



ulm university universität
uulm

**Fakultät für Ingenieurwissenschaften
und Informatik**

Institut für Programmiermethodik
und Compilerbau

Leiter: Prof. Dr. Helmuth Partsch

Variantenmanagement in Anforderungsdokumenten für hochkomplexe und variantenreiche Produkte

Dissertation
zur Erlangung des Doktorgrades
Doktor der Naturwissenschaften (Dr. rer. nat.)
der Fakultät für Ingenieurwissenschaften
und Informatik
der Universität Ulm

Ekaterina Boutkova
aus
Leningrad

2014

Amtierender Dekan:

Prof. Dr. Tina Seufert

Gutachter:

Prof. Dr. H. Partsch

Gutachter:

Prof. Dr. F. Schweiggert

Tag der Promotion

26.06.2014

Danksagung

Meine Arbeit ist fast fertig und es ist an der Zeit, mich bei denjenigen zu bedanken, die mich in dieser spannenden Lebensphase begleitet haben.

Zu besonderem Dank bin ich meinem Doktorvater Professor Partsch verpflichtet, der mich stets mit seinen Anregungen unterstützt hat. Auch beim Herrn Professor Schweigert möchte ich mich sein zweites Gutachten herzlich Bedanken. Ohne ihren wertvollen akademischen Rat wäre diese Arbeit nicht entstanden.

Ebenso geht mein Dank an meine Kollegen von Daimler AG und Carmeq GmbH, die mich in den vergangenen Jahren mit Rat und Tat unterstützt haben. Jörg, Laura, Frank, Nadine, Steffi, Katrin, Frank, Viktor, Helko, Sven, Baraa, Hulya - ohne euch wäre meine Arbeit nie fertig geworden.

Und natürlich möchte ich mich bei meinem Ehemann und meinen beiden Töchtern bedanken, ohne Ihre Geduld und Unterstützung wäre diese Arbeit nicht zu dem Werk geworden, das sie heute ist.

Wolfsburg, im Juli 2014

Ekaterina Boutkova

Für Andromeda und Anthea

Inhaltsverzeichnis

Inhaltsverzeichnis.....	i
Abbildungsverzeichnis.....	iii
Tabellenverzeichnis.....	iv
Zusammenfassung.....	v
Kapitel 1 Einführung.....	1
1.1 Motivation.....	1
1.2 Begriffe.....	3
1.2.1 Anforderungsmanagement.....	3
1.2.2 Produktlinienentwicklung.....	4
1.2.3 Variantenmanagement.....	5
1.3 Problembeschreibung.....	7
1.4 Ziel und Beiträge der Arbeit.....	9
1.5 Aufbau der Arbeit.....	10
Kapitel 2 Stand der Forschung.....	13
2.1 Anforderungsdokumente.....	13
2.1.1 Qualität von Anforderungsdokumenten.....	14
2.1.2 Struktur von Anforderungsdokumenten.....	15
2.1.3 Abstraktionsebenen.....	16
2.1.4 Werkzeuge für Anforderungsmanagement.....	18
2.2 Variantenmanagement in Anforderungsdokumenten.....	18
2.2.1 Variabilitätsdokumentation.....	19
2.2.2 Merkmalsmodellierung.....	22
2.3 Variabilitätsdokumentation in Anforderungsdokumenten.....	29
2.4 Diskussion.....	31
2.4.1 Alternative Ansätze.....	31
2.4.2 Abgrenzung zur alternativen Ansätzen.....	33
Kapitel 3 Stand der Praxis.....	35
3.1 Anforderungsdokumente in der Praxis.....	35
3.1.1 Anforderungsdokumentenlandschaft.....	36
3.1.2 Eigenschaften von Anforderungsdokumenten.....	37
3.1.3 Szenarien für Anforderungswiederverwendung.....	38
3.2 Variantenmanagement in Anforderungsdokumenten.....	39
3.2.1 Variabilitätsdokumentationsansätze.....	39
3.2.2 Herausforderungen für Variantenmanagementansätze.....	43
3.3 Diskussion.....	45
Kapitel 4 Merkmalsidentifizierungsansatz (MIA).....	47
4.1 Manuelle Identifizierungen von Merkmalen.....	47
4.2 Semi-automatischer Ansatz.....	48
4.2.1 Identifikation von Substantiven.....	49
4.2.2 Normalisierung von Substantiven.....	50
4.2.3 Bereinigung der Kandidatenliste.....	52
4.3 MIA-Tool.....	53
4.4 Evaluation des Algorithmus.....	55
4.4.1 Phase I.....	56
4.4.2 Phase 2.....	59
4.5 Diskussion.....	61
4.5.1 Alternative Ansätze.....	61
4.5.2 Abgrenzung zu alternativen Ansätzen.....	63
Kapitel 5 Formalisierung.....	65
5.1 Vorteile der formalen Darstellung.....	65
5.2 Formale Darstellung der Anforderungsdokumente.....	66
5.3 Formale Darstellung eines Variabilitätsmodells.....	67

