziehungen unterstützt die Variantenkonfiguration und kann deshalb als Konfigurationsmodell bezeichnet werden.

Es ist außerdem zu beachten, dass die heutige Modellierung der E/E-Module eine Nutzung dieser bei der Variantenkonfiguration verhindert (Abschnitt 4.1.2). Heute wird jede E/E-Modul-Variante vollständig modelliert.

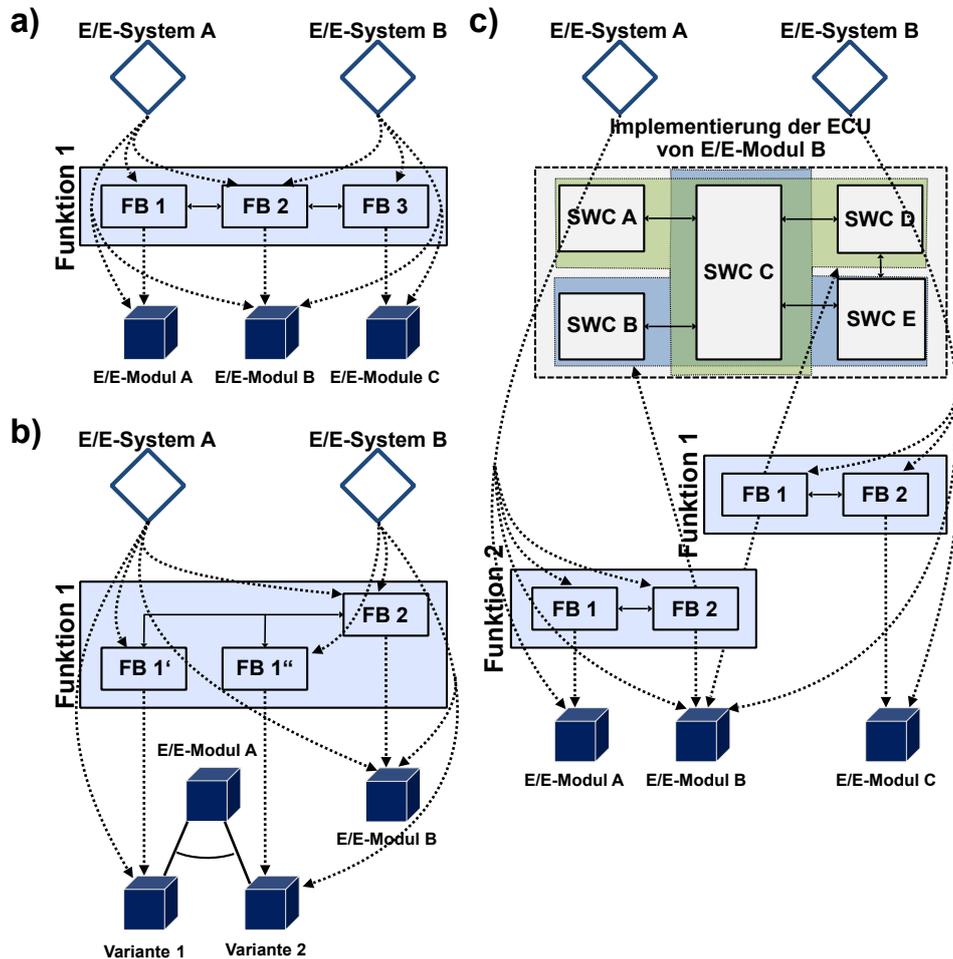


Abbildung 4.7: Beziehung der E/E-Systeme zu Funktionen und E/E-Modulen. Die Zuordnung von E/E-Systemen zu Funktionen und E/E-Modulen muss nicht ausschließlich sein: a) Die Funktionsbeiträge einer verteilten Funktion können zu unterschiedlichen E/E-Systemen gehören. b) Die Wiederverwendung von Funktionsbeiträgen bzw. Implementierungsalternativen können unterschiedlichen E/E-Systemen zugeordnet sein. c) Die gemeinsame Implementierung von zu unterschiedlichen E/E-Systemen gehörenden Funktionsbeiträgen eines E/E-Moduls kann eine versteckte Abhängigkeit zwischen E/E-Systemen erzeugen. Gemeinsame Implementierung von c): Der Funktionsbeitrag FB1 der Funktion 1 wird durch die Softwarekomponenten SWCs A, C, D implementiert (grün hinterlegt) während Funktionsbeitrag FB2 der Funktion 2 durch die SWCs B, C, E implementiert wird (blau hinterlegt). Damit teilen beide Funktionen die SWC C in ihrer Implementierung. Vgl. [125]. FB Funktionsbeitrag; SWC Softwarekomponente

Hardware, die in mehreren E/E-Modul-Varianten enthalten ist, ist somit mehrfach im E/E-Architekturmodell vorhanden. Dies macht eine Anpassung der Modellierung der E/E-Module erforderlich. Die Modellierung eines E/E-Moduls soll somit alle seine E/E-Modul-Varianten (jeweils 100%) umfassen und kann deshalb als 150%-E/E-Modulmodell bezeichnet werden ³.

Ableitung von Designentscheidungen

Insgesamt zeigt sich, dass mehrere Designentscheidungen an die Ausgestaltung des E/E-Systemmodells für die zukünftige E/E-Architekturmodellierung geknüpft sind (Designentscheidung 4.4).

Designentscheidung 4.4 (Ausgestaltung des E/E-Systemmodells): Das E/E-Systemmodell der zukünftigen E/E-Architekturmodellierung muss mehrere Modellierungsebenen überspannen und eine Abgrenzung der E/E-Systeme zueinander, die Integration von 150%-E/E-Modulmodellen und die Modellierung von Beziehungen zwischen den E/E-Systemen in einem Konfigurationsmodell ermöglichen.

4.3.4 Wiederverwendung von E/E-Systemen durch Transfer und Anpassung des Produktlinien Engineering Prozesses

Um die Wiederverwendung von E/E-Systemen (Anforderung 4.4) in der E/E-Architekturmodellierung umzusetzen, wird nachfolgend die bestehende Produktlinienentwicklung (Abschnitt 3.4.3) hinsichtlich erforderlicher Anpassungen und Erweiterungen analysiert und bezüglich einer Übertragung auf die E/E-Architekturmodellierung bewertet.

Erweiterung des Produktlinienansatzes der E/E-Architekturentwicklung

Die Wiederverwendung von E/E-Systemen in der E/E-Architekturmodellierung umfasst sowohl die modulatorientierten Hardware-Anteile als auch die Software-Anteile in Form des Funktionsnetzes. Der bestehende Produktlinienansatz für die E/E-Architekturentwicklung ist heute modulatorientiert und enthält dementsprechend nur die Hardware-Prozesse der E/E-Systementwicklung. Um die Wiederverwendung von E/E-Systemen im Rahmen eines Produktlinienansatzes zu realisieren, muss der heutige Ansatz um die restlichen Prozesse der E/E-Systementwicklung erweitert

³Im Allgemeinen wird im Sprachgebrauch der Modellierung die Übermenge aller vollständigen (100%-)Varianten oft als 150%-Modell der Varianten bezeichnet.

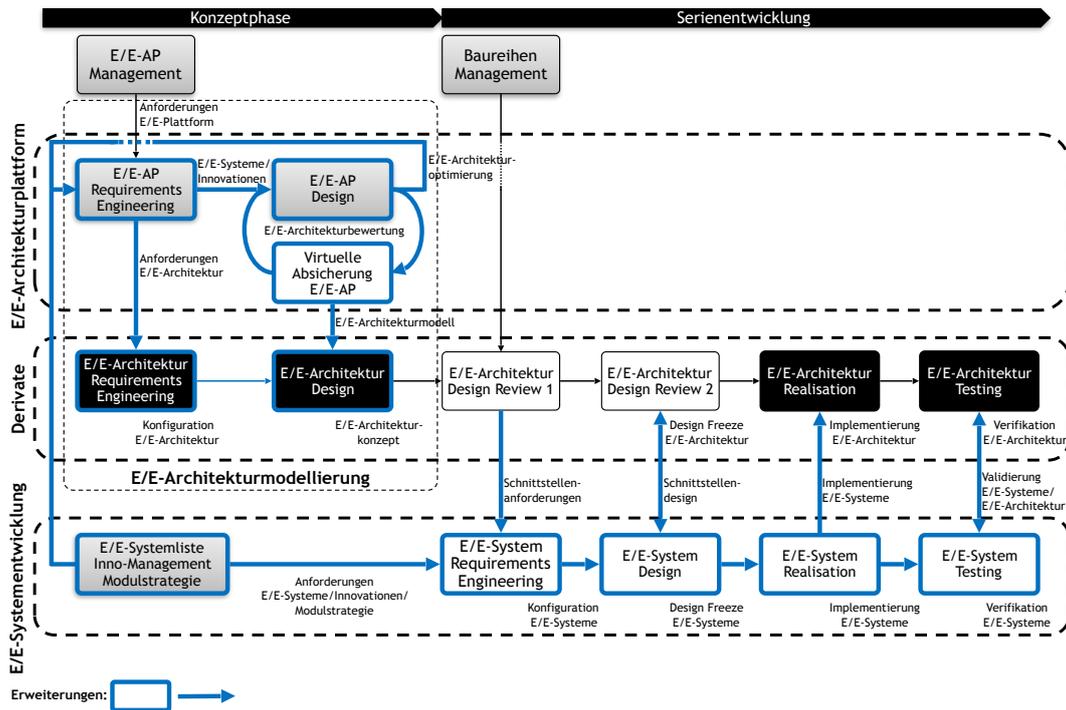


Abbildung 4.8: Erweiterung des Produktlinienansatzes der E/E-Architekturentwicklung. Der Produktlinienansatzes der E/E-Architekturentwicklung [82] wird um die Einbindung der E/E-Systementwicklung und des Innovationsmanagements erweitert. Die Prozesse sind den Entwicklungsphasen zugeordnet. Die gestrichelten Rahmen definieren hierbei die Zugehörigkeit der Prozesse. Grau hinterlegte Prozesse entsprechen dem Domain- und schwarz hinterlegte Prozesse dem Application-Engineering des Software Produktlinien Engineerings [17] aus dem der Produktlinienansatz der E/E-Architekturentwicklung hervorging. Blau gerahmte Prozesse und blaue Pfeile stellen die Erweiterungen dar. E/E-AP E/E-Architektur-Plattform; Inno Innovation

werden (Abbildung 4.8). Dazu werden die zur Modulstrategie zugehörigen Prozesse mit den Prozessen der E/E-Systementwicklung, in denen die Modulstrategie-Prozesse enthalten sind, ersetzt. Gleichzeitig werden die Prozesse des Innovations-Managements und der E/E-Systemliste integriert. Diese Anpassungen erlauben die Einbeziehung von E/E-Systemen in den Prozessen der E/E-Architekturmodellierung. Damit können sowohl bei dem E/E-Architektur-Plattform- als auch bei dem Derivate-Design die E/E-Systeme in der E/E-Architekturmodellierung verwendet und wiederverwendet werden.

Tabelle 4.6: Vergleich der Anwendungsfälle der modulatorientierten E/E-Architekturmodellierung mit neuen Wiederverwendungsmechanismen

Anwendungsfall der modulatorientierten E/E-Architekturmodellierung	Automatisierung der modulatorientierten E/E-Architekturentwicklung	Wiederverwendungsmechanismen des Werkzeugherstellers
Initiale Erstellung von E/E-Modulen aus bestehender E/E-Architekturmodellierung (z.B. dem E/E-Architekturmodell einer weiteren Baureihe)	Mechanismus zum Festlegen eines Bezugsobjekts zum Start zur E/E-Modul-spezifischen oder zur automatisierten Erstellung (im Plug-In <code>createModule</code>)	Nicht unterstützt
Import von E/E-Modulen aus den Modulheften der Modulstrategie	Benutzergeführter manueller Import oder automatisierte Erstellung (im Plug-In <code>importModule</code>)	Nicht unterstützt
Konfiguration von E/E-Modulen	Merkmalsorientierte Konfiguration, Verifikation der Konfiguration und Erweiterbarkeit der Konfiguration (im Plug-In <code>configureModule</code>)	Unterstützt durch Modellierung und Überprüfung von <code>required-</code> und <code>exclude-</code> Beziehungen
Ebenenspezifische Integration der E/E-Module	Benutzergeführte Integration und automatisierte Integration (im Plug-In <code>integrateModule</code>)	Unterstützt durch manuelle Modellierung
Aktualisierung der E/E-Module	Automatisiert (im Plug-In <code>updateModule</code>)	Automatisiert
Innovationsmodellierung	Wiederverwendung von Konzeptmodellen (im Plug-In <code>createInnovation</code>)	Unterstützt durch manuelle Modellierung

Nutzung einer produktiven Werkzeugfunktionalität zur Wiederverwendung

Wird die heutige E/E-Architekturmodellierung um die E/E-Systeme erweitert und sollen diese wiederverwendet werden, so muss

- entweder die bestehende Automatisierung der modulatorientierten E/E-Architekturmodellierung (Abschnitt 3.4.3) erweitert werden
- oder es müssen die neuen Wiederverwendungsmechanismen des Werkzeugherstellers umgesetzt werden.

Diese beiden Optionen werden deshalb verglichen und bewertet (Tabelle 4.6).

Der Vergleich zeigt, dass insgesamt die wichtigsten Automatisierungen der heutigen E/E-Architekturmodellierung einfach durch die neuen Wiederverwendungsmechanismen darstellbar sind. Nicht darstellbar sind lediglich die initiale Erstellung von E/E-Modulen aus bestehenden E/E-Architekturmodellen oder Modulheften der Modulstrategie. Außerdem sind die neuen Wiederverwendungsmechanismen nicht automatisiert, was die Entwicklung einer Automatisierung der Wiederverwendungsmechanismen notwendig macht. Bei der Verwendung der neuen Wiederverwendungsmechanismen werden allerdings prototypische Implementierungen in Form von Plug-Ins durch ins Werkzeug integrierte Lösungen abgelöst. Die Verwendung von Plug-Ins ruft bei einer Versionsänderung des Werkzeugs

- mitunter massive manuelle Anpassung des Quellcodes und
- sich daran anschließende ausgiebige Tests hervor

um die Operativität der Plug-Ins zu gewährleisten. Eine ins Werkzeug integrierte Lösung wird hingegen

- konstant supportet,
- ständig weiterentwickelt und
- auf dem aktuellen Entwicklungsstand einschließlich aller Abhängigkeiten zu anderen Werkzeugerweiterungen und -änderungen gehalten.

Das verringert den Wartungsaufwand und Fehler signifikant.

Ableitung von Designentscheidungen

Damit ist für eine zukünftige E/E-Architekturmodellierung die Verwendung des erweiterten Produktlinienansatzes der E/E-Architekturmodellierung und der produktiven Wiederverwendungsmechanismen inklusive deren Automatisierung zur Wiederverwendung von E/E-Systemen sinnvoll (Designentscheidung 4.5).

Designentscheidung 4.5 (Transfer des Produktlinienengineerings): Für die zukünftige E/E-Architekturmodellierung muss der erweiterte Produktlinienansatz und die Nutzung und Automatisierung der produktiven Wiederverwendungsmechanismen des E/E-Architekturmodellierungswerkzeugs eingeführt werden.

4.4 E/E-System Produktlinien Engineering Konzept

In diesem Kapitel wurden hinsichtlich den Zielsetzungen der Arbeit (Zielsetzung 1.1 und Zielsetzung 1.2) zuvor die Ableitung der Anforderungen aus der Bewertung des Stand der Technik (Abschnitt 4.1) und aus der Analyse der E/E-Architekturmodellierung (Abschnitt 4.2) vorgenommen. Lösungsansätze wurden analysiert und Designentscheidungen getroffen (Abschnitt 4.3). Nachfolgend werden diese Anforderungen und Designentscheidungen in ein Konzept und mehrere Anwendungsfälle überführt, die zusammengenommen als E/E-System Produktlinien Engineering bezeichnet werden (vgl. [124, 125], Abbildung 4.1).

4.4.1 Entwurf der Konzeptionellen Funktionsarchitektur-Ebene zur Bewertung der Evolution von E/E-Systemen

In diesem Abschnitt wird konform zu [124], das Teilkonzept beschrieben, das die Evolution von E/E-Systemen ermöglichen soll (Anforderung 4.1). Dabei soll das Teilkonzept die Heterogenität des Funktionsnetzes (Anforderung 4.5), die Heterogenität der Skalierung von Funktionen (Anforderung 4.6) und die Konkretisierung (Anforderung 4.7) unterstützen. Gleichzeitig soll das Teilkonzept eine Modellierung unabhängiger vom konkreten Funktionsnetz (Anforderung 4.8) erlauben und die Kopplung zwischen den E/E-Modulen transparent machen (Anforderung 4.9). Das Teilkonzept soll dafür eine abstrakte Ebene zur Bewertung der Evolution (Designentscheidung 4.1) beschreiben und eine Verbindung zwischen dieser abstrakten Ebene und Funktionsnetz (Designentscheidung 4.2) ermöglichen.

Dementsprechend wird für das Teilkonzept diese abstrakte Ebene zur Bewertung der Evolution als neue Modellierungsebene definiert und als Konzeptionelle Funktionsarchitektur-Ebene bezeichnet (Definition 4.1). Diese Ebene soll die Überschneidungen auf der Funktionsnetz-Ebene auflösen und durch ihre Abstraktion die Bewertung der Evolution von E/E-Systemen ermöglichen (Abbildung 4.9 a)).

Definition 4.1 (Konzeptionelle Funktionsarchitektur-Ebene): Die Konzeptionelle Funktionsarchitektur-Ebene ist die Modellierungsebene, die die von den E/E-Modulen implementierten verteilten Funktionen, ihre komplexen Beziehungen und ihre versteckten Abhängigkeiten in einer Abstraktion beschreibt, die gesamte E/E-Module einschließlich ihrer Varianten umfasst.

Die Erstellung der Konzeptionellen Funktionsarchitektur-Ebene für Funktionen mit existierender Implementierung erfolgt gemäß der folgenden Vorgehensbeschreibung:

Erstellung der Konzeptionellen Funktionsarchitektur-Ebene:

- Zusammengehörende detaillierte Schnittstellen aus dem Funktionsnetz werden zu einer abstrakten Schnittstelle auf der Konzeptionellen Funktionsarchitektur-Ebene zusammengefasst.
- Die Funktionen des Funktionsnetzes und deren Implementierung wird abstrahiert, indem die Elemente der Funktion beschrieben werden, die die abstrakten Eingaben zu Ausgaben umwandeln.
- Die entstandene abstrakte Beschreibung der Funktionalität soll hierbei klar und intuitiv sein.

- Implementierungsrelevante Informationen werden ignoriert, falls sie nicht E/E-Architektur-relevant sind (wie z.B. eine SWC zur zentralen Koordination).

Dies bildet die Grundlage um neue Funktionalitäten zu integrieren, die in zwei aufeinander folgenden Phasen konkretisiert werden (Abschnitt 4.2.1). Die Modellierung für die erste Phase wird auf der Konzeptionellen Funktionsarchitektur-Ebene durchgeführt und erfolgt gemäß der folgenden Vorgehensbeschreibung:

Erste Phase der Konkretisierung:

- Für jedes wichtige Element der neuen Funktionalität wird ein Element auf der Konzeptionellen Funktionsarchitektur-Ebene erzeugt.
- Die erzeugten Elemente werden mit abstrakten Schnittstellen verbunden, die den Informationsfluss der Funktionalität widerspiegeln.
- Optional kann die Größe der neu erzeugten abstrakten Schnittstellen abgeschätzt und annotiert werden, während die Größe der vorhandenen abstrakten Schnittstellen aus den detaillierten Schnittstellen abgeleitet und annotiert werden kann. Dies kann die Basis für Buslastbewertungen darstellen, die in schrittweiser Optimierung in Dialog mit den Funktions- und E/E-Komponentenexperten einen Konzeptentwurf zur weiteren Konkretisierung liefern kann.

Die Modellierung für die zweite Phase wird auf der Ebene des Funktionsnetzes durchgeführt und erfolgt gemäß der folgenden Vorgehensbeschreibung:

Zweite Phase der Konkretisierung:

- Die Funktionen und ihre Schnittstellen werden ausgearbeitet.
- Diese Detaillierung wird dem Funktionsnetz hinzugefügt.
- Bedarfsweise wird die Abstraktion um die bei dieser Integration vorgenommenen Änderungen angepasst.

Mit diesen Vorgehensbeschreibungen entstehen für detaillierte Schnittstellen des Funktionsnetzes auch abstrakte Schnittstellen auf der Konzeptionellen Funktionsarchitektur-Ebene. Um diese in Relation zueinander zu setzen, wird das Relationsmodell definiert (Definition 4.2, Abbildung 4.9 b)).

Definition 4.2 (Relationsmodell): Das Relationsmodell ist das Modell, das die abstrakten Schnittstellen der Konzeptionellen Funktionsarchitektur-Ebene mit den detaillierten Schnittstellen des Funktionsnetzes in Verbindung setzt.

Bei einer Evolution der abstrakten oder detaillierten Schnittstellen wird auch das Relationsmodell weiterentwickelt.

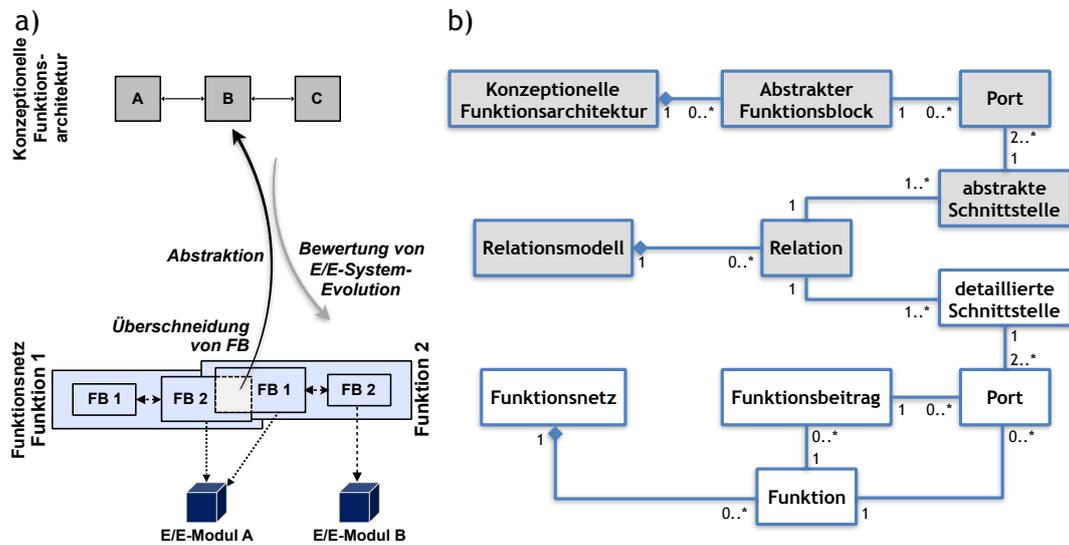


Abbildung 4.9: Konzeptionelle Funktionsarchitektur-Ebene und Relationsmodell. a) die Konzeptionelle Funktionsarchitektur-Ebene abstrahiert die Ebene des Funktionsnetz und soll dadurch eine Bewertung der E/E-System-Evolution erlauben, b) durch das Relationsmodell werden die abstrakten Schnittstellen der Konzeptionellen Funktionsarchitektur zu den detaillierten Schnittstellen des Funktionsnetzes in Verbindung gesetzt; grau gefärbte Blöcke stellen Erweiterungen durch das Teilkonzept dar; FB Funktionsbeitrag; SWC Softwarekomponente

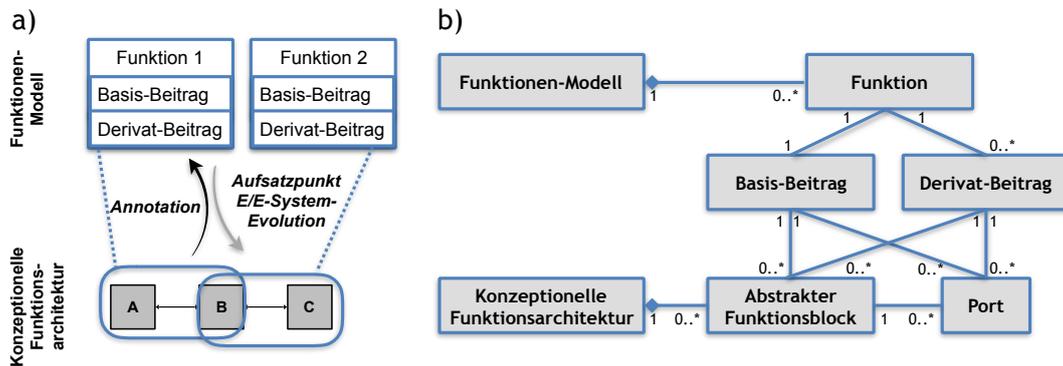


Abbildung 4.10: Funktionen-Modell. a) das Funktionen-Modell wird durch die Annotation der Konzeptionellen Funktionsarchitektur-Ebene erstellt und gibt den Aufsatzpunkt für die E/E-System-Evolution, b) das Funktionen-Modell umfasst die Basis- und Derivat-Beiträge der Funktion, die Elemente und abstrakte Schnittstellen der Konzeptionellen Funktionsarchitektur-Ebene enthalten.

4.4.2 Entwurf des Feature Oriented Architecture Refactorings (FOAR) zur Umsetzung der Evolution von E/E-Systemen

Nachfolgend wird das Teilkonzept beschrieben, das die Evolution von E/E-Systemen ermöglichen soll (Anforderung 4.1). Dabei soll das Teilkonzept die Heterogenität der Skalierung von Funktionen (Anforderung 4.6) unterstützen. Das Teilkonzept soll dafür das Feature-Oriented Refactoring (Abschnitt 3.3) für die Konzeptionelle Funktionsarchitektur-Ebene einsetzen (Designentscheidung 4.3).

Das Teilkonzept wird nachfolgend als Feature Oriented Architecture Refactoring (FOAR) bezeichnet und umfasst ein Funktionen-Modell (Definition 4.3, Abbildung 4.10). Dieses erfasst die Funktionen, die mit der Konzeptionellen Funktionsarchitektur-Ebene beschrieben sind und sich quer über diese verteilen.

Definition 4.3 (Funktionen-Modell): Das Funktionen-Modell ist das Modell, das die Funktionen der Konzeptionellen Funktionsarchitektur-Ebene und ihre Abhängigkeiten zueinander erfasst.

Mit dem FOAR wird eine Funktion im Funktionen-Modell strukturiert. Der Basis-Beitrag umfasst die Anteile auf der Konzeptionellen Funktionsarchitektur, die die Funktion neu einbringt (Definition 4.4). Dementsprechend sind z. B. Funktionsblöcke und Schnittstellen, die zentraler Bestandteil dieser Funktion sind, im Basis-Beitrag enthalten. Jede Funktion hat somit nur einen Basis-Beitrag.

Definition 4.4 (Basis-Beitrag): Ein Basis-Beitrag ist ein Anteil einer Funktion im Funktionen-Modell und enthält diejenigen Elemente und abstrakten Schnittstellen, die diese Funktion der Konzeptionellen Funktionsarchitektur-Ebene hinzufügt.

Allerdings interagieren Funktionen auf der Konzeptionellen Funktionsarchitektur-Ebene miteinander. Somit gibt es Elemente und abstrakte Schnittstellen, die Bestandteil mehrerer Funktionen sind. Ursache dafür ist, dass

- das Element auf der Konzeptionellen Funktionsarchitektur-Ebene einer weiteren Funktion für die eigene Funktionalität benötigt wird, z. B. weil das berechnete Ergebnis einen Input für die eigene Berechnung darstellt, oder
- das Element durch das Vorhandensein der Funktion angepasst wird.

Hierbei kann diese Anpassung auch auf einer anderen Modellierungsebene erfolgen, z. B. indem auf der Ebene des Funktionsnetzes detaillierte Schnittstellen für die abstrakte Schnittstelle auf der Konzeptionellen Funktionsarchitektur-Ebene hinzugefügt werden. Dementsprechend werden in einem Derivat-Beitrag genau diese Anteile eines Basis-Beitrags einer anderen Funktion, mit der interagiert wird, erfasst (Definition 4.5). Da mit mehreren Funktionen interagiert werden kann, kann es auch mehr als einen Derivat-Beitrag für eine Funktion geben.

Definition 4.5 (Derivat-Beitrag): Ein Derivat-Beitrag ist ein Anteil einer Funktion im Funktionen-Modell, der die Interaktion der Funktion mit einer anderen Funktion abbildet. Er enthält diejenigen Elemente und abstrakten Schnittstellen eines Basis-Beitrags einer anderen Funktion, die benötigt oder geändert werden.

Um mehrere Funktionen auf der Konzeptionellen Funktionsarchitektur-Ebene zusammen zu betrachten, ist für das Feature Oriented Architecture Refactoring der $+$ -Operator (Definition 4.6) und der \bullet -Operator (Definition 4.7) notwendig. Mit ihnen kann sichergestellt werden, dass der für die betrachteten Funktionen geltende Umfang auf der Konzeptionellen Funktionsarchitektur-Ebene erfasst werden kann.

Definition 4.6 ($+$ - Operator): Der $+$ - Operator ist die disjunkte Vereinigungsmenge der Elemente und abstrakten Schnittstellen zweier Beiträge.

Definition 4.7 (\bullet - Operator): Der \bullet - Operator ist das Einpflegen einer additiven Änderung an den Elementen und abstrakten Schnittstellen eines Basis-Moduls, das ggf. in einer anderen Modellierungsebene erfolgt.

Dementsprechend wird $\delta b/\delta F \bullet b$ realisiert, indem die Anteile des Derivat-Beitrags Teil der betrachteten Konzeptionellen Funktionsarchitektur werden,

nachdem die Änderung ggf. auch in einem zugehörigen Teil einer anderen Modellierungsebene eingepflegt wurde. Für $f + \delta b / \delta F \bullet b$ wird zusätzlich der entsprechende Basis-Beitrag erstellt und der betrachteten Konzeptionellen Funktionsarchitektur hinzugefügt.

Die initiale Erstellung des Funktionen-Modell erfolgt gemäß der folgenden Vorgehensbeschreibung:

Initiale Erstellung des Funktionen-Modells:

Festlegung Funktionen und Reihenfolge: Im ersten Schritt muss entschieden werden, welche Funktionen im Funktionen-Modell repräsentiert werden sollen. Ein Ansatzpunkt dafür kann eine funktionale Modellierung (Abschnitt 4.2.1) im Funktionsnetz sein. Meist sind die Funktionen jedoch Teil des impliziten Wissens oder wurden bei der Erarbeitung der Konzeptionellen Funktionsarchitektur-Ebene diskutiert. Die Reihenfolge mit der die Funktionen im Funktionen-Modell erstellt werden, ergibt sich größtenteils aus der Skalierung in der Modellierung im Funktionsnetz. Die heterogene Modellierung der Skalierung der Funktionen im Funktionsnetz wird durch die Festlegung der Reihenfolge einheitlich im Funktionen-Modell repräsentiert (Anforderung 4.6).

Annotation: Nachfolgend werden die Elemente und die abstrakten Schnittstellen der Konzeptionellen Funktionsarchitektur-Ebene hinsichtlich ihrer Zugehörigkeit zu den festgelegten Funktionen annotiert.

Erstellung Basis-Beitrag: Entsprechend der festgelegten Reihenfolge werden die annotierten Elemente und abstrakten Schnittstellen den Basis-Beiträgen der Funktionen nacheinander zugeordnet. Nach diesem Schritt gibt es keine Elemente und abstrakten Schnittstellen mehr, die keinem Basis-Beitrag zugeordnet wurden.

Erstellung Derivat-Beiträge: Die Elemente und abstrakten Schnittstellen einer Funktion werden zu einem Derivat-Beitrag zugeordnet, wenn sie Teil des Basis-Beitrags einer anderen Funktion sind. Dabei wird für jeden weiteren Basis-Beitrag ein separater Derivat-Beitrag erstellt.

Werden die E/E-Systeme weiterentwickelt, sind meist mehrere Modellierungsebenen betroffen. Hierbei erfolgt das Umsetzen einer Evolution eines E/E-Systems für das Funktionen-Modell und die Konzeptionelle Funktionsarchitektur-Ebene nach folgender Vorgehensbeschreibung:

Evolution des Funktionen-Modells:

Erfassung betroffener Funktionen: Im ersten Schritt werden diejenigen Funktionen des Funktionen-Modells ausgewählt, die von einer Evolution betroffen sind. Je größer hierbei der Funktionsumfang des weiterzuentwickelnden E/E-Systems ist, desto mehr Funktionen des Funktionen-Modells sind meist betroffen.

Entscheidung zur Auswirkung der Evolution: Anschließend muss entschieden werden, ob die Evolution eine Auswirkung auf das Funktionen-Modell hat. Es ist möglich, dass sich bei einer E/E-System-Evolution lediglich die Zusammenstellung der Funktionen für das E/E-System ändert aber sich die Annotierung der Konzeptionellen Funktionsarchitektur-Ebene nicht ändert. In diesem Fall muss das Funktionen-Modell nicht angepasst werden. Ändert sich durch die E/E-System-Evolution die Annotierung, weil z. B. eine Funktion in zwei Funktionen aufgespalten wird, so muss das Funktionen-Modell durch den nachfolgenden Schritt angepasst werden.

Refaktorisierung: In diesem Schritt erfolgt die Anpassung des Funktionen-Modells, der Konzeptionellen Funktionsarchitektur-Ebene oder beider Modelle. Wird beispielsweise durch die E/E-Systemevolution eine neue Funktion hinzugefügt, werden die entsprechenden Elemente und abstrakten Schnittstellen auf der Konzeptionellen Funktionsarchitektur-Ebene hinzugefügt. Anschließend wird das Funktionen-Modell weiterentwickelt, indem die Basis- und Derivat-Beiträge hinzugefügt, entfernt oder angepasst werden.

Einpfelegen der Änderung (ggf. auf anderer Modellierungsebene):

Nach der Veränderung des Funktionen-Modells und der Konzeptionellen Funktionsarchitektur müssen die Anpassungen ggf. auch auf anderen Modellierungsebenen umgesetzt werden. So muss beispielsweise für eine neue Funktion auch das Funktionsnetz angepasst werden, indem neue Schnittstellen, neue Funktionsbeiträge oder neue Funktionen hinzugefügt werden.

4.4.3 Entwurf eines E/E-Systemmodells für Modellstruktur und Variantenmanagement von E/E-Systemen

Es ist im letzten Abschnitt deutlich geworden, dass das Funktionen-Modell eine enge Verknüpfung zu den E/E-Systemen hat. Deshalb wird nachfolgend das Teilkonzept des E/E-Systemmodells beschrieben mit dem E/E-Systeme in die Modellstruktur der E/E-Architekturmodellierung eingeführt

werden sollen (Anforderung 4.2). Gleichzeitig sollen die E/E-Systeme im Variantenmanagement eingebunden werden (Anforderung 4.3). Dazu soll das E/E-Systemmodell mehrere Modellierungsebenen überspannen, eine Abgrenzung der E/E-Systeme zueinander ermöglichen, die Integration von 150%-E/E-Modulmodellen erlauben und die Verwaltung der Beziehungen zwischen den E/E-Systemen in einem Konfigurationsmodell ermöglichen (Designentscheidung 4.4).

Somit umfasst das Teilkonzept des E/E-Systemmodells ein weiterentwickeltes E/E-Modulmodell und ein Konfigurationsmodell. Der Fokus des Teilkonzepts liegt auf der Separierung der Varianten-Information vom Funktionsnetz bzw. der Hardware-Architektur. Die einzelnen Modelle des Teilkonzepts seien wie folgt definiert:

Definition 4.8 (E/E-Systemmodell): Das E/E-Systemmodell ist das Modell, das die E/E-Systeme zueinander abgrenzt und ihre zugehörigen Funktionen und E/E-Module erfasst.

Definition 4.9 (E/E-Modulmodell): Das E/E-Modulmodell ist das Modell, das die E/E-Module der Modulstrategie und die zugehörige Hardware (ohne deren Mehrfachmodellierung) erfasst.

Definition 4.10 (Konfigurationsmodell): Das Konfigurationsmodell ist das Modell, das die Beziehungen zwischen den E/E-Systemen erfasst und damit Basis für die Konfiguration einer E/E-Architektur-Variante ist.

Für die Ausgestaltung des E/E-Systemmodells müssen neben dem E/E-Modulmodell und dem Konfigurationsmodell noch das Funktionen-Modell (Definition 4.3) und das Funktionsnetz einbezogen werden. Zudem ist eine Strukturierung der E/E-Systeme erforderlich. Ferner ist mit der Anbindung des Variantenmanagements eine Weiterentwicklung der Propagation (Definition 3.1) nötig. Die Ausgestaltung des E/E-Systemmodells mit diesen Punkten erfolgt gemäß der folgenden Beschreibung (Abbildung 4.11):

Ausgestaltung des E/E-Systemmodells:

Strukturierung E/E-Systeme: In Anbetracht der Anzahl der E/E-Systeme und der unterschiedlichen Charakteristik der einzelnen E/E-Architektur-Domänen werden die E/E-Systeme in E/E-Domänen zusammengefasst. Die E/E-Systeme an sich haben mehrere konkrete Varianten, die auch die Ausstattung oder die Generation widerspiegeln können.

Integration Konfigurationsmodell: Um die E/E-Systeme zur Steuerung des Variantenmanagements zu verwenden, werden die Beziehungen zwischen den E/E-Systemen im Konfigurationsmodell hinterlegt. Zwei E/E-Systeme können sich hierbei ausschließen oder bedingen. Gleiches gilt für E/E-System-Varianten. Auch Vermischungen sind durchaus möglich,

d. h. eine E/E-System-Variante erfordert oder bedingt ein E/E-System und umgekehrt.

Weiterentwicklung E/E-Modulmodell: Das E/E-Modulmodell wird weiterentwickelt um es im E/E-Systemmodell einzubinden und E/E-Module im Variantenmanagement in Verbindung mit dem Leitungssatzrouter verwenden zu können. Die Abschaffung der Redundanzen wird erreicht, indem die gesamte Hardware eines E/E-Moduls in einem E/E-Modul-Hardware-Paket zusammengefasst wird. Das E/E-Modul-Hardware-Paket umfasst somit jede E/E-Modul-Variante (jeweils 100% ihrer Hardware) und so insgesamt 150% der Hardware. Die Information über die E/E-Modul-Varianten wird von der Modellierung der Hardware separiert. Die Information, welche Hardware eines E/E-Moduls zu einer E/E-Modul-Variante gehört, wird über eine Beziehung zur Hardware hergestellt und der E/E-Modul-Variante zugeordnet. Die Strukturierung der E/E-Module wird weiterhin anhand der zugehörigen E/E-Modulgruppe vorgenommen. Die Verbindung zu den E/E-Systemen erfolgt, indem eine E/E-Modul-Variante den E/E-System-Varianten zugeordnet wird.

Integration Funktionen-Modell: Die Anteile einer E/E-System-Variante auf der Konzeptionellen Funktionsarchitektur-Ebene wird über das Funktionen-Modell erfasst. Die Funktion des Funktionen-Modells erfasst ihre Umfänge auf der Konzeptionellen Funktionsarchitektur-Ebene. Somit wird die Funktion des Funktionen-Modells der entsprechenden E/E-System-Varianten zugeordnet.

Anbindung Funktionsnetz: Die Anteile der E/E-System-Variante im Funktionsnetz werden zugeordnet, indem die Funktionsbeiträge der einzelnen Funktionen der E/E-System-Variante zugeordnet werden.

Weiterentwicklung Propagation: Um die E/E-Systeme im Variantenmanagement nutzen zu können, wird auch die Propagation (Definition 3.1), mit der eine E/E-Architektur-Variante definiert wird, weiterentwickelt: Als Basis für die weiterentwickelte, gestufte Propagation dienen die E/E-System-Varianten, die einer E/E-Architektur-Variante zugeordnet werden. Die zugeordneten E/E-System-Varianten bilden den Ausgangspunkt für die Propagation. Von ihr aus werden die Anteile der E/E-System-Variante im Funktionsnetz propagiert. Anschließend wird die Hardware über die der E/E-System-Variante zugeordnete E/E-Modul-Variante propagiert.

Mit dem so konzipierten E/E-Systemmodell werden alle betrachteten Modellierungsebenen überspannt:

- Die Konzeptionelle Funktionsarchitektur-Ebene ist über das Funktionen-Modell einbezogen,
- das Funktionsnetz über die Funktionsbeiträge und
- die Hardware-Architektur über E/E-Modul-Varianten.

Die einzelnen E/E-Systeme und E/E-System-Varianten sind auf jeder Modellierungsebene voneinander abgegrenzt:

- Auf der Konzeptionelle Funktionsarchitektur-Ebene kann für jede beliebige Menge an Elementen eine Funktion im Funktionen-Modell erstellt werden und diese einer E/E-System-Variante zugeordnet werden.
- Im Funktionsnetz ist eine Zuordnung einer Funktion zu einem E/E-System nicht ausreichend um kenntlich zu machen, welche Teile der Funktionen zu welchen E/E-Systemen gehören. Mit der Zuordnung der Funktionsbeiträge zu den E/E-System-Varianten wird die korrekte Abgrenzung zu anderen E/E-Systemen und -Varianten sichergestellt.
- Für die Hardware-Architektur sind alle verwendeten Kombinationen der Hardware über die E/E-Modul-Varianten erfasst. Den E/E-System-Varianten sind verwendete E/E-Modul-Varianten zugeordnet. Andere E/E-Systeme und deren E/E-System-Varianten verwenden andere E/E-Module oder E/E-Modul-Varianten. Somit ist in der Hardware-Architektur sichergestellt, dass die E/E-Systeme korrekt voneinander abgegrenzt werden.

4.4.4 Entwurf des Teilkonzepts zur Wiederverwendung von E/E-Systemen

Im nachfolgenden Abschnitt wird das Teilkonzept beschrieben, das die Wiederverwendung von E/E-Systemen ermöglichen soll (Anforderung 4.4). Hierbei soll das Konzept auf dem erweiterten Produktlinienansatz der E/E-Architekturentwicklung und des produktiven Wiederverwendungsmechanismus des E/E-Architekturmodellierungswerkzeugs basieren (Designentscheidung 4.5).

Um ein E/E-System wiederzuverwenden, soll ein mit dem produktiven Wiederverwendungsmechanismus erstellter E/E-System-Modellcontainer benutzt werden. Dabei soll der E/E-System-Modellcontainer alle Modellartefakte umfassen, die einem E/E-System zugeordnet sind (Definition 4.11).