Kapitel 6 Konfigurationsgekoppelte dezentrale Variabilitätsmodellierung	71
6.1 Zentrale vs. dezentrale Variabilitätsmodellierung	71
6.1.1 Zentrales Variabilitätsmodell	71
6.1.2 Dezentrale Variabilitätsmodelle	72
6.2 Der KDVM-Ansatz	73
6.3 Diskussion	81
Kapitel 7 R2F-Ansatz.....	83
7.1 Variabilität in Anforderungsdokumenten	83
7.2 Variantenmanagementprozess	85
7.3 Semi-automatische Tool-Unterstützung	87
7.3.1 Zuordnungsvorschlag (ZV)	87
7.3.2 Zuordnungsüberprüfung (ZÜ)	88
7.3.3 Evaluierung der R2F-Algorithmen	89
7.4 Diskussion	92
Kapitel 8 Unterstützung der Konfigurationserstellung.....	95
8.1 Unterstützung durch optimale Fragenfolge	95
8.2 Bestimmung der optimalen Entscheidungsfolge	96
8.2.1 Berechnung der lokalen Konfigurationsanzahl	98
8.2.2 Berechnung der Auswahl-Wahrscheinlichkeit	106
8.2.3 Berechnung der optimalen Entscheidungsfolge	109
8.2.4 Optimale Entscheidungsfolge bei KDVM	114
8.3 Diskussion	114
Kapitel 9 Einführungs- und Migrationsvorschlag	117
9.1 Einführungsprozess	117
9.1.1 Begriffsklärung	118
9.1.2 Ausprägungen der MBVM-Methode	118
9.1.3 Migrationsprozess	119
9.2 Toolunterstützung	120
9.2.1 Erweiterung der DOORS-Funktionalität	121
9.2.2 Anwendung des Varianten-Moduls	123
9.3 Zusammenfassung	125
Kapitel 10 Evaluation.....	127
10.1 Evaluation mit Studenten an der Universität Ulm	127
10.1.1 Design der Fallstudie	127
10.1.2 Ablauf der Fallstudie	128
10.2 Evaluation mit wissenschaftlichen Mitarbeitern	129
10.2.1 Vergleich des R2F- und Matrix-Ansätze	129
10.2.2 Identifikation der Merkmale	130
10.3 Evaluation bei der Daimler AG	131
10.3.1 Fallstudie mit Werkstudenten	131
10.3.2 Pilotanwendungen	132
Kapitel 11 Zusammenfassung und Ausblick.....	133
11.1 Zusammenfassung	133
11.2 Ausblick	136
Anhang	139
Literaturverzeichnis.....	151