Definition 4.11 (E/E-System-Modellcontainer): Der E/E-System-Modellcontainer ist ein Modellartefakt, das im E/E-Architekturmodell

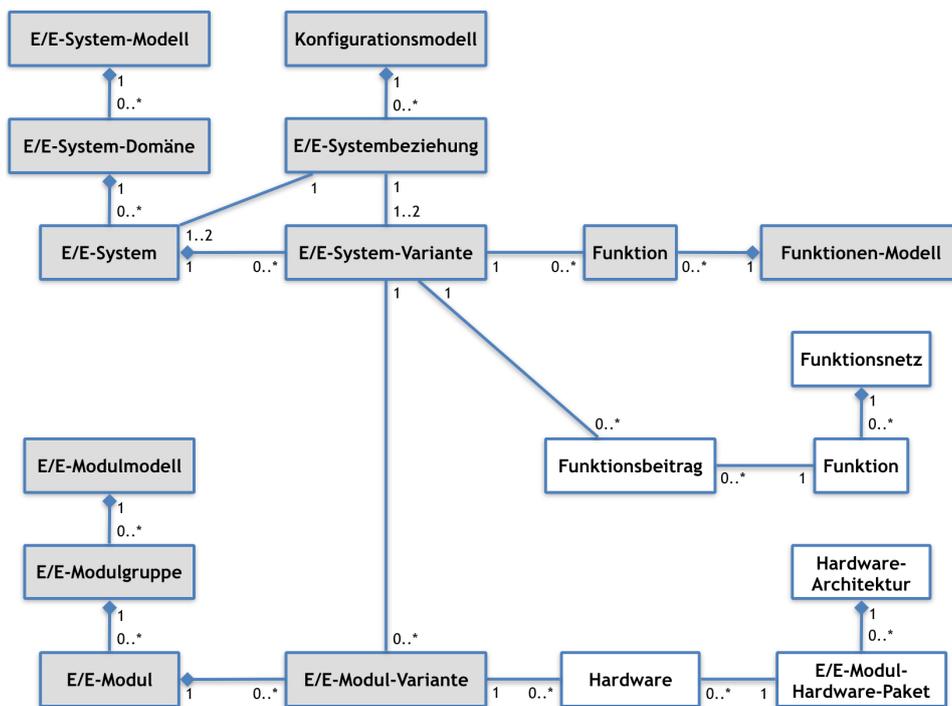


Abbildung 4.11: E/E-Systemmodell und Konfigurationsmodell. Die E/E-Systeme besitzen E/E-System-Varianten und sind in E/E-System-Domänen als Strukturierungselemente zusammengefasst. Die Beziehung zwischen den E/E-Systemen wird im Konfigurationsmodell erfasst. Das Funktionenmodell erfasst die Funktionen der Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 4.4.2). Funktionen des Funktionenmodells und Funktionen des Funktionsnetzes sind E/E-System-Varianten zugeordnet. Die Zuordnung der Hardware erfolgt über das angepasste E/E-Modulmodell, das die E/E-Module mit der zugeordneten Hardware aus der Hardware-Architektur erfasst. Grau gefärbte Blöcke stellen Erweiterungen durch das Teilkonzept dar; weitere Beziehungen zwischen den Modellartefakten, z. B. das Mapping zwischen den Funktionsbeiträgen und der Hardware, sind wegen der Übersichtlichkeit nicht Bestandteil der Abbildung.

hinterlegt wird und alle Modellartefakte beinhaltet, die zu dem E/E-System gehören.

Ein Modellartefakt, das dem E/E-System-Modellcontainer zugewiesen wird, kann hierbei Bestandteil mehrerer E/E-Systeme sein. Dies erschwert insbesondere die Bewertung der Evolution von E/E-Systemen (Abschnitt 4.2.2). Wegen dieser Mehrfachzuordnung müssen bei der Wiederverwendung von E/E-Systemen die Modellartefakte entsprechend unterschieden werden: Einige Modellartefakte, wie z. B. ein Funktionsbeitrag einer Funktion des betrachteten E/E-Systems, sind E/E-System-spezifisch (Definition 4.12). Andere Modellartefakte, wie z. B. ein Bussystem, sind E/E-System-unspezifisch (Definition 4.13).

Definition 4.12 (E/E-System-spezifische Modellartefakte): E/E-System-spezifische Modellartefakte sind genau die Modellartefakte, die ausschließlich einem E/E-System zugeordnet sind und bei einer Wiederverwendung des E/E-Systems zwingend in das Ziel-E/E-Architekturkonzept integriert werden müssen.

Definition 4.13 (E/E-System-unspezifische Modellartefakte): E/E-System-unspezifische Modellartefakte sind die Modellartefakte, die von mehreren E/E-Systemen verwendet werden und deren Integration bei der Wiederverwendung des E/E-Systems optional ist, da sie bereits durch die Integration eines weiteren E/E-Systems vorhanden sein können.

Die Wiederverwendung eines E/E-Systems erfolgt über den E/E-System-Modellcontainer gemäß der folgenden Vorgehensbeschreibung (Abbildung 4.12):

Wiederverwendung eines E/E-Systems in einem E/E-Architektur-Konzept:

Vorbereitung der Wiederverwendung: Im ersten Schritt wird für die Modellartefakte des wiederzuverwendenden E/E-Systems ein E/E-System-Modellcontainer erstellt. Mithilfe des E/E-Systemmodells wird dabei entschieden, ob es sich um ein E/E-System-spezifisches Modellartefakt oder um ein E/E-System-unspezifisches Modellartefakt handelt. Ein E/E-System-spezifisches Modellartefakt wird dadurch erkannt, dass es im E/E-Systemmodell genau einem E/E-System zugeordnet ist. Ein E/E-System-unspezifisches Modellartefakt wird hingegen dadurch erkannt, dass es im E/E-Systemmodell mehr als einem E/E-System zugeordnet ist.

Integration des E/E-Systems: Im zweiten Schritt wird der E/E-System-Modellcontainer in das E/E-Architektur-Konzept, in dem die Wiederverwendung des E/E-Systems erfolgen soll, eingebunden. Dies ermöglicht,

die in ihm enthaltenen Modellartefakte in das E/E-Architektur-Konzept zu übernehmen. Dabei werden die E/E-System-spezifischen Modellartefakte zwingend integriert, da sie nicht durch die Wiederverwendung eines weiteren E/E-System Teil des E/E-Architektur-Konzepts werden können. Die Integration der E/E-System-unspezifischen Modellartefakte in das E/E-Architektur-Konzept kann bereits durch die Wiederverwendung eines weiteren E/E-System erfolgt sein. Deshalb wird das zu integrierende E/E-System-unspezifische Modellartefakt mit dem vorhandenen Modellartefakt verschmolzen und somit als gleichartig definiert.

Austausch des E/E-Systems: Findet eine Evolution des E/E-Systems statt, kann ein Austausch der bereits wiederverwendeten Modellartefakte des E/E-Systems notwendig sein. Um die neue Version des E/E-Systems im E/E-Architektur-Konzept wiederzuverwenden, wird zuerst der E/E-System-Modellcontainer aktualisiert. Dazu wird, wie im ersten Schritt beschrieben, vorgegangen. Anschließend wird der im E/E-Architektur-Konzept eingebundene und veraltete E/E-System-Modellcontainer durch seine neue Version ersetzt. Abschließend werden die enthaltenen Modellartefakte, wie im zweiten Schritt beschrieben, erneut integriert.

4.4.5 Anwendungsfälle für das E/E-System Produktlinien Engineering

In diesem Abschnitt werden für die beschriebenen Teilkonzepte Anwendungsfälle entworfen, die den Anwender beim Einsatz des E/E-System Produktlinien Engineerings unterstützen sollen.

Relation zwischen Konzeptioneller Funktionsarchitektur- und Funktionsnetz-Ebene

Die Konzeptionelle Funktionsarchitektur-Ebene beschreibt die Funktionen auf der Funktionsnetz-Ebene in einer abstrahierten Form (Abschnitt 4.4.1). Im E/E-Architekturmodell kann es somit eine detaillierte und eine abstrakte Beschreibung für die selben Schnittstellen geben. Um eine Nachverfolgbarkeit herzustellen, muss eine Ebenenrelation zwischen der abstrakten und der detaillierten Beschreibung der Schnittstellen modelliert werden. Dies erfolgt im Relationsmodell.

Allerdings umfassen gerade die Schnittstellen einen großen Umfang in den zu einem E/E-System gehörenden E/E-Architekturmodell-Artefakten (Abschnitt 3.2.3). Aus diesem Grund ist eine manuelle Modellierung dieser Ebenenrelationen mit einem hohen Aufwand verbunden. Dementsprechend soll

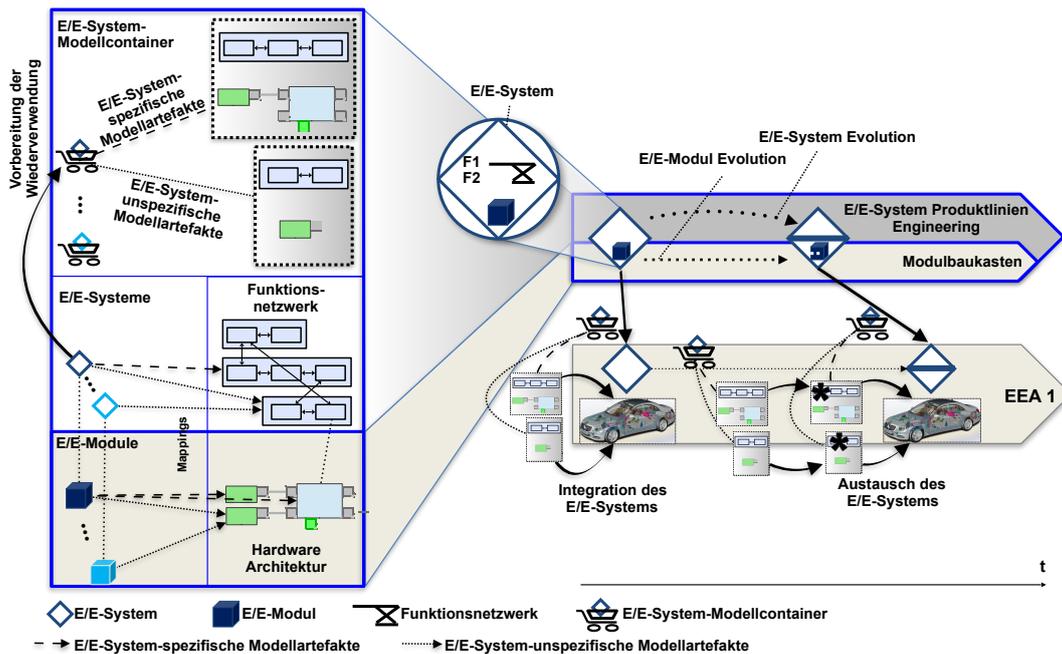


Abbildung 4.12: Konzept zur Wiederverwendung von E/E-Systemen.

Für die Wiederverwendung von E/E-Systemen wird zuerst ein E/E-System-Modellcontainer erstellt (Vorbereitung der Wiederverwendung). Grundlage sind die Zuordnungen (Mappings) zwischen E/E-System und Modellartefakten. So werden dem E/E-System-Modellcontainer E/E-System-spezifische und -unspezifische Modellartefakte zugeordnet. Ein E/E-System wird durch Einbindung des E/E-System-Modellcontainers und Integration der E/E-System-spezifischen und -unspezifischen Modellartefakte integriert. Ein E/E-System wird durch Aktualisierung des E/E-System-Modellcontainers und Integration der Modellartefakte ausgetauscht. EEA E/E-Architektur.

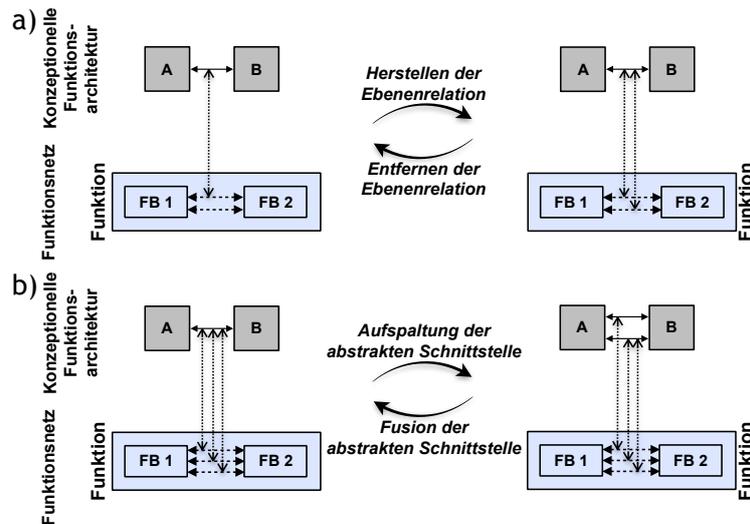


Abbildung 4.13: Anwendungsfälle für die Ebenenrelation (Anwendungsfälle 4.1, 4.2). a) Herstellen und Entfernen der Ebenenrelation (Anwendungsfall 4.1), b) Evolution der Ebenenrelation (Anwendungsfall 4.1)

mit dem nachfolgenden Anwendungsfall eine schnelle und effiziente Erstellung dieser Ebenenrelation ermöglicht werden (Abbildung 4.13 a)).

Anwendungsfall 4.1 (Herstellen und Entfernen der Ebenenrelation): Automatisiertes Herstellen und Entfernen einer Relation zwischen den abstrakten Schnittstellen der Konzeptionellen Funktionsarchitektur-Ebene und den detaillierten Schnittstellen des Kommunikationsmodells der Funktionsnetz-Ebene.

Nachdem diese Ebenenrelation hergestellt ist, unterliegt sie auch der Evolution der E/E-Systeme im E/E-Architekturmodell. Deshalb kann es notwendig werden, diese Ebenenrelation anzupassen, wenn die Schnittstellen der E/E-Systeme weiterentwickelt werden. Insbesondere kann hierbei eine Schnittstelle aufgespalten werden (Abbildung 4.13 b)). Somit entstehen

- zwei abstrakte Schnittstellen,
- zwei Gruppen von detaillierten Schnittstellen und
- zwei Ebenenrelationen.

Letztere verbinden jeweils die abstrakte Schnittstelle mit den detaillierten Schnittstellen. Findet eine Evolution in umgekehrter Abfolge statt, wird von einem Fusionieren gesprochen (Abbildung 4.13 b)).

Der nachfolgende Anwendungsfall soll deshalb ein Vorgehen für diese Evolution der Ebenenrelation bereitstellen um den manuellen Modellierungsaufwand zu reduzieren.

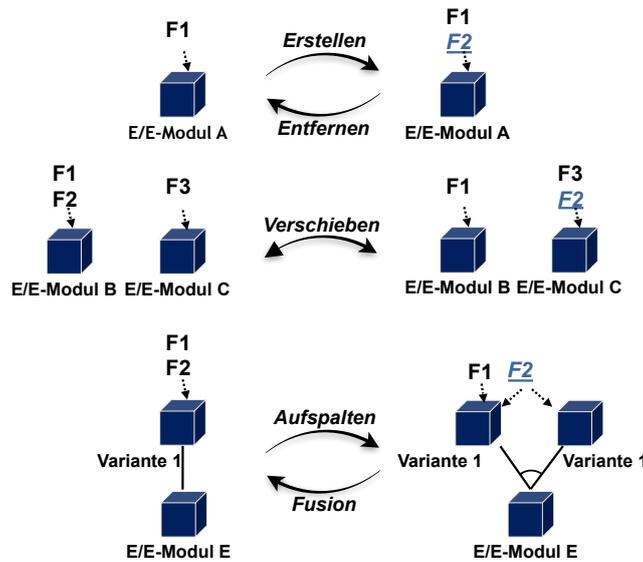


Abbildung 4.14: Anwendungsfälle für die Evolution von E/E-Systemen (Anwendungsfälle 4.3, 4.4, 4.5, 4.6, 4.7). Die Evolution der E/E-Systeme kann in den fünf Arten auftreten. In der Abbildung wird auf E/E-Modul A der Funktionsumfang F2 erstellt bzw. entfernt; der Funktionsumfang F2 von E/E-Module B auf E/E-Module C verschoben; und der Funktionsumfang F2 auf die Variante 2 des E/E-Moduls E aufgespalten bzw. auf die Variante 1 fusioniert. Vgl. [124]. Nachgedruckt, adaptiert, übersetzt mit Genehmigung von Springer. F Funktionsumfang

Anwendungsfall 4.2 (Evolution der Ebenenrelation): Automatisierte Evolution der Relation zwischen der Konzeptionellen Funktionsarchitektur-Ebene und Funktionsnetz-Ebene einschließlich der Aufspaltung einer abstrakten Schnittstelle und der Fusion mehrerer abstrakter Schnittstellen.

Evolution von E/E-Systemen

Eine Analyse von Beispielen der Evolution von E/E-Systemen macht mehrere Muster deutlich (vgl. [124, 125], Abbildung 4.14). Diese werden hier zu mehreren Anwendungsfällen zusammengefasst um den E/E-Architekten und den Modellierern einen geordneten Rahmen und eine Beschreibungsform zu geben, in dem die Evolution gerichtet durchgeführt werden kann und mit dem entsprechenden Teilkonzept dieser Arbeit praktisch handhabbar ist (Abschnitt 4.4.2). Es ist hierbei durchaus möglich, dass mehrere Anwendungsfälle nacheinander ausgeführt werden um zum beabsichtigten Ergebnis zu gelangen.

Bei einer neuen Fahrzeuggeneration erhöht sich im Allgemeinen der Funktionsumfang. Bei dieser Evolution von E/E-Systemen können neue Funktionen oder neue Funktionsbeiträge für vorhandene Funktionen eingeführt werden. In der E/E-Architekturmodellierung müssen deshalb diese neuen Anteile als neue

Modellartefakte erstellt werden damit sie abgesichert werden können. Dementsprechend müssen diese Anteile in die bestehende E/E-Architekturmodell-Struktur eingefügt werden. Hierbei haben sie eine Beziehung zu dem E/E-Modul, auf dem diese neuen Anteile ausgeführt werden. Durch eine Automatisierung soll der für diesen Vorgang notwendige Modellierungsaufwand reduziert werden (Anwendungsfall 4.3).

Anwendungsfall 4.3 (Erstellen): Erstellen von neuen Funktionsumfängen für bereits bestehende E/E-Module.

Umgekehrt können Funktionen abgekündigt werden oder in größere Funktionsumfänge integriert werden, die z.B. von einem Zulieferer entwickelt werden. Beispielsweise wurde so die Tempomat-Funktion in das ESP integriert. In beiden Fällen kann es für die E/E-Architektur-Entwicklung bedeuten, dass die Funktionen nicht mehr explizit modelliert werden müssen. Das Entfernen der Funktion von einem bestehenden E/E-Modul soll somit automatisiert werden (Anwendungsfall 4.4).

Anwendungsfall 4.4 (Entfernen): Entfernen von Funktionsumfängen von bereits bestehenden E/E-Modulen.

Eine weitere Form der E/E-System-Evolution tritt sehr häufig bei Optimierungen auf. Hier kann es zu Funktionsverlagerungen kommen, zum Beispiel im Zuge von Hochintegrationen. Der entstehende Modellierungsaufwand beim Verschieben von Funktionsumfängen auf andere E/E-Module soll durch die Automatisierung verringert werden (Anwendungsfall 4.5).

Anwendungsfall 4.5 (Verschieben): Verschiebung von Funktionsumfängen zu anderen, bereits bestehenden E/E-Modulen.

Allerdings können auch neue Varianten für bestehende E/E-Module entwickelt werden. Meist um

- neue Käufergruppen zu erschließen (z.B. im preissensitiven Kompaktsegment),
- vergleichbaren Angeboten aus dem Wettbewerb ein entsprechendes Angebot entgegenzustellen oder
- kostengünstigere Zwischenlösungen anzubieten, die aufgrund von Optimierungen, Technologieentwicklungen oder Angeboten vom Zulieferer neu ermöglicht wurden.

Hierbei wird oft nur ein Teil der Funktion, die auf der bereits bestehenden Variante implementiert ist, übernommen. Somit kann es zu einer Abspaltung eines Teil-Funktionsumfangs kommen. In der E/E-Architektur-Entwicklung

muss diese Evolution des E/E-Systems frühzeitig abgesichert werden. Aus diesem Grund soll die Modellierung möglichst aufwandsarm durchgeführt werden können (Anwendungsfall 4.6).

Anwendungsfall 4.6 (Aufspalten): Abspalten eines Teil-Funktionsumfangs einer Funktion für eine neue E/E-Modul-Variante, die diesen Teil-Funktionsumfang erhält.

Umgekehrt können Varianten von E/E-Modulen auch reduziert werden. In diesem Fall soll der Teil-Funktionsumfang dieser Variante mit der Funktion einer bestehenden Variante fusioniert werden. Auch hier soll der Modellierungsaufwand durch eine Automatisierung reduziert werden (Anwendungsfall 4.7).

Anwendungsfall 4.7 (Fusion): Fusionierung einer Funktion einer E/E-Modul-Variante, die reduziert werden soll, mit einer bestehenden Funktion einer E/E-Modul-Variante, die fortgeführt wird.

Variantenkonfiguration mit E/E-Systemmodellen

Bei der Konfiguration einer E/E-Architektur-Variante, muss für jedes E/E-Architekturmodell-Artefakt entschieden werden, ob es in einer E/E-Architektur-Variante enthalten ist oder nicht (Abschnitt 3.2.4). Für einen großen Teil der Modellartefakte wird dies durch die Propagationsregeln automatisch durchgeführt. Allerdings muss der E/E-Architekt diese Entscheidung für jeden Funktionsbeitrag einer Funktion auf der Funktionsnetz-Ebene durchführen. Dies ist sehr aufwändig, da es eine hohe Anzahl von Funktionsbeiträgen mit komplexen Abhängigkeiten gibt. Der E/E-Architekt muss dabei seine Kenntnis der Funktionen und sein implizites Wissen anwenden. Auch der Modellierungsaufwand ist umfangreich, da für jede E/E-Architekturmodell-Variante diese Entscheidung modelliert werden muss.

Da die Funktionsbeiträge durch Konzepte dieser Arbeit zu E/E-Systemen zuordenbar sind, kann der Konfigurationsvorgang erleichtert werden (Abschnitt 4.4.3). Dazu soll für ganze E/E-Systeme über eine Konfigurationstabelle entschieden werden, ob sie in einer E/E-Architektur-Variante enthalten sind. Hierbei erfasst das Konfigurationsmodell die Abhängigkeiten zwischen den E/E-Systemen, um das implizite Wissen im E/E-Architekturmodell abzubilden. Dieses Vorgehen soll automatisiert werden (Anwendungsfall 4.8).

Anwendungsfall 4.8 (Variantenkonfiguration mit E/E-Systemmodellen): Konfiguration einer E/E-Architekturmodell-Variante unter Nutzung des vollständigen E/E-Systemmodells und des Konfigurationsmodells.

Wiederverwendung von E/E-Systemen

Für die heutige E/E-Architekturmodellierung gibt es Konzepte, die eine Wiederverwendung von E/E-Modulen erlauben [81, 82, 83, 84]. Jedoch werden in dieser Arbeit zusätzlich die Funktionen des Funktionsnetzes betrachtet. Dieses wird heute separat für die E/E-Architekturen modelliert, in der die Hardware wiederverwendet wird (Abschnitt 4.1). Eine gemeinsame Wiederverwendung hätte den Vorteil, dass der zusätzliche Aufwand durch die separate Modellierung entfällt. Die nachfolgenden Anwendungsfälle sollen die notwendigen Schritte zur gemeinsamen Wiederverwendung, gemäß des Teilkonzepts dieser Arbeit (Abschnitt 4.4.4), beschreiben.

Zuerst muss für das betrachtete E/E-System ein E/E-System-Modellcontainer für die E/E-Architekturmodell-Artefakte definiert werden. Dieser E/E-System-Modellcontainer stellt die Grundlage für die Wiederverwendung des E/E-Systems dar. Allerdings ist mit der Erstellung des E/E-System-Modellcontainers ein gewisser Modellierungsaufwand verbunden: Es muss schon bei der Definition des E/E-System-Containers für jedes Modellartefakt entschieden werden, ob es später zwingend integriert werden muss. Wenn ja, dann handelt es sich um ein E/E-System-spezifisches Modellartefakt (z.B. ein Funktionsbeitrag einer Funktion des betrachteten E/E-Systems). Andernfalls handelt es sich um ein E/E-System-unspezifisches Modellartefakt, das für mehrere E/E-Systeme notwendig ist (z.B. ein Bussystem).

Wenn ein E/E-System-Modellcontainer erstellt ist, kann er anschließend in ein E/E-Architektur-Konzept integriert werden. Hierbei muss jedes E/E-System-spezifische Modellartefakt zwingend übernommen werden. Bei der Integration muss zuerst entscheiden werden, wo es in der vorhandenen Modellstruktur integriert werden soll. Anschließend muss es manuell aus dem E/E-System-Modellcontainer heraus an die entsprechende Stelle platziert werden. Für jedes E/E-System-unspezifische Modellartefakt muss entschieden werden, ob schon ein passendes Modellartefakt in der Produktlinie enthalten ist. Falls ja, dann wird das Modellartefakt mit dem vorhandenen Modellartefakt verschmolzen. Andernfalls wird es wie ein E/E-System-spezifisches Modellartefakt integriert.

Diese beiden aufwändigen Vorgänge sollen automatisiert werden (Anwendungsfall 4.9).

Anwendungsfall 4.9 (Wiederverwendung von E/E-Systemmodellen): Automatisierte Erstellung eines E/E-System-Modellcontainers dynamisch für ein gegebenes E/E-System als Grundlage für die Wiederverwendung der Modellartefakte eines E/E-Systems und automatisierte Integration aller Modellartefakte eines E/E-Systemmodellcontainers in ein gegebenes E/E-Architektur-Konzept.

Die E/E-Systeme unterliegen jedoch einer ständigen Weiterentwicklung, die auch Auswirkung auf die Wiederverwendung eines E/E-Systems haben kann. Wenn eine Änderung an einem E/E-System auch in das E/E-Architektur-Konzept übernommen werden soll, müssen die Modellartefakte auf den neusten Stand aktualisiert werden. Dieser Vorgang, in dem die veralteten Modellartefakte durch die Neueren ausgetauscht werden, soll automatisiert werden (Anwendungsfall 4.10).

Anwendungsfall 4.10 (Austausch von E/E-Systemmodellen): Automatisierter Austausch der Modellartefakte eines integrierten E/E-Systems nach einer Änderung im Systembaukasten.

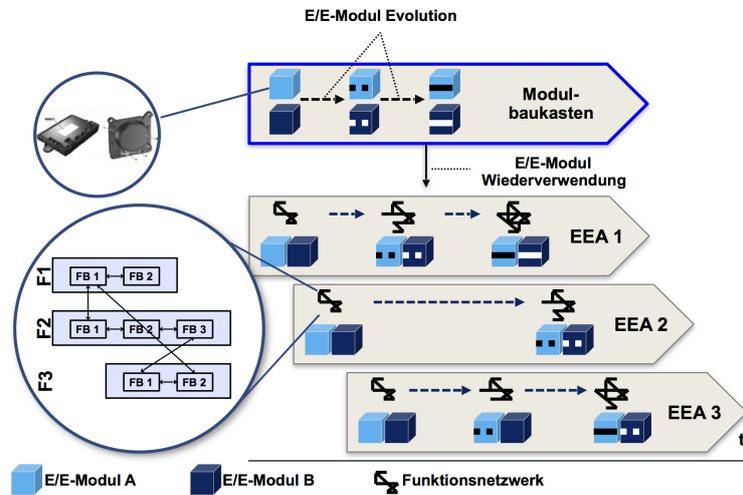
4.5 Zusammenfassung

In Kapitel 4 wird das E/E-System Produktlinien Engineering als Erweiterung der heutigen E/E-Architekturmodellierung konzeptioniert (vgl. Abbildung 4.1). Dafür wird der Stand der Technik bewertet und Anforderungen aus dieser Bewertung (Abschnitt 4.1) und der Analyse der E/E-Architekturmodellierung (Abschnitt 4.2) abgeleitet. Nach einer Analyse von Lösungsansätzen und Ableitung von Designentscheidungen (Abschnitt 4.3) werden die verschiedenen Konzepte und Anwendungsfälle des E/E-System Produktlinien Engineerings beschrieben (Abschnitt 4.4). Nachfolgend wird die Konzeption der einzelnen Aspekte kurz zusammengefasst und veranschaulicht (Abbildung 4.15).

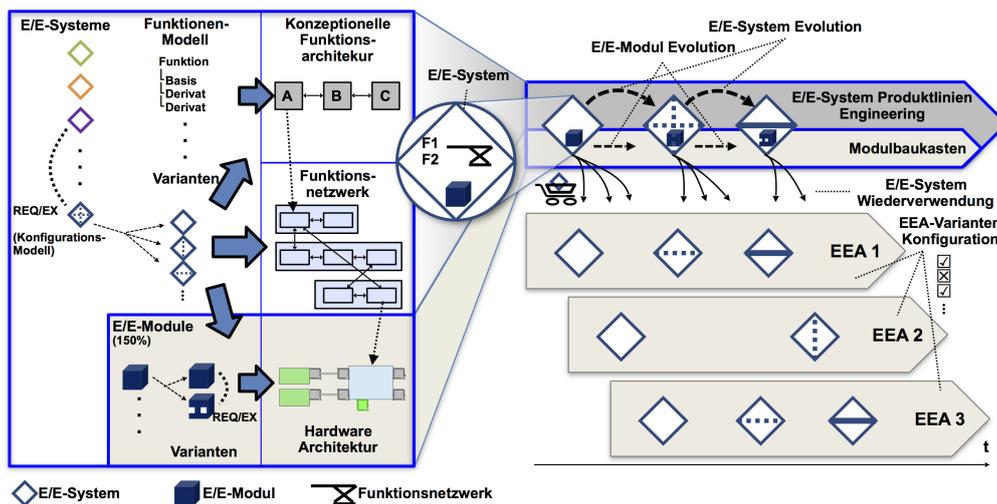
Evolution von E/E-Systemen In der heutigen E/E-Architekturmodellierung existiert kein Vorgehen um die E/E-System-Evolution vollständig zu bewerten (Abschnitt 4.1.1). Deshalb werden bei der Evolution die Ziele der E/E-Architekturmodellierung nicht erfüllt, weshalb heute auf ein Vorgehen nach *Best Engineering Practice* zurückgegriffen werden muss. Die Ursache dafür liegt in den komplexen Abhängigkeiten zwischen den E/E-Modulen, die durch die Funktionen hervorgerufen werden.

In dieser Arbeit wurde für die Erfassung dieser Abhängigkeiten die Konzeptionelle Funktionsarchitektur-Ebene als abstrakte Beschreibung der Funktionen eingeführt (Abschnitt 4.4.1). Damit soll die Bewertung einer Evolution als auch die Konkretisierung einer Innovation ermöglicht werden. Die Konzeptionelle Funktionsarchitektur-Ebene und das Funktionsnetz werden über die Modellierung von Relationen in Beziehung zueinander gesetzt.

Im Anschluss an eine Bewertung einer Evolution eines E/E-Systems muss diese in der E/E-Architekturmodellierung umgesetzt werden. In der heutigen E/E-Architekturmodellierung wird die Evolution separat für die jeweiligen E/E-Architekturmodelle umgesetzt (Abbildung 4.15a).



(a) Heutige E/E-Architekturmodellierung



(b) E/E-System Produktlinien Engineering

Abbildung 4.15: Erweiterung der heutigen E/E-Architekturmodellierung zum E/E-System Produktlinien Engineering. Um die Evolution von E/E-Modulen zu bewerten wird die heutige E/E-Architekturmodellierung um die Konzeptionelle Funktionsarchitektur-Ebene erweitert (Abschnitt 4.4.1). Die Einführung eines E/E-Systemmodells (Abschnitt 4.4.3) erneuert die Beschreibung von E/E-Modulen zu einer 150%-Repräsentation und erlaubt die vollständige Beschreibung eines E/E-Systems. Die E/E-System Evolution wird durch das Feature Oriented Architecture Refactoring (Abschnitt 4.4.2) umgesetzt. Die Einführung eines vollständigen E/E-Systemmodells erlaubt die zentrale Wiederverwendung von E/E-Systemen (Abschnitt 4.4.4). Vgl. [125]. EEA E/E-Architektur; F Funktion; FB Funktionsbeitrag

In dieser Arbeit wurde für die zentrale Umsetzung der Evolution von E/E-Systemen (Abbildung 4.15b) das Feature Oriented Architecture Refactoring (Abschnitt 4.4.2) konzipiert. Kern dafür ist ein umfassendes Funktionen-Modell einschließlich mehrerer Operationen auf diesem Modell.

Modellstruktur und Variantenmanagement für E/E-Systeme In der heutigen E/E-Architekturmodellierung sind lediglich Funktionen und E/E-Module als Teilaspekte von E/E-Systemen erfasst (Abschnitt 4.1.2, Abbildung 4.15a). Jede Variante eines E/E-Moduls wird dabei einzeln modelliert, was zu einer Vervielfältigung von gemeinsam genutzten Modellartefakten führt.

In dieser Arbeit wurde für die zentrale Verwaltung von E/E-Systemen ein E/E-Systemmodell entwickelt (Abschnitt 4.4.3), das eine vollständige Erfassung aller Aspekte eines E/E-Systems vorsieht (Abbildung 4.15b). Dazu gehört

- eine Anpassung der Repräsentation der E/E-Module zu einer 150%-Repräsentation,
- die Erweiterung des Hardware-Modulbaukastens auf die Software-Umfänge einschließlich des Funktionsnetzes und der Konzeptionellen Funktionsarchitektur-Ebene,
- die Zuordnung von Funktionen und E/E-Modulen zu E/E-Systemen,
- die Erfassung der Abhängigkeiten zwischen den E/E-Systemen als auch
- die Steuerung von E/E-Architekturmodell-Varianten über E/E-Systeme.

Wiederverwendung von E/E-Systemen In der heutigen E/E-Architekturmodellierung existiert kein Konzept zur Wiederverwendung von E/E-Systemen, was in einer heterogenen und mehrfachen Modellierung der Funktionsnetz-Umfänge resultiert (Abschnitt 4.1.3, Abbildung 4.15a).

Für die Ablösung dieses Vorgehens wurde in dieser Arbeit die Wiederverwendung für E/E-Systeme (Abschnitt 4.4.4) konzipiert. Durch die vollständige Erfassung von E/E-Systemen wird eine zentrale und gekoppelte Wiederverwendung von E/E-Modulen und Funktionen ermöglicht (Abbildung 4.15b).

Kapitel 5

Prototypische Umsetzung des E/E-System Produktlinien Engineerings

In diesem Kapitel wird die in Kapitel 4 beschriebene Konzeption des E/E-System Produktlinien Engineerings umgesetzt. Die nachfolgenden Abschnitte umfassen somit die Umsetzung für die Evolution (Abschnitt 5.1), für die Modellstruktur und das Variantenmanagement (Abschnitt 5.2) und für die Wiederverwendung (Abschnitt 5.3) von E/E-Systemen. Wie das E/E-System Produktlinien Engineering im Werkzeug für die E/E-Architekturmodellierung (Abschnitt 3.2.5) umgesetzt wird, ist in Abschnitt 5.4 beschrieben.

5.1 Evolution von E/E-Systemen

Für die Umsetzung der Evolution von E/E-Systemen im Rahmen des E/E-System Produktlinien Engineerings ist die eingeführte konzeptionelle Funktionsarchitektur-Ebene einschließlich des Relationsmodells (Abschnitt 4.4.1) und das Feature Oriented Architecture Refactoring (FOAR, Abschnitt 4.4.2) von zentraler Bedeutung. Relevant ist auch das eingeführte E/E-Systemmodell (Abschnitt 4.4.3), das bei einer E/E-System-Evolution entsprechend angepasst wird.

Den eingeführten Teilkonzepten entsprechend umfasst das Vorgehen für die Anwendungsfälle der Evolution von E/E-Systemen allgemein (Abbildung 5.1):

- die Erfassung der von der Evolution betroffenen Funktionen,
- die Auswahl des passenden Anwendungsfalles für die E/E-Systemevolution (Anwendungsfälle 4.3, 4.4, 4.5, 4.6 und 4.7),

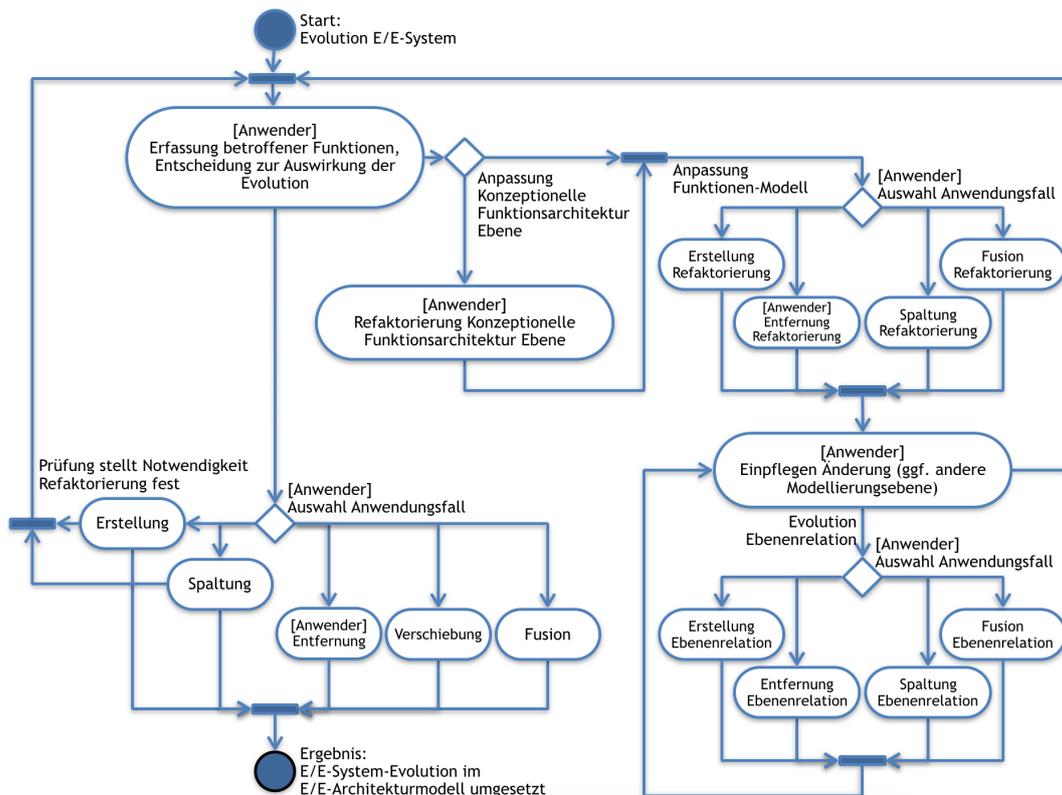


Abbildung 5.1: Aktivitätsdiagramm für das allgemeine Vorgehen bei den Anwendungsfällen zur Evolution (Anwendungsfälle 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7). Das Aktivitätsdiagramm beschreibt das Vorgehen und die Benutzerinteraktion für die Anwendungsfälle bei der Evolution von E/E-Systemen.

- falls sich Funktionsumfänge oder deren Zuordnungen ändern, die Durchführung der Refaktoriierung der Konzeptionellen Funktionsarchitektur-Ebene,
- die anschließende Anpassung des Funktionenmodells,
- das Einpflegen der Änderung, die auch auf einer anderen Modellierungsebene erfolgen kann. Hierzu gehört auch die Evolution der Ebenenrelation (Anwendungsfall 4.2).

Bei der Refaktoriierung der Konzeptionellen Funktionsarchitektur werden die Elemente und abstrakten Schnittstellen entfernt, verändert oder neue hinzugefügt. Bei der ggf. anschließenden Anpassung des Funktionenmodells werden Basis- und Derivat-Beiträge für die Funktionen erstellt. Basis-Beiträge enthalten die durch die Funktion hinzugefügten funktionalen Anteile und Derivat-Beiträge enthalten die Umfänge von Basis-Beiträgen anderer Funktionen. Gemäß des FOAR (Abschnitt 4.4.2) werden die Anpas-

sungen an Basis-Beiträgen ggf. auf einer anderen Modellierungsebene durchgeführt. Dazu gehört auch die Verbindung zwischen den Modellierungsebenen, wie z. B. die Relation der abstrakten Schnittstellen auf der Konzeptionellen Funktionsarchitektur-Ebene zu den detaillierten Schnittstellen des Funktionsnetzes (Abschnitt 4.4.1).

Die Automatisierung für die Durchführung der einzelnen Anwendungsfälle ist in den nachfolgenden Abschnitten detailliert. Hierbei ist zu beachten, dass die Automatisierungen generell entworfen wurden um dem Anwender ein breiteres Spektrum der Anwendungsmöglichkeit zu bieten. Es ist hierbei durchaus möglich, dass mehrere Anwendungsfälle nacheinander ausgeführt werden um zum beabsichtigten Ergebnis zu gelangen. Dies gilt insbesondere wenn mehrere Funktionen betroffen sind. Sollen z. B. mehrere Funktionen abgespalten werden, wird erst die Operation Spaltung (Abschnitt 5.1.4) ausgeführt, die eine Funktion abspaltet und die E/E-System- und E/E-Modul-Variante erstellt, und die weiteren Funktionen danach mit der Operation Verschiebung (Abschnitt 5.1.3) verschoben.

5.1.1 Operation Erstellung

Die Erstellung neuer Funktionsumfänge auf bestehenden E/E-Modulen (Anwendungsfall 4.3) umfasst mehrere Schritte und erfolgt gemäß der folgenden Vorgehensbeschreibung:

Erfassung betroffener Funktionen: Im ersten Schritt wird die Funktion identifiziert, die von den neuen Funktionsumfängen betroffen ist. Da es sich um neue Funktionsumfänge handelt, muss keine Funktion vorhanden sein und eine neue Funktion soll erstellt werden. Erweitern die neuen Funktionsumfänge eine bereits bestehende Funktion, die bereits auf der Konzeptionellen Funktionsarchitektur-Ebene und im Funktionen-Modell repräsentiert ist, wird zuerst für die neuen Funktionsumfänge eine neue Funktion erstellt. Anschließend wird eine Fusion der neu erstellten Funktion mit der bestehenden Funktion durchgeführt.

Refaktorisierung der Konzeptionellen Funktionsarchitektur-Ebene:

In diesem Schritt wird die Konzeptionelle Funktionsarchitektur-Ebene um neue Elemente und abstrakte Schnittstellen erweitert, die die neue Funktionalität repräsentieren.

Erstellung Refaktorisierung: Als nächstes wird das Funktionen-Modell mit der in Abbildung 5.2 beschriebenen Vorgehensweise um die neue Funktion erweitert. Die neue Funktion enthält die neuen Funktionsumfänge und die Elemente und abstrakten Schnittstellen mit denen die neuen Funktionsumfänge interagieren.

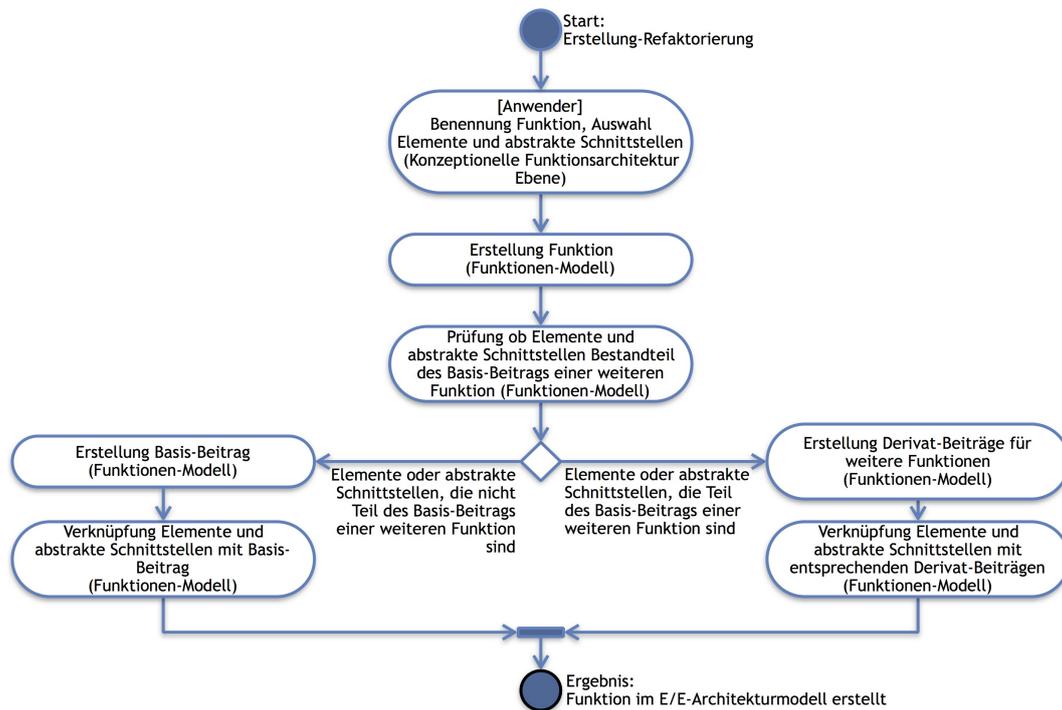


Abbildung 5.2: Aktivitätsdiagramm für die Erstellung von Funktionen (Anwendungsfall 4.3).

Einpflegen der Änderung (u. a. Erstellung Ebenenrelation): Danach werden die neuen Funktionsumfänge in weiteren Modellierungsebenen eingepflegt. Es wird das Funktionsnetz erweitert, eine Ebenenrelation mit der in Abbildung 5.3 beschriebenen Vorgehensweise erstellt (Anwendungsfall 4.1), die Hardware-Ebene und der Hardware-Modulbaukasten angepasst.

Erstellung im E/E-Systemmodell: Im letzten Schritt wird das E/E-Systemmodell angepasst, indem ein neues E/E-System mit der in Abbildung 5.4 beschriebenen Vorgehensweise angelegt wird. Falls die neuen Funktionen ein bestehendes E/E-System erweitern sollen, wird anschließend eine Fusion des neu erstellten E/E-Systems mit dem bestehenden E/E-System durchgeführt. Abschließend werden die für das E/E-System relevanten Modellartefakte der anderen Modellierungsebenen im E/E-Systemmodell (Abschnitt 5.2.2) mit dem neu erstellten E/E-System verknüpft.

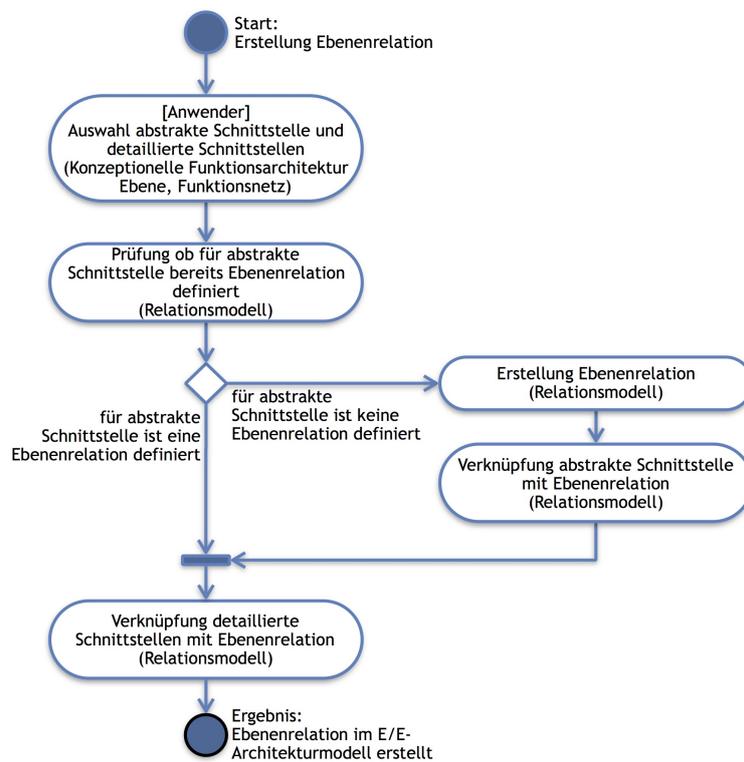


Abbildung 5.3: Aktivitätsdiagramm für die Erstellung von Ebenenrelationen (Anwendungsfall 4.1).

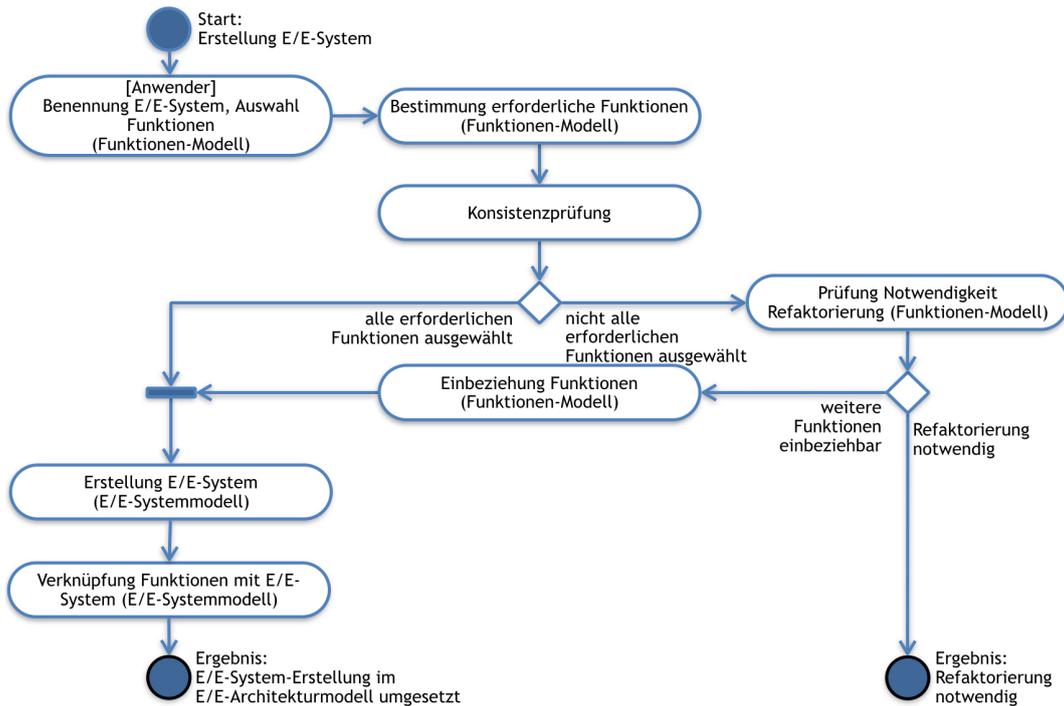


Abbildung 5.4: Aktivitätsdiagramm für die Erstellung von E/E-Systemen (Anwendungsfall 4.3).

5.1.2 Operation Entfernung

Bestehende Funktionsumfänge werden von E/E-Modulen gemäß der folgenden Vorgehensbeschreibung entfernt (Anwendungsfall 4.4):

Erfassung betroffener Funktionen: Im ersten Schritt wird die Funktion erfasst, die entfernt wird oder für die Funktionsumfänge entfallen. Soll nicht die komplette Funktion entfallen, wird zuerst eine Spaltung-Operation (Abschnitt 5.1.4) durchgeführt, so dass die zu entfallenden Funktionsumfänge in einer Funktion isoliert sind.

Refaktoriierung der Konzeptionellen Funktionsarchitektur-Ebene:

Danach werden die Funktionsumfänge, die vorher in einer Funktion isoliert wurden, aus dem Modell der Konzeptionellen Funktionsarchitektur-Ebene entfernt. Dazu werden die Elemente und abstrakte Schnittstellen der zu entfallenden Funktionalität gelöscht.

Entfernung Refaktoriierung zur Löschung der Funktion:

Anschließend wird der Funktionsumfang aus dem Funktionen-Modell entfernt. Dazu wird die Funktion, die die Funktionsumfänge isoliert, einschließlich ihres Basis-Beitrags und all ihrer Derivat-Beiträge aus dem Funktionen-Modell gelöscht. Zudem werden Derivat-Beiträge für

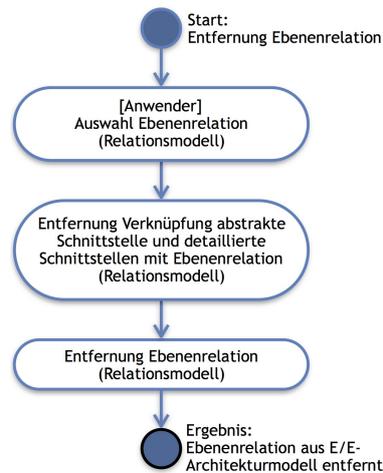


Abbildung 5.5: Aktivitätsdiagramm für die Entfernung von Ebenenrelationen (Anwendungsfall 4.1).

gelöschte Funktion entfernt, falls sie bei weiteren Funktionen existieren sollten.

Einpflegen der Änderung (u. a. Entfernung Ebenenrelation): In diesem Schritt wird der Funktionsumfang in weiteren Modellierungsebenen entfernt. Dafür werden die entsprechenden Teile des Funktionsnetzes entfernt, die Ebenenrelationen mit der in Abbildung 5.5 beschriebenen Vorgehensweise entfernt (Anwendungsfall 4.1), die Hardware-Ebene und der E/E-Modulbaukasten angepasst.

Entfernung aus E/E-Systemmodell: Zuletzt wird das E/E-Systemmodell angepasst. Dafür wird die komplette Funktion bzw. die Funktion, die die Funktionsumfänge isoliert, aus dem zugehörigen E/E-System gelöscht. Falls diese Funktion die einzige Funktion des E/E-Systems war, dann wird das ganze E/E-System aus dem E/E-Systemmodell entfernt. Abschließend werden die Verknüpfungen mit den für das E/E-System relevanten Modellartefakten der anderen Modellierungsebenen im E/E-Systemmodell (Abschnitt 5.2.2) gelöscht.

5.1.3 Operation Verschiebung

Funktionsumfänge werden zu anderen, bereits bestehenden E/E-Modulen gemäß der folgenden Vorgehensbeschreibung verschoben (Anwendungsfall 4.5):

Erfassung betroffener Funktionen: Es wird in diesem Schritt die Funktion erfasst, die verschoben wird oder deren Funktionsumfänge verschoben werden. Soll nicht eine komplette Funktion verschoben werden, wird

zuerst eine Spaltung-Operation (Abschnitt 5.1.4) durchgeführt. Damit werden die zu verschiebenden Funktionsumfänge in einer Funktion isoliert.

Refaktorisierung der Konzeptionellen Funktionsarchitektur-Ebene:

Als nächstes werden die in einer Funktion isolierten Funktionsumfänge im Modell der Konzeptionellen Funktionsarchitektur-Ebene entsprechend verschoben.

Fusion Refaktorisierung zur Verschiebung der Funktion: In diesem Schritt wird der Funktionsumfang im Funktionen-Modell verschoben. Sollen die Funktionsumfänge zu einer bestehenden Funktion verschoben werden, wird die Funktion, die die Funktionsumfänge isoliert, mit der bestehenden Funktion fusioniert. Wird eine komplette Funktion verschoben, wird in diesem Schritt keine Änderung vorgekommen, da diese bereits korrekt im Funktionen-Modell repräsentiert ist.

Einpflegen der Änderung: Als nächstes wird das Verschieben der Funktionsumfänge auf den weiteren Modellierungsebenen umgesetzt. Dementsprechend wird das Funktionsnetz angepasst und die Ebenenrelationen überprüft. Entsprechend der Änderungen auf dem Funktionsnetz wird sie angepasst. Zusätzlich dazu wird die Hardware-Ebene und der E/E-Modulbaukasten angepasst.

Entfernung aus E/E-Systemmodell: Im letzten Schritt wird das E/E-Systemmodell angepasst. Falls die in einer Funktion isolierten Funktionsumfänge bereits im dritten Schritt zu einer bestehenden Funktion verschoben wurden, sind sie bereits korrekt im E/E-Systemmodell repräsentiert. Andernfalls, also wenn die isolierte Funktion eigenständig einem E/E-System zugeordnet oder eine komplette Funktion zu einem anderen E/E-System verschoben werden soll, wird das E/E-Systemmodell angepasst. Dazu wird die isolierte oder komplette Funktion mit der in Abbildung 5.6 beschriebenen Vorgehensweise zu einem anderen E/E-System verschoben. Sollte die Funktion die einzige Funktion des Quell-E/E-Systems sein, dann wird das E/E-System aus dem E/E-Systemmodell entfernt. Abschließend werden die Modellartefakte der anderen Modellierungsebenen im E/E-Systemmodell (Abschnitt 5.2.2) mit dem E/E-System, zu dem die Funktion verschoben wurde, verknüpft.

5.1.4 Operation Spaltung

Funktionsumfänge werden von E/E-Modulen gemäß der folgenden Vorgehensbeschreibung abgespalten (Anwendungsfall 4.6):

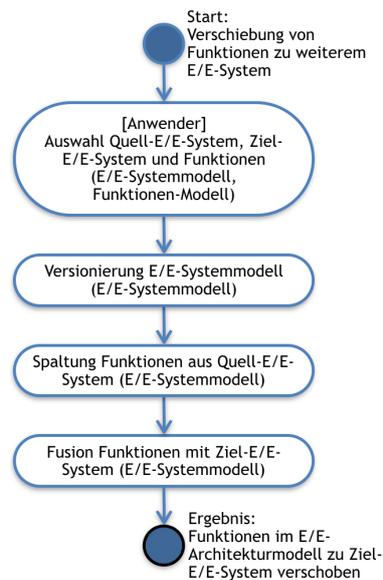


Abbildung 5.6: Aktivitätsdiagramm für die Verschiebung von E/E-Systemen (Anwendungsfall 4.5).

Erfassung betroffener Funktionen: Zuerst wird die Funktion erfasst, von der Funktionsumfänge abgespalten werden sollen. Sollte es für die Spaltung notwendig sein, die Konzeptionelle Funktionsarchitektur-Ebene anzupassen, so wird die Modellierungsebene im Vorfeld durch die Operationen Erstellung (Abschnitt 5.1.1) und Entfernung (Abschnitt 5.1.2) angepasst. Dabei unterstützt die in Abbildung 5.7 beschriebene Vorgehensweise zur Spaltung der Ebenenrelation (Anwendungsfall 4.2).

Refaktorisierung der Konzeptionellen Funktionsarchitektur-Ebene:

Die Konzeptionelle Funktionsarchitektur-Ebene ist durch den vorherigen Schritt bereits so angepasst, dass alle Elemente und abstrakten Schnittstellen korrekt repräsentiert sind.

Spaltung Refaktorisierung: In diesem Schritt wird die Abspaltung des Funktionsumfangs im Funktionen-Modell umgesetzt. Dafür wird die Funktion mit der in Abbildung 5.8 beschriebenen Vorgehensweise aufgeteilt, so dass eine separate Funktion für die abzuspaltenden Funktionsumfänge entsteht.

Einpflegen der Änderung: Als nächstes wird die Spaltung in weiteren Modellierungsebenen umgesetzt. Das Funktionsnetz, die Ebenenrelationen und die Hardware-Ebene werden angepasst und eine neue E/E-Modul-Variante im E/E-Modulbaukasten erstellt.

Spaltung im E/E-Systemmodell: Abschließend wird das E/E-Systemmodell angepasst. Das bisherige E/E-System wird um die Verknüpfung zur

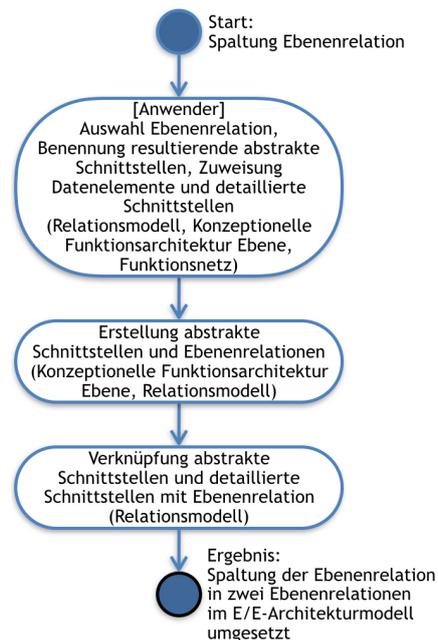


Abbildung 5.7: Aktivitätsdiagramm für die Spaltung von Ebenenrelationen (Anwendungsfall 4.2).

separaten Funktion ergänzt und mit der in Abbildung 5.9 beschriebenen Vorgehensweise aufgespalten. Zudem werden die Verknüpfungen für das bisherige E/E-System mit den relevanten Modellartefakten der anderen Modellierungsebenen im E/E-Systemmodell (Abschnitt 5.2.2) entsprechend angepasst und für das neue E/E-System erstellt.

5.1.5 Operation Fusion

Die Funktion einer zu reduzierenden E/E-Modul-Variante wird mit einer bestehenden Funktion einer bestehenden E/E-Modul-Variante gemäß der folgenden Vorgehensbeschreibung fusioniert (Anwendungsfall 4.7):

Erfassung betroffener Funktionen: Als Erstes werden die Funktionen erfasst, die fusioniert werden sollen. Soll bei der Fusion auch die Konzeptionelle Funktionsarchitektur-Ebene angepasst werden, wird die Modellierungsebene im Vorfeld durch die Operationen Entfernung (Abschnitt 5.1.2) und Erstellung (Abschnitt 5.1.1) angepasst. Die in Abbildung 5.10 beschriebene Vorgehensweise zur Fusion der Ebenenrelation (Anwendungsfall 4.2) kann bei der Anpassung der abstrakten Schnittstellen unterstützen.

Refaktorisierung der Konzeptionellen Funktionsarchitektur-Ebene:

Die Ebene der Konzeptionellen Funktionsarchitektur wurde bereits so

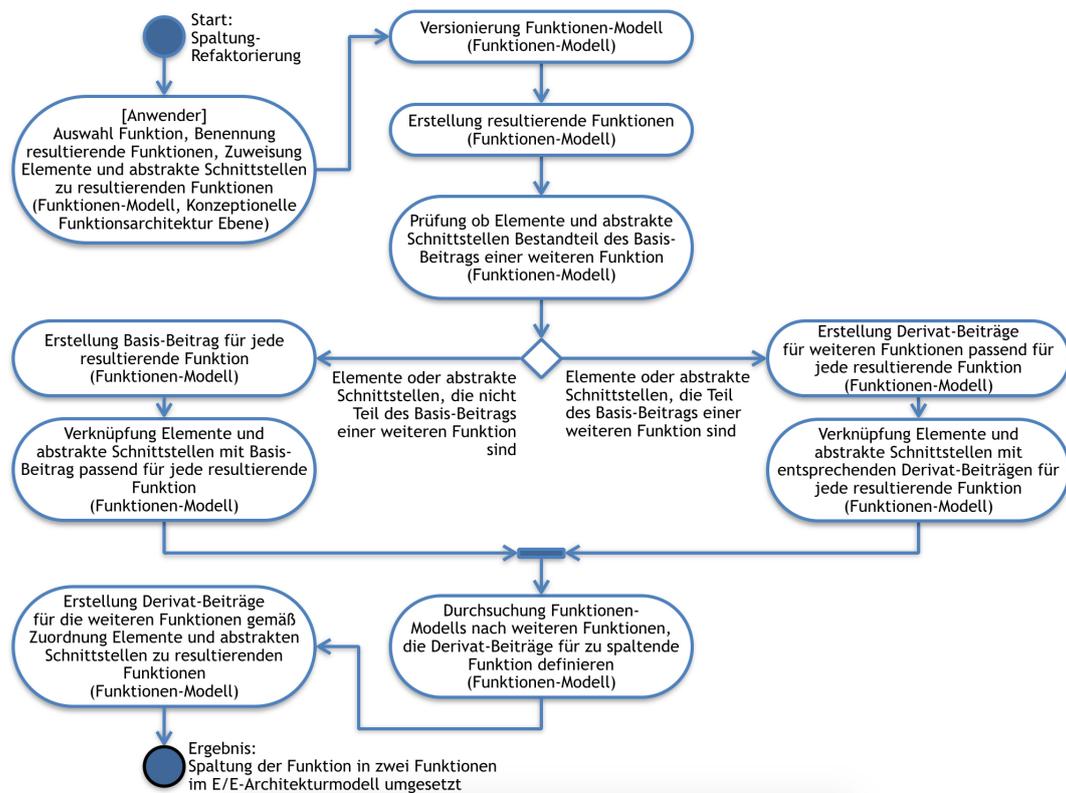


Abbildung 5.8: Aktivitätsdiagramm für die Spaltung von Funktionen (Anwendungsfall 4.6).

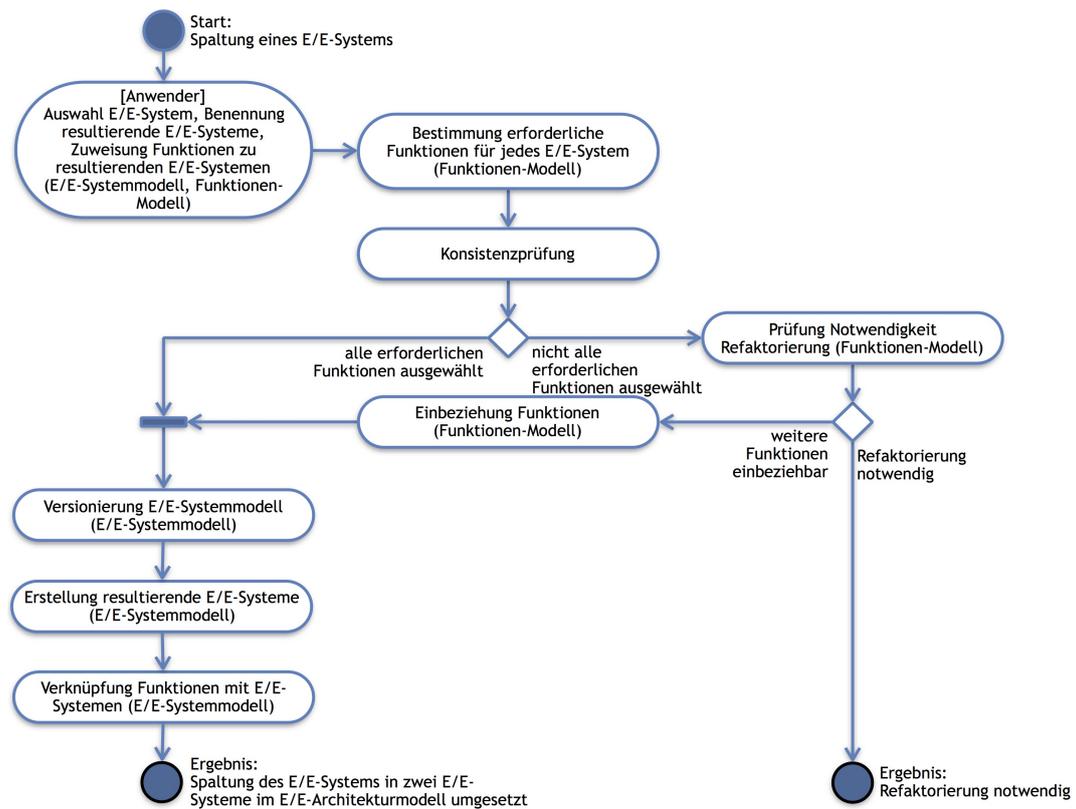


Abbildung 5.9: Aktivitätsdiagramm für die Spaltung von E/E-Systemen (Anwendungsfall 4.6).

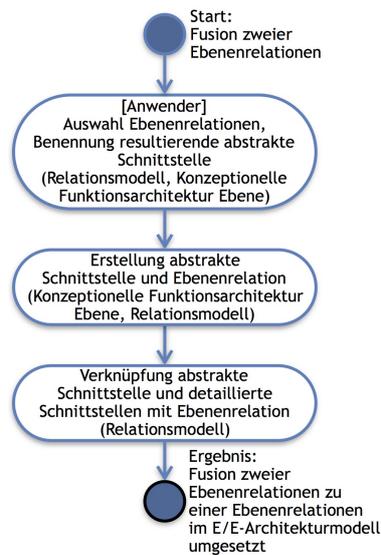


Abbildung 5.10: Aktivitätsdiagramm für die Fusion von Ebenenrelationen (Anwendungsfall 4.2).

angepasst, dass alle Elemente und abstrakten Schnittstellen korrekt repräsentiert sind.

Fusion Refaktorisierung: Danach wird die Fusion der Funktionen im Funktionen-Modell umgesetzt. Dafür werden die Funktionen mit der in Abbildung 5.11 beschriebenen Vorgehensweise fusioniert, so dass eine Funktion für die gesamten Funktionsumfänge entsteht.

Einpfelegen der Änderung: In diesem Schritt wird die Fusion in weiteren Modellierungsebenen umgesetzt. Das Funktionsnetz und die Ebenenrelationen werden angepasst. Ferner wird, falls notwendig, die Hardware-Ebene und, falls noch nicht geschehen, die E/E-Modul-Variante aus dem E/E-Modulbaukasten entfernt.

Fusion im E/E-Systemmodell: Zuletzt wird das E/E-Systemmodell angepasst. Die Verknüpfung der entfallenen Funktionen zum E/E-Systemmodell wird entfernt. Dann wird die neue Funktion mit dem E/E-Systemmodell entsprechend verknüpft. Danach werden die E/E-Systeme mit der in Abbildung 5.12 beschriebenen Vorgehensweise fusioniert. Abschließend werden die Verknüpfungen für das fusionierte E/E-System mit den relevanten Modellartefakten der anderen Modellierungsebenen im E/E-Systemmodell (Abschnitt 5.2.2) entfernt bzw. entsprechend erstellt.

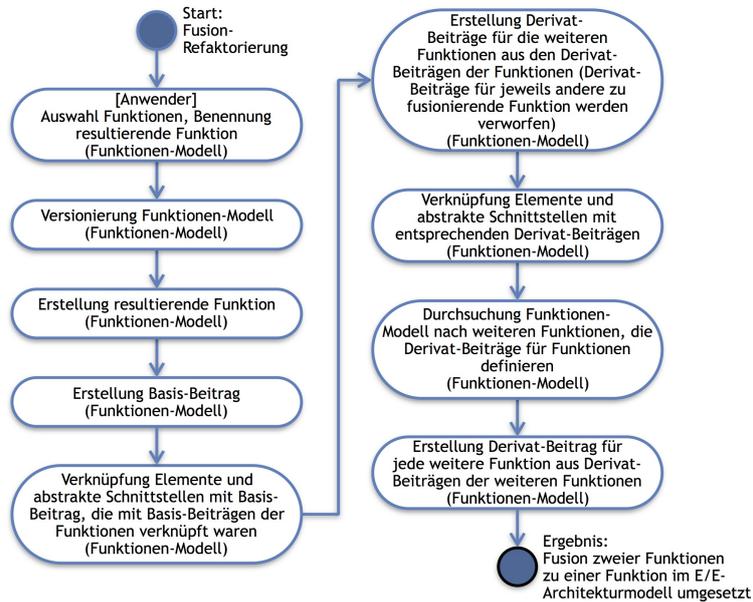


Abbildung 5.11: Aktivitätsdiagramm für die Fusion von Funktionen (Anwendungsfall 4.7).

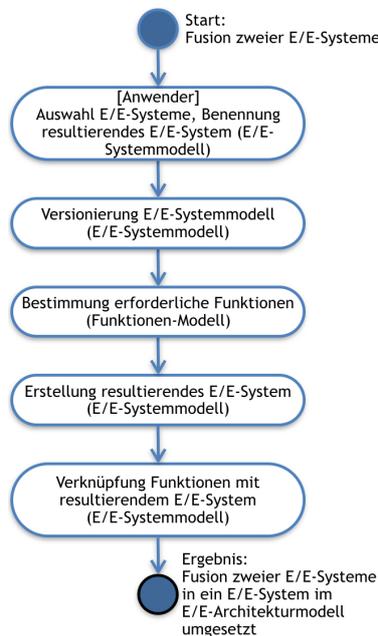


Abbildung 5.12: Aktivitätsdiagramm für die Fusion von E/E-Systemen (Anwendungsfall 4.7).

5.2 Modellstruktur und Variantenmanagement für E/E-Systeme

Für die Umsetzung der E/E-Systeme in der Modellstruktur und bei der Variantensteuerung des E/E-System Produktlinien Engineerings sind die Teil-Konzepte zur Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 4.4.1) einschließlich des Relationsmodells und des Funktionen-Modells für das FOAR (Abschnitt 4.4.2) relevant. Auch das E/E-Systemmodell (Abschnitt 4.4.3), das das E/E-Modulmodell und das Konfigurationsmodell einschließt, ist wesentlich. Nachfolgend werden die Teil-Konzepte nacheinander in der Modellstruktur und im Variantenmanagement umgesetzt.

5.2.1 Modell der Konzeptionellen Funktionsarchitektur-Ebene

Gemäß des Konzepts der Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 4.4.1) werden Elemente und abstrakte Schnittstellen vorgesehen, um die Abstraktion zu beschreiben. In Rahmen des E/E-System Produktlinien Engineerings werden sie wie folgt umgesetzt:

Abstraktes Funktions-Paket: Das Abstrakte Funktionen-Paket stellt ein Element zur Strukturierung dar, das eine nach Domänen getrennte Modellierung der Abstrakten Funktionen erlaubt.

Abstrakter Funktionsblock: Die Elemente, mit denen die abstrakten Funktionen auf der Konzeptionellen Funktionsarchitektur-Ebene modelliert werden, sind die Abstrakten Funktionsblöcke.

Port eines Abstrakten Funktionsblocks: Die Abstrakten Funktionsblöcke haben wie die Funktionen des Funktionsnetzes p-Ports (provided Ports) und r-Ports (required Ports) für die Modellierung der Schnittstellen zwischen Abstrakten Funktionsblöcken.

Abstrakte Schnittstelle: Die Abstrakte Schnittstelle verbindet einen p-Port und mindestens einen r-Port zwischen verschiedenen Abstrakten Funktionsblöcken.

Relation zwischen den Funktionsarchitekturebenen: Die abstrakten Schnittstellen werden mit den detaillierten Schnittstellen des Funktionsnetzes in Verbindung gesetzt. Diese Verbindung zwischen den Schnittstellen ist eine Relation im Relationenmodell.

5.2.2 E/E-Systemmodell

Das Konzept des E/E-Systemmodells (Abschnitt 4.4.3) sieht vor, dass die Varianten-Information vom Funktionsnetz und der Hardware-Architektur separiert wird. In Rahmen des E/E-System Produktlinien Engineerings wird das E/E-Systemmodell wie folgt umgesetzt:

Umsetzung E/E-Systeme: Die einzelnen E/E-Systeme, die E/E-Domänen, denen sie zugeordnet sind und die E/E-System-Varianten werden als Merkmale im E/E-Architekturmodell umgesetzt.

Integration Konfigurationsmodell: Die Beziehungen zwischen den E/E-Systemen werden im E/E-Architekturmodell als Bedingungen zwischen den Merkmalen umgesetzt.

Umsetzung Weiterentwicklung E/E-Modulmodell: Gleiches gilt für das E/E-Modulmodell. Auch hier werden das E/E-Modul-Hardware-Paket, die E/E-Module und deren E/E-Modul-Varianten als Merkmale im E/E-Architekturmodell hinterlegt. Die Strukturierung der E/E-Module erfolgt dabei gemäß der Modulstrategie. Die Hardware-Modellartefakte werden über Mappings zwischen den einzelnen Modellartefakten und den Merkmalen angebunden.

Umsetzung Funktionen-Modell: Die Funktionen des Funktionen-Modells, deren Basis- und deren Derivat-Beiträge werden auch als Merkmale im E/E-Architekturmodell umgesetzt. Die Anbindung der einzelnen Modellartefakte der Konzeptionellen Funktionsarchitektur erfolgt über Mappings zwischen den Merkmalen und den Modellartefakten.

Umsetzung Anbindung Funktionsnetz: Die Funktionen des Funktionsnetzes werden auch über Mappings zwischen den Funktionsbeiträgen und den Merkmalen für die E/E-System-Varianten angebunden.

5.2.3 Erstellung von E/E-Architekturmodell-Varianten

Für die Erstellung von E/E-Architekturmodell-Varianten unterstützen

- die Übersicht E/E-Modul-Inhalte, eine Tabelle, in der die E/E-System-Varianten der E/E-Module-Hardware gegenübergestellt sind und
- das Konfigurationsmodell-Diagramm, das die Merkmale und die Bedingungen zwischen den Merkmalen darstellt.

Die Erstellung einer E/E-Architekturmodell-Variante (Anwendungsfall 4.8) erfolgt gemäß der in Abbildung 5.13 beschriebenen Vorgehensweise:

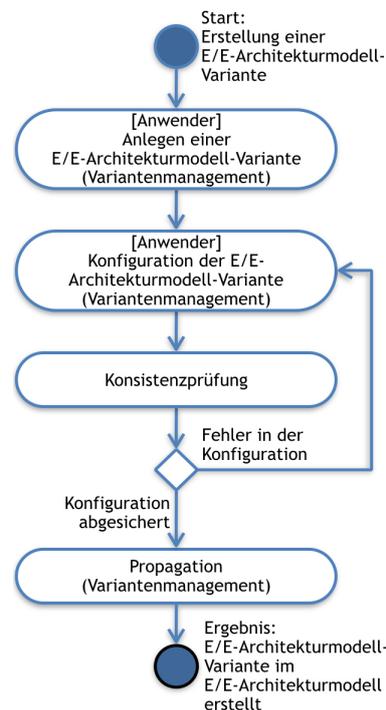


Abbildung 5.13: Aktivitätsdiagramm für die Erstellung einer E/E-Architekturmodell-Variante (Anwendungsfall 4.8).

Anlegen einer E/E-Architekturmodell-Variante: Im ersten Schritt wird im E/E-Architekturmodell eine neue E/E-Architekturmodell-Variante angelegt und eindeutig benannt, damit sie klar von den weiteren E/E-Architekturmodell-Varianten unterschieden werden kann.

Konfiguration der E/E-Architekturmodell-Variante: Die neue E/E-Architekturmodell-Variante wird danach konfiguriert, indem ihr genau die E/E-Systeme-Varianten zugewiesen werden, die in der E/E-Architekturmodell-Variante enthalten sein sollen.

Absicherung der Konfiguration: In diesem Schritt werden die hinterlegten Bedingungen zwischen den Merkmalen abgeprüft um sicherzustellen, dass die Konfiguration gültig ist. So kann z. B. vermieden werden, dass der E/E-Architekturmodell-Variante zwei E/E-System-Varianten zugewiesen sind, die sich gegenseitig ausschließen. Wird ein Fehler in der Konfiguration erkannt, muss dieser entsprechend behoben werden.

Propagation: Abschließend wird die Propagation gestartet um die restlichen Modellartefakte der E/E-Architekturmodell-Variante zuzuordnen.

5.2.4 Propagation

Die Weiterentwicklung der Propagation ist eine gestufte Propagation, für die die Zuordnung von E/E-System-Varianten zu E/E-Architektur-Varianten als Basis dient. Die Propagationskeime sind somit die Merkmale. In der ersten Stufe wird das Funktionsnetz und in der zweiten Stufe wird die Hardware wie folgt propagiert:

Stufe 1: Propagation des Funktionsnetzes

1. Propagation der Funktionen und Funktionsbeiträge, die ein Mapping eines Merkmals haben.
 - (a) Propagation des Mappings zwischen Funktionen bzw. Funktionsbeiträgen und den ECUs der Hardware-Architektur.
2. Propagation der Funktionen bzw. Funktionsbeiträge, die kein Mapping eines Merkmals haben.
 - (a) Für die Funktionsbeiträge, die ein Mapping eines Merkmals haben, wird die zugehörige Funktion (die kein Mapping eines Merkmals hat) propagiert.
 - (b) Für die Funktionen, die ein Mapping eines Merkmals haben, werden alle Funktionsbeiträge (die kein Mapping eines Merkmals haben), die ein Mapping auf eine ECU in der Hardware-Architektur haben, propagiert. Das Mapping zwischen diesen Funktionsbeiträgen und der ECU wird ebenfalls propagiert.
3. Propagation der Ports und Assembly Connectors zwischen den Funktionsbeiträgen.
4. Propagation der Delegation-Ports und Assembly Connectors zwischen den Funktionen.

Optional: Es können Elemente auf der Konzeptionellen Funktionsarchitektur-Ebene analog propagiert werden.

Stufe 2: Propagation der Hardware

1. Propagation der Hardware, die ein Mapping eines Merkmals haben.
2. Propagation der Anbindungen und Verbindungen zwischen zwei Hardware-Komponenten.
3. Propagation der Schaltplan-Modellelemente.
 - (a) Propagation der Schaltplanverbindungen und der Schaltplains zwischen den Hardware-Komponenten
 - (b) Ausgehend von den Anbindungen und Verbindungen werden Leitungen, Pins, Stecker und andere Leitungssatzelemente propagiert.

4. Ausgehend von den Hardware-Komponenten werden die Einbauorte und entsprechende Segmente, Ausbindungen und Trennstellen propagiert.

5.3 Wiederverwendung von E/E-Systemen

Für die Umsetzung der Wiederverwendung von E/E-Systemen im Rahmen des E/E-System Produktlinien Engineerings ist das Konzept zur Wiederverwendung von E/E-Systemen (Abschnitt 4.4.4) relevant. Die Grundlage für dieses Konzept bildet hierbei das eingeführte E/E-Systemmodell (Abschnitt 4.4.3).

Im Zentrum des Konzepts steht der E/E-System-Modellcontainer, der alle Modellartefakte eines E/E-Systems umfasst und erlaubt, sie in einem E/E-Architektur-Konzept wiederzuverwenden.

Die Automatisierung für die Durchführung der einzelnen Anwendungsfälle ist in den nachfolgenden Abschnitten detailliert.

5.3.1 Dynamische Erstellung von E/E-System-Modellcontainern

Um ein E/E-System in einem E/E-Architektur-Konzept wiederzuverwenden (Anwendungsfall 4.9), wird zuerst dynamisch ein E/E-System-Modellcontainer für ein gegebenes E/E-System erstellt. Die Erstellung erfolgt gemäß der in Abbildung 5.14 beschriebenen Vorgehensweise:

Überprüfung der E/E-System-Modellcontainer: Im ersten Schritt wird überprüft, ob für das E/E-System bereits ein E/E-System-Modellcontainer vorhanden ist. Dazu wird das Paket im E/E-Architekturmodell, in dem die E/E-System-Modellcontainer verortet sind, durchsucht.

Aktualisierung/Erstellung des E/E-System-Modellcontainers:

Falls ein E/E-System-Modellcontainer vorliegt, kann der bestehende E/E-System-Modellcontainer mit den ggf. geänderten Modellartefakten des E/E-Systems aktualisiert werden. Anderenfalls wird ein E/E-System-Modellcontainer für das E/E-System initial erstellt.

Zuweisung der Modellartefakte: In diesem Schritt werden auf Basis des E/E-Systemmodells die dem E/E-System zugeordneten Modellartefakte dem E/E-System-Modellcontainer für einer Wiederverwendung zugeordnet. Dafür wird für alle Modellartefakte des E/E-Systems überprüft, ob sie weiteren E/E-Systemen zugeordnet sind: Für die E/E-Module, Funktionen des Funktionsnetzes und Funktionen im Funktionen-Modell

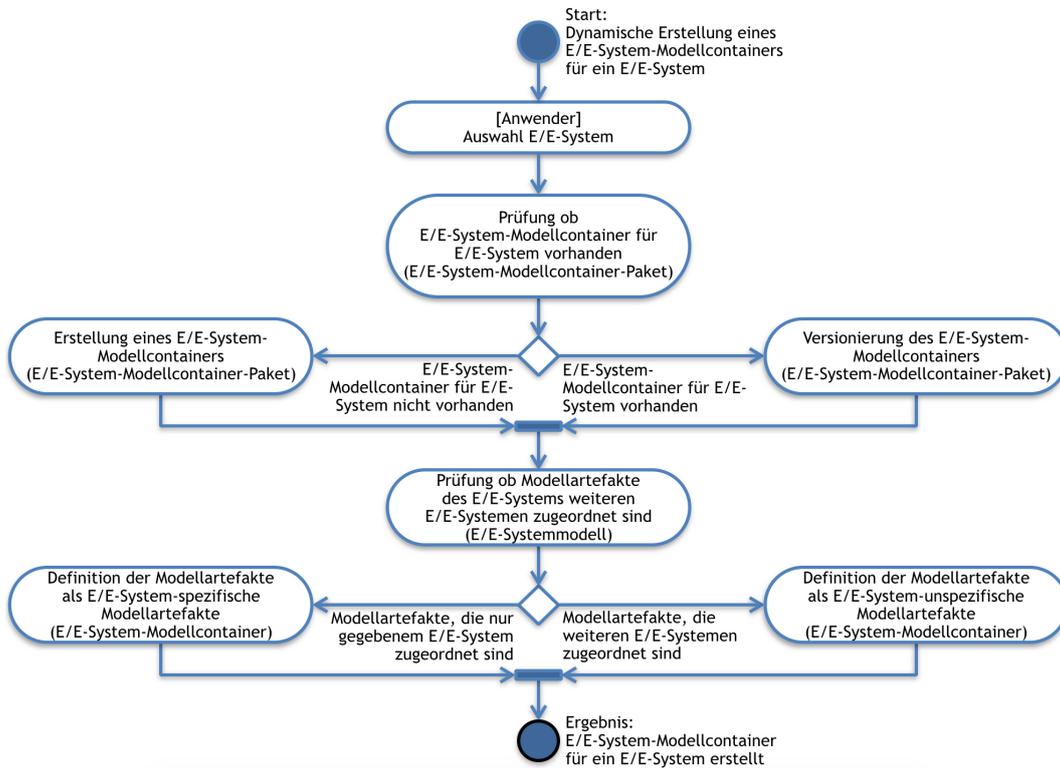


Abbildung 5.14: Aktivitätsdiagramm für die Erstellung eines E/E-System-Modellcontainers als Grundlage für die Wiederverwendung eines E/E-Systems (Anwendungsfall 4.9).

wird überprüft, ob weitere E/E-Systeme auf diese referenzieren. Falls sie nur dem gegebenen E/E-System zugeordnet sind, werden sie als E/E-System-spezifische Modellartefakte dem E/E-System-Modellcontainer hinzugefügt. Falls die Modellartefakte auch von weiteren E/E-Systemen verwendet werden, werden sie als E/E-System-unspezifische Modellartefakte dem E/E-System-Modellcontainer zugeordnet. Bei der Zuordnung der Funktionen des Funktionen-Modells werden hierbei nur die Modellartefakte im Basis-Beitrag dem E/E-System-Modellcontainer zugeordnet, da die in den Derivat-Beiträgen enthaltenen Modellartefakte in dem Basis-Beitrag einer weiteren Funktion des Funktionen-Modells enthalten sind.

5.3.2 Integration von E/E-Systemen

Für die Wiederverwendung des E/E-Systems (Anwendungsfall 4.9) erfolgt die Einbindung in das E/E-Architektur-Konzept, in dem das E/E-System wiederverwendet wird, und die Integration der Modellartefakte des E/E-Systems gemäß der in Abbildung 5.15 beschriebenen Vorgehensweise:

Konsistenzprüfung: Im ersten Schritt wird überprüft, ob der E/E-System-Modellcontainer korrekt in das E/E-Architektur-Konzept eingebunden ist. Ferner werden die Ziele in der Modellstruktur des E/E-Architektur-Konzepts, bei denen die Integration der Modellartefakte des E/E-Systems erfolgen soll, überprüft. Falls sie die Modellartefakte des E/E-Systems aufgrund eines falschen Typs des Integrationsziels nicht aufnehmen können, muss eine Korrektur erfolgen.

Integration der Modellartefakte: In diesem Schritt wird abhängig vom Typ des Modellartefakts (z. B. SWComp oder ECU), das passende Integrationsziel in der Modellstruktur des E/E-Architektur-Konzepts ermittelt. Falls das Modellartefakt bereits beim Integrationsziel enthalten ist, wird das Modellartefakt mit dem vorhandenen Modellartefakt verschmolzen und somit als gleichartig definiert. Das Modellartefakt kann bereits enthalten sein, weil es als E/E-System-unspezifisches Modellartefakt durch die Wiederverwendung eines weiteren E/E-Systems in das E/E-Architektur-Konzept integriert wurde oder weil eine neue Version des E/E-System-Modellcontainers integriert wird. Falls das Modellartefakt nicht beim Integrationsziel enthalten ist, wird das Modellartefakt dort integriert.