Abbildungsverzeichnis

Abbildung 1: Begriffslandschaft	4
Abbildung 2: Metamodell der Variabilitätsmodellierung nach [PoBL05]	20
Abbildung 3: Metamodell der Variabilitätsmodellierung	20
Abbildung 4: Beispiel des Merkmalsmodells mit FODA-Notation	25
Abbildung 5: Beispiel für die kardinalitätsbasierte Notation eines Merkmalsmodells	26
Abbildung 6: Beispiel des Merkmalsmodells mit FODA-Notation	27
Abbildung 7: Redundante Modellierung	28
Abbildung 8: Objekt-Hierarchie-Ansatz	29
Abbildung 9: Nachvollziehbarkeitslinks	30
Abbildung 10: Attribuierung von Anforderungen	31
Abbildung 11: Informationsfluss zwischen unterschiedlichen Anforderungsdokumententypen	37
Abbildung 12: Merkmalsidentifikationsansatz (MIA)	48
Abbildung 13: Evaluation von Stanford Tagger	50
Abbildung 14: Normalisierung in MIA	51
Abbildung 15: MIA-Tool – Festlegung des Imports	53
Abbildung 16: MIA-Tool – Festlegung der Filterwerte	54
Abbildung 17: MIA-Tool	55
Abbildung 18: Beispiel für den KDVM-Ansatz	75
Abbildung 19: Morphologischer Katen	77
Abbildung 20: Beispiel einer Verträglichkeitsmatrix	77
Abbildung 21: Konfigurationsmatrix	78
Abbildung 22: Beispiel zur Verwendung von „Configuration Links“ [Reis08]	79
Abbildung 23: Auszug des Variabilitätsmodells der Komponente <i>Sonnenblende</i>	85
Abbildung 24: R2F-Prozess bei der Einführung des R2F-Ansatzes	86
Abbildung 25: R2F-Prozess bei der Lastenhefterstellung	87
Abbildung 26: Hierarchieebenen in einem Merkmalmodell	98
Abbildung 27: Beispielsmerkmalsmodell	99
Abbildung 28: Erweiterung des Beispiels um Hilfsmerkmale	101
Abbildung 29: Beispiel-Merkmalmodell mit Abhängigkeitsbeziehungen	110
Abbildung 30: Dokumentation der Variabilitätskriterien im Varianten-Modul	123
Abbildung 31: Spezifikationsmodul mit Zuordnungen	124

Tabellenverzeichnis

Tabelle 1: Abstraktionsebenen	17
Tabelle 2: Textuelle Ansatz.....	40
Tabelle 3: Matrix-Ansatz	41
Tabelle 4: Spalten-Ansatz	41
Tabelle 5: Code-Regel-Ansatz	43
Tabelle 6: Bewertung existierender Ansätze	43
Tabelle 7: Evaluierungsdatensätze	55
Tabelle 8: Erster Durchlauf in der Phase I	57
Tabelle 9: Zweiter Durchlauf in der Phase I	58
Tabelle 10: Erster Durchlauf in der Phase II.....	59
Tabelle 11: Zweiter Durchlauf in der Phase II	60
Tabelle 12: Klassifikation für die Trefferquote und Genauigkeit nach [HaDS05].....	61
Tabelle 13: Elemente eines Merkmalsmodells	67
Tabelle 14: Notationsformen.....	84
Tabelle 15: Evaluierungsdatensätze	89
Tabelle 16: Evaluierung des R2F-Zuordnungsalgorithmus für DS2	90
Tabelle 17: Evaluierung des ZÜ-Algorithmus.	91
Tabelle 18: Erfüllung der Anforderungen an VM-Ansatz.....	94
Tabelle 19: Entscheidungsfolge ohne Beziehungsberücksichtigung.....	110
Tabelle 20: Entscheidungsfolge mit Beziehungsberücksichtigung	111
Tabelle 21: Optimale Entscheidungsfolge nach Phase 1	112
Tabelle 22: Optimale Entscheidungsfolge nach Phase 2.....	112
Tabelle 23: Optimale Entscheidungsfolge nach Phase e	112
Tabelle 24: Optimale Entscheidungsfolge nach Phase 4.....	113
Tabelle 25: Optimale Entscheidungsfolge.....	114
Tabelle 26: Begriffe der Variabilitätsbeschreibung.....	118
Tabelle 27: Analyse der VM-Werkzeuge.....	121
Tabelle 28: Attribut Object Type	122
Tabelle 29: Erstellung der Produktspezifikationen	130
Tabelle 30: Erstellung der Produktspezifikationen	130
Tabelle 31: Erstellung der Produktspezifikationen	131
Tabelle 32: Fallstudie MIA bei Daimler AG.....	131

Zusammenfassung

In der modernen Produktentwicklung spielen Produktvarianten eine zentrale Rolle. Aus einer Produktidee werden in der Regel eine ganze Reihe ähnlicher Produkte entwickelt. Die Ursachen für Produktvarianten sind dabei mannigfaltig. Ein Grund für Variantenvielfalt liegt in der Bestrebung der Hersteller, Wettbewerbsvorteile durch gezielte Kundenansprache zu erzielen. Mit der Produktvariantenvielfalt versuchen die Hersteller sowohl neue Kunden zu gewinnen, als auch die alten Kunden zu halten. Dafür werden bei der Variantenbildung beispielsweise länderspezifische Aspekte, kundenspezifische Wünsche und technische Neuigkeiten berücksichtigt.

Die Reduktion der Entwicklungszeit durch Wiederverwendung ist der Leitgedanke der letzten Jahrzehnte, besonders in der Automobilindustrie. Andererseits erfordert die große Variantenanzahl längere Entwicklungszeiten, die aufgrund der aktuellen Wettbewerbssituation nicht zur Verfügung stehen. Um dieses Paradoxon zu lösen werden aus einer Produktidee oft eine ganze Reihe ähnlicher Produkte (sogenannte Produktfamilien) entwickelt. Um die Gemeinsamkeiten und Unterschiede zwischen einzelnen Produkten einer Produktfamilie verwalten zu können, wird Variantenmanagement eingesetzt.

Mit der steigenden Komplexität der Produkte sind auch die Anforderungen an die einzelnen Komponenten und Systeme immer vielfältiger geworden. Reichte früher eine Zeichnung und eine Stückliste für die Anforderungsdokumentation einer Komponente, so ist heute ein umfassendes Anforderungsdokument erforderlich. Die Anforderungsdokumente für einzelne Produkte einer Produktfamilie bieten ein großes Wiederverwendungspotential. Der Schlüssel für eine effiziente Wiederverwendung ist das Variantenmanagement. Ein effektives Variantenmanagement für Anforderungsdokumente muss den Entwicklern die Methoden zur Verfügung stellen, welche die Identifikation, Dokumentation und Verwaltung von Anforderungsvarianten mit geringem Zeitaufwand sowie tool-gestützte Prüfungen ermöglichen.

Im Bereich des Anforderungsmanagements fehlt oft der systematische Einsatz des Variantenmanagements. Der zentrale Grund hierfür ist, dass Variantenmanagement meist als separate Aktivität betrieben wird. Die Verbindung mit dem Anforderungsmanagement findet typischerweise nicht oder nur auf der obersten Ebene (z.B. Teilsysteme) statt. Für die Dokumentation von Variabilität in Anforderungsdokumenten existiert zwar eine Reihe von Ansätzen, diese sind aber sehr aufwändig und erfüllen nicht alle Bedürfnisse der Industrie.

Die Nutzung von Merkmalsmodellierung in Anforderungsdokumenten ist eine Ausnahme in der Industrie. Die Gründe dafür bilden mehrere offene Problemfelder, die die Nutzung von Merkmalsmodellen in Anforderungsdokumenten für die Industrie ineffizient bzw. unmöglich machen. Beispiele für die existierende Problemfelder sind: die Komplexität der Variabilitätsmodelle, nicht-effektive Ansätze zur Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten und fehlende Beispiele von Einführungsstrategien. Diese und weitere Problemfelder werden im Rahmen dieser Arbeit vorgestellt, diskutiert und bearbeitet.