Verbinden der integrierten Modellartefakte: Abschließend müssen die mit der Wiederverwendung neu integrierten Modellartefakte mit den bestehenden Modellartefakten des E/E-Architektur-Konzepts verbunden werden. So wird z. B. die integrierte ECU an ein Bussystem angeschlossen.

5.3.3 Austausch von E/E-Systemen

Wird eine Änderung am E/E-System vorgenommen, erfolgt ein Austausch des wiederverwendeten E/E-Systems (Anwendungsfall 4.10) gemäß der in Abbildung 5.16 beschriebenen Vorgehensweise:

Änderung des E/E-System-Modellcontainers: Einem Austausch eines wiederverwendeten E/E-System-Modellcontainers liegt eine Evolution des E/E-Systems zugrunde. Für das weiterentwickelte E/E-System wird im ersten Schritt mit der obigen Vorgehensbeschreibung (Abschnitt 5.3.1) der E/E-System-Modellcontainer aktualisiert. Alternativ kann bei geringem Änderungsumfang (z. B. Entfernung eines einzigen Funktionsbeitrags) die Änderung auch direkt im bestehenden E/E-System-Modellcontainer durch Hinzufügungen und Entfernen der entsprechenden Modellartefakte umgesetzt werden.

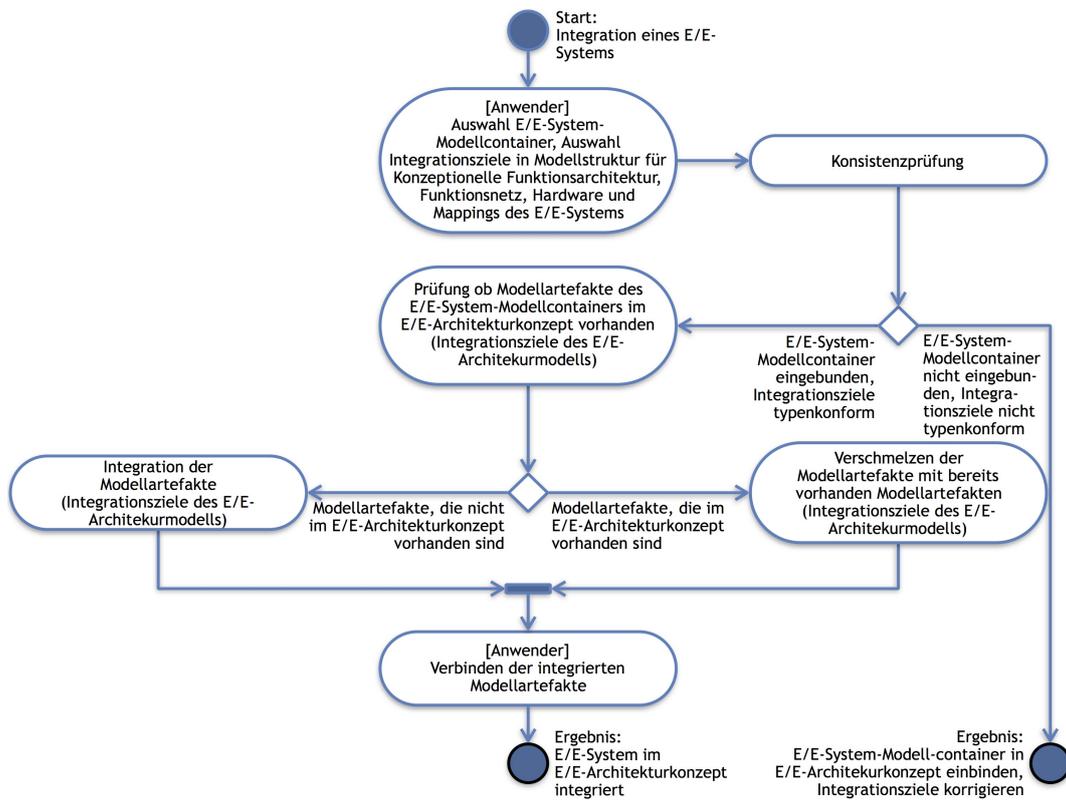


Abbildung 5.15: Aktivitätsdiagramm für die Integration eines E/E-System-Modellcontainers zur Wiederverwendung des E/E-Systems (Anwendungsfall 4.9).

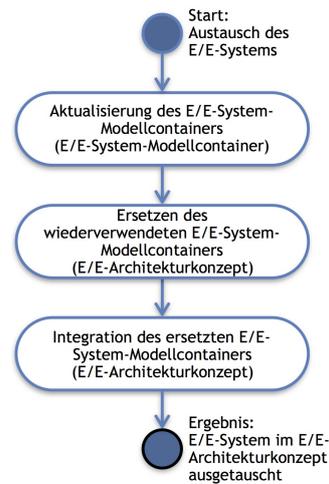


Abbildung 5.16: Aktivitätsdiagramm für Austausch eines wiederverwendeten E/E-Systems (Anwendungsfall 4.10).

Ersetzen des wiederverwendeten E/E-System-Modellcontainers: In diesem Schritt wird der in einem E/E-Architektur-Konzept wiederverwendeten E/E-System-Modellcontainer mit dem im vorherigen Schritt aktualisierten E/E-System-Modellcontainer ersetzt.

Integration des ersetzten E/E-System-Modellcontainers: Zuletzt wird die oben beschriebene Vorgehensweise zur Integration des E/E-System-Modellcontainers (Abschnitt 5.3.2) angewendet um den Austausch abzuschließen.

5.4 Umsetzung im Werkzeug für die E/E-Architekturmodellierung

Für die produktive E/E-Architekturmodellierung und -Absicherung von E/E-Architektur-Konzepten wird bei Mercedes-Benz Cars das Werkzeug PREEvision (Abschnitt 3.2.5) eingesetzt. Dieses Werkzeug eignet sich für die Umsetzung der Konzepte des E/E-System Produktlinien Engineerings, da

- die Modellierung in früher Phase und von E/E-Modulen durchgeführt wird, wobei ein E/E-Architekturmodell pro E/E-Architektur-Plattform mit einem Modellumfang von etwa 1,5 Millionen Modellartefakten (Modellobjekten) erstellt wird,
- das Werkzeug ständig erweitert wird und viele Funktionen umfasst, die den Anwender bei der Modellierung unterstützen und ohne Anpassung

bei der Umsetzung der Konzepte verwendet werden können, so fanden z. B. die Konzepte für den Modulbaukasten [81, 82, 83, 84] eine kommerzielle Umsetzung als `ReuseUnit` in `PREEvision`,

- die unterschiedlichen Ebenen der E/E-Architekturmodellierung (Abschnitt 3.2.2) in Verwendung, aktuell und miteinander verknüpft sind, denn das Werkzeug ist im Produktiveinsatz für die Optimierung, die Beschreibung, die Konzeptbewertung und den Technologieentscheid von Vernetzung, Leistungsverteilung und Leitungssatz,
- diese produktive Nutzung des Werkzeugs die Evaluierung (Kapitel 6) an Modellartefakten für ein reales Beispiel (Abschnitt 6.1) erlaubt.

Nachfolgend wird beschrieben, wie die neue Modellstruktur in das Werkzeug für die E/E-Architekturmodellierung integriert wird (Abschnitt 5.4.1) und wie die Anwendungsfälle umgesetzt wurden (Abschnitt 5.4.2).

5.4.1 Integration der neuen Modellstruktur

Die Konzeptionelle Funktionsarchitektur-Ebene wird in `PREEvision` auf der Logischen Architekturebene (Abschnitt 3.2.2) implementiert. Dabei ist das Modell der Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 5.2.1) wie folgt implementiert:

- Das Abstrakte Funktions-Paket wird als `Logical Function Package`,
- der Abstrakte Funktionsblock wird als `Logical Function`,
- der Port eines Abstrakten Funktionsblocks wird als `Logical Provided Port` bzw. `Logical Required Port`,
- die Abstrakte Schnittstelle wird als `Assembly Connector` und
- die Relation zwischen den Funktionsarchitekturebenen als `Condition` des Typs `Trace` implementiert (Abbildung 5.17 d)).

Ein Diagramm der Konzeptionellen Funktionsarchitektur-Ebene für das E/E-System Produktlinien Engineering in `PREEvision` ist in Abbildung 6.2 (Abschnitt 6.1.3) dargestellt. Es enthält die `Logical Functions`, ihre `Logical Provided Ports` bzw. `Logical Required Ports` und die `Assembly Connectors` zwischen diesen.

Um die Merkmale der neuen Modelle (Abschnitt 5.2.2) zu implementieren, kommen als Modellartefakte in `PREEvision` sowohl `CustomerFeatures` als auch `Requirements` in Frage:

- **CustomerFeatures** sollen eigentlich kundenwahrnehmbare Funktionen und explizit keine technische Beschreibung modellieren (Abschnitt 3.2.2) und
- können zur Steuerung des Variantenmanagements verwendet werden.
- **Requirements** sind zur Modellierung der Anforderungen im Rahmen der Spezifikation (Abschnitt 3.2.2) und
- können mit Daten in DOORS (Dynamic Object Oriented Requirements System) abgeglichen werden¹.

Da die Merkmale für die E/E-Systeme und E/E-System-Varianten für die Steuerung des Variantenmanagements verwendet werden sollen, werden die Merkmale trotz ihrer technischen Natur in PREEvision mittels **CustomerFeatures** implementiert. Für die Implementierung des E/E-Systemmodells heißt dies (Abbildung 5.17):

Implementierung E/E-Systeme: Die E/E-Systeme, die E/E-Domänen und die E/E-System-Varianten werden als **CustomerFeatures** implementiert.

Integration Konfigurationsmodell: Die Beziehungen zwischen den E/E-Systemen werden als **Conditions** zwischen den **CustomerFeatures** auf der **CustomerFeature**-Ebene implementiert. Ein Diagramm des Konfigurationsmodells für das E/E-System Produktlinien Engineering in PREEvision ist in Abbildung 6.3 (Abschnitt 6.1.3) dargestellt. Es enthält **CustomerFeatures** als auch die **Conditions** zwischen ihnen.

Implementierung Weiterentwicklung E/E-Modulmodell: Das E/E-Modul-Hardware-Paket, die E/E-Module und deren E/E-Modul-Varianten werden als **CustomerFeatures** implementiert. Die Hardware-Modellartefakte werden über **CF-X-Mappings** angebunden. Ein **TableBasedEditor** implementiert eine Übersicht, die alle E/E-Modul-Varianten und die angebundenen Hardware-Modellartefakte darstellt.

¹Dies ist für diese Arbeit relevant, da die Daimler AG für den gesamten Bereich R&D eine gemeinsame Systemliste aufgebaut hat und diese in DOORS pflegt. Diese Systemliste in PREEvision zu importieren und zu synchronisieren ist unabdingbar. Da die Merkmale für die E/E-Systeme mittels **CustomerFeatures** modelliert werden, wurde ein Konzept umgesetzt mit dem die Systeme der Systemliste von DOORS zu Systemen, modelliert als **Requirements**, in PREEvision importiert und synchronisiert werden können und diese anschließend in PREEvision in **CustomerFeatures** transformiert werden. Allerdings soll dieses Konzept nicht im Fokus dieser Ausarbeitung sein und wird deshalb nicht im Detail beschrieben.

Implementierung Funktionen-Modell: Die Funktionen des Funktionen-Modells, die Basis- und die Derivat-Beiträge werden als `CustomerFeatures` implementiert. Die Modellartefakte der Konzeptionellen Funktionsarchitektur werden über `CF-X-Mappings` angebunden.

Implementierung Anbindung Funktionsnetz: Die Funktionen des Funktionsnetzes werden über `CF-X-Mappings` angebunden.

Das mit Merkmalen definierte E/E-Systemmodell wird für die Erstellung von E/E-Architekturmodell-Varianten verwendet (Abschnitt 5.2.3). Dafür wird wie bisher auch die `Variant Management perspective` einschließlich des `Template Views` verwendet.

Nach der Konfiguration der E/E-Architekturmodell-Variante erfolgt die gestufte Propagation (Abschnitt 5.2.4), die

- durch `PropagationRules` implementiert wird. Mit der Propagationsregel werden Artefakte des E/E-Architekturmodells über Suchpatterns gefunden und der E/E-Architekturmodell-Variante hinzugefügt.
- Hierbei werden die Propagationsregeln für jede Stufe in einer `RoleGroup` zusammengefasst.

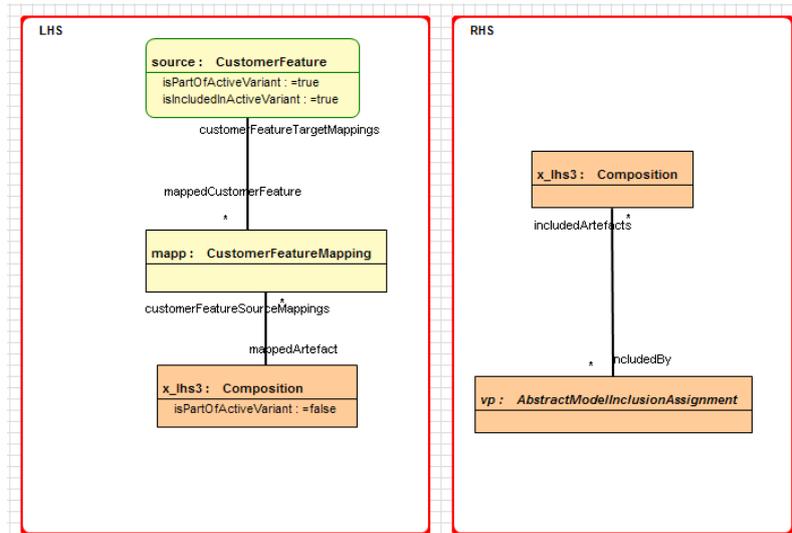
Exemplarisch werden zwei Propagationsregeln in Abbildung 5.18 dargestellt.

Für die Wiederverwendung von E/E-Systemen wird der E/E-System-Modellcontainer (Abschnitt 5.3) gemäß Designentscheidung 4.5 durch einen produktiven Wiederverwendungsmechanismus des Werkzeugherstellers umgesetzt. Dieser Wiederverwendungsmechanismus wird erstmals mit Version 5.5 des Werkzeugs `PREEvision` vorgestellt und als `ReuseUnit` bezeichnet (Abschnitt 3.4.3). Hierbei werden

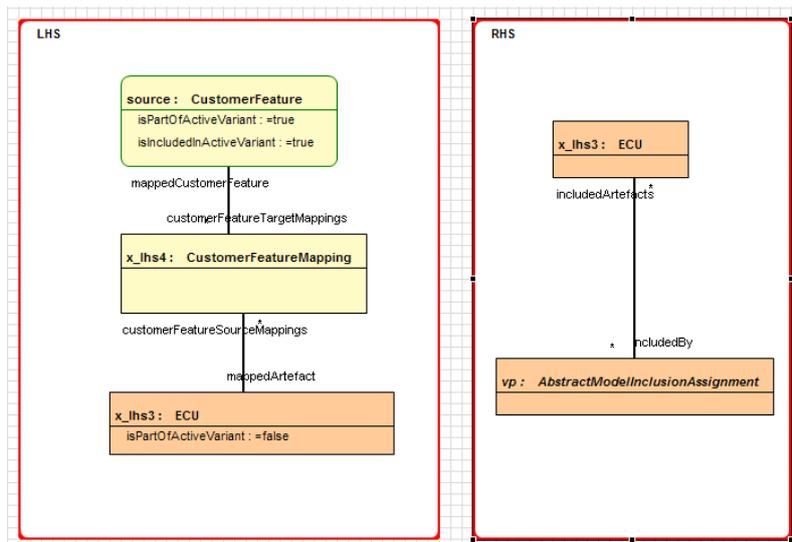
- die E/E-System-spezifischen Modellartefakte mittels der `CoreArtefacts` einer `ReuseUnit` und
- die E/E-System-unspezifischen Modellartefakte mittels der `BoundaryArtefacts` einer `ReuseUnit`

zugeordnet. Die Wiederverwendung eines E/E-Systems erfolgt in einem E/E-Architektur-Konzept (Abschnitt 5.3). In `PREEvision` entspricht dies einer `Productline`.

In Abbildung 5.19 ist der Wiederverwendungsmechanismus in `PREEvision` dargestellt.



(a) Propagationsregel zur Propagation der Funktionen und Funktionsbeiträge, die ein Mapping eines Merkmals haben



(b) Propagationsregel zur Propagation der Hardware, die ein Mapping eines Merkmals hat

Abbildung 5.18: Propagationsregeln des E/E-System Produktlinien Engineering für die Propagation ausgehend von Merkmalen in PREEvision.

Die Funktionen und Funktionsbeiträge werden in der Stufe 1 der Propagation mit der `PropagationRule CF2CO` (oben) propagiert. Hierbei werden durch die left-hand side (LHS) der Regel die `Compositions` (Funktionen bzw. Funktionsbeiträge) ausgewählt, die nicht in der E/E-Architekturmodell-Variante enthalten sind und auf die ein `CustomerFeature` (Merkmal) gemappt ist, das in der E/E-Architekturmodell-Variante enthalten ist. Diese `Compositions` werden dann über die right-hand side (RHS) der Regel in die E/E-Architekturmodell-Variante aufgenommen. In der Stufe 2 der Propagation wird die Hardware mit der `PropagationRule CF2ECU` (unten) propagiert. Dafür werden die `ECUs` ausgewählt, die nicht in der E/E-Architekturmodell-Variante enthalten sind und auf die ein `CustomerFeature` (Merkmal) gemappt ist, das in der E/E-Architekturmodell-Variante enthalten ist (LHS der Regel). Durch die RHS der Regel werden diese in die E/E-Architekturmodell-Variante mit aufgenommen.

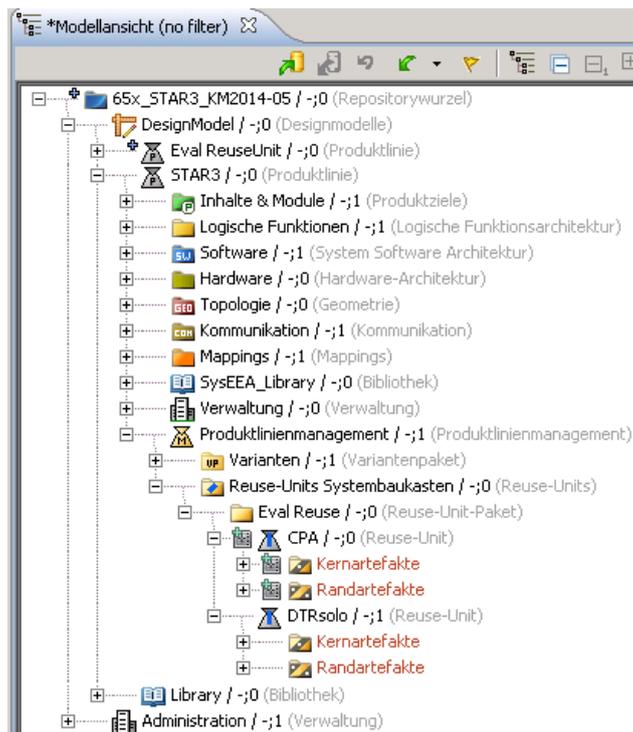


Abbildung 5.19: Der Wiederverwendungsmechanismus für das E/E-System Produktlinien Engineering in PREEvision. Dargestellt ist der Modellbaum, in dem die ReuseUnits einem ReuseUnitPackage zugeordnet sind. Den Kernartefakten (CoreArtefacts) sind die E/E-System-spezifischen Modellartefakte und den Randartefakten (BoundaryArtefacts) sind die E/E-System-unspezifischen Modellartefakte zugeordnet. Die so definierten ReuseUnits werden in einer anderen ProductLine wiederverwendet.

5.4.2 Werkzeugunterstützung für die Anwendungsfälle

Zur Unterstützung des Anwenders bei den einzelnen Anwendungsfällen (Abschnitt 4.4.5) des E/E-System Produktlinien Engineerings wurde das Werkzeug für die E/E-Architekturmodellierung PREEvision erweitert. Die Erweiterung um die Automatisierung für die Durchführung der einzelnen Anwendungsfälle wird über Metriken umgesetzt, da

- sie eine grafische Beschreibung umfassen, die dem Anwender einen einfachen Zugang bietet (so werden Diagramme mit Konnektoren, Eingabe-, Berechnungs- und Ausgabeblocks verwendet),
- sie Java-Anwendungscode kapseln können, der Berechnungen und Modelloperationen mit Zugriff auf das gesamte E/E-Architekturmodell beinhalten kann,
- sie über eine Java-API (engl. *Application Programming Interface*, Schnittstelle zur Anwendungsprogrammierung) Zugriff auf PREEvision-Methoden zum Einlesen des Anwender-Inputs, Veränderung des E/E-Architekturmodells und Erzeugung eines Anwender-Outputs bereitstellen, und
- sie im Vergleich zu Java-PlugIns einen geringeren Migrationsaufwand bei einer Weiterentwicklung des Werkzeugs haben.

Der Anwender findet einen Einstieg in die implementierten Metriken und die Modellartefakte des E/E-System Produktlinien Engineerings über ein dafür geschaffenes Framework (Abbildung 5.20). Das Framework ist über ein **Vernetzungs-Diagramm** implementiert. In den nachfolgenden Abschnitten werden die einzelnen Metriken für die Anwendungsfälle detailliert.

Evolution von E/E-Systemen

Für die Evolution von E/E-Systemen wurden acht Metriken implementiert, die den Anwender bei der Durchführung der Anwendungsfälle unterstützen.

Für die Operation Erstellung (Abschnitt 5.1.1, Anwendungsfall 4.3))

- ist für die Erstellung von Funktionen im E/E-Architekturmodell die Metrik **CreateFeature** (Abbildung 5.21) implementiert,
- wird die Ebenenrelation mit der Metrik **CreateInterfaceRelationship** (Abbildung 5.22) erstellt und
- ist für die Erstellung des E/E-Systems im E/E-Systemmodell die Metrik **New** (Abbildung 5.23) implementiert.

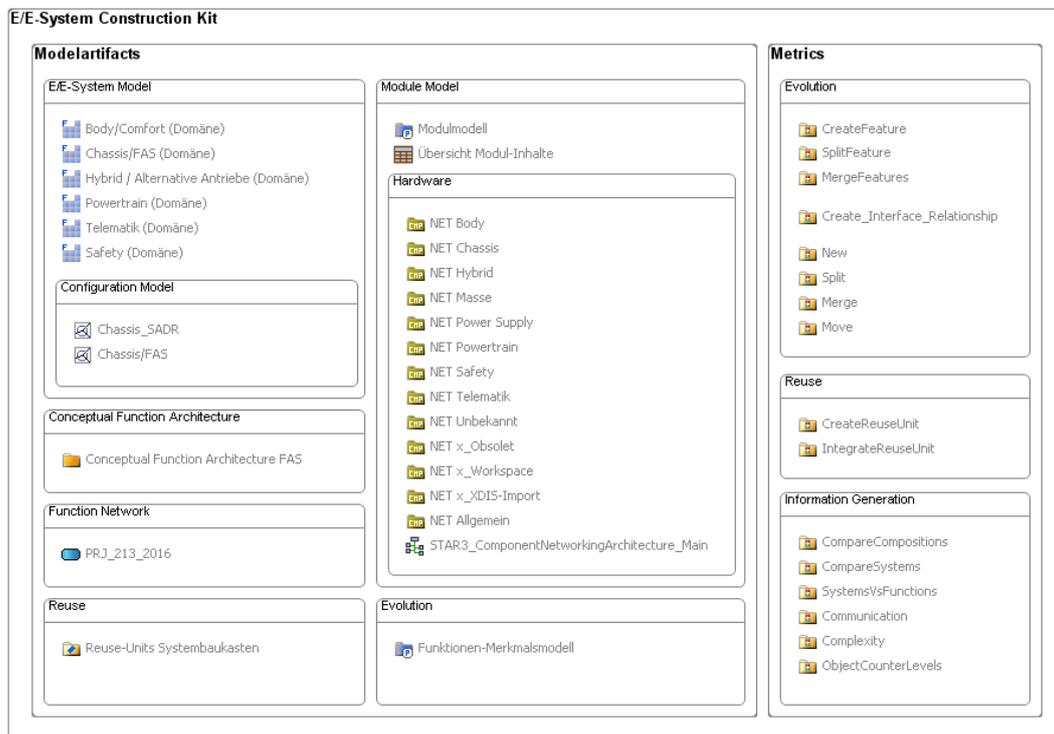


Abbildung 5.20: Framework für das E/E-System Produktlinien Engineering in PREEvision. Das Framework verknüpft Modellartefakte (links, mittig) und Metriken (rechts) des E/E-System Produktlinien Engineerings in einer Übersicht. Der die Modellartefakte darstellende Teil umfasst dabei das E/E-Systemmodell, unterteilt nach den einzelnen E/E-Architektur-Domänen, das Konfigurationsmodell, die Konzeptionelle Funktionsarchitektur-Ebene, das Funktionsnetzwerk, **ReuseUnits** für die Wiederverwendung von E/E-Systemen, den Hardware-Modulbaukasten und das Funktionen-Modell für die Evolution von E/E-Systemen. Der die Metriken darstellende Teil umfasst die Metriken für die Evolution und Wiederverwendung von E/E-Systemen. Die Metriken zur Modellauswertung und deren Konzepte sollen nicht im Fokus dieser Ausarbeitung sein und sind deshalb nicht im Detail beschrieben.

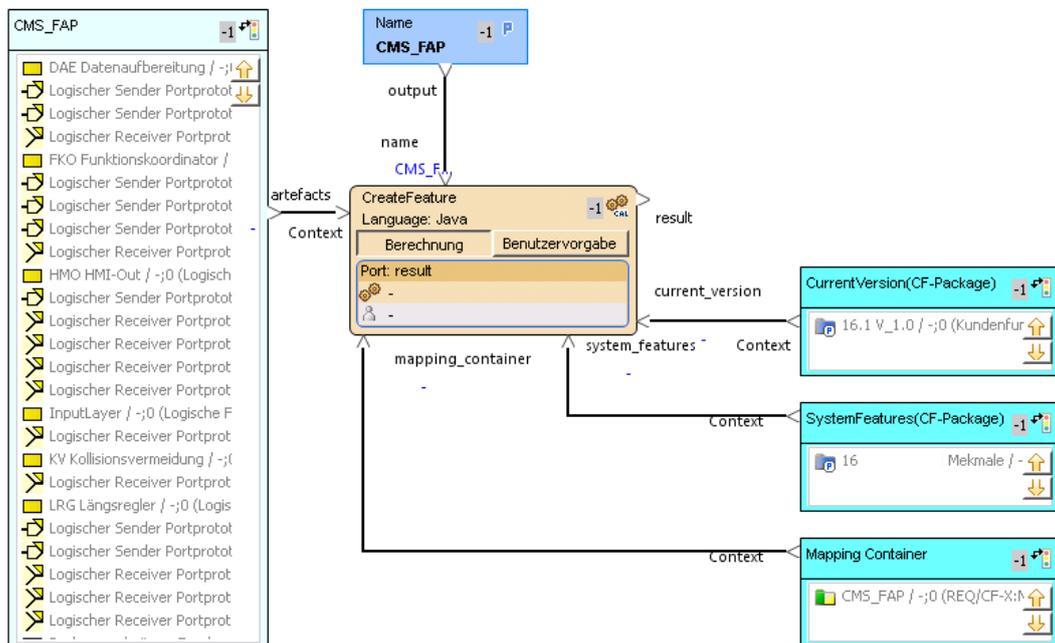
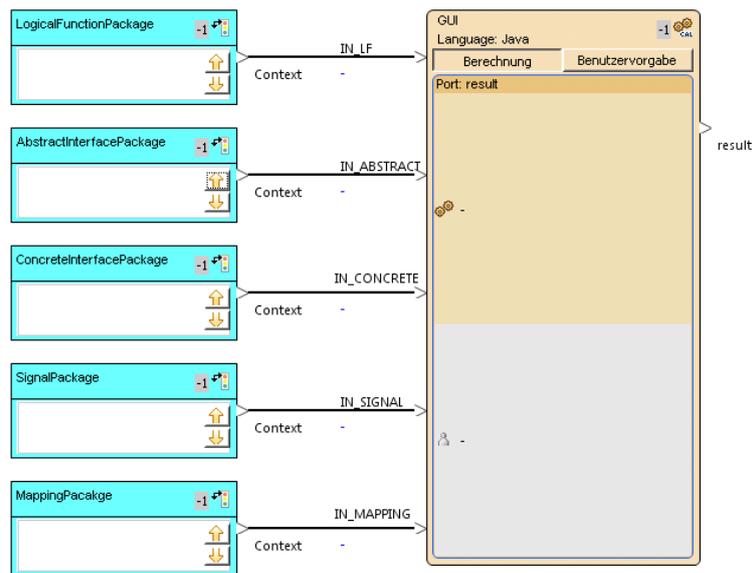
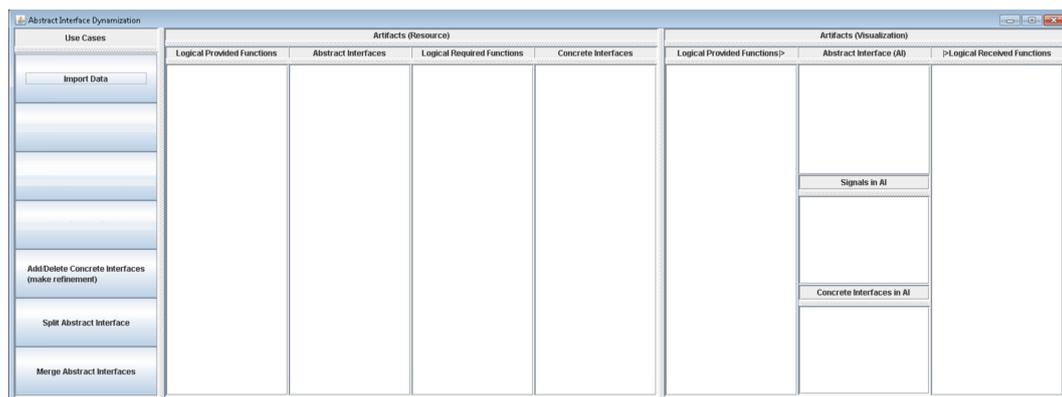


Abbildung 5.21: Metrik CreateFeature für die Erstellung von Funktionen in PREEvision (Anwendungsfall 4.3). Gemäß Abschnitt 5.1.1 wird eine Funktion durch den Berechnungsblock im Zentrum erstellt. Benutzereingaben: Links: Modellartefakte, die zur Funktion gehören; Oben: Name für die Funktion; Rechts (v. o. n. u.): Die Version, das Merkmalsmodell und das Mapping-Paket für die zu erstellenden Mappings.



(a) Metrik CreateInterfaceRelationship



(b) GUI der Metrik CreateInterfaceRelationship

**Abbildung 5.22: Metrik CreateInterfaceRelationship und GUI für die Erstellung, Entfernung und Evolution einer Ebenenrelation in PREE-
vision (Anwendungsfälle 4.1, 4.2).** Die GUI wird durch den Berechnungsblock rechts erzeugt. Benutzereingaben Links (v. o. n. u.): Modell der Konzeptionellen Funktionsarchitektur-Ebene, Paket für abstrakte Schnittstellen, Paket für detaillierte Schnittstellen, Paket für Signale, Paket für Mappings. Die GUI zeigt links die einzelnen Anwendungen (**Use Cases**), mittig die veränderbaren Modellartefakte (**Artifacts (Resource)**) und rechts die Modellartefakte nach einer Anwendung (**Artifacts (Vizualisation)**). Die Anwendungen der GUI links sind im einzelnen (v. o. n. u.): **Import Data** zur Initialisierung der GUI, **Add/Delete Concrete Interface** zur Erstellung einer Ebenenrelation gemäß Abschnitt 5.1.1 bzw. zur Entfernung einer Ebenenrelation gemäß Abschnitt 5.1.2, **Split Abstract Interface** zur Spaltung der Ebenenrelation gemäß Abschnitt 5.1.4 und **Merge Abstract Interfaces** zur Fusion der Ebenenrelation gemäß Abschnitt 5.1.5. Die veränderbaren Modellartefakte zeigen (v. l. n. r.) die Blöcke der Konzeptionellen Funktionsarchitektur-Ebene von denen eine abstrakte Schnittstelle ausgeht, die abstrakte Schnittstelle, den Block mit der die abstrakte Schnittstelle verbunden ist und die zugehörigen detaillierten Schnittstellen. Für die Modellartefakte nach der Anwendung analog.

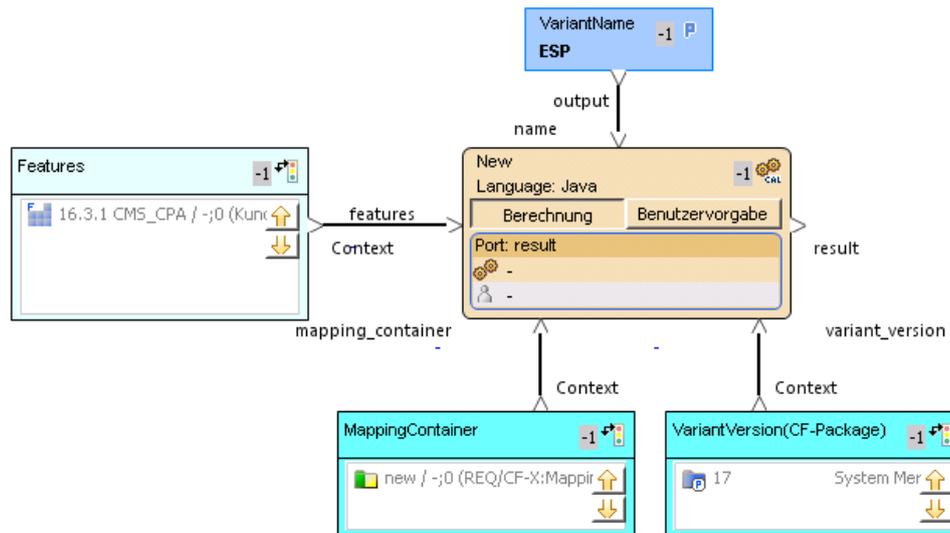


Abbildung 5.23: Metrik New für die Erstellung von E/E-Systemen in PREEvision (Anwendungsfall 4.3). Gemäß Abschnitt 5.1.1 wird ein E/E-System durch den Berechnungsblock im Zentrum erstellt. Benutzereingaben: Links: Funktionen, die zum E/E-System gehören; Oben: Name für das E/E-System; Unten (v. l. n. r.): Das Mapping-Paket für die zu erstellenden Mappings und das E/E-Systemmodell.

Bei der Operation Entfernung (Abschnitt 5.1.2, Anwendungsfall 4.4))

- wird die Funktion und das E/E-System im E/E-Architekturmodell durch den Benutzer gelöscht und
- die Ebenenrelation wird mit der Metrik `CreateInterfaceRelationship` (Abbildung 5.22) entfernt.

Bei der Operation Verschiebung (Abschnitt 5.1.3, Anwendungsfall 4.5)

- werden durch den Benutzer die Funktionen verschoben,
- wird die Ebenenrelation überprüft und ggf. mit der Metrik `CreateInterfaceRelationship` (Abbildung 5.22) angepasst und
- wird die Verschiebung von Funktionen im E/E-Systemmodell mit der Metrik `Move` (Abbildung 5.24) durchgeführt.

Für die Operation Spaltung (Abschnitt 5.1.4, Anwendungsfall 4.6)

- unterstützt die Metrik `CreateInterfaceRelationship` (Abbildung 5.22) bei der Anpassung der Konzeptionellen Funktionsarchitektur-Ebene,

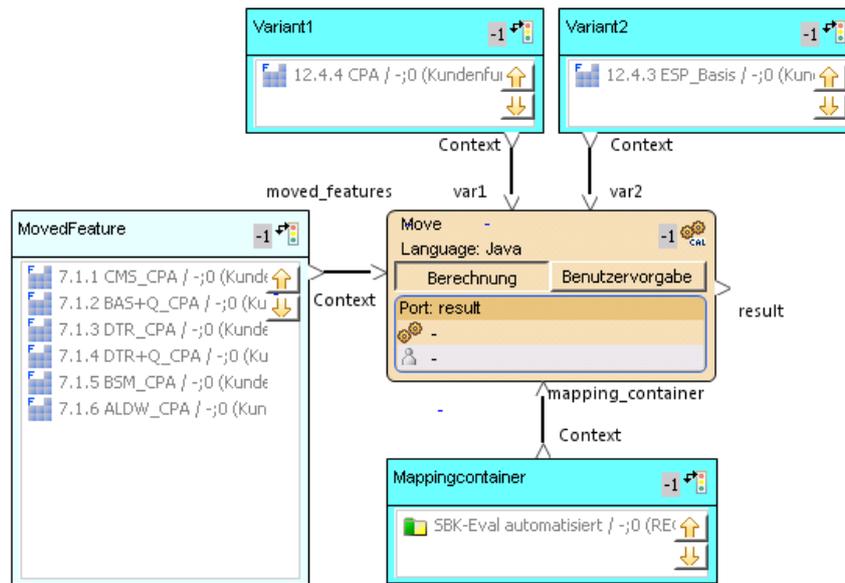


Abbildung 5.24: Metrik Move für die Verschiebung von Funktionen eines E/E-Systems in PREEvision (Anwendungsfall 4.5). Gemäß Abschnitt 5.1.3 werden die Funktionen eines E/E-Systems durch den Berechnungsblock im Zentrum verschoben. Benutzereingaben: Links: Funktionen, die verschoben werden sollen; Oben (v. l. n. r.): Quell-E/E-System, Ziel-E/E-System; Unten: Das Mapping-Paket für die zu erstellenden Mappings.

- ist zur Abspaltung der Funktionsumfänge in eine separate Funktion die Metrik `SplitFeature` (Abbildung 5.25) implementiert und
- erfolgt nach der Verknüpfung der separaten Funktion mit dem E/E-System dessen Aufspaltung mit der Metrik `Split` (Abbildung 5.26).

Für die Operation Fusion (Abschnitt 5.1.5, Anwendungsfall 4.7)

- unterstützt die Metrik `CreateInterfaceRelationship` (Abbildung 5.22) bei der Anpassung der Konzeptionellen Funktionsarchitekturebene,
- ist zur Fusion der Funktionen die Metrik `MergeFeatures` (Abbildung 5.27) implementiert und
- erfolgt nach der Verknüpfung der Funktion mit dem E/E-System durch den Benutzer die Fusion der E/E-Systeme mit der implementierten Metrik `Merge` (Abbildung 5.28)

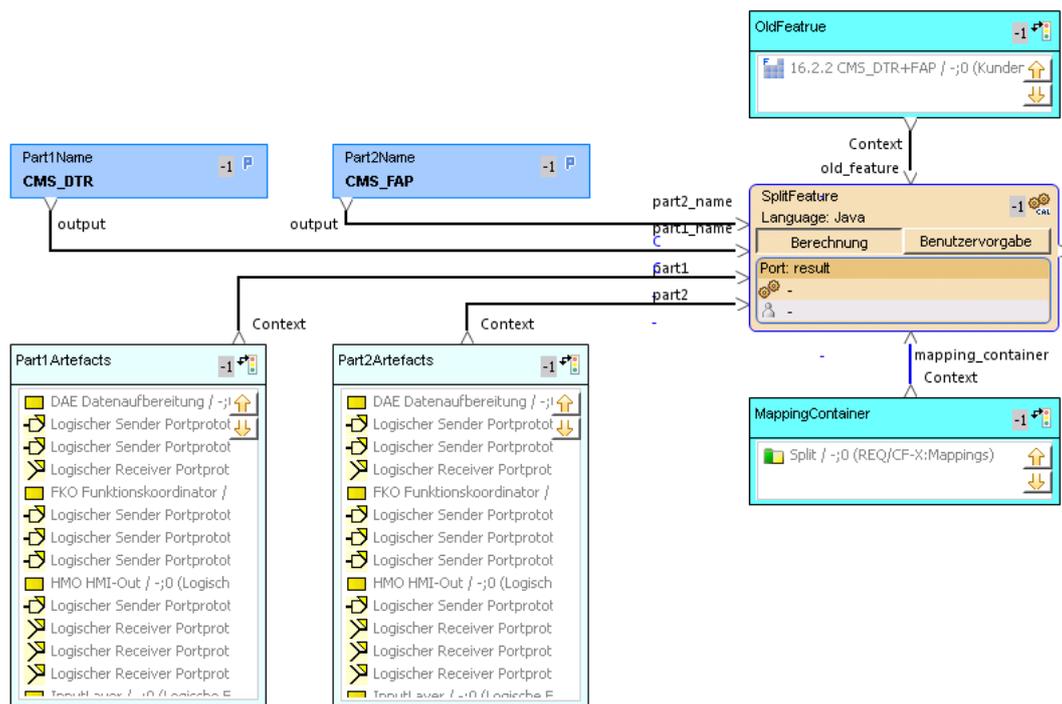


Abbildung 5.25: Metrik SplitFeature für die Abspaltung von Funktionsumfängen in eine separate Funktion in PREEvision (Anwendungsfall 4.6). Gemäß Abschnitt 5.1.4 werden die Funktionsumfänge in eine separate Funktion durch den Berechnungsblock im Zentrum abgespalten. Benutzereingaben: Oben (v. l. n. r.): Name der separaten Funktion, Name der Funktion für die restlichen Funktionsumfänge, ursprüngliche Funktion; Unten (v. l. n. r.): Modellartefakte für die abzuspaltenden Funktionsumfänge, restliche Modellartefakte, Mapping-Paket für die zu erstellenden Mappings.

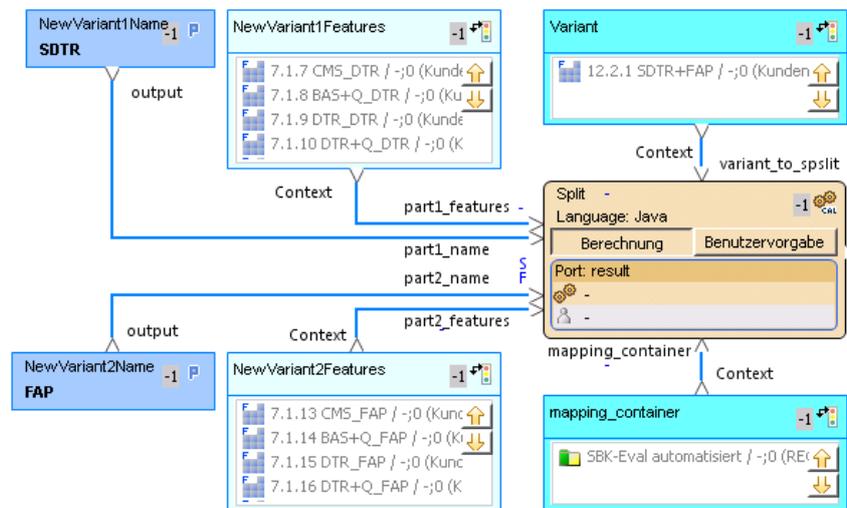


Abbildung 5.26: Metrik Split für die Abspaltung eines E/E-Systems in PREEvision (Anwendungsfall 4.6). Gemäß Abschnitt 5.1.4 wird das E/E-System durch den Berechnungsblock im Zentrum aufgespalten. Benutzereingaben: Oben (v. l. n. r.): Name des ersten resultierenden E/E-Systems, Funktionen des ersten resultierenden E/E-Systems, ursprüngliches E/E-System; Unten (v. l. n. r.): Name für das zweite resultierende E/E-System, Funktionen für das zweite resultierende E/E-System, Mapping-Paket für die zu erstellenden Mappings.