In der vorliegenden Arbeit wurden mehrere Ansätze zur Unterstützung des Variantenmanagements in Anforderungsdokumenten bei der hoch-komplexen und variantenreichen Produktentwicklung auf Basis der Merkmalsmodellierung erarbeitet. Die zentralen Beiträge diese Arbeit sind:

- (i) Ein konfigurationsbasierter dezentraler Variabilitätsmodellierungsansatz für die Beherrschung der Komplexität von Merkmalsmodellen in großen Organisationen. Bei der dezentralen Variabilitätsmodellierung wird die Variabilität auf den unterschiedlichen Abstraktionsebenen von Anforderungsdokumenten dokumentiert. Dabei wird für die globalen Produkteigenschaften eine zusätzliche Ebene eingeführt. Die Kopplung zwischen einzelnen Ebenen wird über Konfigurationen realisiert.

- (ii) Für die Identifikation von Merkmalen in existierenden Anforderungsdokumenten bietet diese Arbeit einen semi-automatischen Merkmalsidentifikationsansatz (MIA). Mithilfe des MIA wird automatisch eine Merkmalskandidatenliste erstellt, die dem Benutzer eine Unterstützung bei der Merkmalsidentifikation bietet.
- (iii) Ein zentraler Aspekt dieser Arbeit ist der praktische Ansatz zur Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten sowie zur Verbindung von Anforderungsdokumenten mit Merkmalsmodellen.
- (iv) Für die Unterstützung des Konfigurationsprozesses bietet diese Arbeit einen Ansatz, der es auf der Basis einer optimalen Entscheidungsfolge dem Benutzer ermöglicht, innerhalb kürzester Zeit eine benötigte Produktkonfiguration zu erstellen.
- (v) Der letzte Beitrag dieser Arbeit sind Empfehlungen für den Aufbau und die Einführung von merkmalsbasierter Variabilitätsmodellierung in gewachsenen Umfeldern. Dabei werden ein gestuftes Einführungsmodell und mögliche Migrationsvorgehen für die meistverbreiteten Variabilitätsdokumentationsansätze in der Praxis vorgeschlagen.

Die erarbeiteten Ansätze können sowohl einzeln als auch kombiniert eingesetzt werden, um die effiziente Wiederverwendung von Anforderungen zu ermöglichen und die Qualitätssteigerung von Anforderungsdokumenten zu unterstützen.

Kapitel 1

Einführung

Dieses Kapitel gibt eine Einführung in das Thema Variantenmanagement in Anforderungsdokumenten. Nach der Motivation werden die grundlegenden Begriffe aus den Bereichen des Anforderungs- und Variantenmanagements definiert, die in dieser Arbeit verwendet werden. Im zweiten Teil dieses Kapitels werden die existierenden Problemfelder identifiziert und diskutiert. Im Anschluss daran werden die Ziele dieser Dissertation sowie ihre Beiträge für die Wissenschaft und Praxis vorgestellt. Abschließend wird der Aufbau dieser Arbeit präsentiert.

1.1 Motivation

In der modernen Produktentwicklung spielen Produktvarianten eine zentrale Rolle. Aus einer Produktidee werden in der Regel eine ganze Reihe ähnlicher Produkte, also eine so genannte Produktlinie¹, entwickelt. Die Ursachen für Produktvarianten sind dabei mannigfaltig [PoBL05]. Der Wunsch, mit einem im Grunde identischen Produkt unterschiedliche Märkte zu bedienen, kann eine Quelle für Variabilität sein. Teilweise bedingen unterschiedliche Märkte unterschiedliche Ausprägungen desselben Produktes (z.B. durch unterschiedliche Gesetze und Normen). Aber auch unterschiedliche Vorlieben in verschiedenen Märkten bestimmen unterschiedliche Produktausprägungen. Zwei Beispiele sollen dies kurz verdeutlichen:

- *Die Gesetzgebung in den Golfstaaten verlangt einen akustischen Warnhinweis, sobald ein Fahrzeug schneller als 120 km/h fährt. In anderen Ländern ist ein solcher Warnton hingegen nicht akzeptiert bzw. sogar untersagt.*
- *In südamerikanischen Ländern werden typischerweise ausgeprägt maskuline Designvarianten bevorzugt. Ein Kuhfänger an einem Geländewagen würde beispielsweise als*

¹ Die Verwendung der Begriffe Produktfamilie und Produktlinie ist in der Literatur nicht einheitlich. In der vorliegenden Arbeit wird der Begriff aus [CINo01] verwendet, wonach eine Produktlinie eine Menge von Produkten mit ähnlichen Eigenschaften und Merkmalen ist.

obligatorische Ausstattung angesehen werden, während dieses Accessoire bei einem Fahrzeug in Europa nur als optionales Element angeboten werden kann.

Ein weiterer Grund für die Variabilität ist der Wunsch, verschiedene Kundengruppen mit ähnlichen, aber kundengruppen-spezifisch angepassten Produkten zu bedienen. *Ein bekanntes Beispiel aus dem Hause Mercedes-Benz wären die E-Klasse Limousine, die E-Klasse T-Modell (Kombi) und die E-Klasse Coupé.* Wie bereits die Namensgebung zeigt, handelt es sich um relativ ähnliche Fahrzeuge, die auch viele Gemeinsamkeiten haben, aber durch unterschiedlichen Karosserietypen jeweils unterschiedliche Käufergruppen ansprechen.

Die Reduktion der Entwicklungszeit durch Wiederverwendung ist der Leitgedanke der letzten Jahrzehnte in der Industrie, besonders in der Automobilindustrie. Dem entgegen erfordert die große Variantenanzahl längere Entwicklungszeiten, die aber aufgrund der aktuellen Wettbewerbssituation nicht zur Verfügung stehen. Um dieses Paradoxon zu lösen wird aus einer Produktidee oft eine Produktlinie, eine ganze Reihe ähnlicher Produkte, durch gezielte, systematische Wiederverwendung von Teil-Komponenten entwickelt. Bei der Entwicklung einer Produktlinie spielt das Variantenmanagement, also die Identifikation und Verwaltung von Produktvarianten eine zentrale Rolle.

Mit der steigenden Komplexität der Produkte sind auch die Anforderungen an die einzelnen Komponenten und Systeme immer vielfältiger geworden. Reichte früher eine Zeichnung und eine Stückliste für die Anforderungsdokumentation einer Komponente, so ist heute ein umfassendes Anforderungsdokument erforderlich. Die Anforderungsdokumente einer ganzen Produktlinie bieten ein großes Wiederverwendungspotential für die Anforderungsdokumente des einzelnen Produkts [Houd03]. Der Schlüssel für eine effiziente Wiederverwendung ist auch hier das Variantenmanagement [ChAt07]. Ein effektives Variantenmanagement für Anforderungsdokumente muss eine Methode zur Verfügung stellen, die die Identifikation, Dokumentation und Verwaltung von Anforderungsvarianten mit geringem Zeitaufwand sowie tool-gestützte Validierungen ermöglicht.

Bereits seit den frühen 90er Jahren gibt es Ansätze zur Beschreibung von Variabilität. Exemplarisch sei hier FODA (Feature-Oriented Domain Analysis) erwähnt [KCHN90]. Der praktische Einsatz, vor allem im Produktmanagement und der Software-Realisierung, zeigt deren grundsätzliche Industrietauglichkeit [MaLi04a].