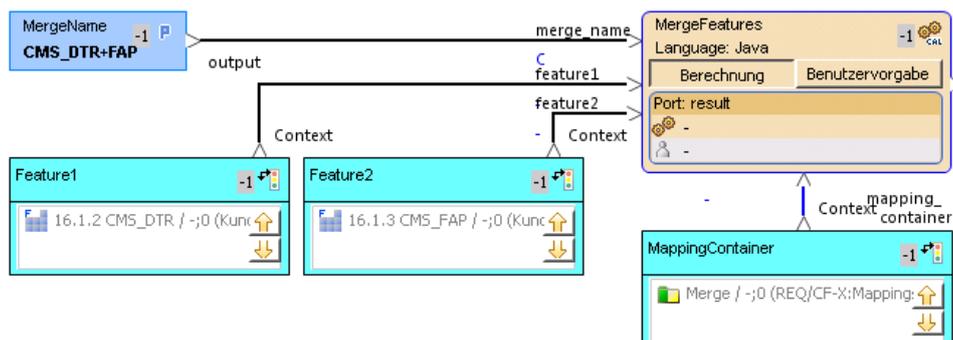


Abbildung 5.27: Metrik MergeFeatures für die Fusion von Funktionen in PREEvision (Anwendungsfall 4.7). Gemäß Abschnitt 5.1.5 werden die Funktionen durch den Berechnungsblock oben rechts fusioniert. Benutzereingaben: Oben links: Name der fusionierten Funktion; Unten (v. l. n. r.): Erste zu fusionierende Funktion, zweite zu fusionierende Funktion, Mapping-Paket für die zu erstellenden Mappings.

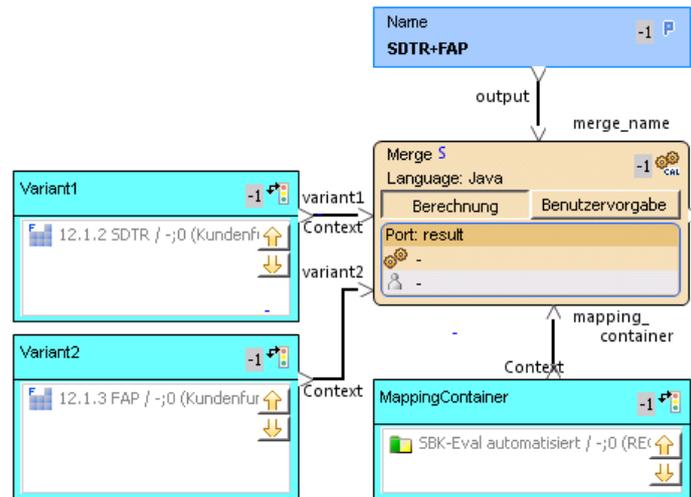


Abbildung 5.28: Metrik Merge für die Fusion von E/E-Systemen in PREEvision (Anwendungsfall 4.7). Gemäß Abschnitt 5.1.5 werden die E/E-Systeme durch den Berechnungsblock im Zentrum fusioniert. Benutzereingaben: Oben: Name des resultierenden E/E-Systems; Links (v. o. n. u.): Das erste zu fusionierende E/E-System, das zweite zu fusionierende E/E-System; Unten rechts: Mapping-Paket für die zu erstellenden Mappings.

Variantenkonfiguration

Für die Konfiguration der E/E-Architekturmodell-Varianten (Abschnitt 5.2.3, Anwendungsfall 4.8) verwendet der Anwender die **Variant Management perspective**, deren Bestandteil der **Template View** ist. Dort werden

- zur Konfiguration der E/E-Architekturmodell-Varianten die E/E-Systeme zugewiesen und
- zur Absicherung der Konfiguration mit dem Konfigurationsmodell abgeglichen (automatisch durch PREEvision) und Konfigurationsfehler hervorgehoben.

In Abbildung 5.29 ist dargestellt, wie die E/E-Architekturmodell-Varianten mit E/E-Systemen definiert werden.

Wiederverwendung von E/E-Systemen

Um den Anwender bei der Durchführung der Anwendungsfälle zur Wiederverwendung von E/E-Systemen (Anwendungsfälle 4.9, 4.10) zu unterstützen, wurden zwei Metriken implementiert:

- Für die Erstellung eines E/E-System-Modellcontainers (Abschnitt 5.3.1) ist die Metrik **CreateReuseUnit** (Abbildung 5.30) implementiert, die für ein gegebenes E/E-System eine **ReuseUnit** erzeugt.

E..	Sets	Modus	206 Min (Alternati
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Body/Comfort / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Chassis/FAS / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/> / -;0 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Aktiver Spurhalte-Assistent (Kamera) / -;1 (Kunden...	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> / -;1 (Kundenfunktion)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> / -;1 (K...	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Bremsregelsystem / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Collision Prevention Assist / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> CPA / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> / -;1 (Kunden...	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Distronic solo / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> DTRsolo / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Elektrisches Lenksystem / -;1 (Kundenfunktion)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Fahrerassistenzpaket / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Fernlichtassistent / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Luftfederung / -;1 (Kundenfunktion)	<input type="checkbox"/>	<input type="checkbox"/>

Abbildung 5.29: Konfiguration und Absicherung von E/E-Architekturmodell-Varianten in PREEvision (Anwendungsfall 4.8). Gezeigt wird ein Ausschnitt des Template Views der in der linken Spalte die E/E-Systeme, in der mittleren Spalte den Modus und in der rechten Spalte die Konfiguration der E/E-Architekturmodell-Variante darstellt. Rechts daneben würden die weiteren E/E-Architekturmodell-Varianten und deren Konfiguration angezeigt werden. Ein E/E-System wird der E/E-Architekturmodell-Variante zugewiesen, indem es angehakt wird. Soll ein E/E-System einer E/E-Architektur-Variante immer zugewiesen werden, wird es in der Spalte Modus angehakt. Liegt ein Konfigurationsfehler vor, werden die entsprechenden Zeilen hervorgehoben. Hier werden die Zeilen hervorgehoben weil sich die E/E-Systeme CPA und DTR gegenseitig ausschließen (Abschnitt 6.1). Einige E/E-Systeme wurden aus Gründen der Vertraulichkeit nicht dargestellt.

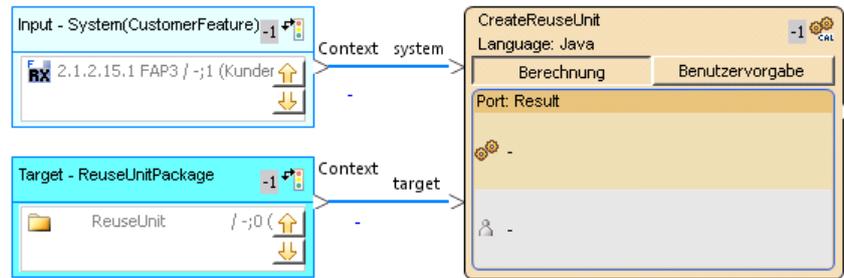


Abbildung 5.30: Metrik CreateReuseUnit für die Erstellung einer ReuseUnit für gegebenes E/E-System in PREEvision (Anwendungsfälle 4.9, 4.10). Gemäß Abschnitt 5.3.1 wird die ReuseUnit durch den Berechnungsblock rechts erstellt. Benutzereingaben: Links (v. o. n. u.): E/E-System für das eine ReuseUnit erstellt werden soll, ReuseUnit-Paket, in dem die ReuseUnit erstellt werden soll.

- Die Integration des E/E-System-Modellcontainers (Abschnitt 5.3.2) erfolgt mit der implementierten Metrik `IntegrateReuseUnit`² (Abbildung 5.31), die die Modellartefakte einer ReuseUnit in eine Productline integriert.
- Beim Austausch des E/E-Systems (Abschnitt 5.3.3) unterstützt die Metrik `CreateReuseUnit` (Abbildung 5.30) bei der Aktualisierung des E/E-System-Modellcontainers und die Metrik `IntegrateReuseUnit` (Abbildung 5.31) bei der erneuten Integration der Modellartefakte in die Productline.

5.5 Zusammenfassung

In diesem Kapitel wird das Konzept des E/E-System Produktlinien Engineerings (Kapitel 4) umgesetzt. Dazu werden Vorgehensweisen für alle Anwendungsfälle (Abschnitt 4.4.5) entwickelt und das E/E-Architekturmodell erweitert um die entsprechenden Voraussetzungen für die Anwendungsfälle zu erschaffen.

²Die Verwendung von AUTOSAR-SWC führt in Verbindung mit ReuseUnits zu fehlerhaften Konsistenzchecks: Ursache dafür sind die `SWCType-SWCPrototype`-Beziehung, die für eine Productline in der Library von PREEvision abgebildet sind. Wird eine SWC wiederverwendet, gibt es in der Library keinen `SWCType` und keinen `SWCPrototype` für die SWC. Als Folge erzeugt der ständige Konsistenzcheck über dem E/E-Architekturmodell einen Fehler. Zur Lösung dieses Konflikts wurden vier Konzepte entworfen und analysiert. Nach Bewertung wurde das Konzept zur Anpassung der Library implementiert: Mit der Wiederverwendung der SWC werden passende `SWCType`- und `SWCPrototype`-Modellartefakte in der Library erzeugt und entsprechend in Beziehung gesetzt. Das E/E-Architekturmodell wird somit konsistent.

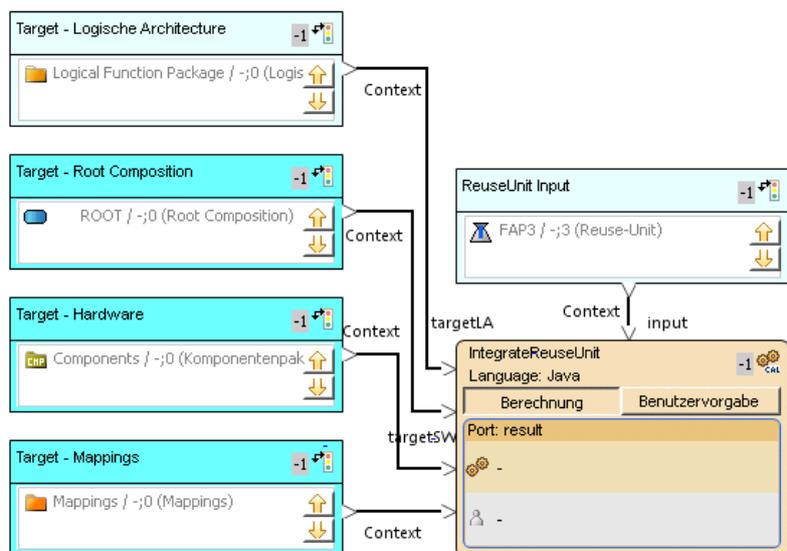


Abbildung 5.31: Metrik IntegrateReuseUnit für die Integration einer ReuseUnit in eine Productline in PREEvision (Anwendungsfälle 4.9, 4.10). Gemäß Abschnitt 5.3.2 werden die Modellartefakte der ReuseUnit durch den Berechnungsblock rechts unten erstellt. Benutzereingaben: Links (v. o. n. u.): Integrationsziel für die Modellartefakte der Konzeptionellen Funktionsarchitektur-Ebene (`LogicalFunctionPackage`), Integrationsziel für die Funktionen des Funktionsnetzes (`RootComposition`), Integrationsziel für die E/E-Module (`ComponentPackage`), Mapping-Paket für die zu erstellenden Mappings; Rechts oben: `ReuseUnit`, deren Modellartefakte an den Integrationszielen integriert werden sollen.

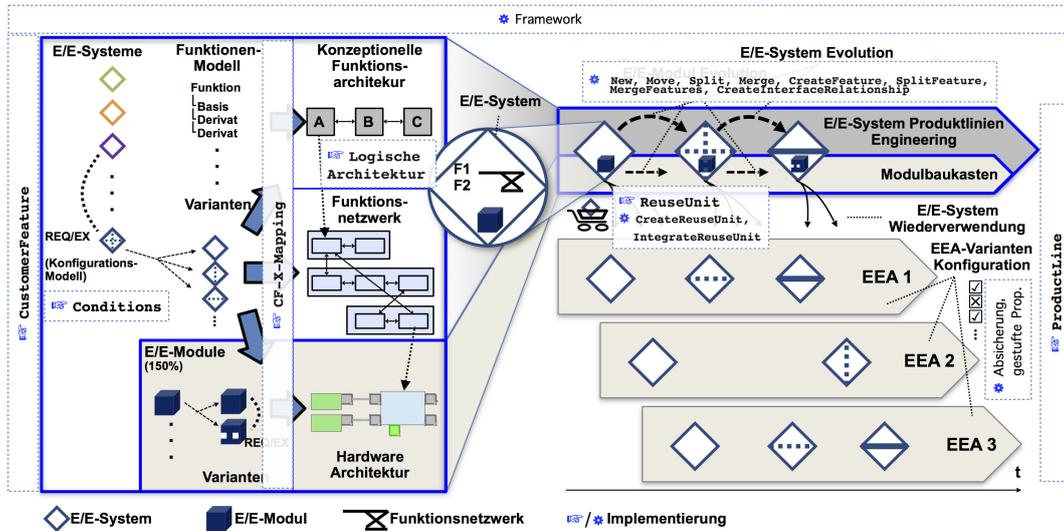


Abbildung 5.32: Implementierung des E/E-System Produktlinien Engineerings. Im Werkzeug für die E/E-Architekturmodellierung wird das E/E-System Produktlinien Engineering in einem Framework implementiert. Eingebettet in dieses werden für die Evolution von E/E-Systemen 5 Operationen entwickelt und 8 Metriken implementiert. Die Basis bildet die Konzeptionelle Funktionsarchitektur-Ebene, implementiert auf der Logische Architektur-Ebene und das Funktionen-Modell, das mit CustomerFeatures umgesetzt ist. Weitere Anteile zur Modellstruktur und Variantenmanagement werden ebenfalls mit CustomerFeatures und dem entsprechend CF-X-Mappings, Conditions und PropagationRules umgesetzt. Ein Werkzeugmechanismus nutzt das Konfigurationsmodell für die Absicherung der E/E-Architektur-Varianten. Für die Wiederverwendung werden ReuseUnits verwendet und zwei Metriken implementiert. EEA E/E-Architektur; F Funktion

Das E/E-System Produktlinien Engineering wird im Werkzeug für die E/E-Architekturmodellierung in einem Framework implementiert. Es enthält die Erweiterungen des E/E-Architekturmodells und Metriken für aufwändige Aktivitäten (Abschnitt 5.4).

Eine Übersicht findet sich in Abbildung 5.32. Nachfolgend wird die Umsetzung der Konzepte kurz zusammengefasst.

Evolution von E/E-Systemen Für die Anwendungsfälle der Evolution von E/E-Systemen wird ein allgemeines Vorgehen beschrieben, in das die einzelnen Operationen Erstellung, Entfernung, Verschiebung, Spaltung und Fusion eingebettet sind (Abschnitt 5.1). Gemäß diesem werden bei jeder Operation betroffene Funktionen erfasst, die Konzeptionelle Funktionsarchitektur-Ebene refaktoriert, das Funktionen-Modell refaktoriert, die Änderungen eingepflegt und das E/E-Systemmodell angepasst. Die Operationen sind hierbei so aus-

gelegt, dass es bei einer funktionalen Änderung durchaus zur Ausführung verschiedener Operationen kommen kann. Wird z. B. eine Funktion verschoben, werden entstehende funktionale Lücken durch die Erstellung neuer Funktionalität ersetzt.

Insgesamt wurden zur Unterstützung des Anwenders acht Metriken im Werkzeug für die E/E-Architekturmodellierung implementiert (Abschnitt 5.4.2). Die Metriken unterstützen bei der Erstellung, Verschiebung, Abspaltung und Fusion von Funktionen im E/E-Systemmodell, bei der Anpassung der Konzeptionellen Funktionsarchitektur-Ebene und der Erstellung, Anpassung und Entfernung von Ebenenrelationen bzw. bei der Erstellung, Abspaltung und Fusion von E/E-Systemen im E/E-Systemmodell.

Modellstruktur und Variantenmanagement für E/E-Systeme Zur Umsetzung der Modellstruktur des E/E-System Produktlinien Engineerings gehört das Modell der Konzeptionellen Funktionsarchitektur-Ebene und das E/E-Systemmodell (Abschnitt 5.2). Mit letzterem werden E/E-Systeme beschrieben, die Beziehungen zwischen den E/E-Systemen im Konfigurationsmodell erfasst, die Weiterentwicklung des E/E-Modulmodells eingebunden, die Funktionen im Funktionen-Modell beschrieben und das Funktionsnetz an die E/E-Systeme angebunden. Für den Anwendungsfall zur Erstellung einer E/E-Architekturmodell-Variante (Anwendungsfall 4.8) wird eine Vorgehensweise beschrieben mit der eine E/E-Architekturmodell-Variante angelegt, konfiguriert, abgesichert und propagiert wird. Hierfür wird die Weiterentwicklung der Propagation umgesetzt, die gestuft ist und deren Basis das E/E-Systemmodell bildet.

Die Modellstruktur wird in das Werkzeug für die E/E-Architekturmodellierung integriert (Abschnitt 5.4.1). Dafür wird die Konzeptionelle Funktionsarchitektur-Ebene auf der Logischen Architekturebene implementiert. Zudem werden die Merkmale des E/E-Systemmodells und ihre Beziehungen trotz ihrer technischen Natur mit **CustomerFeatures** implementiert. Für die gestufte Propagation werden neue Propagationsregeln implementiert. Somit werden für die Konfiguration einer E/E-Architekturmodell-Variante nur die E/E-Systeme zugewiesen (Abschnitt 5.4.2). Die Absicherung der Konfiguration erfolgt unmittelbar durch ein Abgleich mit dem Konfigurationsmodell und dem Hervorheben der Konfigurationsfehler.

Wiederverwendung von E/E-Systemen Um E/E-Systeme wiederzuverwenden werden ein E/E-System-Modellcontainer verwendet und mehrere Vorgehensweisen genutzt (Abschnitt 5.3). Für ein E/E-System wird dynamisch ein E/E-System-Modellcontainer erstellt, wofür bestehende E/E-System-Modellcontainer überprüft, aktualisiert bzw. neu erstellt und die Modellartefakte zugewiesen werden. Die eigentliche Wiederverwendung erfolgt

durch Einbindung des E/E-System-Modellcontainers in ein E/E-Architektur-Konzept und Integration der Modellartefakte. Für letztere werden die Konsistenz geprüft, die Modellartefakte ins E/E-Architektur-Konzept integriert und verbunden. Ein Austausch des wiederverwendeten E/E-Systems erfolgt durch Änderung, Ersetzen und Integration des E/E-System-Modellcontainers.

Als E/E-System-Modellcontainer wird ein produktiver Wiederverwendungsmechanismus des Werkzeugherstellers verwendet, der als **ReuseUnit** bezeichnet wird (Abschnitt 5.4.1). Für die Unterstützung bei der Verwendung des E/E-System-Modellcontainers werden zwei Metriken implementiert, die dessen Erstellung und die Integration automatisieren (Abschnitt 5.4.2).

Kapitel 6

Evaluierung

Das Konzept für das E/E-System Produktlinien Engineering aus Kapitel 4 mit dessen prototypischer Umsetzung aus Kapitel 5 wird in diesem Kapitel evaluiert. Als Grundlage werden dafür E/E-Systeme und Szenarien aus der Praxis der E/E-Architekturentwicklung in der Domäne Chassis/Fahrerassistenz bei Mercedes-Benz Cars bilden. Diese werden in Abschnitt 6.1 vorgestellt. Daran anschließend werden die Ergebnisse eines Vergleichs mit der heutigen E/E-Architekturmodellierung (Abschnitt 6.2) und eine Fallstudie (Abschnitt 6.3) beschrieben.

6.1 Fallbeispiel der Arbeit

Diese Arbeit behandelt konkrete Fragestellungen aus der Praxis der E/E-Architekturentwicklung für Kraftfahrzeuge (Abschnitt 3.2.1). Dementsprechend werden in dieser Arbeit E/E-Systeme und Szenarien aus der Praxis der E/E-Architekturentwicklung zur exemplarischen Erklärung und zur Evaluierung der Konzepte herangezogen. Nachfolgend wird deshalb die dafür verwendete E/E-Architektur-Domäne Chassis/Fahrerassistenz und E/E-Systeme dieser E/E-Architektur-Domäne, die Radarsysteme, vorgestellt.

6.1.1 E/E-Architektur-Domäne Chassis/Fahrerassistenz

Die E/E-Architektur-Domäne Chassis/Fahrerassistenz umfasst E/E-Systeme, die das Fahrwerk beeinflussen. Zum Fahrwerk gehören die Teile, die den Kontakt mit der Fahrbahn herstellen, z. B. Reifen, Federn oder Stoßdämpfer). Im Allgemeinen werden 3 Ebenen unterschieden, auf denen diese E/E-Systeme agieren können [174, S. 1]:

- Auf die Stabilisierungsebene wirkende E/E-Systeme helfen dem Fahrer, einen komfortablen Zustand herzustellen oder die Gewalt über das Fahrzeug zu behalten, wie z. B. die E/E-Systeme Magic Body Control [172] oder ESP (Elektronisches Stabilitätsprogramm).
- Auf die Bahnführungsebene wirkende E/E-Systeme unterstützen bei der Trajektorienwahl und -haltung, wie z. B. E/E-Systeme zur Unterstützung des Parkvorgangs [140], zur Kollisionsvermeidung (z. B. *Collision Prevention Assist*) oder das E/E-System Solo-Distronic (Abstandsregeltempomat).
- Auf die Navigationsebene wirkende E/E-Systeme unterstützen den Fahrer bei der Orientierung im Straßenverkehr, wie z. B. das E/E-System Verkehrszeichenassistent.

An der Umsetzung der Funktionen sind Aktuatoren und Sensoren beteiligt. Es gibt oftmals mehrere konkurrierende Konzepte, die die Funktion mit einer unterschiedlichen Auswahl an Aktuatoren und Sensoren umsetzen. Abhängig vom Hersteller werden z. B. Kollisionswarnsysteme mit Lidar-, Radar- und/oder Kamerasensorik umgesetzt [174, S. 1]. Die Funktionen dieser E/E-Architektur-Domäne haben generell harte Echtzeitanforderungen und strenge Anforderungen an die Verfügbarkeit [23].

Im Allgemeinen werden heute für einen geringen Funktionsumfang in dieser E/E-Architektur-Domäne Lösungen eingesetzt, bei denen die ECUs über Bussysteme vernetzt sind und Daten austauschen [146]. Für einen hohen Funktionsumfang werden Master/Slave-Netzwerke eingesetzt, bei der eine ECU als Master agiert und die Slaves in der Ausführung ihrer Aufgaben synchronisiert [146]. Die Wahrscheinlichkeit, dass eine weitere Hochintegration der Funktionen in sogenannten Domänenkontrollern (DCU, engl. *Domain Control Unit*) stattfindet, die lastfrei sein können, wird abhängig vom Marktsegment für die E/E-Architektur-Domäne Chassis/Fahrerassistenz

- in dem Volumensegment als mittelmäßig und
- in dem Premiumsegment als hoch

angesehen [146].

Neue Funktionen sind in dieser E/E-Architektur-Domäne ständig in der Entwicklung, z. B. Assistenzen in Baustellen oder beim Ausweichen in Notsituationen, wobei die Informationen der Sensoren zu einem Umfeldmodell fusioniert werden [90, 143]. Für die Zukunft ist das autonome Fahren eine Vision der Automobilindustrie. Prototypische autonome Fahrten im Straßenverkehr wurden sowohl von OEMs als auch von Tier-1-Zulieferern durchgeführt [70, 176]. Als einer der erste Meilensteine gilt dabei das Projekt PROMETHEUS (Programm

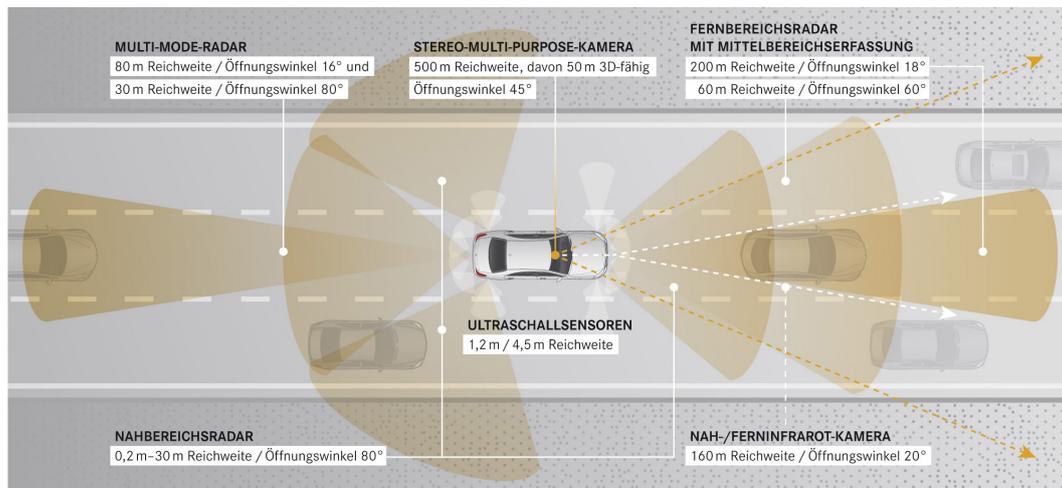


Abbildung 6.1: Sensorik der Fahrerassistenzsysteme. Die Fahrerassistenzsysteme verwenden Ultraschallsensoren, Radare (Nahbereichsradar, Multi-Mode-Radar, Fernbereichsradar) und Kameras (Stereo Multi Purpose Kamera, Infrarot-Kamera) als Sensoren zur Erfassung des Fahrzeugumfelds. Quelle: Daimler AG.

für ein europäisches Transportwesen mit höchster Effizienz und unerreichter Sicherheit), das 1986 von der damaligen Daimler-Benz AG auf den Weg gebracht wurde und als Kooperation mehrerer europäischer Autohersteller, Elektronik- und Zulieferfirmen, Universitäten und Institute über acht Jahre lief.

In der E/E-Architecturentwicklung bei Mercedes-Benz Cars werden aktuell in der E/E-Architektur-Domäne Chassis/Fahrerassistenz

- mehr als 25 E/E-Systeme mit
- insgesamt rund 100 Funktionsbeiträgen

betrachtet. Innovationen führen dazu, dass die Anzahl tendenziell steigt.

6.1.2 Radarsysteme

Drei eng miteinander verwobene Radarsysteme, die zum E/E-Modul Fahrerassistenzsysteme gehören, werden in dieser Arbeit detaillierter betrachtet. Jedes davon stellt eine unterschiedliche Skalierungsstufe auf einer Skala von einem einfachen E/E-System mit eingeschränkter Funktionalität bis zu einem umfassenden E/E-System mit vollem Funktionsumfang dar:

- Das E/E-System *Collision Prevention Assist* (CPA, engl., Kollisionsvermeidungs-Assistent) kombiniert eine Abstandswarnfunktionalität mit einem adaptiven Bremsassistenten [141]. Die

Abstandswarnfunktionalität löst eine Warnung bei einer konstanten Folgefahrt mit zu geringen Abstand oder einer gefährlichen Auffahrsituation aus. Der adaptive Bremsassistent erhöht ggf. die Verzögerung um einen Unfall zu vermeiden. Diese Funktionalitäten werden mit Hilfe eines Mittelbereichs-Radarsensor umgesetzt.

- Das E/E-System Solo-Distronic (SDTR) erweitert die Funktionalitäten des CPA und kombiniert eine Abstandsregelung zum vorausfahrenden Fahrzeug mit einer Geschwindigkeitsregelung [141]. Das E/E-System verwendet hierfür einen Fernbereichs-Radarsensor, in den die Längsregelung integriert wurde.
- Das E/E-System Fahrerassistenzpaket (FAP) erweitert die Funktionalitäten der SDTR um eine Lenk-Assistenz mit Stop&Go Pilot [142]. Auch hier wird die Geschwindigkeit auf das vorausfahrende Fahrzeug geregelt, wobei zusätzlich die Fahrbahnmarkierungen und das vorausfahrende Fahrzeug zur Querführung genutzt werden. Für diese Funktionen werden zwei Nahbereichs-Radarsensoren, ein Fernbereichs-Radarsensor und die *Stereo Multi Purpose Camera* (SMPC, engl., Stereo-Mehrzweckkamera) verwendet (Abbildung 6.1). Es werden die Sensordaten aus den Radaren und der SMPC erfasst und fusioniert um das Fahrzeugumfeld zu erfassen. D.h. räumlich ausgedehnte Objekte werden so über die Zeit verfolgt und in der anschließenden Situationsbewertung auf dem zentralen Fahrerassistenz-Steuergerät, der *Radar Decision Unit* (RDU, engl., Radarauswerteeinheit), analysiert. Ferner wird daraus eine Reaktion abgeleitet und für die Längsregelung die Motorleistung, Getriebe und Bremse bzw. für die Querregelung die elektrische Lenkung gesteuert.

Die Funktionalitäten dieser E/E-Systeme werden auf sechs Funktionen aufgeteilt: COLLISIONMITIGATIONSYSTEM, BRAKEASSIST+Q, DISTRONIC, DISTRONIC+Q, BLINDSPOTMONITORING und ADVANCEDLANEDEPARTUREWARNING. Jedes E/E-System verwendet einen unterschiedlichen Umfang dieser Funktionen zur Darstellung der Funktionalität: Das E/E-System CPA verwendet den kleinsten Umfang und damit auch die geringste Anzahl an Schnittstellen der Funktionen, während das E/E-System FAP den gesamten Umfang und damit den größten Schnittstellenumfang der Funktionen verwendet.

Die verwendeten Hardware-Komponenten der oben beschriebenen E/E-Systeme gehören zu mehreren E/E-Modulen. Das E/E-Modul Radarsysteme hat drei Varianten. Der Mittelbereichs-Radarsensor des CPA gehört zur ersten Variante, der Fernbereichs-Radarsensor gehört zur zweiten und dritten Variante. Die weiteren Nahbereichs-Radarsensoren gehören auch zur dritten Variante dieses E/E-Moduls. Die SMPC ist dem E/E-Modul Kamerasysteme zugeordnet.

6.1.3 Untersuchung der Konzepte am Fallbeispiel

Die Einführung der Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 4.4.1) wurde anhand des FAP der Generation 3 der Baureihe 222 (Mercedes-Benz S-Klasse 2013) untersucht (Abbildung 6.2). Eine Konzeptionelle Funktionsarchitektur-Ebene wurde dem Konzept folgend aufgebaut:

- Die detaillierten Interfaces des Funktionsnetzes wurden für die Schnittstellen des E/E-Systems zu anderen E/E-Systemen abstrahiert.
- Die Software-Architektur des E/E-Systems wurde reengineert.
- Die erhaltene Software-Architektur wurde abstrahiert um die internen Elemente und internen Schnittstellen zu erstellen. Dabei wurden die implementierungsrelevanten Informationen ignoriert während die E/E-Architektur-relevanten Informationen einbezogen wurden.

Überdies wurde diese Konzeptionelle Funktionsarchitektur in das Produktivmodell übernommen und um Innovationen der nächsten Generation der Mercedes-Benz S-Klasse angereichert. Das Ergebnis wurde als Diskussionsgrundlage für Expertengespräche verwendet. Es zeigte sich, dass die mehrere Funktionen einbeziehende 150%-Sicht detailliert erklärt werden musste, da beim Systementwickler klar der Fokus auf der selbst verantworteten Funktion bzw. E/E-System liegt und weitere Skalierungsstufen nicht notwendigerweise betrachtet werden. Bei der Diskussion der Konzeptionellen Funktionsarchitektur fiel auf, dass die Darstellung des Signalfusses von Sensorik (z. B. links) zur Aktuatorik (z. B. rechts) hilfreich für das Verständnis waren. Anteile, die nicht gemäß dieses Signalfusses angeordnet waren, wurden vermisst oder mussten erst gesucht werden. Für die Schnittstellen zeigte sich, dass abstrakte Schnittstellen den Einstieg in die Diskussion sehr beförderten während Schnittstellen die detailliert waren aufgrund der Unübersichtlichkeit störten. Mit dem Aufbau der Konzeptionellen Funktionsarchitektur für das FAP konnte die Überschneidung auf dem Funktionsnetz aufgelöst und eine generationenübergreifende Beschreibung gefunden werden. Auch wurde das Funktionsnetz komplett aktualisiert. Diese Aktualisierung der detaillierten Schnittstellen hatte aufgrund der Entkopplung durch Nutzung einer anderen Modellierungsebene keinen Einfluss auf die modellierte Konzeptionelle Funktionsarchitektur. Diese war somit robust gegenüber dieser Änderung.

Die Einführung des E/E-Systemmodells (Abschnitt 4.4.3) wurde für alle E/E-Systeme der Domäne Chassis/Fahrerassistenz durchgeführt. Die Basis dafür war eine Analyse der Beziehungen zwischen den E/E-Systemen, Funktionen und E/E-Modulen, deren Ergebnis die Zuordnung der Funktionsbeiträge und E/E-Modulen zu E/E-Systemen gemäß des Konzepts erlaubte. Die

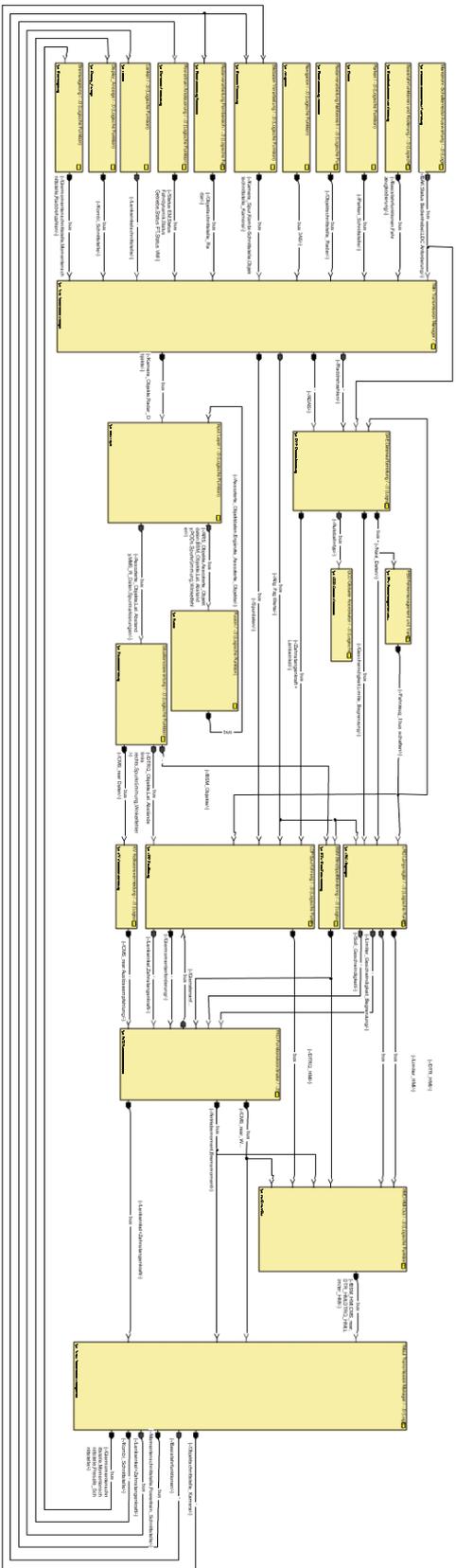


Abbildung 6.2: Die Konzeptionelle Funktionsarchitektur-Ebene als Datengrundlage für die Evaluierung. Für die Evaluierung wurden Arbeiten im Rahmen des Fallbeispiels der Arbeit (Abschnitt 6.1) durchgeführt. Hierbei ist die Konzeptuelle Funktionsarchitektur-Ebene für das FAP-Gen-3 der BR222 entstanden. Aus Gründen der Vertraulichkeit sind die Details nicht lesbar.

Varianz in den Funktionen der Domäne Chassis/Fahrerassistenz konnte aufgelöst werden. Die explizite Modellierung der Beziehungen zwischen den E/E-Systemen durch das Konfigurationsmodell (Abbildung 6.3) erlaubte das Variantenmanagement zu steuern. Diese Umsetzung findet Einsatz im Produktivmodell. Hierbei wird auch auf die Werkzeugunterstützung (**Resolver**) zur Absicherung der gewählten Konfiguration zurückgegriffen, die durch die Modellierung der Beziehungen eingesetzt werden konnte.

Die beschriebene Konzeptionelle Funktionsarchitektur-Ebene des FAP wurde annotiert, indem ein Funktionen-Modell gemäß des Konzepts des Feature Oriented Architecture Refactoring (Abschnitt 4.4.2) für alle sechs Funktionen des E/E-Modul Radarsysteme in allen drei Funktionsumfängen erstellt wurde. Besonderer Fokus wurde anschließend auf das Herauslösen der DISTRONIC-Funktion aus dem FAP unter Wiederverwendung von Schnittstellen und Hardware gelegt. Hierbei wurden der bestehende Datenstand im Modell und die aufgebaute Konzeptionelle Funktionsarchitektur-Ebene genutzt um Evolution zu modellieren.

Die Wiederverwendung für E/E-Systeme (Abschnitt 4.4.4) wurde untersucht, indem die E/E-Systeme des E/E-Moduls Radarsysteme einschließlich deren Überschneidungen auf der Konzeptionellen Funktionsarchitektur-Ebene in ein E/E-Architekturkonzept (Produktlinie) integriert wurden.

6.2 Qualitative Evaluierung

Um das E/E-System Produktlinien Engineering zu evaluieren, wird ein qualitativer Vergleich mit der heutigen E/E-Architekturmodellierung bei Mercedes-Benz Cars durchgeführt. Weitere systemorientierte Ansätze in der E/E-Architekturmodellierung sind nicht bekannt (Abschnitt 4.1) und werden dementsprechend nicht in den Vergleich eingeschlossen. Der nachfolgende Vergleich betrachtet hierbei

- die Ziele der E/E-Architekturmodellierung (Abschnitt 3.2.1),
- die Automatisierungsstufe und
- weitere Qualitätsattribute.

Diese qualitative Evaluierung erfolgte im Rahmen des Fallbeispiels der Arbeit (Abschnitt 6.1). Nach einer Übersicht der in diesem Abschnitt beschriebenen qualitativen Ergebnisse (Tabelle 6.1), wird nachfolgend jeweils der Vergleich für ein Attribut erläutert. Die Diskussion der qualitativen Evaluierung findet sich in Abschnitt 6.4.

Tabelle 6.1: Übersicht der qualitativen Evaluierung des E/E-System Produktlinien Engineerings

Attribut	Evolution von E/E-Systemen	Modellstruktur und Variantenmanagement	Wiederverwendung von E/E-Systemen
Realisierung der Ziele der E/E-Architekturmodellierung	++	+	+
Automatisierung	++	++	+
Nachverfolgbarkeit	+	+	+
Robustheit	+	0	+
Modellqualität	++	++	+

Den in diesem Abschnitt beschriebenen Attributen für den Vergleich des E/E-System Produktlinien Engineerings mit der heutigen E/E-Architekturmodellierung sind die allgemeinen Anforderungen aus Abschnitt 4.1 gegenübergestellt. Hierbei beschreibt + eine Verbesserung, – eine Verschlechterung und 0 keine Änderung. Ein mehrfaches + oder – verdeutlicht eine stärkere Verbesserung oder Verschlechterung.

6.2.1 Realisierung der Ziele der E/E-Architekturmodellierung

Die E/E-Architekturmodellierungsziele sind in der heutigen E/E-Architekturmodellierung insbesondere für die Evolution von E/E-Systemen nicht erfüllbar. Für die Strukturierung und das Management von E/E-Systemen, für die Wiederverwendung von E/E-Systemen und für die Modellauswertung sind sie nur schwer und mit hohem Aufwand erfüllbar (Abschnitt 4.1).

Im Rahmen des E/E-System Produktlinien Engineerings wurden mehrere Konzepte entworfen um diesen Zustand zu ändern. So ermöglicht die Konzeptionelle Funktionsarchitektur-Ebene (Abschnitt 4.4.1) in Kombination mit dem FOAR (Abschnitt 4.4.2) die Evolution von E/E-Systemen in der E/E-Architekturmodellierung (Anforderung 4.1). Ferner wurde das E/E-Systemmodell konzipiert (Abschnitt 4.4.3), das erlaubt die E/E-Systeme zu modellieren und für die Variantenkonfiguration zu nutzen (Anforderungen 4.2, 4.3). Weiterhin wurde die Wiederverwendung für E/E-Systeme entworfen (Abschnitt 4.4.4) um die Wiederverwendung von kompletten E/E-Systemen einschließlich ihrer E/E-Module und Funktionen aus Anforderung 4.4 zu realisieren.

Insgesamt werden durch diese Konzepte des E/E-System Produktlinien Engineerings erstmals die Ziele für die E/E-Architekturmodellierung nicht nur für die Hardware- bzw. E/E-Modul-Ebene ermöglicht sondern für die vollständige Betrachtung eines E/E-Systems einschließlich der funktionalen Anteile auf dem Funktionsnetz.

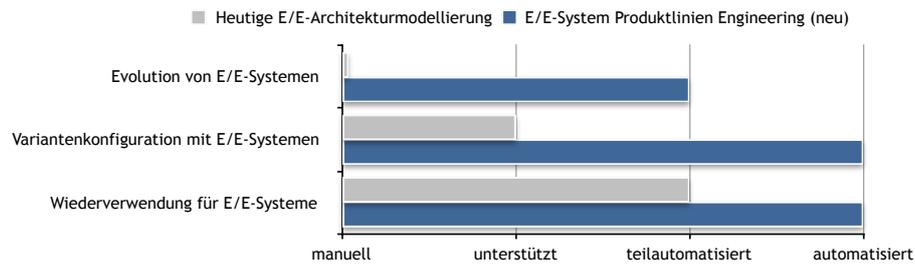


Abbildung 6.4: Qualitativer Vergleich des Automatisierungsgrads der allgemeinen Anforderungen. Steigender Automatisierungsgrad von links nach rechts.

6.2.2 Automatisierungsstufe der allgemeinen Anforderungen

In dieser Arbeit werden die allgemeinen Anforderungen für das E/E-System Produktlinien Engineering (Abschnitt 4.1) automatisiert umgesetzt. Anhand dieser wird nachfolgend die Automatisierungsstufe (vgl. [32]) der heutigen E/E-Architekturmodellierung mit dem E/E-System Produktlinien Engineering verglichen (Abbildung 6.4). Hierbei soll

manuell ein Vorgehen mit *hohem* Modellierungsaufwand bei *mittlerer* Fehleranfälligkeit, in dem der Nutzer alle Schritte selbst durchführt;

unterstützt ein Vorgehen mit *mittlerem* Modellierungsaufwand bei *mittlerer* Fehleranfälligkeit, in dem der Anwender vereinzelt auf herkömmliche Assistenz zurückgreifen kann (z. B. Kopieren und Einfügen von Modellartefakten);

teilautomatisiert ein Vorgehen mit *geringem* Modellierungsaufwand bei *geringer* Fehleranfälligkeit, in dem der Anwender teilweise auf eine Automatisierung zurückgreifen kann aber einige Umfänge selbst durchführen muss; und

automatisiert ein Vorgehen *ohne* Modellierungsaufwand bei *geringer* Fehleranfälligkeit, bei dem der Nutzer auf eine umfangreiche Automatisierung zurückgreifen kann ohne selbst Umfänge durchzuführen;

bezeichnen.

Evolution von E/E-Systemen In der heutigen E/E-Architekturmodellierung wird die Evolution von E/E-Systemen (Anforderung 4.1) manuell durchgeführt. Es werden die Modellartefakte auf allen Ebenen des E/E-Architekturmodells manuell identifiziert und entsprechend geändert.

Die Evolution von E/E-Systemen im E/E-System Produktlinien Engineering wird teilautomatisiert durchgeführt:

- Die Refaktorisierung der Konzeptionellen Funktionsarchitektur-Ebene wird manuell und die Anpassung des Funktionen- und des E/E-Systemmodells automatisiert durchgeführt.
- Die Relation zum Funktionsnetz wird benutzergeführt und automatisiert umgesetzt.
- Gegebenenfalls wird manuell eine Anpassung der Modellartefakte auf dem Funktionsnetz und der Hardware-Architektur vorgenommen.

Variantenkonfiguration In der heutigen E/E-Architekturmodellierung wird eine E/E-Architekturmodell-Variante durch das Variantenmanagement unterstützt konfiguriert (Anforderung 4.2, Anforderung 4.3). Für alle Modellartefakte auf der Funktionsnetz-Ebene muss entschieden werden, ob sie in der entsprechenden E/E-Architekturmodell-Variante enthalten sind oder nicht. Die Entscheidung, ob Modellartefakte der Hardware-Ebene zu einer E/E-Architekturmodell-Variante gehören, wird durch eine herkömmliche Propagation vorgenommen.

Im E/E-System Produktlinien Engineering wird aufgrund der jetzt vorhandenen E/E-System-Sicht für ein komplettes E/E-System entschieden, ob es in der E/E-Architekturmodell-Variante enthalten ist oder nicht. Dem Variantenmanagement wird somit automatisiert übermittelt welche Modellartefakte auf der Funktionsnetz-Ebene wie betroffen sind. Die abschließende Propagation wird vorgenommen um die Hardware mit einzubinden. Somit ist im E/E-System Produktlinien Engineering die Variantenkonfiguration automatisiert.

Wiederverwendung von E/E-Systemen Um in der heutigen E/E-Architekturmodellierung eine Wiederverwendung von E/E-Systemen (Anforderung 4.4) durchzuführen, werden die Modellartefakte für jede Ebene manuell im E/E-Architekturmodell ausfindig gemacht und in E/E-System-spezifische und -unspezifische Modellartefakte unterschieden (Abschnitt 4.4.4). Dementsprechend unterschieden werden sie an den Wiederverwendungsmechanismus übergeben, der die Wiederverwendung in ein anderes E/E-Architekturmodell automatisiert. Eine Integration der Artefakte in das E/E-Architekturmodell muss anschließend manuell vorgenommen werden. Die Wiederverwendung für E/E-Systeme in der heutigen E/E-Architekturmodellierung ist somit teilautomatisiert.

Im Vergleich dazu wird im E/E-System Produktlinien Engineering die Unterteilung in E/E-System-spezifische und -unspezifische Modellartefakte für

ein gegebenes E/E-System automatisiert vorgenommen und an den Wiederverwendungsmechanismus übergeben. Gleichzeitig werden die Artefakte automatisiert in das andere E/E-Architekturmodell integriert. Somit ist die Wiederverwendung von E/E-Systemen im E/E-System Produktlinien Engineering automatisiert.

6.2.3 Nachverfolgbarkeit

Durch die Konzepte des E/E-System Produktlinien Engineerings wird die Nachverfolgbarkeit im Vergleich zur heutigen E/E-Architekturmodellierung wesentlich gesteigert. Diese Verbesserung wird nachfolgend anhand der allgemeinen Anforderungen (Abschnitt 4.1) deutlich gemacht.

Evolution von E/E-Systemen In der heutigen E/E-Architekturmodellierung müssen die funktionalen Anteile von Innovationen auf der Ebene des Funktionsnetzes modelliert werden. Bis auf Namenskonventionen besteht somit keine klare Trennung zwischen den Weiterentwicklungen und den ursprünglichen Inhalten der E/E-Architektur.

Das E/E-System Produktlinien Engineering verbessert die nachverfolgbarkeit. Durch die Einführung der Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 4.4.1) wird eine abstrakte Beschreibung bestehender Funktionen des Funktionsnetzes erzeugt. Damit wird die Überschneidung auf dem Funktionsnetz aufgelöst. Dadurch entsteht eine generationenübergreifende Beschreibung der Funktionen. Werden die Innovationen auf dieser Ebene im Zuge der ersten Schritte einer Konkretisierung beschrieben, dann wird dadurch eine klare und nachvollziehbare Trennung zwischen den E/E-Architekturgenerationen erzeugt. Auf dem Funktionsnetz finden sich alle Anteile der ursprünglichen, und als Ausgangslage für die Weiterentwicklung verwendeten, E/E-Architektur. Auf der Konzeptionellen Funktionsarchitektur-Ebene befinden sich die Innovationen der Weiterentwicklung.

Modellstruktur und Variantenmanagement Um in der heutigen E/E-Architekturmodellierung alle Anteile eines E/E-Systems auf Hardware-Ebene und auf der Ebene des Funktionsnetzes zu identifizieren, ist Spezialwissen erforderlich. Allein aus dem E/E-Architekturmodell ist nicht ersichtlich, welche Funktionsbeiträge und welche Hardware zu einem E/E-System gehören (Abschnitt 4.1). Überdies sind Beziehungen zwischen den Funktionsbeiträgen bzw. zwischen den Hardware-Anteilen nicht modelliert, weshalb auch die Konfiguration von E/E-Architekturmodell-Varianten Expertenwissen erfordert.

In dem E/E-System Produktlinien Engineering wurde ein E/E-Systemmodell eingeführt, das nachvollziehbar macht, welche Anteile des E/E-Architekturmodells zu welchem E/E-System gehören. Darüber hinaus wird

erstmalig die Modellierung der Beziehungen zwischen den E/E-Systemen eingeführt um die Konfiguration von E/E-Architekturmodell-Varianten klar zu gestalten.

Wiederverwendung von E/E-Systemen Für verschiedene E/E-Architekturen werden in der heutigen E/E-Architekturmodellierung verschiedene E/E-Architekturmodelle verwendet. Beim Export eines E/E-Systems aus einem E/E-Architekturmodell und Import des E/E-Systems in ein weiteres E/E-Architekturmodell, wird keine Beziehung zwischen diesen beiden E/E-Architekturmodellen für dieses E/E-System hergestellt. Dadurch ist nicht einfach nachvollziehbar, in welchen E/E-Architekturen ein E/E-System wiederverwendet wurde.

Diese Nachverfolgbarkeit wird durch die neuen Konzepte der Wiederverwendung von E/E-Systemen verbessert. Das Produktlinien Engineering für E/E-Systeme wurde so umgesetzt, dass eine Unterstützung im E/E-Architekturmodellierungswerkzeug besteht um die Verwendung eines E/E-Systems in anderen E/E-Architekturen aufzuzeigen. Dadurch ist klar nachvollziehbar in welchen E/E-Architekturen das E/E-System wiederverwendet wurde.

6.2.4 Robustheit

Im Vergleich zur heutigen E/E-Architekturmodellierung gibt es mit dem E/E-System Produktlinien Engineering eine Verbesserung in der Robustheit der Modellierung. Nachfolgend wird diese für die Evolution und die Wiederverwendung von E/E-Systemen beschrieben.

Evolution von E/E-Systemen In der heutigen E/E-Architekturmodellierung ist die Aktualisierung des Funktionsnetzes auf einen neuen Stand mit großem Aufwand verbunden. Dieser entsteht, da bei der Modellierung von Funktionen keine Trennung zwischen Innovationen und Übernahmen existiert.

Im Gegensatz dazu werden im E/E-System Produktlinien Engineering die Innovationen separat auf der Konzeptionellen Funktionsarchitektur-Ebene modelliert. Dadurch hat eine Aktualisierung des Funktionsnetzes viel weniger Rückwirkungen und lässt sich einfacher durchführen.

Modellstruktur und Variantenmanagement In der heutigen E/E-Architekturmodellierung sind das Funktionsnetz und die Hardware nur über Mappings der Funktionen auf die Hardware verbunden. Bei Änderungen an Funktionen oder Hardware muss meist die jeweils andere Modellierungsebene angepasst werden. Fehler bei dieser Anpassung werden nur implizit durch fehlerhafte Mappings erkennbar.

Im E/E-System Produktlinien Engineering hingegen existiert eine E/E-System-Sicht im Modell mit der klar nachvollziehbar ist, welche Funktionen und Hardware zu welchem E/E-System gehören. Werden Funktionen oder Hardware geändert, lassen sich Fehler bei der Anpassung der jeweils anderen Ebene explizit über diese Sicht erkennen.

Wiederverwendung von E/E-Systemen In der heutigen E/E-Architekturmodellierung gibt es keine Beziehung zwischen E/E-Architekturmodellen, in denen die gleichen E/E-Systeme durch Export und Import wiederverwendet wurden. Wird somit ein E/E-System in einem dieser E/E-Architekturmodelle geändert, dann erzeugt dies eine Inkonsistenz zu den anderen Wiederverwendungen dieses E/E-Systems.

Mit dem E/E-System Produktlinien Engineering wird bei einer Wiederverwendung eine Beziehung zwischen dem E/E-System und der Wiederverwendung des E/E-Systems hergestellt. Wird eine Änderung an dem E/E-System vorgenommen, werden die Unterschiede zum wiederverwendeten E/E-System deutlich gemacht. Eine Werkzeugunterstützung kann benutzt werden um die Änderung an dem E/E-System in die Wiederverwendungen zu übernehmen (Anwendungsfall 4.10).

6.2.5 Modellqualität

Im Vergleich zur heutigen E/E-Architekturmodellierung wird durch das E/E-System Produktlinien Engineering die Qualität der E/E-Architekturmodellierung verbessert. Dies lässt sich an den nachfolgend beschriebenen Punkten festmachen.

Evolution von E/E-Systemen Durch die Einführung der Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 4.4.1) wird die Beschreibung einer Funktion in einer abstrakten Weise ermöglicht (Designentscheidung 4.1). Diese abstrakte Beschreibung der Funktionen erlaubt es dem E/E-Architekten, komplexe Zusammenhänge schnell zu verstehen und damit die Evolution eines E/E-Systems schlagkräftig zu bewerten. Damit erhöht sich im Vergleich zur heutigen E/E-Architekturmodellierung die Qualität des E/E-Architekturmodells, da der E/E-Architekt durch das E/E-Architekturmodell eine verbesserte Entscheidungsgrundlage hat. Außerdem ermöglicht die Modellierung auf der Konzeptionellen Funktionsarchitektur-Ebene eine Auflösung der Überschneidung der Funktionen auf dem Funktionsnetz. Dadurch werden die Abhängigkeiten, die zwischen den Funktionen auf dem Funktionsnetz bestehen, für den E/E-Architekten transparent gemacht. Die Beschreibung auf der Konzeptionellen Funktionsarchitektur-Ebene hat zudem einen generationenübergreifenden Charakter. Dies ist ein weiterer Faktor, der die Modellqualität erhöht.

Ferner wird durch das Feature Oriented Architecture Refactoring (FOAR, Abschnitt 4.4.2) eine Methode eingeführt, die eine geregelte Evolution der E/E-Systeme mit dedizierten Operationen ermöglicht. Dabei wird das E/E-Architekturmodell um zusätzlichen Informationen angereichert. Hierbei erhöht sich die Modellqualität, da die genaue Verteilung der Funktionsanteile auf die Elemente der Konzeptionellen Funktionsarchitektur modelliert ist.

Modellstruktur und Variantenmanagement Verglichen mit der heutigen E/E-Architekturmodellierung wurde diese um ein E/E-Systemmodell (Abschnitt 4.4.3) angereichert. Durch die Einführung dieses Systemmodells wird ein E/E-System vollständig im E/E-Architekturmodell beschrieben, d. h. einschließlich der verwendeten Hardware bzw. E/E-Module und der funktionalen Anteile. Durch diese vollständige Erfassung der E/E-Systeme erhöht sich die Modellqualität. Gleichzeitig wird die Konfiguration der E/E-Architekturmodell-Varianten geändert, wodurch transparent wird, welche E/E-Systeme welcher E/E-Architekturmodell-Variante zugeordnet sind. Auch dies erhöht die Qualität des E/E-Architekturmodells, denn in der heutigen Vorgehensweise muss diese Zuordnung von den einer E/E-Architekturmodell-Variante zugeordneten Funktionsbeiträgen und Hardware-Anteilen aufwendig abgeleitet werden. Darüber hinaus werden erstmalig die Beziehungen zwischen den E/E-Systemen, die bisher Expertenwissen waren, modelliert. Dies erlaubt, die Abhängigkeiten besser zu erfassen und beim Variantenmanagement auf Werkzeuge zurückzugreifen, die die Konfiguration absichern. Dies reduziert die Fehleranfälligkeit und erhöht die Qualität des Modells.

Wiederverwendung von E/E-Systemen Auch durch die Konzepte, die für die Wiederverwendung von E/E-Systemen eingeführt werden, erhöht sich die Modellqualität. Grund hierfür ist, dass für die Wiederverwendung mehrere unterschiedliche E/E-Architekturen in einem E/E-Architekturmodell gepflegt werden. Dadurch wird dem E/E-Architekten einfacher ersichtlich, welche Variante eines E/E-Systems in welcher E/E-Architektur verbaut ist. Somit ist erstmals die Verblockung von E/E-Systemen einschließlich ihrer funktionalen Anteile modellierbar.

6.3 Quantitative Evaluierung mittels einer Fallstudie

Nachfolgend werden die Ergebnisse einer Fallstudie vorgestellt, die im Rahmen des Fallbeispiels der Arbeit (Abschnitt 6.1) durchgeführt wird. Für die Modellierung der Evolution von E/E-Systemen (Abschnitt 6.3.1), der Variantenkonfiguration (Abschnitt 6.3.2) und der Wiederverwendung von E/E-Systemen

(Abschnitt 6.3.3) wird eine Expertenstudie durchgeführt. Ein Modellierungsexperte, ein Field-Application-Engineer des Modellierungswerkzeugherstellers mit einer Erfahrung von 4,5 Jahren in diesem Werkzeug, wird mit den entsprechenden Szenarien konfrontiert und erhält die notwendigen Informationen für die Modellierung, d. h.

- Listen von Schnittstellen,
- Tabellarische Zuordnung von E/E-Komponentenbeiträgen zu E/E-Systemen,
- E/E-Systemdiagramme, usw.

Sowohl die Zeit, die der Experte mit der Vorgehensweise der heutigen E/E-Architekturmodellierung benötigt als auch die Zeit die mit den Konzepten dieser Arbeit bzw. den prototypischen Implementierungen werden anschließend erfasst.

Eine Durchführung der Studie mit vielen Teilnehmern ist nicht möglich. Der Umgang mit dem E/E-Architekturmodellierungswerkzeug ist nicht trivial und es sind umfangreiche Erfahrung oder sehr intensive Schulung und Einarbeitung erforderlich. Erforderlich ist auch eine Einarbeitung in die Modellierungssystematik, die bei Mercedes-Benz Cars angewendet wird, und ein Kenntnis über die Modellierung der E/E-Architektur-Domäne Chassis/Fahrerassistenz. Eine höhere Anzahl von Personen mit diesen spezifischen Kenntnissen ist nicht verfügbar. Dennoch lässt sich die Studie aufgrund der folgenden Punkte als charakteristisch einstufen:

- Die Modellierung im E/E-Architekturmodellierungswerkzeug wird von Experten durchgeführt, meist von Field-Application-Engineers oder spezialisierten Dienstleistern.
- Zwar kann die Erfahrung anderer Modellierer mit dem E/E-Architekturmodellierungswerkzeug abweichen, aber sie wirkt sich sowohl auf die Durchführung mit der Vorgehensweise der heutigen E/E-Architekturmodellierung als auch auf die Durchführung mit dem E/E-System Produktlinien Engineering aus.
- Die für die Studie wichtige Kenntnis der Modellierungssystematik bei Mercedes-Benz Cars und der Modellierung der E/E-Architektur-Domäne Chassis/Fahrerassistenz liegt vor.

Unabhängig davon wird eine Fallstudie zur Evaluierung der Automatisierung durchgeführt (Abschnitt 6.3.4). Die Diskussion der qualitativen Evaluierung findet sich in Abschnitt 6.4.

6.3.1 Evolution von E/E-Systemen

Für die Evaluierung der E/E-System-Evolution werden die unterschiedlichen Vorgehensweisen für alle Operationen durch den Modellierungsexperten modelliert. Die Modellierung erfolgt im Rahmen des Fallbeispiels der Arbeit (Abschnitt 6.1) unter Nutzung der aufgebauten Konzeptionellen Funktionsarchitektur.

Bei der Evaluierung wurde ein Schritt in der Vorgehensweise für das Feature Oriented Architecture Refactoring (FOAR, Abschnitt 4.4.2) nicht durchgeführt. Das Einpflegen einer Änderung, das ggf. auf einer anderen Modellierungsebene erfolgt, muss bei einer Weiterentwicklung mit oder ohne FOAR genau gleich durchgeführt werden. Deshalb wird dieses Einpflegen für diese Evaluierung als konstante Größe veranschlagt.

Erstellen neuer Funktionsumfänge

Das Erstellen (Anwendungsfall 4.3) von neuen Funktionsumfängen für ein E/E-Modul wird anhand einer Funktion (COLLISIONMITIGATIONSYSTEM, CMS) aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) untersucht.

In der heutigen E/E-Architekturmodellierung werden für diese Funktion zwei Funktionen auf der SW-Ebene angelegt.

- Eine dieser Funktionen enthält hierbei alle Umfänge dieser Funktion für das E/E-System CPA.
- Die andere Funktion enthält alle Umfänge dieser Funktion für das E/E-Systemen SDTR als auch die Umfänge für das E/E-System FAP.

Für beide Funktionen erhält der Modellierungsexperte ein Diagramm vom E/E-Architekten, das die Funktionsbeiträge mit deren Schnittstellen dieser Funktion beschreibt. Die Modellierung wird nach dem Anlegen eines SW-Projekts auf der SW-Ebene jeweils durchgeführt, indem

- eine **Software-Composition** angelegt,
- für diese ein **Softwareinstanz-Diagramm** erzeugt,
- im Diagramm die **Software-Compositions** inklusive **Types** und **Prototypes** für die entsprechenden Funktionsbeiträge angelegt,
- entsprechend viele **Receive-** und **Provided-Ports** angelegt,
- die **Ports** entsprechend umbenannt und verbunden,
- und für die **Assemblies** ein **Interface** mit **Datenelement** angelegt und

- zuletzt `Delegation Ports` erstellt

werden. Bei der Modellierung der zweiten Funktion werden durch den Modellierungsexperten bereits für die erste Funktion erstellte `Interfaces` wiederverwendet. Deshalb kann hier auf Werkzeugunterstützung zurückgegriffen werden.

Im E/E-System Produktlinien Engineering wird durch den Modellierungsexperten

- die Konzeptionelle Funktionsarchitektur-Ebene (Abschnitt 4.4.2) nach der Vorlage eines Diagramms durch E/E-Architekten modelliert,
- diese Konzeptionelle Funktionsarchitektur durch das Anlegen eines Funktionen-Modells im Rahmen des Feature Oriented Architecture Refactoring (Abschnitt 4.4.2) annotiert. Diese Annotierung wird durch das Initialisieren und Ausführen einer Metrik (Abschnitt 5.4.2) für alle drei Umfänge der Funktion, d. h. CPA, SDTR und FAP durchgeführt.
- Anschließend werden die entsprechenden Merkmale zu den entsprechenden E/E-Systemen hinzugefügt.

Das Erfassen der Modellierungsdauer (Tabelle 6.2) zeigt für die Erstellung von neuen Funktionsumfängen eine Erhöhung der Modellierungsdauer um 1807s, was tendenziell eine Verringerung der Modellierungseffizienz um den Faktor 2 entspricht. Hierbei ist der Faktor gerundet und wird wie folgt bei Verringerung der Modellierungseffizienz

$$\text{Faktor} = \frac{\text{Modellierungsdauer}_{\text{Heutige E/E-Architekturmodellierung}}}{\text{Modellierungsdauer}_{\text{E/E-System Produktlinien Engineering (neu)}}} \quad (6.1)$$

und wie folgt bei Steigerung der Modellierungseffizienz:

$$\text{Faktor} = \frac{\text{Modellierungsdauer}_{\text{E/E-System Produktlinien Engineering (neu)}}}{\text{Modellierungsdauer}_{\text{Heutige E/E-Architekturmodellierung}}} \quad (6.2)$$

ermittelt.

Fusion

Nach der Erstellung der Funktionsinhalte werden die Umfänge der E/E-Systeme SDTR und FAP des E/E-Moduls Radarsysteme fusioniert (Anwendungsfall 4.7).

Tabelle 6.2: Vergleich der Modellierungsdauer für das Erstellen von neuen Funktionsumfängen (Anwendungsfall 4.3)

Heutige E/E-Architekturmodellierung				E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.3 (Erstellen)			
Tätigkeit	Daten-grundlage	#	Zeit [s]	Tätigkeit	Daten-grundlage	#	Zeit [s]
neues SW-Projekt anlegen	-	2 Modellartefakte angelegt	89	Konzeptionelle Funktionsarchitektur auf LA-Ebene modellieren	Diagramm	42 Assemblies, 86 Ports, 12 Funktionsblöcke, 12 Types, 35 Interfaces, 35 Datenelemente	3090
Funktion auf SW-Ebene modellieren: VE-SYS CPA	Systemdiagramm	4 Types, 4 Prototypes, 4 Instanzen, 23 Ports, 16 Interfaces, 16 Datenelemente, 23 Assemblies, 15 Delegation Ports, 1 Diagramm	1609	Metrik zur Erstellung des Feature-Modell ausführen: CMS_CPA	Annotiertes Diagramm	45 Mapping und 3 CF	869
Funktion auf SW-Ebene modellieren: VE-SYS CMS	Systemdiagramm	6 Types, 6 Prototypes, 6 Instanzen, 29 Ports, 2 Interfaces, 2 Datenelemente, 25 Assemblies, 12 Delegation Ports, 1 Diagramm	1176	Metrik zur Erstellung des Feature-Modell ausführen: CMS_DTR	Annotiertes Diagramm	50 Mappings und 3 CF	248
				Metrik zur Erstellung des Feature-Modell ausführen: CMS_FAP	Annotiertes Diagramm	65 Mappings und 4 CF	269
				Metrik zur Erstellung der E/E-Systeme ausführen	-	3 CF und 6 Mappings	116
$\Sigma=2785s$				$\Sigma=4592s$			

Für das Erstellen neuer Funktionsumfänge wird die Modellierungsdauer mit der Vorgehensweise der heutigen E/E-Architekturmodellierung dem E/E-System Produktlinien Engineering gegenübergestellt. Hierzu wird die Funktion COLLISIONMITIGATIONSYSTEM (CMS) aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) modelliert, die für die E/E-Systeme *Collision Prevention Assist* (CPA), Solo-Distrionic (SDTR) und Fahrerassistenzpaket (FAP) unterschiedliche Umfänge aufweist. # gibt die Anzahl an neu modellierten Modellartefakten an. VE_SYS Vorentwicklungs-System (interne Namenskonvention zur Modellierung von Innovationen); CF CustomerFeature; SW Software (Modellierungsebene); LA Logical Architecture (Modellierungsebene)

In der heutigen E/E-Architekturmodellierung erhält der Modellierungsexperte ein Diagramm, das die fusionierte Funktion COLLISIONMITIGATIONSYSTEM und alle ihre Funktionsbeiträge einschließlich der Schnittstellen beschreibt. Anhand dieser Vorlage führt der Modellierungsexperte die Modellierung durch indem

- die Funktionsbeiträge, die sich nicht ändern, übernommen werden und
- der fusionierte Funktionsbeitrag neu modelliert wird.

Vor der manuellen Modellierung des fusionierten Funktionsbeitrags versucht der Modellierungsexperte eine kurze Zeit lang erfolglos eine Werkzeugunterstützung für diese Aufgabe zu verwenden.

Tabelle 6.3: Vergleich der Modellierungsdauer für die Fusion (Anwendungsfall 4.7)

Heutige E/E-Architekturmodellierung				E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.7 (Fusion)			
Tätigkeit	Daten- grundlage	#	Zeit [s]	Tätigkeit	Daten- grundlage	#	Zeit [s]
VE-SYS CMS_Merge	System- diagramm	1 SWC mit 15 Ports	512	CMS_DTR+FAP	-	6 CF und 110 Mappings	40
				Systeme erstellen	-	3 CF mit 3 Mappings	59
$\Sigma=512s$				$\Sigma=99s$			

Für die Fusion wird die Modellierungsdauer in der heutigen E/E-Architekturmodellierung dem E/E-System Produktlinien Engineering gegenübergestellt. Hierzu wird die Funktion COLLISIONMITIGATIONSYSTEM (CMS) aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Die Anteile dieser Funktion für die E/E-Systeme Solo-Distrionic (SDTR) und Fahrerassistenzpaket (FAP) werden hierbei fusioniert. # gibt die Anzahl an neu modellierten Modellartefakten an. VE_SYS Vorentwicklungs-System (interne Namenskonvention zur Modellierung von Innovationen); CF CustomerFeature; SWC Software Component

Die gleiche Aufgabe führt der Modellierungsexperte im E/E-System Produktlinien Engineering durch, indem

- die passende Metrik initialisiert und anschließend ausführt wird (Abschnitt 5.4.2). Diese passt das Funktionen-Modell entsprechend an.
- Abschließend werden die neuen Merkmale den entsprechenden E/E-Systemen hinzugefügt.

Für beide Vorgehensweisen wurde die Modellierungsdauer erfasst (Tabelle 6.3). Mit dem E/E-System Produktlinien Engineering reduziert sich die Modellierungsdauer um 413s, was einer Steigerung der Modellierungseffizienz um den Faktor 5 entspricht.

Aufspalten

Die fusionierte Funktion wird aufgespalten um für das E/E-Modul Radarsysteme wieder die Umfänge für die E/E-Systeme SDTR und FAP zu erzeugen (Anwendungsfall 4.6).

Um diese Modellierung mit der Vorgehensweise der heutigen E/E-Architekturmodellierung durchzuführen, erhält der Modellierungsexperte dafür eine Vorlage vom E/E-Architekten. Anschließend werden

- eine Kopie der fusionierten Software Components erstellt,

Tabelle 6.4: Vergleich der Modellierungsdauer für das Aufspalten (Anwendungsfall 4.6)

Heutige E/E-Architekturmodellierung				E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.6 (Aufspalten)			
Tätigkeit	Daten-grundlage	#	Zeit [s]	Tätigkeit	Daten-grundlage	#	Zeit [s]
VE-SYS CMS_Split	System-diagramm	2 Types, 2 Prototypes, 2 Instanzen, 28 Ports, 16 Assemblies	315	CMS_DTR+CMS_FAP	-	160 Mappings und 10 CF	391
				Systeme erstellen	-	4 CF mit 6 Mappings	75
$\Sigma=315s$				$\Sigma=466s$			

Für das Aufspalten wird die Modellierungsdauer in der heutigen E/E-Architekturmodellierung dem E/E-System Produktlinien Engineering gegenübergestellt. Hierzu wird die Funktion COLLISIONMITIGATIONSYSTEM (CMS) aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Die Anteile dieser Funktion für die E/E-Systeme Solo-Distrionic (SDTR) und Fahrerassistenzpaket (FAP) werden hierbei aus einer fusionierten Funktion erzeugt. # gibt die Anzahl an neu modellierten Modellartefakten an. VE_SYS Vorentwicklungssystem (interne Namenskonvention zur Modellierung von Innovationen); CF CustomerFeature; SWC Software Component

- die notwendigen Ports verbunden und
- die übrigen Ports gelöscht.

Im E/E-System Produktlinien Engineering

- initialisiert der Modellierungsexperte eine Metrik (Abschnitt 5.4.2) nach Vorlage des E/E-Architekten. Diese beschreibt, welche Umfänge zu dem E/E-System SDTR bzw. FAP gehören.
- Anschließend werden die Merkmale entsprechend den E/E-Systemen hinzugefügt.

Ein Vergleich der Modellierungsdauer (Tabelle 6.4) zeigt eine Erhöhung um 151s, was einer Reduzierung der Modellierungseffizienz um den Faktor 2 entspricht.

Verschieben

Für das Verschieben (Anwendungsfall 4.5) werden die Umfänge der CMS-Funktion des CPA auf das E/E-Modul ESP verschoben. In der heutigen E/E-Architekturmodellierung wird dafür das Mapping der Master-Software Component angepasst. Für das E/E-System Produktlinien Engineering werden die Merkmale dem E/E-System ESP hinzugefügt.

Ein Vergleich der Modellierungsdauer (Tabelle 6.5) zeigt mit 5s nur geringe Unterschiede zugunsten der heutigen E/E-Architekturmodellierung.

Tabelle 6.5: Vergleich der Modellierungsdauer für das Verschieben (Anwendungsfall 4.5)

Heutige E/E-Architekturmodellierung				E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.5 (Verschieben)			
Tätigkeit	Daten-grundlage	#	Zeit [s]	Tätigkeit	Daten-grundlage	#	Zeit [s]
VE-SYS CPA_ESP	System-diagramm	1 Mapping	37	Systeme erstellen	-	2 CF mit 2 Mappings	42
$\Sigma=37s$				$\Sigma=42s$			

Für das Verschieben wird die Modellierungsdauer in der heutigen E/E-Architekturmodellierung dem E/E-System Produktlinien Engineering gegenübergestellt. Hierzu wird die Funktion COLLISIONMITIGATIONSYSTEM (CMS) aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Die Anteile dieser Funktion für das E/E-System *Collision Prevention Assist* (CPA) werden hierbei zum E/E-System ESP verschoben. # gibt die Anzahl an neu modellierten Modellartefakten an. VE_SYS Vorentwicklungs-System (interne Namenskonvention zur Modellierung von Innovationen); CF CustomerFeature

Entfernen

Um das Entfernen von Funktionsumfängen von einem E/E-Modul (Anwendungsfall 4.4) zu evaluieren, werden die Umfänge der CMS-Funktion des CPA vom E/E-Modul ESP entfernt. Dafür werden in der heutigen E/E-Architekturmodellierung die Mappings der Funktionsbeiträge dieser Funktion entfernt. Im E/E-System Produktlinien Engineering wird die Zuordnung der Merkmale zum E/E-System ESP wieder entfernt.

Wird Modellierungsdauer für beide Vorgehensweisen verglichen (Tabelle 6.6), dann zeigt sich mit 11s nur ein geringer Unterschied der beiden Vorgehensweisen zugunsten des E/E-System Produktlinien Engineerings .

6.3.2 Variantenkonfiguration im Variantenmanagement

Um die Variantenkonfiguration (Anwendungsfall 4.8) zu evaluieren werden drei E/E-Architekturmodell-Varianten in den unterschiedlichen Vorgehensweisen konfiguriert. Hierbei wird

- eine MIN-Variante, die die minimale Ausstattung,
- eine MID-Variante, die eine mittlere Ausstattung und
- eine MAX-Variante, die eine maximale Ausstattung darstellt,

konfiguriert. Dabei werden alle E/E-Systeme der Domäne Chassis/Fahrerassistenz herangezogen und für jedes E/E-System entschieden und modelliert, ob ein E/E-System der E/E-Architekturmodell-Variante zugeordnet wird und

Tabelle 6.6: Vergleich der Modellierungsdauer für das Entfernen (Anwendungsfall 4.4)

Heutige E/E-Architekturmodellierung				E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.4 (Entfernen)			
Tätigkeit	Daten- grundlage	#	Zeit [s]	Tätigkeit	Daten- grundlage	#	Zeit [s]
VE-SYS CPA_ESP	-	1 Mapping	15	Systeme löschen	-	2 CF mit 2 Mappings	4
$\Sigma=15s$				$\Sigma=4s$			

Für das Entfernen wird die Modellierungsdauer der heutigen E/E-Architekturmodellierung dem E/E-System Produktlinien Engineering gegenübergestellt. Hierzu wird die Funktion COLLISIONMITIGATIONSYSTEM (CMS) aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Die Anteile dieser Funktion für das E/E-Systeme *Collision Prevention Assist* (CPA) werden hierbei vom E/E-System ESP entfernt. # gibt die Anzahl an neu modellierten Modellartefakten an. VE_SYS Vorentwicklungs-System (interne Namenskonvention zur Modellierung von Innovationen); CF CustomerFeature

wenn ja, welche E/E-Komponentenbeiträge des E/E-Systems davon betroffen sind. Um die Entscheidung durchführen zu können, müssen Kenntnisse der E/E-Systeme vorliegen. Aus diesem Grund trifft der E/E-Architekt mit seinem Architekturwissen die Entscheidung während der Modellierer diese im Modellierungswerkzeug umsetzt.

Abhängig von der Vorgehensweise werden unterschiedliche Tätigkeiten durchgeführt, wobei einige Tätigkeiten einmalig und andere Tätigkeiten bei jeder Konfiguration einer E/E-Architekturmodell-Variante durchgeführt werden müssen. In der heutigen E/E-Architekturmodellierung

- wählt der E/E-Architekt aus einer Liste aller E/E-Komponentenbeiträge eine Menge an E/E-Komponentenbeiträgen aus, die der entsprechenden E/E-Architekturmodell-Variante zuzuordnen sind. Das Ergebnis wird in Form einer Tabelle dem Modellierungsexperten übergeben.
- Der Experte startet mit einem einmaligen Vorbereitungsschritt, in dem er für jede Innovation ein **Set** manuell angelegt und für die Übernahme-Funktionen ein **Set** durch eine Metrik anlegen lässt.
- Für die Konfiguration einer E/E-Architekturmodell-Variante wird der Modellierungsexperte zuerst anhand der Konfigurations-Tabelle des E/E-Architekten modellieren, ob eine Funktion in der E/E-Architekturmodell-Variante verwendet wird. Wenn eine Funktion in der E/E-Architekturmodell-Variante verwendet wird, ordnet er das **Set**, das alle E/E-Komponentenbeiträge dieser Funktion enthält, der E/E-Architekturmodell-Variante zu.

- Als zweites müssen diejenigen E/E-Komponentenbeiträge, die zwar zur Funktion gehören aber nicht der E/E-Architekturmodell-Variante zugeordnet werden sollen, von der Variantenkonfiguration ausgeschlossen werden. Dazu wird ein **Exclude-Set** modelliert, dessen Inhalt von der Variantenbildung ausgeschlossen wird. Die entsprechenden E/E-Komponentenbeiträge werden dann dem **Exclude-Set** zugeordnet.

Im E/E-System Produktlinien Engineering wird erst einmalig das E/E-Systemmodell umgesetzt (Abschnitt 5.2.2) und anschließend die Konfiguration der E/E-Architekturmodell-Varianten durchgeführt. Um das E/E-Systemmodell zu erstellen,

- ordnet der E/E-Architekt tabellarisch alle E/E-Komponentenbeiträge E/E-Systemen zu,
- beschreibt die Beziehungen zwischen den E/E-Systemen in Form eines Diagramms und
- ordnet die E/E-Module den E/E-Systemen zu.

Diese Informationen werden durch den Modellierungsexperten modelliert, indem

- die E/E-Systeme als **CustomerFeature**,
- deren Beziehungen zueinander,
- die **Mappings** zu den entsprechenden E/E-Komponentenbeiträgen auf der SW-Ebene bzw. auf der LA-Ebene für Innovationen und
- **Links** zu den dazugehörigen E/E-Modulen angelegt

werden. Nach dieser Vorbereitung werden für jede E/E-Architekturmodell-Variante

- durch den E/E-Architekten ausgewählt, welches E/E-System in der E/E-Architekturmodell-Variante enthalten ist und
- durch den Modellierungsexperten in der **Vorlagenansicht** diese E/E-Systeme ausgewählt.

Sowohl die benötigte Zeit für die Tätigkeiten des E/E-Architekten als auch die des Modellierungsexperten werden für beide Vorgehensweisen erfasst (Tabelle 6.7). Im Vergleich erhöht sich die Zeit für die Vorbereitung sowohl für den E/E-Architekten (von 0s auf 1631s, einmalige Tätigkeiten) als auch für den Modellierungsexperten (von 336s auf 2869s, einmalige Tätigkeiten) deutlich. Im Gegensatz dazu reduziert sich die durchschnittliche Zeit

für die Konfiguration einer E/E-Architekturmodell-Variante, sowohl für den E/E-Architekten (von 362s auf 75s) als auch für den Modellierungsexperten (von 1156s auf 112s). Als Trend ergibt sich eine Steigerung der Effizienz für die Konfiguration einer E/E-Architekturmodell-Variante um den Faktor 5 für den E/E-Architekten und um 10 für den Modellierungsexperten. Bezieht man den hohen Initialaufwand mit ein, dann hat das E/E-System Produktlinien Engineering für den E/E-Architekten ab 6 konfigurierten E/E-Architekturmodell-Varianten und für den Modellierungsexperten ab 3 E/E-Architekturmodell-Varianten einen zeitlichen Vorteil. Insgesamt lohnt sich somit die Einführung von Anwendungsfall 4.8 ab 4 konfigurierten E/E-Architekturmodell-Varianten. In der Praxis werden zum Vergleich für E/E-Architektur-Plattform-Entwicklungsprojekte zwischen 3 und 13 E/E-Architekturmodell-Varianten betrachtet (Abschnitt 3.2.4). Die Konfiguration der E/E-Architekturmodell-Varianten wird bei den Modellständen (z. B. 4 mal im Jahr) wiederholt.

6.3.3 Wiederverwendung von E/E-Systemen

Um die Wiederverwendung von E/E-Systemen zu evaluieren, wird der Modellierungsexperte eine Modellierung anhand des Fallbeispiels der Arbeit (Abschnitt 6.1) durchführen. Dazu wird die Wiederverwendung des E/E-Systems FAP in die Vorgänger-E/E-Architektur-Plattform untersucht.

- Für die Modellierung in der heutigen E/E-Architekturmodellierung wird dabei das FAP in ein separates Modell der Baureihe 213 (Mercedes-Benz E-Klasse 2016), die ein Vertreter dieser E/E-Architektur-Plattform ist, wiederverwendet.
- Im E/E-System Produktlinien Engineering wird diese Untersuchung in einer weiteren Produktlinie des selben Modells durchgeführt.

Wiederverwendung von E/E-Systemmodellen

Um die Wiederverwendung (Anwendungsfall 4.9) in der heutigen E/E-Architekturmodellierung zu realisieren, muss das E/E-System FAP durch den Modellierungsexperten manuell exportiert und im separaten Modell importiert werden. Da die Vorgänge nicht automatisiert sind, gestaltet sich die Durchführung als schwierig und fehleranfällig.

- Der Modellierungsexperte beginnt mit der Sammlung der zum E/E-System FAP gehörenden Modellartefakte. Diese werden einer vom E/E-Architekten bereitgestellten Matrix, in der die entsprechenden E/E-Komponentenbeiträge aufgelistet sind, entnommen. Es wird

Tabelle 6.7: Vergleich der Modellierungsdauer für die Variantenkonfiguration (Anwendungsfall 4.8)

Heutige E/E-Architekturmodellierung					E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.8 (Variantenkonfiguration mit E/E-Systemmodellen)				
Rolle	Tätigkeit	Daten- grundlage	#	Zeit [s]	Rolle	Tätigkeit	Daten- grundlage	#	Zeit [s]
A	Konfiguration MIN-Var.	KB-Liste	7 Funktionen, 20 KB	209	A (E)	KB zu E/E- Systeme zuordnen	KB-E/E-Sys.- Matrix mit Varianten- konfiguration	163 Komponentenbeiträge von 32 Funktionen zu 21 Systemen zugeordnet	942
	Konfiguration MID-Var.		14 Funktionen, 52 KB	372		Systembez. erstellen	Liste E/E- Systeme	15 Systeme (4 variant)	508
	Konfiguration MAX-Var.		16 Funktionen, 65 KB	504		E/E-Module zuordnen	Liste E/E-Module	15 Systeme (4 variant)	181
M (E)	Vorbereitung		19 E/E-Sys., 18 Inno.-Sys.	336	M (E)	E/E-Systeme anlegen	Liste E/E- Systeme	25 CF für 15 Systeme (4 davon variant)	458
M	Konfiguration MIN-Var.	Tabelle einer KB- Liste mit Varianten- konfiguration	36 Mappings in Excl.-Set	887	M (E)	Systembez. anlegen	Handschriftliches Diagramm	24 Beziehungen in einem Diagramm	1045
	Konfiguration MID-Var.		70 Mappings in Excl.-Set	1121		KB zuordnen	Tabelle mit Zuordnung	219 Mappings SW/LA auf CF	1057
	Konfiguration MAX-Var.		43 Mappings in Excl.-Set	1461		E/E-Module zuordnen	Tabelle mit Zuordnung	18 Links von E/E-Systemen zu E/E-Modul	309
A	Konfiguration MIN-Var.	Ausdruck Variantenvorlage des Architektur- modells			A	Konfiguration MIN-Var.		5 (1 variant)	59
	Konfiguration MID-Var.					Konfiguration MID-Var.		9 (1 variant)	75
	Konfiguration MAX-Var.					Konfiguration MAX-Var.		9 (4 variant)	91
M	Konfiguration MIN-Var.	Ausgefüllte Variantenvorlage			M	Konfiguration MIN-Var.		5 (1 variant)	149
	Konfiguration MID-Var.					Konfiguration MID-Var.		9 (1 variant)	91
	Konfiguration MAX-Var.					Konfiguration MAX-Var.		9 (4 variant)	95
A	Variantenkonfiguration: $\Sigma=1085s$ $\sigma=362s$				A	Einmalig: $\Sigma=1631s$; Variantenkonfiguration: $\Sigma=225s$ $\sigma=75s$; Gesamt: $\Sigma=1856s$ $\sigma=619s$			
M	Einmalig: $\Sigma=336s$; Variantenkonfiguration: $\Sigma=3469s$ $\sigma=1156s$				M	Einmalig: $\Sigma=2869s$; Variantenkonfiguration: $\Sigma=335s$ $\sigma=112s$; Gesamt: $\Sigma=3204s$ $\sigma=1068s$			

Für die Variantenkonfiguration wird die Modellierungsdauer mit der heutigen E/E-Architekturmodellierung dem E/E-System Produktlinien Engineering gegenübergestellt. Dabei werden drei E/E-Architekturmodell-Varianten für die E/E-Systeme der Domäne Chassis/Fahrerassistenz im Rahmen des Fallbeispiels der Arbeit (Abschnitt 6.1) konfiguriert: Eine MIN-Variante mit minimaler Ausstattung, eine MID-Variante mit mittlerer Ausstattung und eine MAX-Variante mit maximaler Ausstattung. An der Variantenkonfiguration sind mehrere Rollen beteiligt: A E/E-Architekt, M Modellierer. Mit E sind Tätigkeiten gekennzeichnet, die einmalig durchgeführt werden müssen. # gibt die Anzahl an getroffenen Zuordnungen bzw. neu modellierter Modellartefakte an. KB E/E-Komponentenbeitrag; Inno. Innovation; Excl.-Set Exclude-Set; CF CustomerFeature; SW/LA Software/Logical Architecture (Modellierungsebene)

ein Set erstellt und die **Software-Compositions**, **Software-Types**, **Software-Prototypes** und **SW-HW-Mappings** diesem zuordnet.

- Daraufhin wird ein Export für dieses Set, das **Hardware-Paket** und das **Interface-Paket** ausgeführt.
- Da der Exporter Inkonsistenzen beim Export anzeigt, wird anschließend vom Modellierungsexperten eine Fehlersuche durchgeführt.
- Unter der Annahme, dass ein **Software-Type** vergessen wurde, wird für einen weiteren Exportversuch die Selektion der **Software-Types** erweitert
- und anschließend der Export durchgeführt.
- Bei der anschließenden Ausführung des Importers werden wieder Inkonsistenzen angezeigt und der Import automatisch abgebrochen.
- Dies umgeht der Modellierungsexperte, indem er auf interne Entwickler-Dokumentationen des Werkzeugherstellers zurückgreift und
- den automatischen Konsistenzcheck bei einem Import deaktiviert.
- Nach dem Import stellt sich heraus, dass einige exportierte **Assemblies** die Ursache für die Inkonsistenz waren. Diese **Assemblies** verbinden die exportierten **Software-Compositions** mit nicht exportierten **Software-Compositions**. Damit waren diesen **Assemblies** nur ein statt zwei **Ports** zugewiesen, was bei dem Konsistenzcheck einen Fehler hervorruft. Anschließend führt der Modellierungsexperte den Import des E/E-Moduls und der Funktionsbeiträge durch eine Drag&Drop-Aktion in die geeigneten Stellen der Modellstruktur durch.
- Danach stellt der Modellierungsexperte die Konsistenz her, indem die **Software-Prototypes**, die keinen **Software-Type** haben, gelöscht werden, die **Assemblies**, bei denen mindestens ein **Port** nicht zugewiesen ist, entfernt werden und die noch fehlenden **Mappings** der **Software-Compositions** hergestellt werden.
- Abschließend werden die importierten Anteile verbunden, indem die **Software-Compositions** mittels einer Werkzeugunterstützung (Port Connector) verbunden und die Hardware des E/E-Moduls an die entsprechenden Bussysteme angebunden werden.

Für das E/E-System Produktlinien Engineering wird vom Modellierungsexperten

- zuerst das E/E-Systemmodell (Abschnitt 4.4.3) für das E/E-System FAP angelegt.
- Anschließend wird die Metrik zur Erstellung von E/E-System-Modellcontainern (Abschnitt 5.4.2) initialisiert und ausgeführt.
- Der erzeugte E/E-System-Modellcontainer, eine **Reuse-Unit**, wird vom Modellierungsexperten eingecheckt (und damit zur Datenbank transferiert) und in einer weiteren **Productline** wiederverwendet.
- Anschließend wird die Metrik zur Integration des E/E-System-Modellcontainers (Abschnitt 5.4.2) initialisiert und ausgeführt.
- Abschließend werden die neuen Modellartefakte verbunden.

Für beide Vorgehensweisen wurde der Modellierungsexperte begleitet und die Modellierungsdauer erfasst (Tabelle 6.8). Das Verbinden der Modellartefakte musste hier nicht separat erfasst werden, da für diese Tätigkeit in beiden Vorgehensweisen analog vorgegangen wird. Mit der Vorgehensweise des E/E-System Produktlinien Engineerings reduziert sich die Modellierungsdauer in diesem konkreten Fall von 7177s auf 1393s, was einer Steigerung der Modellierungseffizienz um den Faktor 5 entspricht. Unter der Annahme, dass während der Modellierung keine Fehler in der heutigen Vorgehensweise aufgetreten wären, würde sich die Modellierungsdauer von 3303s auf 1393s (Faktor 2) reduzieren.

Austausch von E/E-Systemmodellen

Nach einer Änderung des E/E-Systems FAP müssen die Wiederverwendungen durch die neue Änderung ausgetauscht werden (Anwendungsfall 4.10). Als Änderung im E/E-System FAP werden zwei neue **Ports** und ein neuer Sensor angelegt bei gleichzeitigem Entfall eines weiteren Sensors.

In der heutigen E/E-Architekturmodellierung werden durch den Modellierungsexperten dafür

- die Software-Anteile exportiert, indem ein ECU-Extract (AUTOSAR-Export) erzeugt wird.
- Der anschließende Import in das separate Modell erlaubt ein Update der geänderten **Software-Composition** und importiert viele Modellartefakte des Kommunikationsmodells. Zusätzlich dazu werden einige **Ports** aufgrund unterschiedlicher IDs¹ der zu importierenden Modellartefakte gegenüber den Modellartefakten des Modells dupliziert.

¹Im E/E-Architekturmodell wird jedem Modellartefakt eine ID zugewiesen über die es eindeutig identifizierbar ist.

Tabelle 6.8: Vergleich der Modellierungsdauer für das Wiederverwenden (Anwendungsfall 4.9)

Heutige E/E-Architekturmodellierung				E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.9 (Wiederverwendung von E/E-Systemmodellen)			
Tätigkeit	Daten- grundlage	#	Zeit [s]	Tätigkeit	Daten- grundlage	#	Zeit [s]
Zusammensuchen der Artefakte	KB-E/E- Sys.-Matrix	1 Set	952	FAP-CF modellieren	KB-E/E- Sys.-Matrix	1 CF mit 29 Mappings	338
Export HW-Modul, SW-Anteile	-	14,9MB komprimierte Exportdatei	199	Metrik RU erstellen	-	1 RU mit 36 Kern- und 21 Randartefakten	36
Fehlersuche: Inkonsistenzen Export	-	-	939	Check-In, RU Wiederverwenden	-	-	118
Weiterer Exportversuch: neue Selektion	-	32,3MB komprimiertes Export-File	151	Modell Locken + Metrik Integration	-	36 Kern- und 21 Randartefakten	102
Weiterer Exportversuch: Exporter	-	-	631	Verbinden*	-	-	799
Importer ausführen	-	47 Mappings, 47 Composition-Tyes, 285 Sender- und 1012 Receiver-Porttypes, 47 Compositions, 285 Sender- und 1012 Receiver-Ports, 47 Composition-Prototypes, 285 Sender- und 1012 Receiver-Portprototypes, 12 ECUs erstellt	101				
Entwickler-Doku. nach Spezial-Flags durchsuchen und Flags zum Deakt. des autom. Konsistenzchecks setzen	-	-	562				
Ausführung des Importers ohne SW-Konsistenzcheck	-	-	631				
Import HW-Modul, SW-Anteile durch Drag&Drop	-	-	663				
Umsetzen der Drop-Aktionen durch Importer	-	-	59				
Konsistenz herstellen	-	-	1490				
Verbinden der SW über den Port-Connector und der HW durch Zuweisung des Bussystems an die Busanbindung	-	73 Assemblies, 12 Busanbindungen mit dem Bussystem verlinkt	799				
$\Sigma=7177s$; Ohne Fehler: 3303s				$\Sigma=1393s$			

Für die Wiederverwendung wird die Modellierungsdauer der heutigen E/E-Architekturmodellierung und dem E/E-System Produktlinien Engineering gegenübergestellt. Hierzu wird das E/E-System FAP des E/E-Moduls Radarsysteme aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Das E/E-System wird in der heutigen E/E-Architekturmodellierung in einem separaten Modell der Baureihe BR213 (Mercedes-Benz E-Klasse 2016) wiederverwendet, während im E/E-System Produktlinien Engineering das E/E-System FAP in einer weiteren **Productline** des selben Modells wiederverwendet wird. Die mit * gekennzeichnete Tätigkeit muss nicht separat erfasst werden, da sie analog zur heutigen E/E-Architekturmodellierung durchzuführen ist und deshalb vergleichbar ist. # gibt die Anzahl an neu modellierten Modellartefakten an. KB E/E-Komponentenbeitrag; HW Hardware, SW Software; CF CustomerFeature; RU ReuseUnit

- Diese duplizierten Ports werden anschließend durch den Modellierungsexperten gelöscht und
- die Änderungen gespeichert (Commit).
- Danach wird die neue Hardware exportiert,
- die nicht mehr benötigte Hardware manuell gelöscht
- und die neue Hardware importiert.
- Abschließend verbindet der Modellierungsexperte die neuen Ports und den Sensor entsprechend.

Um diesen Austausch durch den Modellierungsexperten im E/E-System Produktlinien Engineering durchzuführen,

- wird ein Check-out der Software-Types der zu ändernden Software-Composition durchgeführt, was einen Check-out des E/E-System-Modellcontainers (ReuseUnit) hervorruft.
- Nach der Änderung und
- einem Check-in der Änderung,
- wird der wiederverwendete E/E-System-Modellcontainer ausgetauscht. Dafür ersetzt der Modellierungsexperte durch die Werkzeugunterstützung die ReuseUnit mit der aktuellen Version und führt die Metrik zur Integration des E/E-System-Modellcontainers (Abschnitt 5.4.2) aus.

Für beide Vorgehen wurde die Modellierungsdauer erfasst (Tabelle 6.9). Da die Tätigkeiten für die Änderung und das Verbinden in beiden Vorgehensweisen analog durchgeführt werden, müssen sie nicht separat erfasst werden. Im Vergleich zeigt sich, dass die Modellierungsdauer mit dem E/E-System Produktlinien Engineering um 5646s sinkt, was einer Steigerung der Modellierungseffizienz um den Faktor 10 entspricht.

6.3.4 Evaluierung der Automatisierung

Nachfolgend werden die Ergebnisse der Fallstudie zur Evaluierung der Automatisierung beschrieben. Damit konzentriert sich diese Fallstudie darauf, wie stark eine Automatisierung die Modellierung der Anwendungsfälle (Abschnitt 4.4.5) unterstützen kann.

Tabelle 6.9: Vergleich der Modellierungsdauer für das Austauschen (Anwendungsfall 4.10)

Heutige E/E-Architekturmodellierung				E/E-System Produktlinien Engineering (neu): Anwendungsfall 4.10 (Austausch von E/E-Systemmodellen)			
Tätigkeit	Daten- grundlage	#	Zeit [s]	Tätigkeit	Daten- grundlage	#	Zeit [s]
Änderung modellieren*	Vorgabe Architekt	2 Ports	113	Check-Out der Typen der enthaltenen SWCs (automatischer Check-Out der RU)	-	-	19
Export SW-Anteile	-	65MB	1649	Änderung modellieren	Vorgabe Architekt	2 Ports, 1 Sensor	113
Import SW-Anteile	-	889 neue, 739 geänderte, 714 gelöschte Artefakte	3316	Check-In	-	-	236
Löschen der verdoppelten Anteile	-	89 Duplikate-Ports löschen	487	Update	-	2 Ports, 1 Sensor	123
Commit	-	-	243	Verbinden der SW und HW*	-	4 Delegation Ports, 6 Assemblies, 1 Busanbindungen mit dem Bussystem verlinkt	160
Export HW-Anteile	-	1 Sensor	65				
Manuelles Löschen des HW-Sensors	-	-	5				
Import HW-Anteile	-	1 Sensor	259				
Verbinden der SW und HW	-	4 Delegation Ports, 6 Assemblies, 1 Busanbindungen mit dem Bussystem verlinkt	160				
$\Sigma=6297s$				$\Sigma=651s$			

Für den Austausch wird die Modellierungsdauer der heutigen E/E-Architekturmodellierung dem E/E-System Produktlinien Engineering gegenübergestellt. Hierzu wird das E/E-System FAP des E/E-Moduls Radarsysteme aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Das E/E-System wird geändert und die bereits bestehende Wiederverwendung im separaten Modell bzw. in einer weiteren **Productline** durch die geänderte Version ausgetauscht. Die mit * gekennzeichneten Tätigkeiten müssen nicht separat erfasst werden, da sie analog durchzuführen und deshalb vergleichbar sind. # gibt die Anzahl an neu modellierten Modellartefakten an. KB E/E-Komponentenbeitrag; HW Hardware, SW Software; CF CustomerFeature; RU ReuseUnit

Dafür wird eine manuelle Modellierung der Anwendungsfälle mit der prototypischen Umsetzung (Kapitel 5) verglichen. Dieser Vergleich wird von unterschiedlichen Personen durchgeführt, die fortgeschrittene Kenntnisse des Modellierungswerkzeugs und E/E-Architekturwissen über die Domäne Chassis/Fahrerassistenz bzw. einzelner Aspekte besitzen. Für beide Vorgehensweisen wird die Modellierungsdauer erfasst (Tabelle 6.10).

Relation zwischen Konzeptioneller Funktionsarchitektur- und Funktionsnetz-Ebene Um die Automatisierung für die Relation zwischen den Modellierungsebenen zu evaluieren, wird die Objektschnittstelle des FAP aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) untersucht. Das Herstellen der Relation (Anwendungsfall 4.1) ist mit der Automatisierung bis zu 12 mal schneller während das Entfernen etwa gleich schnell ist. Die Evolution (Anwendungsfall 4.2) der Relation ist mit der Automatisierung tendenziell 5 - 17 mal schneller.

Evolution von E/E-Systemen Die Automatisierung der E/E-System-Evolution wird anhand der E/E-Systeme CPA, SDTR und FAP (Abschnitt 6.1) evaluiert.

Für die Erstellung von neuen Funktionsumfängen (Anwendungsfall 4.3) wird in der manuellen Vorgehensweise das Funktionen-Modell angelegt, indem die nötigen Merkmale erstellt werden und die Modellartefakte der Konzeptionellen Funktionsarchitektur entsprechend zugeordnet werden. In der automatisierten Vorgehensweise werden diese Schritte durch Metriken durchgeführt. Für beide Vorgehensweisen ist der größte Teil des Aufwands das Treffen der Entscheidung, zu welcher Funktion ein Modellartefakt gehört. Im Vergleich zu den anderen Anwendungsfällen der E/E-System-Evolution ist die Reduzierung der Modellierungsdauer für die Erstellung mit einem Faktor von 2 deshalb am geringsten. Für die übrigen Anwendungsfälle zeigt sich eine Reduzierung durch die Automatisierung bis Faktor 9 (Anwendungsfall 4.7).

Ferner zeigt sich, dass die manuelle Erstellung und Änderung des Funktionen-Modells fehleranfällig ist. Bei fast allen Anwendungsfällen entstanden Fehler bei der Zuordnung zu den Merkmalen. Diese sind bei der automatisierten Vorgehensweise nicht entstanden.

Variantenkonfiguration im Variantenmanagement Die Automatisierung der Variantenkonfiguration (Anwendungsfall 4.8) wird evaluiert, indem drei E/E-Architekturmodell-Varianten für die E/E-Systeme der Domäne Chassis/Fahrerassistenz (Abschnitt 6.1) erstellt werden. In der manuellen Vorgehensweise wird dafür eine neue **Alternative** angelegt und ausgewählt, welche Funktionen zur E/E-Architekturmodell-Variante gehören. Anschließend wird ein **Exclude-Set** angelegt, zur E/E-Architekturmodell-Variante hinzugefügt

Tabelle 6.10: Modellierungsdauer bei manuellem Vorgehen gegenüber Automatisierung

Anwendungsfall	Datengrundlage	#	Zeit Manuelle Modellierung [s]	Zeit E/E-System Produktlinien Engineering [s]	Faktor
4.1 Herstellen Ebenenrelation	Objektschnittstelle FAP	1 Relation	37	3	=12,3
4.1 Entfernen Ebenenrelation	Objektschnittstelle FAP	1 Relation	2	3	=0,7
4.2 Evolution Ebenenrelation Aufspalten	Objektschnittstelle FAP	2 Relationen	63	12	=5,3
4.2 Evolution Ebenenrelation Fusion	Objektschnittstelle FAP	1 Relation	34	2	=17,0
4.3 Evolution Erstellen	CPA, SDTR, FAP	1129 Mapings, 98 CF	11468 (7 Fehler)	7882	=1,5
4.5 Evolution Verschieben	CPA → ESP	29 Mapings, 5 CF	107	61	=1,8
4.6 Evolution Aufspalten	CPA, SDTR / FAP	1100 Mapings 94 CF	6259 (16 Fehler)	3186	=2,0
4.7 Evolution Fusion	CPA, SDTR+FAP	822 Mapings 64 CF	2306 (6 Fehler)	263	=8,8
4.8 Variantenkonfiguration	MIN-Variante aus allen FW/FAS-E/E-Systemen	6 Systeme (2 variant) bzw. 7 Funktionen, 20 FB	488	80	=6,1
4.8 Variantenkonfiguration	MID-Variante aus allen FW/FAS-E/E-Systemen	10 Systeme (2 variant) bzw. 14 Funktionen, 52 FB	704	124	=5,7
4.8 Variantenkonfiguration	MAX-Variante aus allen FW/FAS-E/E-Systemen	12 Systeme (6 variant) bzw. 16 Funktionen, 65 FB	941	189	=5,0
4.9 Autom. Erstellung	CPA	6 Kern-, 16 Randartefakte	267	59	=4,5
4.9 Autom. Erstellung	SDTR	9 Kern-, 23 Randartefakte	378 (3 Fehler)	60	=6,2
4.9 Autom. Erstellung	FAP	28 Kern-, 67 Randartefakte	767 (11 Fehler)	61	=12,8
4.9 Integration	CPA	6 Kern-, 16 Randartefakte	118	72	=1,6
4.9 Integration	SDTR	9 Kern-, 23 Randartefakte	144	65	=2,2
4.9 Integration	FAP	28 Kern-, 67 Randartefakte	265	114	=2,3

Für die Anwendungsfälle (Abschnitt 4.4.5) wird die Modellierungsdauer im E/E-System Produktlinien Engineering bei manueller Modellierung und bei Automatisierung gegenübergestellt. Es werden die E/E-Systeme aus dem Fallbeispiel der Arbeit (Abschnitt 6.1) verwendet. Für einige Anwendungsfälle wurden mehrere Messungen durchgeführt, die sich in der Anzahl der verwendeten E/E-Systeme bzw. durch den Umfang der E/E-Systeme unterscheiden. Wurden bei manuellem Vorgehen Modellierungsfehler beobachtet, sind diese in der Spalte für die Modellierungsdauer in der manuellen Modellierung dokumentiert. # gibt den Modellierungsumfang an. KB E/E-Komponentenbeitrag; CF CustomerFeature;

und alle unnötigen E/E-Komponentenbeiträge dieser Funktionen, die von der Variantenbildung ausgeschlossen werden sollen, hinzugefügt. Mit der Automatisierung wird eine **Variante** erstellt und die gewünschten E/E-Systeme in der **Vorlagenansicht** ausgewählt. Je nach Umfang der erstellten E/E-Architekturmodell-Variante, ist die Variantenbildung um den Faktor 5 schneller.

Wiederverwendung von E/E-Systemen Die Automatisierung der Wiederverwendung von E/E-Systemen (Anwendungsfall 4.9) wird mit den E/E-Systemen CPA, SDTR und FAP (Abschnitt 6.1) evaluiert. Hierbei zeigt sich, dass die Reduzierung für die Erstellung des E/E-System-Modellcontainers (zwischen Faktor 5 und 13) abhängig von Umfang der E/E-Systeme ist. Je größer das E/E-System ist, desto höher ist der Effizienzgewinn durch die Automatisierung. Gleiches gilt für die automatische Integration des E/E-System-Modellcontainers, die mit der Automatisierung im Vergleich zum manuellen Vorgehen um den Faktor 2 schneller ist.

Bei der Erstellung des E/E-System-Modellcontainers wurden bei manuellem Vorgehen Fehler verzeichnet. Die Fehler betreffen dabei die fälschliche Zuordnungen zu Kern- oder Randartefakten bzw. das Vergessen oder die überschüssige Zuordnung von Modellartefakten. Da die Automatisierung durch das E/E-System Produktlinien Engineering diese Zuordnung automatisiert, traten hier keine Fehler auf.

Der Austausch (Anwendungsfall 4.10) wird nicht betrachtet da, da dieser in beiden Vorgehensweisen analog vorgenommen wird.

6.4 Zusammenfassung und Diskussion

In Kapitel 6 wird das Konzept des E/E-System Produktlinien Engineerings aus Kapitel 4, dessen prototypische Umsetzung in Kapitel 5 beschrieben ist, evaluiert. Neben einem Vergleich mit der heutigen E/E-Architekturmodellierung (Abschnitt 6.2) wird eine Fallstudie durchgeführt (Abschnitt 6.3), deren Grundlage die E/E-Systeme und Szenarien aus der Praxis der E/E-Architekturentwicklung in der Domäne Chassis/Fahrerassistenz bei Mercedes-Benz Cars bilden (Abschnitt 6.1).

Modellierungsziele und Automatisierung

Im Vergleich mit der heutigen E/E-Architekturmodellierung erlaubt das E/E-System Produktlinien Engineering eine Realisierung der Ziele der E/E-Architekturmodellierung (Abschnitt 3.2.1). Gerade im Hinblick auf die steigende Komplexität von E/E-Architekturen (Abschnitt 1.2) ist es notwendig

über eine Vorgehensweise zu verfügen um die E/E-Architektur und deren Weiterentwicklung zu modellieren.

Ferner ermöglicht die neue Vorgehensweise eine Erhöhung der Automatisierungsstufe für die allgemeinen Anforderungen der Arbeit (Abschnitt 4.1). Insbesondere wird die Evolution teilautomatisiert und die Variantenkonfiguration und die Wiederverwendung automatisiert. Dies führt für das E/E-System Produktlinien Engineering zu einer Reduktion des Modellierungsaufwands und der Fehleranfälligkeit im Vergleich mit der heutigen E/E-Architekturmodellierung. Beides ist bedeutend für den Einsatz in der Praxis, da die Modellierungskapazitäten begrenzt sind bzw. einen Kostenfaktor darstellen.

Nachverfolgbarkeit, Modellqualität und Robustheit

Durch die Konzepte des E/E-System Produktlinien Engineering wird sowohl die Nachverfolgbarkeit, die Modellqualität als auch die Robustheit der E/E-Architekturmodellierung gesteigert. Insbesondere profitierte die Nachverfolgbarkeit und die Modellqualität durch das Auflösen der Varianz in den Funktionen infolge der Einführung eines E/E-Systemmodells (Abschnitt 4.4.3). Die Robustheit hingegen verbessert sich vorwiegend durch die Einführung der Konzeptionellen Funktionsarchitektur-Ebene (Abschnitt 4.4.1). Zusammen genommen sind diese Kriterien bedeutsam für den Nutzen, den ein E/E-Architekt aus einem E/E-Architekturmodell ziehen kann, da sie entscheiden, für welche Fragestellungen das E/E-Architekturmodell eingesetzt werden kann und wie gut die Modellierung von sonst manuell verwalteten Aspekten entlastet.

Evolution von E/E-Systemen

Für die Evolution von E/E-Systemen wird ein neues Vorgehen durch eine Fallstudie evaluiert, das bei der Beherrschung der Komplexität einer hohen Anzahl von Funktionen unterstützt und damit eine Basis für Design-Entscheidungen bildet. Allerdings erfordert dieses Vorgehen die Modellierung von Mehrinhalten. Dies führt zu einer geringen Verringerung der Modellierungseffizienz für 3 von 5 Anwendungsfällen. Alle übrigen Anwendungsfälle verzeichnen eine Effizienzsteigerung um den Faktor 4 bis 5. Insbesondere hängt der Modellierungsaufwand für die Ersterstellung der zusätzlichen Inhalte von dem gewählten Detaillierungsgrad und damit von der Anzahl zu erstellender Modellartefakte ab. Für einen geringeren Detaillierungsgrad könnte somit die Modellierungseffizienz steigen. Insgesamt überwiegt klar der Vorteil der Bewertbarkeit einer Evolution von E/E-Systemen gegenüber der geringfügig verringerten Modellierungseffizienz.

Variantenkonfiguration im Variantenmanagement

Die Konfiguration von E/E-Architekturmodell-Varianten wurde in einer Fallstudie evaluiert. Die Ergebnisse für die Konfiguration einer E/E-Architekturmodell-Variante belegen eine signifikante Steigerung der Effizienz für den Modellierer um den Faktor 10 als auch für den E/E-Architekten um den Faktor 5. Allerdings führt die Modellierung des E/E-Systemmodells zu einem um den Faktor 9 höheren Initialaufwand. Dieser lässt sich allerdings durchaus motivieren, da die Ersparnis bei der Konfiguration der E/E-Architekturmodell-Varianten deutlich überwiegt. Schon ab 4 Variantenkonfigurationen lohnt sich die neue Vorgehensweise insgesamt. In der Praxis wird bei der E/E-Plattform-Entwicklung etwa alle 3 Monate ein neuer Modellstand erzeugt, der mehr als 10 E/E-Architekturmodell-Varianten (Ausstattungsvarianten) enthält. Wie viele E/E-Architekturmodell-Varianten in jedem Modellstand neu angelegt oder angepasst werden ist unterschiedlich.

Insgesamt wird eine einfachere, mächtige und systematische Vorgehensweise aufgezeigt, die den E/E-Architekten von Themen entlastet, die er bisher manuell verwalten musste. Zudem wird die Effizienz der E/E-Architekturmodellierung erhöht, die einen Kostenfaktor darstellt.

Wiederverwendung von E/E-Systemen

In der Fallstudie wird die heutige Vorgehensweise mit den neuen Konzepten zur Wiederverwendung verglichen und evaluiert. Vorteil der heutigen Vorgehensweise ist die verringerte Größe des E/E-Architekturmodells da die Modellierung in separaten E/E-Architekturmodellen vorgenommen wird. Demgegenüber erhöht sich die Modellierungseffizienz um den Faktor 5 für die erstmalige Wiederverwendung eines E/E-Systems mit dem neuen E/E-System Produktlinien Engineering. Für den Austausch nach einer Weiterentwicklung des E/E-Systems erhöht sich die Modellierungseffizienz um den Faktor 12.

Ursache für die starke Steigerung der Modellierungseffizienz für die erstmalige Wiederverwendung sind auch Fehler, die beim Export bzw. Import der Modellartefakte in der heutigen Vorgehensweise auftreten. Diese müssen vom Modellierer mühsam analysiert und behoben werden. Es ist hervorzuheben, dass der Modellierer hierbei Techniken anwendete, die ausschließlich den Experten in der Entwicklung des Modellierungswerkzeug-Herstellers zur Verfügung stehen. Das heutige Vorgehen an sich ist deshalb durchaus als herausfordernd zu bewerten. Dennoch ist es nicht ausgeschlossen, dass der Modellierer bei einer weiteren Anwendung effizienter vorgeht. Unter der Annahme, dass gar keine Fehler auftraten, reduziert sich die Steigerung der Modellierungseffizienz für die Wiederverwendung eines E/E-Systems nicht weiter als auf den Faktor 2. Die Modellierungseffizienz für Austausch eines E/E-Systems ist hingegen nicht betroffen.

Trotz der erhöhten Modellgröße überwiegen somit insgesamt die Vorteile des E/E-System Produktlinien Engineerings.

Nutzen des E/E-System Produktlinien Engineerings in der Praxis

Der Nutzen, den das E/E-System Produktlinien Engineering einträgt, wurde mit der Durchführung einer Fallstudie evaluiert. Die Fallstudie weist neben der generellen Machbarkeit der Konzepte und Anwendungsfälle auch die Auswirkung auf die Modellierungseffizienz nach, die für fast alle Anwendungsfälle signifikant steigt. Weitere Verbesserungen in der Nachverfolgbarkeit, Robustheit, Modellqualität und Durchgängigkeit in der E/E-Architekturmodellierung wurden qualitativ evaluiert.

Durch die Konzepte dieser Arbeit werden wichtige Fragen in der Praxis der E/E-Architekturmodellierung (Abschnitt 3.2.1) adressiert und eine Lösung erarbeitet. Neben den entwickelten Anforderungen und Konzepten sind praktische Modellierungswerkzeug-Erweiterungen entstanden. Die Konzepte werden entweder bereits in der produktiven E/E-Architecturentwicklung eingesetzt oder lassen sich für die zukünftigen E/E-Architekturprojekte einführen. Eine Einführung kann im Allgemeinen nur schrittweise erfolgen. Ursache dafür sind die Komplexität der Organisation und starke Abhängigkeiten zwischen Projektmeilensteinen verschiedenster E/E-Architektur-Projekte mit wenigen Zeitfenstern für die Einführung neuer Methoden. Erleichtert wird eine Einführung durch minimalinvasive Konzepte mit keinen oder minimalen Einschränkungen für bestehende Anwendungsfälle. Zudem sollten die Modellierungswerkzeug-Erweiterungen migrationsarm sein und somit bei Aktualisierungen des Modellierungswerkzeugs bzw. des Meta-Modells nur möglichst wenige und leicht umsetzbare Anpassungen erfordern.

Konkret sind bisher Konzepte in die E/E-Architekturmodellierung des E/E-Plattform-Entwicklungsprojekts STAR3 eingeführt. Es umfasst die nächste Generation der Mercedes-Benz S-Klasse und nachfolgender Kraftfahrzeuge bei Mercedes-Benz Cars. Insbesondere ist das E/E-Systemmodell einschließlich der vorgestellten Variantenkonfiguration für die gesamte E/E-Architektur-Domäne Chassis/Fahrerassistenz im Einsatz. Ferner ist exemplarisch die umfassendste Innovation dieser E/E-Architektur-Domäne mit Hilfe der Konzeptionellen Funktionsarchitektur-Ebene beschrieben und bewertet worden. Somit sind alle Voraussetzungen gegeben um mit dem Produktlinien Engineering und dem Feature Oriented Architecture Refactorings anzuknüpfen. Um den Migrationsaufwand so gering wie möglich zu halten, wurden die Modellierungswerkzeug-Erweiterungen der Konzepte in Form von Metriken umgesetzt.

Die Modellierung für die Konzeptionelle Funktionsarchitektur-Ebene erfordert zuvor ein Erarbeiten der Inhalte durch den E/E-Architekten der E/E-Architektur-Domäne im Dialog mit den entsprechenden Funktions- und E/E-

Systemverantwortlichen. Dies ist aufwändig und sollte deshalb erst für eine Auswahl mit wichtigen Weiterentwicklungen eines E/E-Moduls, die einen hohen Abstimmungsaufwand und starke Auswirkungen haben, durchgeführt werden. Anschließend können sukzessiv weitere Schwerpunkte erschlossen werden und schließlich eine Gesamtübersicht erzeugt werden.

Kapitel 7

Zusammenfassung und Ausblick

Der E/E-Architekturentwurf ist stärker denn je mit steigender Komplexität und Variabilität konfrontiert. Zwar wird der Entwurf modellbasiert durchgeführt, aber es gibt keine Vorgehensweise zur gesamtheitlichen Betrachtung von E/E Systemen, weshalb Variabilität und Komplexität nicht mehr ausreichend beherrscht werden können. E/E-Module als Hardware der E/E-Systeme werden durch Modulstrategien in der mechanischen Fahrzeugentwicklung als Teil eines vereinfachten, schnelleren Produktionsprozess definiert, jedoch das Zusammenspiel mit Funktionen der E/E-Systemen wird methodisch nicht durch die Modulstrategien abgedeckt. Eine Absicherung dieses Zusammenspiels ist für die E/E-Integration ist jedoch ausschlaggebend, da Software die Hauptquelle für Innovationen [112] ist und sich mechanische Innovationen mehr und mehr zu Software-Innovationen verlagern [135].

Ziel dieser Arbeit ist somit die Entwicklung einer Vorgehensweise zur Bewertung und Umsetzung der Evolution (Weiterentwicklung) von E/E-Systemen (Zielsetzung 1.1), die mit der heutigen E/E-Architekturmodellierung nicht bewertbar ist. Zudem soll mit dieser Arbeit die heutige E/E-Architekturmodellierung um E/E-Systeme (Zielsetzung 1.2), die vollständig alle Aspekte einer Kunden- oder technischen Funktionalität umfassen, erweitert werden. Mit diesen Zielen setzt diese Arbeit das E/E-System Produktlinien Engineering der steigenden Komplexität und Variabilität als Lösungskonzept für eine zukünftige E/E-Architekturmodellierung entgegen.

In Anbetracht der diversen Anforderungen an das E/E-System Produktlinien Engineering (Abschnitte 4.1, 4.2) wurden Designentscheidungen für das Lösungskonzept unter der Prämisse einer Umsetzbarkeit im industriellen Kontext getroffen (Abschnitt 4.3). So umfasst das E/E-System Produktlinien Engineering eine neue E/E-Architekturebene, die Konzeptionelle Funktionsarchitektur-Ebene (Abschnitt 4.4.1), zur Bewertung der Evolution und das Feature Oriented Architecture Refactoring (Abschnitt 4.4.2), eine Adaption des Feature Oriented Refactoring aus der Software-Entwicklung

(Designentscheidung 4.3), für die Umsetzung der Evolution. In dieser Kombination ermöglicht das E/E-System Produktlinien Engineering die Evolution von E/E-Systemen. Zudem wurde das E/E-Systemmodell entwickelt (Abschnitt 4.4.3), das erlaubt, die E/E-Systeme zu modellieren und für die Variantenkonfiguration zu nutzen. Weiterhin wurde ein um E/E-Systeme erweiterter Produktlinienansatz entworfen (Abschnitt 4.4.4) um die Wiederverwendung von kompletten E/E-Systemen einschließlich ihrer E/E-Module und Funktionen zu realisieren. Für die konkreten Problemstellungen eines Anwenders bei der E/E-Architekturmodellierung wurden Anwendungsfälle entworfen (Abschnitt 4.4.5). So können mit dem E/E-System Produktlinien Engineering

- für die Evolution von E/E-Systemen die Verbindung zwischen den E/E-Architekturebenen hergestellt, entfernt und weiterentwickelt werden (Anwendungsfälle 4.1, 4.2) und Funktionsumfänge für E/E-Module erstellt (Anwendungsfall 4.3), entfernt (Anwendungsfall 4.4), verschoben (Anwendungsfall 4.5), aufgespalten (Anwendungsfall 4.6) und fusioniert (Anwendungsfall 4.7) werden,
- die Variantenkonfiguration des E/E-Architekturmodells mit E/E-Systemmodellen durchgeführt (Anwendungsfall 4.8) und
- für die Wiederverwendung von E/E-Systemen die E/E-Systemmodelle in weiteren E/E-Architekturmodellen wiederverwendet (Anwendungsfall 4.9) und die Wiederverwendung nach einer Änderung ausgetauscht (Anwendungsfall 4.10) werden.

Zur Unterstützung des Anwenders mit einer Automatisierung bei diesen Anwendungsfällen wurde das Werkzeug für die E/E-Architekturmodellierung PREEvision mit Metriken erweitert (Kapitel 5). Die Erkenntnisse aus Fragestellungen und Lösungsansätzen dieser im industriellen Kontext entstandenen Arbeit wurden regelmäßig als Input für die Fehlerbehebung (vgl. Anmerkungen zu `ReuseUnit` in Abschnitt 3.4.3) und die Weiterentwicklung von PREEvision gegeben. Den Einstieg in die Metriken und Modellartefakte des E/E-System Produktlinien Engineering finden die Anwender über ein dafür geschaffenes Framework.

Mit dem E/E-System Produktlinien Engineering werden erstmals die Ziele für die E/E-Architekturmodellierung nicht nur für die Hardware- bzw. E/E-Module erreicht sondern eine vollständige Betrachtung eines E/E-Systems einschließlich der funktionalen Anteile ermöglicht (Abschnitt 6.2). Zudem wird im Vergleich zur heutigen Modellierung die Nachverfolgbarkeit, Robustheit und die Qualität der Modellierung gesteigert und der Modellierungsaufwand und die Fehleranfälligkeit reduziert. In einer Fallstudie wurde ein Modellierungsexperte, ein Field-Application-Engineer des Modellierungswerkzeugherstellers, mit den entsprechenden Szenarien in der heutigen E/E-

Architekturmodellierung und im E/E-System Produktlinien Engineering konfrontiert (Abschnitt 6.3). Die Ergebnisse weisen die generelle Machbarkeit und die Steigerung der Modellierungseffizienz für fast alle Anwendungsfälle nach. Konkret wurde

- für die Evolution von E/E-Systemen eine leichte Verschlechterung der Modellierungseffizienz für 3 von 5 Anwendungsfällen und eine Steigerung um Faktor 4 bis 5 für die übrigen Anwendungsfälle,
- für die Variantenkonfiguration mit E/E-Systemen eine signifikante Steigerung der Effizienz für den Modellierer um den Faktor 10 und für den E/E-Architekten um den Faktor 5 bei einem um Faktor 9 höheren Initialaufwand und
- für die Wiederverwendung von E/E-Systemen eine Erhöhung der Modellierungseffizienz um den Faktor 2 bis 5 und eine Erhöhung um den Faktor 12 für den Austausch eines wiederverwendeten E/E-Systems

verzeichnet. Für die Evolution von E/E-Systemen ist die Modellierung von Mehrinhalten erforderlich, weshalb der Detaillierungsgrad direkte Auswirkungen auf die Modellierungseffizienz hat. Hier liegt allerdings klar der Vorteil der Bewertbarkeit einer Evolution gegenüber einer geringfügig verringerten Modellierungseffizienz. Bei der Variantenkonfiguration überwiegt die Ersparnis durch die gesteigerte Modellierungseffizienz deutlich gegenüber einem erhöhten Initialaufwand. Bei der Wiederverwendung von E/E-Systemen überwiegen die Vorteile durch die signifikante Steigerung der Modellierungseffizienz gegenüber einer erhöhten Modellgröße. Insgesamt konnten mit dem E/E-System Produktlinien Engineering somit die Ziele, die für diese Arbeit formuliert wurden (Zielsetzung 1.1 und Zielsetzung 1.2), erreicht werden.

Mit dieser Arbeit werden wichtige Fragestellungen aus der Praxis der E/E-Architekturmodellierung adressiert und ein Lösungskonzept angeboten. Einige Konzepte des E/E-System Produktlinien Engineering werden bereits in der produktiven E/E-Architecturentwicklung im E/E-Plattform-Entwicklungsprojekt STAR3, das die nächste Generation der Mercedes-Benz S-Klasse und nachfolgender Kraftfahrzeuge bei Mercedes-Benz Cars umfasst, eingesetzt. Hierbei ist das E/E-Systemmodell einschließlich der vorgestellten Variantenkonfiguration für alle E/E-Systeme der E/E-Architektur-Domäne Chassis/Fahrerassistenz (Abschnitt 6.1) im Einsatz. Zudem ist die größte Innovation in diesem Bereich mit der Konzeptionellen Funktionsarchitektur-Ebene beschrieben und bewertet worden. Als nächster Schritt bietet sich die Anknüpfung mit dem Feature Oriented Architecture Refactoring an und eine Erschließung weiterer Schwerpunkte mit großer Auswirkung auf die E/E-Architektur.

Für weitere Arbeiten im Bereich der E/E-Architekturmodellierung werden folgende Schwerpunkte in Anknüpfung an das E/E-System Produktlinien Engineering erkannt und für eine weitere Betrachtung unter Einbeziehung theoretischer, wissenschaftlicher Betrachtungswinkel vorgeschlagen:

Auswertung von E/E-Architekturmodellen Die heutige E/E-Architekturmodellierung zeichnet sich durch eine hohe Komplexität mit vielfältigen intransparenten Abhängigkeiten aus. Während heute die Hardware-Aspekte sehr leicht aus der E/E-Architekturmodellierung analysierbar sind, ist aufgrund der hohen Anzahl an Modellartefakten und der Komplexität der Modellierung die Auswertung der E/E-Architekturmodellierung von systemischen Fragestellungen nur mit hohem Aufwand möglich. Eine automatisierte Auswertung des E/E-Architekturmodells zur Analyse der modellierten Inhalte hätte das Potential, die Transparenz zu erhöhen, die Überprüfbarkeit des E/E-Architekturmodells zu verbessern und den Aufwand zu reduzieren, der momentan notwendig ist um gezielte Einblicke über komplexe Zusammenhänge zu erstellen. Hierbei interessieren insbesondere Auswertungen zu den Kommunikationspartnern eines E/E-Systems, Vergleichen von E/E-System-Schnittstellen, Vergleichen von Funktionsbeitrag-Schnittstellen und Komplexität eines E/E-Systems. Letzteres kann als weitere konkrete Größe verwendet werden um die Alternativen für E/E-Systemschnitte zu vergleichen: Wird die E/E-Architektur weiterentwickelt, so betrifft dies auch meist die Funktionsarchitektur. Eine Änderung in der Funktionsarchitektur resultiert ggf. in einer veränderten Verteilung der Funktionen. Dies wiederum kann eine Auswirkung auf den Schnitt der E/E-Systeme und der E/E-Systemgrenzen haben. Gleichzeitig entstehen während des E/E-Architekturdesigns oft verschiedene Alternativen für den E/E-Systemschnitt und die E/E-Systemgrenzen. Für die Bewertung der Alternativen wird in der frühen Phase derzeit modellbasiert hauptsächlich die Buslast herangezogen (Abschnitt 3.2.3). Der Komplexitätswert hingegen ermöglicht eine qualitative Aussage zur Anzahl an zum E/E-System gehöriger Hardware, Funktionen und Schnittstellen. Die Arbeiten hierzu sind bereits fortgeschritten und basieren auf einer Adaption der Übertragung der Halstead-Metrik für die modellbasierte Entwicklung mit Simulink- und TargetLink-Modellen [148]. Allerdings wurden diese Anteile aufgrund des Umfangs nicht mit in diese Ausarbeitung einbezogen.

Erweiterung der E/E-Systeme auf die TOP-Ebene E/E-Komponenten haben Anforderungen an die Verbauorte im Fahrzeug. Insbesondere bei Sensoren und Aktoren führt dies dazu, dass die Einbauorte über viele Baureihen und Derivate hinweg identisch sind. Für alle anderen E/E-Komponenten sind die Einbauorte einer Baureihe über die Derivate

hinweg ähnlich. Somit könnte eine Erweiterung des E/E-System Produktlinien Engineerings um die Einbauorte für die E/E-Komponenten den Modellierungsaufwand reduzieren. Hierfür müssten die Konzepte um die Topologieebene der E/E-Architekturmodellierung, in der die geometrischen Eigenschaften der Bauräume und Leitungsführungen modelliert werden, erweitert werden.

Modellbasierter Spezifikationsprozess Die heutige Entwicklung von E/E-Systemen ist weitestgehend ein dokumentbasierter Entwicklungsprozess, in dem System- und Komponentenlastenhefte für den Zulieferer spezifiziert werden. Die Spezifikation erfolgt in tabellenbasierten Anforderungswerkzeugen und eine Übernahme bzw. ein Austausch der Modelle des E/E-Systems mit der E/E-Architekturmodellierung ist heute nicht vorgesehen. Ein Austausch ist einer Übernahme vorzuziehen, da es nach technischen Verhandlungen mit Lieferanten zu Änderungen kommen kann. Zukünftig sollten geeignete Schnittstellen für eine Übernahme bzw. einen Austausch geschaffen werden oder das Werkzeug für die E/E-Architekturmodellierung um die Funktionen für die Spezifikation und Verknüpfung mit den E/E-System-Modellen erweitert werden. Insbesondere beim letzten Punkt sieht der Werkzeughersteller Potentiale [133] aber in der Praxis ist zu erarbeiten

- wie eine Harmonisierung der heterogenen Werkzeuglandschaft der verschiedenen Fachbereiche erfolgen kann,
- wie die Performance bei hohen Nutzerzahlen sichergestellt werden kann,
- wie mit den großen Datenmengen umgegangen werden kann und
- wie ein rechtlich belastbarer Austausch mit dem Zulieferer erfolgen kann.

Demandprozess In großen Organisationen stammen die Anforderungen, die an die E/E-Architektur gestellt werden aus den unterschiedlichsten Quellen. Es gibt u. a.

- Ideen- und Innovationsprozesse,
- Modulstrategien,
- marktspezifische Anforderungen,
- Zertifizierungsanforderungen,
- Anforderungen von Organisationseinheiten, die nur Teile der E/E-Architektur oder einzelne E/E-Komponenten nutzen,
- Anforderungen durch die Fachbereiche, die den Fokus auf viele ähnliche Funktionen haben und

- Anforderungen durch Baureihen, die ein Fahrzeug für einen Kunden mit einem gewissen Profil entwickeln.

Hieraus können neue Funktionen oder Änderungen an bestehenden Funktionen hervorgehen. Bei den Anforderungen kann es auch zu Dopplungen oder zu Unterschieden in lediglich Nuancen kommen. Insgesamt ist es schwierig

- diese Masse an Anforderungen zu konsolidieren,
- Treiber zu bestimmen, die sich dem Thema annehmen und über viele Organisationseinheiten hinweg die technische Umsetzbarkeit abklären und
- das Thema zu entscheiden, nachdem die Auswirkungen bewertet wurden.

Bei der steigenden Komplexität und Varianz bei gleichzeitig kürzeren Entwicklungszyklen ist ein schlanker Demand-Prozess für die Aufnahme, Konsultierung, Prüfung und Entscheidung für neue Anforderungen notwendig. Potentiale für solch einen Demandprozess sind eine gesteigerte Geschwindigkeit mit der Funktionen in der E/E-Architektur abgesichert und verortet werden können, eine gesteigerte Transparenz in der Entwicklung und eine Reduktion der Blindleistung. Als Aufsatzpunkt für eine derart neue Vorgehensweise ist eine Beschränkung auf kundenrelevante Funktionen, mit denen der Benutzer in der Regel interagiert. In Kopplung mit Organisationseinheiten, die Schalter, Lenkräder, Displays, Infotainment und User-Interface verantworten, kann so für diese große Gruppe an Funktionen sichergestellt werden, dass Funktionen und Änderungen schnell sinnvoll entschieden und entwickelt werden.

Anhang A

Modularisierung

A.1 Effekte der Modularisierung

Einer durchgeführten Modularisierung werden mehrere Effekte zugesprochen, die sowohl Vor- als auch Nachteile hervorrufen [52]:

Entkopplung Durch die Modularisierung wird eine Entkopplung erreicht, die aufgrund der wenigen Beziehungen zwischen den Modulen entsteht. Einerseits ist die Konstruktion und Spezifikation der Schnittstellen zwischen den Modulen sehr aufwändig. Dadurch wird aber andererseits eine Parallelisierung in der Entwicklung ermöglicht. Zudem wird die Anzahl der in der Montage zu handhabenden Schnittstellen reduziert [134].

Wiederverwendung Ferner wird eine Wiederverwendung der Module ermöglicht. Dies reduziert den Entwicklungsaufwand und senkt durch Skalen- und Lernkurveneffekte die Kosten und die Fehlerrate. Dem gegenüber stehen allerdings geringere Möglichkeiten um verschiedene Produkte, in denen eine Wiederverwendung eines Moduls stattfindet, voneinander zu differenzieren. Es ist möglich die Module marken-, baureihen- und derivateübergreifend wiederzuverwenden [72, S. 130].

Austauschbarkeit Ein entscheidender Effekt, der die Erzeugung von Varianz erlaubt, ist die Austauschbarkeit. Module können ausgetauscht werden, was einen einfachen Weg darstellt um das Produkt zu verändern. Auch die Reparatur wird dadurch vereinfacht, dass nur das fehlerhafte Modul ausgetauscht werden muss. Jedoch ist man in der Reparatur gegebenenfalls nur auf den Austausch des gesamten Moduls beschränkt obwohl der Austausch einer einzelnen Komponente des Moduls genügen würde.

Erweiterbarkeit Weiterhin wird durch eine Modularisierung eine Erweiterbarkeit des Produkts erlaubt. Durch das Hinzufügen neuer Module lässt

sich die Gesamtfunktion des Produktes erweitern wobei das Herstellen einer Produktintegrität herausfordernd ist. Unter Produktintegrität wird das optimale Zusammenspiel der einzelnen Module miteinander verstanden [52]. Außerdem ist Modularität eine Möglichkeit um die Änderungen und Anpassungen im Produktionsprozess und am Produktdesign, die für bestimmte Märkte durchgeführt werden müssen, vorwegzunehmen [134].

Standardisierbarkeit Ferner erlaubt die Modularisierung eine Standardisierung der Module. Durch die Vereinheitlichung der Schnittstellen lassen sich auf dem Markt existierende Lösungen verwenden. Aufgrund konkurrierende Anbieter können so günstigere Preise sowie eine höhere Verfügbarkeit erzielt werden. Ein Nachteil bei der Verwendung von im Markt verfügbaren Komponenten ist die geringere Originalität des Produkts.

Kontrollierbarkeit Die Modularisierung hat außerdem einen Einfluss auf die Kontrollierbarkeit der Module. Funktionstests werden vereinfacht und defekte Module können leichter identifiziert werden. Führt man diese Prüfung vor dem Einbau durch, hat das Produkt weniger Fehler. Jedoch ist anzumerken, dass man aus der Funktionstüchtigkeit der einzelnen Module noch nicht auf die Funktionstüchtigkeit der Gesamtfunktion des Produkts schließen kann. Einen weiteren Vorteil liegt in der Reduzierung der Unsicherheit im Produktionsprozess und im Produktdesign durch eine Modularisierung [134].

Kombinierbarkeit Ein weiterer Effekt einer Modularisierung ist die Kombinierbarkeit. Module können nach dem Baukastenprinzip miteinander kombiniert werden. Neben der Austauschbarkeit lässt sich auch dadurch eine Varianz für das Produkt erzeugen. Jedoch ist die Erstellung eines Baukastens mit kombinierbaren Modulen aufwändig.

A.2 Klassen von Produktarchitekturen

Ein weiterer und wichtiger Punkt, der im Zuge von Modularisierung betrachtet werden muss und erstmals von Ulrich [160] thematisiert wurde, betrachtet den Zusammenhang zwischen Funktion und Komponente. Dieser Zusammenhang wird über die Produktarchitektur hergestellt, die von Ulrich [160] wie folgt definiert wird:

Die Produktarchitektur ist das Schema, mit dem

- 1) funktionale Elemente ausgelegt,
- 2) funktionale Elemente zu physischen Komponenten durch ein Mapping zugeordnet und

3) Schnittstellen zwischen den physischen Komponenten spezifiziert

werden. Ferner differenziert Ulrich [160] zwischen zwei Klassen einer Produktarchitektur. Genauer, in eine modulare Produktarchitektur, die ein eins-zu-eins-Mapping zwischen funktionalen Elementen und physischen Komponenten aufweist, und in eine integrale Produktarchitektur. Letztere weist gerade kein eins-zu-eins Mapping auf.

Ein weiteres Kennzeichen einer modularen Produktarchitektur sind laut Ulrich [160] entkoppelte Schnittstellen zwischen den physischen Komponenten. Analog dazu ist eine integrale Produktarchitektur durch gekoppelte Schnittstellen zwischen den physischen Komponenten gekennzeichnet. Unter gekoppelten Schnittstellen versteht Ulrich [160] hierbei Schnittstellen, die bei Änderung einer Komponente eine Änderung in einer weiteren, über diese Schnittstelle verbundene, Komponente hervorrufen.

Diese Thematik wird von Göpfert und Steinbrecher [52] über die Unabhängigkeit beschrieben, die im Gegensatz zu Ulrich [160] auch die funktionalen Elemente betrachtet. Die Unabhängigkeit wird deshalb in zwei Dimensionen aufgeteilt [52]:

- 1) Funktionale Unabhängigkeit ist vorhanden, wenn eine physische Komponente eine Funktion unabhängig von einer anderen Komponente erfüllt.
- 2) Physische Unabhängigkeit ist vorhanden, wenn eine physische Komponente von anderen durch eine geeignete Gestaltung der Schnittstelle physisch trennbar (und somit im Sinne von Ulrich [160] entkoppelt) ist.

Dies erlaubt Göpfert und Steinbrecher [52] eine weitergehende Differenzierung in insgesamt vier Klassen (Abbildung A.1). Es wird somit neben der modularen und der integralen Produktarchitektur noch in die funktionalmodulare Produktarchitektur und die physischmodulare Produktarchitektur unterschieden. Insgesamt ergibt sich somit, dass je höher der Grad an Unabhängigkeit in beiden Dimensionen ist, desto größer ist auch das Maß an Modularität.

Zu beachten ist jedoch, dass nicht die Maximierung der Modularität das Ziel einer Modularisierung ist, sondern die Optimierung für die unterschiedlichen Zielsetzungen (Abschnitt 2.3.3) [116, S. 685]. Eine Maximierung würde eine unnötige Erhöhung der Anzahl von Schnittstellen verursachen. Auch Hüttenrauch und Baum [72] schlagen vor, ein Modul bei der Integration mit anderen Modulen entweder in einer bestimmten Lage oder in einer bestimmten Funktion einzusetzen [72, S. 130].

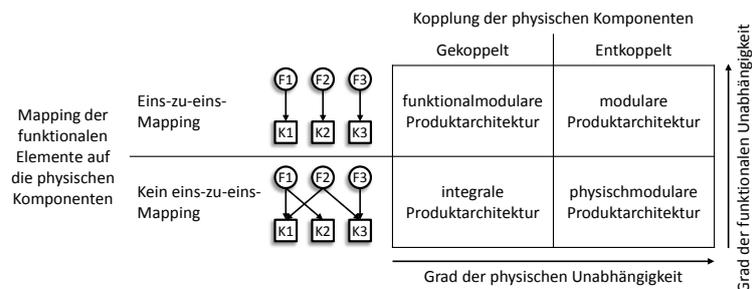


Abbildung A.1: Klassen von Produktarchitekturen. Göpfert und Steinbrecher [52] erweitern die von Ulrich [160] getroffene Klassifizierung von Produktarchitekturen. Abgeleitet von [52, 160]. □ physische Komponente, ○ funktionales Element.

Literaturverzeichnis

- [1] N. Adler, P. Graf und K. D. Müller-Glaser. *Model-Based Consistency Checks of Electric and Electronic Architectures against Requirements*, Seiten 262–275. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-29645-1.
- [2] N. Adler, M. Hillenbrand, K. D. Müller-Glaser, E. Metzker und C. Reichmann. Graphically notated fault modeling and safety analysis in the context of electric and electronic architecture development and functional safety. In *23rd IEEE International Symposium on Rapid System Prototyping (RSP)*, Seiten 36–42. IEEE, 2012.
- [3] A. Ahmad, P. Jamshidi und C. Pahl. Classification and comparison of architecture evolution reuse knowledge—a systematic review. *Journal of Software: Evolution and Process*, 26(7):654–691, 2014.
- [4] D. Ahrens, A. Frey, A. Pfeiffer und T. Bertram. Designing reusable and scalable software architectures for automotive embedded systems in driver assistance. Technical report, SAE Technical Paper, 2010.
- [5] S. Apel und D. Batory. When to use features and aspects?: a case study. In *Proceedings of the 5th international conference on Generative programming and component engineering*, Seiten 59–68. ACM, 2006.
- [6] AUTOSAR Consortium. Technical Overview (Release 3.2), 2011.
- [7] J. Axelsson. Cost models for electronic architecture trade studies. In *Proceedings Sixth IEEE International Conference on Engineering of Complex Computer Systems - ICECCS 2000*, Seiten 229–239, 2000.
- [8] J. Axelsson. Evolutionary architecting of embedded automotive product lines: An industrial case study. In *Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) & European Conference on Software Architecture (ECSA)*, Seiten 101–110. IEEE, 2009.
- [9] J. Axelsson. A transformation-based model of evolutionary architecting for embedded system product lines. In *Third International Workshop on*

- Model Based Architecting and Construction of Embedded Systems*. Oslo, 2010.
- [10] J. Axelsson. On how to deal with uncertainty when architecting embedded software and systems. In *European Conference on Software Architecture*, Seiten 199–202. Springer, 2011.
- [11] J. Axelsson. Improving the evolutionary architecting process for embedded system product lines. In *Systems Conference (SysCon), 2011 IEEE International*, Seiten 334–341. IEEE, 2011.
- [12] J. Axelsson, J. Fröberg, H. Hansson, C. Norström, K. Sandström und B. Villing. Correlating bussines needs and network architectures in automotive applications—a comparative case study. In *Proc of FET*, volume 3, Seiten 219–228, 2003.
- [13] D. Batory. A tutorial on feature oriented programming and product-lines. In *Proceedings of the 25th International Conference on Software Engineering*, Seiten 753–754. IEEE Computer Society, 2003.
- [14] D. Batory, J. N. Sarvela und A. Rauschmayer. Scaling step-wise refinement. In *Proceedings of the 25th International Conference on Software Engineering, ICSE '03*, Seiten 187–197, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1877-X.
- [15] R. Belschner, J. Berwanger, C. Bracklo, C. Ebner, B. Hedenetz, W. Kuffner, P. Lohrmann, J. Minuth, M. Peller, A. Schedl und V. Seefried. Anforderungen an ein zukünftiges Bussystem für fehlertolerante Anwendungen aus Sicht Kfz-Hersteller. *VDI-Berichte*, Seiten 23–41, 2000.
- [16] R. Belschner, J. Freess und M. Mroßko. Gesamtheitlicher Entwicklungsansatz für Entwurf, Dokumentation und Bewertung von E/E-Architekturen. *12. Internationaler Kongress Elektronik im Kraftfahrzeug, Baden Baden, VDI Berichte Nr.1907*, Seite 511, 2005.
- [17] G. Böckle, F. J. van der Linden und K. Pohl. *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer Science & Business Media, 2005. ISBN 978-3-540-28901-2.
- [18] J. Bortolazzi. Systems Engineering for Automotive Electronics. Vorlesung, Karlsruher Institut für Technologie (KIT) - Institut für Technik der Informationsverarbeitung (ITIV), 2010.
- [19] J. Bradbury, J. Cordy, J. Dingel und M. Wermelinger. A classification of formal specifications for dynamic software architectures. In *International Workshop on Self-Managed Systems*, volume 7, 2004.

- [20] H. P. Breivold, I. Crnkovic und M. Larsson. A systematic review of software architecture evolution research. *Information and Software Technology*, 54(1):16–40, 2012.
- [21] A.-P. Bröhl und W. Dröschel. *Das V-Modell: Der Standard für die Softwareentwicklung mit Praxisleitfaden*. Oldenbourg-Verlag, München, 1993. ISBN 978-3-4862-2207-4.
- [22] M. Broy und B. Rumpe. Modulare hierarchische Modellierung als Grundlage der Software- und Systementwicklung. *Informatik-Spektrum*, 30(1): 3–18, 2007.
- [23] M. Broy, I. H. Kruger, A. Pretschner und C. Salzmann. Engineering automotive software. *Proceedings of the IEEE*, 95(2):356–373, 2007.
- [24] M. Broy, G. Reichart und L. Rothhardt. Architekturen softwarebasierter Funktionen im Fahrzeug: von den Anforderungen zur Umsetzung. *Informatik-Spektrum*, 34(1):42–59, 2011.
- [25] A. Burkert. Gefährliches Datenleck. *ATZ-Automobiltechnische Zeitschrift*, 114(4):306–311, 2012.
- [26] M. Calder, M. Kolberg, E. H. Magill und S. Reiff-Marganiec. Feature interaction: a critical review and considered forecast. *Computer Networks*, 41(1):115–141, 2003.
- [27] B. Chown und M. Lange. Modernizing system development: Requirements-based, model-driven design, implementation and test. In *Embedded Real Time Software and Systems (ERTS) Congress 2012*, 2012.
- [28] M. Conrad, I. Fey, M. Grochtmann und T. Klein. Modellbasierte Entwicklung eingebetteter Fahrzeugsoftware bei DaimlerChrysler. *Informatik-Forschung und Entwicklung*, 20(1-2):3–10, 2005.
- [29] M. Cooperation. MOST Specification Rev. 3.0 Errata 2. Standard, www.mostcooperation.com, 2010.
- [30] R. C. De Boer, P. Lago, A. Telea und H. Van Vliet. Ontology-driven visualization of architectural design decisions. In *Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECISA 2009. Joint Working IEEE/IFIP Conference on*, Seiten 51–60. IEEE, 2009.
- [31] J. Delange, C. Honvault und J. Windsor. Model-based engineering approach for system architecture exploration. In *Proceedings of the Embedded Real Time Software and Systems Conference (ERTS), Toulouse, France*, 2012.

- [32] DIN 19 233. Automat, Automatisierung; Begriffe, 1972.
- [33] Y. Duhr und T. Ringler. Product line development of models with MATLAB/Simulink. In *27th Annual European Simulation and Modelling Conference - ESM'2013*, 2013.
- [34] C. Dziobek, J. Loew, W. Przystas und J. Weiland. Von Vielfalt und Variabilität–Handhabung von Funktionsvarianten in Simulink-Modellen. *Elektronik automotive*, 2008.
- [35] C. Dziobek, T. Ringler und F. Wohlgemuth. Herausforderungen bei der modellbasierten Entwicklung verteilter Fahrzeugfunktionen in einer verteilten Entwicklungsorganisation. In *Dagstuhl-Workshop MBEES: Modellbasierte Entwicklung eingebetteter Systeme VIII*, Seiten 1–10, 2012.
- [36] C. Ebert. Improving engineering efficiency with PLM/ALM. *Software & Systems Modeling*, 12(3):443–449, 2013.
- [37] C. Ebert. *Cyber Security Requirements Engineering*, Seiten 209–228. Springer International Publishing, 2017. ISBN 978-3-319-51310-2.
- [38] U. Eklund, Ö. Askerdal, J. Granholm, A. Alminger und J. Axelsson. Experience of introducing reference architectures in the development of automotive electronic systems. *ACM SIGSOFT Software Engineering Notes*, 30(4):1–6, 2005.
- [39] R. Farenhorst, J. F. Hoorn, P. Lago und H. Van Vliet. The lonesome architect. In *Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on*, Seiten 61–70. IEEE, 2009.
- [40] P. H. Feiler, B. Lewis, S. Vestal und E. Colbert. *An Overview of the SAE Architecture Analysis & Design Language (AADL) Standard: A Basis for Model-Based Architecture-Driven Embedded Systems Engineering*, Seiten 3–15. Springer US, Boston, MA, 2005. ISBN 978-0-387-24590-4.
- [41] G. Fischer, T. Hartstock und T. Recknagel. Projektmanagement und Projektsteuerung. *ATZextra*, 16:22–27, 2011.
- [42] G. Fischer, R. Theurer und T. Recknagel. Projektmanagement und Projektsteuerung. *ATZextra*, 17:28–33, 2012.
- [43] C. Florent. Strategic perspectives on modularity. In *DRUID Tenth Anniversary Summer Conference 2005 on Dynamics Of Industry And Innovation: Organizations, Networks And Systems*, Seiten 27–29, 2005.

- [44] E. Frank, R. Wilhelm, R. Ernst, A. Sangiovanni-Vincentelli und M. Di Natale. Methods, tools and standards for the analysis, evaluation and design of modern automotive architectures. In *Proceedings of the conference on Design, automation and test (DATE) in Europe*, Seiten 659–663. ACM, 2008.
- [45] U. Freund, M. von der Beeck, P. Braun und M. Rappl. Architecture centric modeling of automotive control software. Technical report, SAE Technical Paper, 2003.
- [46] Frost&Sullivan. Analysis of vehicle platform strategies of key global OEM. Technical report, Frost&Sullivan M726-18, 2012.
- [47] D. Gebauer, J. Matheis, C. Reichmann und K. Müller-Glaser. Ebenenübertreifende, variantengerechte Beschreibung von Elektrik/Elektronik-Architekturen. *Diagnose in mechatronischen Fahrzeugsystemen*, Seiten 142–151, 2008.
- [48] D. Gebauer, J. Matheis, M. Kühl und K. Müller-Glaser. Integrierter, graphisch notierter Ansatz zur Bewertung von Elektrik/Elektronik-Architekturen im Fahrzeug. *Elektronik im Kraftfahrzeug*, 2009.
- [49] R. Gemmerich, S. Semmelrodt, A. Zündorf, C. Reckord, J. Leohold, J. Trippler, L. Brabetz, D. Müller, U. Schrey und H. Weil. Ein ganzheitlicher Ansatz zur Generierung und Optimierung von Fahrzeugbordnetzen. In *VDI Berichte Nr. 1907 (12th International Conference and Exhibition Electronic Systems for Vehicles Baden-Baden)*, Seiten 597–608, 2005.
- [50] H. Gentner, T. Demmeler, S. Hohmann und O. Wronn. Architekturen für vernetzte aktive und passive Sicherheit. *VDI-Berichte*, (2075), 2009.
- [51] H. Glathe, M. Bittner, M.-O. Reiser und M. Weber. Artefaktübergreifendes Varianten-Management. *Elektronik automotive 1/2*, Seiten 30–33, 2010.
- [52] J. Göpfert und M. Steinbrecher. Modulare Produktentwicklung leistet mehr. *Harvard Business Manager*, 22:20–48, 2000.
- [53] E. Greiner, C. Steuer und J. Schaser. Variabilität durch Standardisierung. *ATZextra*, 17(4):24–27, 2012.
- [54] K. Grimm. Software technology in an automotive company: major challenges. In *Proceedings of the 25th international conference on Software Engineering*, Seiten 498–503. IEEE Computer Society, 2003.

- [55] H. Grönniger, J. Hartmann, H. Krahn, S. Kriebel, L. Rothhardt und B. Rumpe. Modelling automotive function nets with views for features, variants, and modes. *CoRR*, abs/1409.6628, 2014.
- [56] H. Grönniger, J. Hartmann, H. Krahn, S. Kriebel, L. Rothhardt und B. Rumpe. View-centric modeling of automotive logical architectures. *CoRR*, abs/1409.6630, 2014.
- [57] H. Grönniger, H. Krahn, C. Pinkernell und B. Rumpe. Modeling variants of automotive systems using views. *CoRR*, abs/1409.6629, 2014.
- [58] H. Gustavsson und J. Axelsson. Evaluating flexibility in embedded automotive product lines using real options. In *SPLC'08 12th International Software Product Line Conference*, Seiten 235–242. IEEE, 2008.
- [59] H. Gustavsson und J. Axelsson. Evaluation of design options in embedded automotive product lines. In *Applied Software Product Line Engineering*, Seiten 477–496. Auerbach Publications, 2010.
- [60] H. Gustavsson und J. Axelsson. Architecting complex embedded systems: An industrial case study. In *IEEE International Systems Conference (SysCon)*, Seiten 472–478. IEEE, 2011.
- [61] H. Gustavsson und J. Axelsson. A comparative case study of architecting practices in the embedded software industry. In *18th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, Seiten 226–231. IEEE, 2011.
- [62] H. Gustavsson und U. Eklund. Architecting automotive product lines: Industrial practice. In *International Conference on Software Product Lines*, Seiten 92–105. Springer, 2010.
- [63] P. Hank, S. Müller, O. Vermesan und J. Van Den Keybus. Automotive ethernet: in-vehicle networking and smart mobility. In *Proceedings of the Conference on Design, Automation and Test in Europe*, Seiten 1735–1739. EDA Consortium, 2013.
- [64] B. Hedenetz. *Entwurf von verteilten fehlertoleranten Elektronikarchitekturen in Kraftfahrzeugen*. Dissertation, Eberhard-Karls-Universität Tübingen, 2001.
- [65] M. Helmling. Integrierte Entwicklung einer vollständigen E/E-Architektur. *Hanser automotive*, (01-02):28–31, 2017.
- [66] M. Hemprich und G. Schwefer. Model based E/E-architecture development at Daimler ... and a look at the broader picture. In *6. Vector Congress*, 2012.

- [67] B. Hense. Entwurf zukünftiger E/E-Architekturen im Kraftfahrzeug. Vorlesung, TU Dresden, 2009.
- [68] P. Heymans, P.-Y. Schobbens, J.-C. Trigaux, Y. Bontemps, R. Matulevičius und A. Classen. Evaluating formal properties of feature diagram languages. *IET software*, 2(3):281–302, 2008.
- [69] J. Higgins, D. Siebert, G. Sobczyk und D. Turner. SMART tool based analysis of E/E subsystems. Technical report, SAE Technical Paper, 2007.
- [70] A. Hohm, F. Lotz, O. Fochler, S. Lueke und H. Winner. Automated driving in real traffic: from current technical approaches towards architectural perspectives. Technical report, SAE Technical Paper, 2014.
- [71] C. Hopp, F. Wolf, H. Rendel, B. Rumpe und V. A. Braunschweig. Einführung eines Produktlinienansatzes in die automotiv Softwareentwicklung am Beispiel von Steuergerätesoftware. In *Software Engineering*, Seiten 181–192, 2012.
- [72] M. Hüttenrauch und M. Baum. *Effiziente Vielfalt*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-72115-4.
- [73] IEEE. IEEE Standard for Ethernet. Standard, IEEE 802.3-2015, 2015.
- [74] ISO. Road vehicles – Controller area network (CAN) – Part 1 - Part 6. Standard, ISO 11898, 2003-2016.
- [75] ISO. Road vehicles – Diagnostic communication over Internet Protocol (DoIP) – Part 1 - Part 4. Standard, ISO 13400, 2011-2016.
- [76] ISO. Road vehicles – FlexRay communications system – Part 1 - Part 5. Standard, ISO 11898:2013, 2013.
- [77] ISO. Road vehicles – Local Interconnect Network (LIN) – Part 1 - Part 7. Standard, ISO 17987:2016, 2016.
- [78] ISO/IEC. Information technology – Meta Object Facility (MOF). Standard, ISO/IEC 19502:2005, 2005.
- [79] ISO/IEC/IEEE. Systems and software engineering – Architecture description. Standard, ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), 2011.
- [80] M. Jackson. Aspects of abstraction in software development. *Software & Systems Modeling*, 11(4):495–511, 2012.

- [81] M. Jaensch. *Modulorientiertes Produktlinien Engineering für den modellbasierten Elektrik/Elektronik-Architekturentwurf*. Dissertation, Karlsruher Institut für Technologie (KIT), Fakultät für Elektrotechnik und Informationstechnik, 2012.
- [82] M. Jaensch, B. Hedenetz, M. Conrath und K. D. Müller-Glaser. Transfer von Prozessen des Software-Produktlinien Engineering in die Elektrik/Elektronik-Architekturentwicklung von Fahrzeugen. In *GI Jahrestagung (2)*, Seiten 497–502, 2010.
- [83] M. Jaensch, M. Conrath, B. Hedenetz und K. D. Müller-Glaser. Integration of Electrical/Electronic-relevant modules into the model-based design of Electrical/Electronic-architectures. In *1. Energy Efficient Vehicle Conference (EEVC)*, Seiten 84–92, 2011.
- [84] M. Jaensch, P. Prehl, G. Schwefer, K. D. Müller-Glaser und M. Conrath. Model-Based Design of E/E Architectures 2.0. *ATZelektronik worldwide eMagazine*, 6(4):16–21, 2011.
- [85] J.-M. Jézéquel. Model driven design and aspect weaving. *Software & Systems Modeling*, 7(2):209–218, 2008.
- [86] A. Kampker, A. Gulden und C. Deutschens. Kostenpotenziale modular aufgebauter Elektrofahrzeuge. *ATZ-Automobiltechnische Zeitschrift*, 112(4):258–263, 2010.
- [87] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak und A. S. Peterson. Feature-oriented domain analysis (FODA) feasibility study (No. CMU/SEI-90-TR-21). Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst., 1990.
- [88] K. C. Kang, J. Lee und P. Donohoe. Feature-oriented product line engineering. *IEEE software*, 19(4):58–65, 2002.
- [89] C. Kästner, S. Apel, M. Rosenmüller, D. Batory, G. Saake et al. On the impact of the optional feature problem: Analysis and case studies. In *Proceedings of the 13th International Software Product Line Conference*, Seiten 181–190. Carnegie Mellon University, 2009.
- [90] E. Kelling und T. Raste. Trends in der Systemvernetzung am Beispiel einer skalierbaren E/E-Architektur für die Domänen Vehicle Motion und Safety. In *VDI-Kongress Elektronik im Kraftfahrzeug, Baden-Baden*, 2011.
- [91] R. S. Khokar, N. Robson und P. Dégardins. Eine Elektrik/Elektronik-Plattform für Low-cost und High-tech zugleich. *ATZelektronik*, 4(6): 22–29, 2009.

- [92] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier und J. Irwin. Aspect-oriented programming. In *ECOOP'97—Object-oriented programming*, Seiten 220–242. Springer, 1997.
- [93] M. Krämer, A. Röhringer, W. Kirchner, T. Vogel, J. Rochlitzer und B. Heiniger. Programm-Management und Projektsteuerung. *ATZextra*, 14:150–155, 2009.
- [94] S. Kubica, W. Friess, A. Christian und T. Koelzow. Domain-crossing software product lines in embedded, automotive systems. *From Specification to Embedded Systems Application*, Seiten 1–12, 2005.
- [95] P. Larses. Factors influencing dependable modular architectures for automotive applications. Technical report, Department of Machine Design, Royal Institute of Technology, 2005.
- [96] G. Leen und D. Heffernan. Expanding automotive electronic systems. *Computer*, 35(1):88–93, 2002.
- [97] M. M. Lehman. Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE*, 68(9):1060–1076, 1980.
- [98] K. Lind und R. Heldal. Automotive system development using reference architectures. In *35th Annual IEEE Software Engineering Workshop (SEW)*, Seiten 42–51. IEEE, 2012.
- [99] J. Liu, D. Batory und C. Lengauer. Feature oriented refactoring of legacy applications. In *Proceedings of the 28th International Conference on Software Engineering*, Seiten 112–121. ACM, 2006.
- [100] H. Lönn, T. Saxena, M. Nolin und M. Törngren. Far east: Modeling an automotive software architecture using the EAST ADL. In *ICSE 2004 workshop on Software Engineering for Automotive Systems (SEAS)*, Seiten 43–50. Institution of Engineering and Technology (IET), 2004.
- [101] A. Lübke und H.-C. Reuss. Optimierung der Bordnetzarchitektur mit Hilfe genetischer Algorithmen. *ATZ-Automobiltechnische Zeitschrift*, 102(6):436–442, 2000.
- [102] J. Mancke und M. Kardos. Product line engineering - a key for reuse. In *6. Vector Congress*, 2012. URL http://vector.com/vi_congress12_de.html.
- [103] J. Matheis, D. Gebauer, M. Kühl, K. Reichmann und K. D. Müller-Glaser. Vorstellung einer Methodik zur E/E-Architektur-Modellierung

- und -Bewertung in der frühen Konzeptphase. *Moderne Elektronik im Kraftfahrzeug*, 1:66–80, 2006.
- [104] J. Matheis, D. Gebauer, C. Reichmann und K. Müller-Glaser. Ganzheitliche abstraktionsebenenübergreifende Beschreibung konsistenter Elektrik/Elektronik-Architekturen. In *Systems Engineering Infrastructure Conference Seisconf*, 2008.
- [105] J. D. McGregor. Software product lines. *Journal of Object Technology*, 3(3):65–74, 2004.
- [106] K. Melin. Volvo S80: Electrical system of the future. *Volvo Technology Report*, 1:3–7, 1998.
- [107] A. Meroth und B. Tolg. *Infotainmentsysteme im Kraftfahrzeug: Grundlagen, Komponenten, Systeme und Anwendungen*. Vieweg+Teubner Verlag, 2008. ISBN 978-3-834-89430-4.
- [108] D. Merschen, Y. Duhr, T. Ringler, B. Hedenetz und S. Kowalewski. Model-based analysis of design artefacts applying an annotation concept. In *Software Engineering*, Seiten 169–180, 2012.
- [109] A. Michailidis, T. Ringler, B. Hedenetz und S. Kowalewski. Virtuelle Integration modellbasierter Fahrzeugfunktionen unter AUTOSAR. *AT-Zelektronik*, 5(1):32–37, 2010.
- [110] A. Michailidis, U. Spieth, T. Ringler, B. Hedenetz und S. Kowalewski. Test front loading in early stages of automotive software development based on autosar. In *Proceedings of the Conference on Design, Automation and Test in Europe*, Seiten 435–440. European Design and Automation Association, 2010.
- [111] S. Mischo, I. Krauter, U. Kersken, R. Schöttle et al. E/E Architectures at the crossroads. *ATZelektronik worldwide*, 3(5):10–13, 2008.
- [112] J. Mössinger. Software in automotive systems. *IEEE Software*, 27(2), 2010.
- [113] L. M. Northrop. SEI’s software product line tenets. *IEEE software*, 19(4):32–40, 2002.
- [114] I. Ozkaya, A. Diaz-Pace, A. Gurfinkel und S. Chaki. Using architecturally significant requirements for guiding system evolution. In *14th European Conference on Software Maintenance and Reengineering (CSMR)*, Seiten 127–136. IEEE, 2010.

- [115] C. Pahl, S. Giesecke und W. Hasselbring. Ontology-based modelling of architectural styles. *Information and Software Technology*, 51(12):1739–1749, 2009.
- [116] G. Pahl, W. Beitz, J. Feldhusen und K.-H. Grote. *Konstruktionslehre*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-34060-7.
- [117] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972.
- [118] D. L. Parnas. On the design and development of program families. *IEEE Transactions on software engineering*, (1):1–9, 1976.
- [119] P. Popp, M. Di Natale, P. Giusto, S. Kanajan und C. Pinello. Towards a methodology for the quantitative evaluation of automotive architectures. In *Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE'07*, Seiten 1–6. IEEE, 2007.
- [120] S. Powolny, T. Friedrich, S. Mischo, R. Kornhaas, H. Mangold und C. Reichmann. Model based top down process for automotive E/E-architecture development. Technical report, SAE Technical Paper, 2008.
- [121] PREEvision Version 5.5. Manual. Benutzerhandbuch, Vector Informatik GmbH, 2012.
- [122] C. Prehofer. Feature-oriented programming: A fresh look at objects. In *ECOOP'97—Object-Oriented Programming*, Seiten 419–443. Springer, 1997.
- [123] M. Rahmani, J. Hillebrand, W. Hintermaier, R. Bogenberger und E. Steinbach. A novel network architecture for in-vehicle audio and video communication. In *2nd IEEE/IFIP International Workshop on Broadband Convergence Networks (BcN'07)*, Seiten 1–12. IEEE, 2007.
- [124] S. Raue, M. Conrath, B. Hedenetz und W. Rosenstiel. Assessing the evolution of E/E hardware modules with conceptual function architectures. In J. Fischer-Wolfarth und G. Meyer, editors, *Advanced Microsystems for Automotive Applications 2014*, Lecture Notes in Mobility, Seiten 61–69. Springer International Publishing, 2014. ISBN 978-3-319-08086-4.
- [125] S. Raue, M. Conrath, B. Hedenetz und W. Rosenstiel. A system-oriented approach towards an integrated module-oriented model-based Electrical/Electronic architecture design. In *3. International Conference on Energy Efficient Vehicles (ICEEV)*, Juni 2014.

- [126] G. Reichart und J. Bielefeld. Einflüsse von Fahrerassistenzsystemen auf die Systemarchitektur im Kraftfahrzeug. In *Handbuch Fahrerassistenzsysteme*, Seiten 84–92. Springer, 2009.
- [127] G. Reichart und M. Haneberg. Key drivers for a future system architecture in vehicles. Technical report, SAE Technical Paper, 2004.
- [128] K. Reif. *Automobilelektronik: Eine Einführung für Ingenieure*. Springer Fachmedien Wiesbaden, Wiesbaden, 2014. ISBN 978-3-658-05048-1.
- [129] N. Reindl. E/E-Fahrzeugarchitekturen der Zukunft. In *ELMOS Workshop - Mobilität 2020 ff ...*, 2010.
- [130] D. Reinhardt, D. Kaule und M. Kucera. Achieving a scalable E/E-architecture using autosar and virtualization. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 6(2013-01-1399): 489–497, 2013.
- [131] T. Ringler, M. Simons und R. Beck. Reifegradsteigerung durch methodischen Architekturentwurf mit dem E/E-Konzeptwerkzeug. *13. Internationaler Kongress Elektronik im Kraftfahrzeug, Baden-Baden, VDI-Berichte Nr. 2000*, 2007.
- [132] T. Ringler, M. Simons und R. Beck. An approach to tool-based design of Electrics/Electronics architectures. *ATZechnik worldwide*, 3(1): 42–46, 2008.
- [133] R. Rotter. Die nächste Evolutionsstufe im Requirements Engineering. *Elektronik automotive*, (11):40–43, 2017.
- [134] M. S. Salerno und A. V. C. Dias. Product design modularity, modular production, modular organization: the evolution of modular concepts. *Acets du GERPISA n° 33*, Seiten 61–73, 2002.
- [135] C. Salzmann und T. Stauner. Automotive software engineering. *Languages for system specification*, Seiten 333–347, 2004.
- [136] J. Schäuuffele. Moderne E/E-Architekturen optimieren. *automobil elektronik*, (11-12):42–45, 2016.
- [137] J. Schäuuffele. E/E architectural design and optimization using PREEvision. Technical report, SAE Technical Paper, 2016.
- [138] J. Schäuuffele und T. Zurawka. *Automotive Software Engineering*. Springer Fachmedien Wiesbaden, Wiesbaden, 2013. ISBN 978-3-8348-2469-1.

- [139] S. Schmerler, T. Ringler, B. Hedenetz, U. Grüner, F. Wohlgemuth, C. Dziobek und P. Lohrmann. Mit AUTOSAR zu einem integrierten und durchgängigen Entwicklungsprozess. In *VDI-Kongress Elektronik im Kraftfahrzeug, Baden-Baden*, 2011.
- [140] M. Schopper und J. Missel. Mehr Sicherheit durch vernetzte Systeme. *ATZextra*, 19(8):76–83, 2014.
- [141] M. Schopper, H.-J. Koch, O. Späth und M. Feldmann. Radarbasierte Assistenzsysteme. *ATZextra*, 17(4):82–85, 2012.
- [142] M. Schopper, L. Henle und T. Wohland. Intelligent Drive Vernetzte Intelligenz für mehr Sicherheit. *ATZextra*, 18(5):106–114, 2013.
- [143] M. Schöttle. Zukunft der Fahrerassistenz mit neuen E/E-Architekturen. *ATZelektronik*, 6(4), 2011.
- [144] V. Schulte-Coerne, A. Thums und J. Quante. Challenges in reengineering automotive software. In *Software Maintenance and Reengineering, 2009. CSMR'09. 13th European Conference on*, Seiten 315–316. IEEE, 2009.
- [145] A. P. Slowak und M. Itohisa. Who profits from automotive electronics standards? Facing cost pressures. *VDI-Berichte*, 2132:639–650, 2011.
- [146] W. Stolz, R. Kornhaas, R. Krause und T. Sommer. Domain control units - the solution for future E/E architectures? Technical report, SAE Technical Paper, 2010.
- [147] T. Streichert und M. Traub. *Elektrik/Elektronik-Architekturen im Kraftfahrzeug*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-25477-2.
- [148] I. Stürmer, H. Pohlheim und T. Rogier. Berechnung und Visualisierung der Modellkomplexität bei der modellbasierten Entwicklung sicherheitsrelevanter Software. *Automotive-Safety & Security*, Seiten 69–82, 2010.
- [149] A. Takeishi und T. Fujimoto. Modularisation in the auto industry: inter-linked multiple hierarchies of product, production and supplier systems. *International Journal of Automotive Technology and Management*, 1(4): 379–396, 2001.
- [150] D. Tamzalit und T. Mens. Guiding architectural restructuring through architectural styles. In *Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on*, Seiten 69–78. IEEE, 2010.

- [151] V. Tannenberger und M. Paetz. Modularisierung ermöglicht Individualisierung. *ATZextra*, 17(6):64–71, 2012.
- [152] J.-L. Terraillon, A. Jung, P. Arberet, S. Montenegro, A. Rossignol, G. Garcia, A. I. Rodriguez, S. Mazzini, P. Hougaard, S. Fowell et al. Space on-board software reference architecture. In *DASIA 2010 Data Systems In Aerospace*, volume 682, 2010.
- [153] S. Thiel und A. Hein. Modelling and using product line variability in automotive systems. *IEEE software*, 19(4):66–72, 2002.
- [154] S. Thiel, S. Ferber, T. Fischer, A. Hein und M. Schlick. A case study in applying a product line approach for car periphery supervision systems. Technical report, SAE Technical Paper, 2001.
- [155] S. Thiel, M. A. Babar, G. Botterweck und L. O’Brien. Software product lines in automotive systems engineering. *SAE international journal of passenger cars-electronic and electrical systems*, 1(2008-01-1449):531–543, 2008.
- [156] J. Thomas, C. Dziobek und B. Hedenetz. Variability management in the AUTOSAR-based development of applications for in-vehicle systems. In *Proceedings of the 5th Workshop on Variability Modeling of Software-Intensive Systems*, Seiten 137–140. ACM, 2011.
- [157] J. Thyssen, D. Ratiu, W. Schwitzer, A. Harhurin, M. Feilkas und E. Thaden. A system for seamless abstraction layers for model-based development of embedded software. In *Software Engineering*, 2010.
- [158] C. Tischer, A. Muller, M. Ketterer und L. Geyer. Why does it take that long? Establishing product lines in the automotive domain. In *Software Product Line Conference, 2007. SPLC 2007. 11th International*, Seiten 269–274. IEEE, 2007.
- [159] D. D. Turner. Determining the optimal distributed electronic module solution of an automotive system while incorporating harness routing alternatives in an Electrical/Electronic architecture tool environment. Technical report, SAE Technical Paper, 2008.
- [160] K. Ulrich. The role of product architecture in the manufacturing firm. *Research policy*, 24(3):419–440, 1995.
- [161] T. Unselt. Herausforderungen im Spannungsfeld Sicherheit, Leichtbau und Wirtschaftlichkeit. *ATZextra*, 17(9):12–15, 2012.
- [162] F. Van der Linden. Software product families in europe: the esaps & café projects. *IEEE software*, 19(4):41–49, 2002.

- [163] VDA. Harness Description List (KBL). VDA-Recommendation 4964, Project Group “Car Electric” of VDA Working Group “PLM”, 2014.
- [164] Vector Informatik GmbH. Vector Informatik GmbH Website, 2018. URL <https://vector.com/>.
- [165] S. Voget. Future trends in software architectures for automotive systems. *Advanced Microsystems for Automotive Applications 2003*, Seiten 457–469, 2003.
- [166] S. Voget und M. Becker. Establishing a software product line in an immature domain. In *International Conference on Software Product Lines*, Seiten 60–67. Springer, 2002.
- [167] M. von der Beeck. Development of logical and technical architectures for automotive systems. *Software & Systems Modeling*, 6(2):205–219, 2007.
- [168] P. Wallin und J. Axelsson. A case study of issues related to automotive E/E system architecture development. In *Engineering of Computer Based Systems, 2008. ECBS 2008. 15th Annual IEEE International Conference and Workshop on the*, Seiten 87–95. IEEE, 2008.
- [169] P. Wallin, S. Larsson, J. Fröberg und J. Axelsson. Problems and their mitigation in system and software architecting. *Information and Software Technology*, 54(7):686–700, 2012.
- [170] P. Waszecki, M. Lukasiewicz, A. Masrur und S. Chakraborty. How to engineer tool-chains for automotive E/E architectures? *ACM SIGBED Review*, 10(4):6–15, 2013.
- [171] T. Weidner, D. Stüker, S. Wender, R. Katzwinkel und A. Vukotich. Skalierbare E/E-Architekturen als Enabler innovativer Fahrerassistenzfunktionen bis zum hochautomatisierten Fahren. In *VDI-Kongress Elektronik im Kraftfahrzeug, Baden-Baden*, 2013.
- [172] U. Weist, J. Missel, S. Cytrynski, D. Mehren, T. Schwarz und S. Kern. Fahrkomfort der Extraklasse. *ATZextra*, 18(5):124–128, 2013.
- [173] B. J. Williams und J. C. Carver. Characterizing software architecture changes: A systematic review. *Information and Software Technology*, 52(1):31–51, 2010.
- [174] H. Winner, S. Hakuli und G. Wolf. *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Vieweg+Teubner, 2009. ISBN 978-3-8348-9977-4.

- [175] D. Ziegenbein, P. Braun, U. Freund, A. Bauer, J. Romberg und B. Schatz. Automode-model-based development of automotive software. In *Design, Automation and Test in Europe, 2005. Proceedings*, Seiten 171–176. IEEE, 2005.
- [176] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller et al. Making Bertha drive - An autonomous journey on a historic route. *Intelligent Transportation Systems Magazine, IEEE*, 6(2):8–20, 2014.
- [177] G. Zimmermann. PREEvision goes to series development. In *7. Vector Congress*, 2014.
- [178] W. Zimmermann und R. Schmidgall. *Bussysteme in der Fahrzeugtechnik: Protokolle, Standards und Softwarearchitektur*. Springer Fachmedien Wiesbaden, Wiesbaden, 2014. ISBN 978-3-658-02419-2.