Im Bereich des Anforderungsmanagements in großen Organisationen sucht man hingegen vergeblich nach dem Einsatz eines systematischen Variantenmanagements [WeRe09]. Die Gründe hierfür liegen in den folgenden Bereichen:

- Variantenmanagement wird meist als separate Aktivität betrieben. Die Verbindung mit dem Anforderungsmanagement findet typischerweise nicht oder nur auf oberer Ebene (z.B. Systemebene) statt. Die Nutzung von Merkmalen innerhalb von Anforderungsspezifikationen ist die Ausnahme. Es gibt mittlerweile zwar erste Werkzeuge, die auch die Übernahme von Merkmalen in Anforderungsmanagement-Tools ermöglichen, deren Einsatz ist jedoch nicht unproblematisch.
- Die bekannten und erprobten Ansätze des Variantenmanagements basieren auf einem zentralen Variabilitätsmodell. In großen und komplexen Organisationen sind solche zentralen Variabilitätsmodelle praktisch unbrauchbar und damit nicht realistisch.
- Die existierenden Ansätze zum Variantenmanagement beschäftigen sich nur sehr unzureichend mit der Frage der Veränderung in dem Entwicklungsprozess und den daraus resultierenden Konsequenzen auf Entwicklungsartefakte. Typischerweise muss das gesamte Variabilitätsmodell einer Produktlinie einem Versionsmanagement unterliegen, d.h. es gibt mehrere aufeinanderfolgende Stände des Variabilitätsmodells, die jeweils freigegeben werden müssen. Diese „Big-Bang“-Herangehensweise ist je-

doch nicht praxistauglich, sobald die Variabilitätsmodelle eine gewisse Größe und Komplexität überschreiten.

1.2 Begriffe

In diesem Abschnitt werden die grundlegenden Begriffe in den Bereichen des Anforderungsmanagements, der Produktlinienentwicklung und des Variantenmanagements definiert, die in dieser Arbeit verwendet werden. Abbildung 1 gibt einen Überblick über die Begriffe, die in diesem Kapitel definiert werden und stellt die Zusammenhänge zwischen einzelnen Begriffen dar.

1.2.1 Anforderungsmanagement

Das Anforderungsmanagement (engl. Requirements Management) ist ein Teil des Produktentwicklungsprozesses und befasst sich mit der Ermittlung, Dokumentation, Prüfung und dem Verwalten von Anforderungen. Durch vielfache industrielle Erfahrung weiß man, dass das Anforderungsmanagement einen entscheidenden Einfluss auf den Erfolg der Entwicklungsprojekte hat [Stan09].

Der Standard IEEE 610.12-1990 definiert den Begriff *Anforderung* wie folgt [IEEE610]:

Definition 1-1: *Anforderung (engl. Requirement)*

- (1) *A condition or capability needed by a user to solve a problem or achieve an objective.*
- (2) *A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.*
- (3) *A documented representation of a condition or capability as in (1) or (2).*

Nach dieser Definition definieren die Anforderungen sowohl die Wünsche und Ziele von Benutzern als auch die Bedingungen und Eigenschaften des zu entwickelnden Systems oder Systemkomponenten (vgl. [Poh07]).

In der Literatur werden drei Basis-Klassen von Anforderungen definiert: funktionale Anforderungen, Qualitätsanforderungen und Rahmenbedingungen. Diese drei Basis-Klassen können verfeinert und bei Bedarf an die Bedürfnisse einer bestimmten Domäne angepasst werden [Pohl07]. Außerdem können Anforderungen nach [Part10] anhand mehrerer Eigenschaften von anderen Anforderungen unterschieden werden (z.B. Detaillierungsgrad, Darstellungsform, Verbindlichkeit, Priorisierung).

Die einzelnen Anforderungen an die neuen Produkte werden in den Anforderungsdokumenten dokumentiert und verwaltet (in Anlehnung an [Pohl07]).

Definition 1-2: *Anforderungsdokument (Synonym: Anforderungsartefakt)*

In einem Anforderungsdokument werden alle Anforderungen an ein zu entwickelndes Produkt dokumentiert.

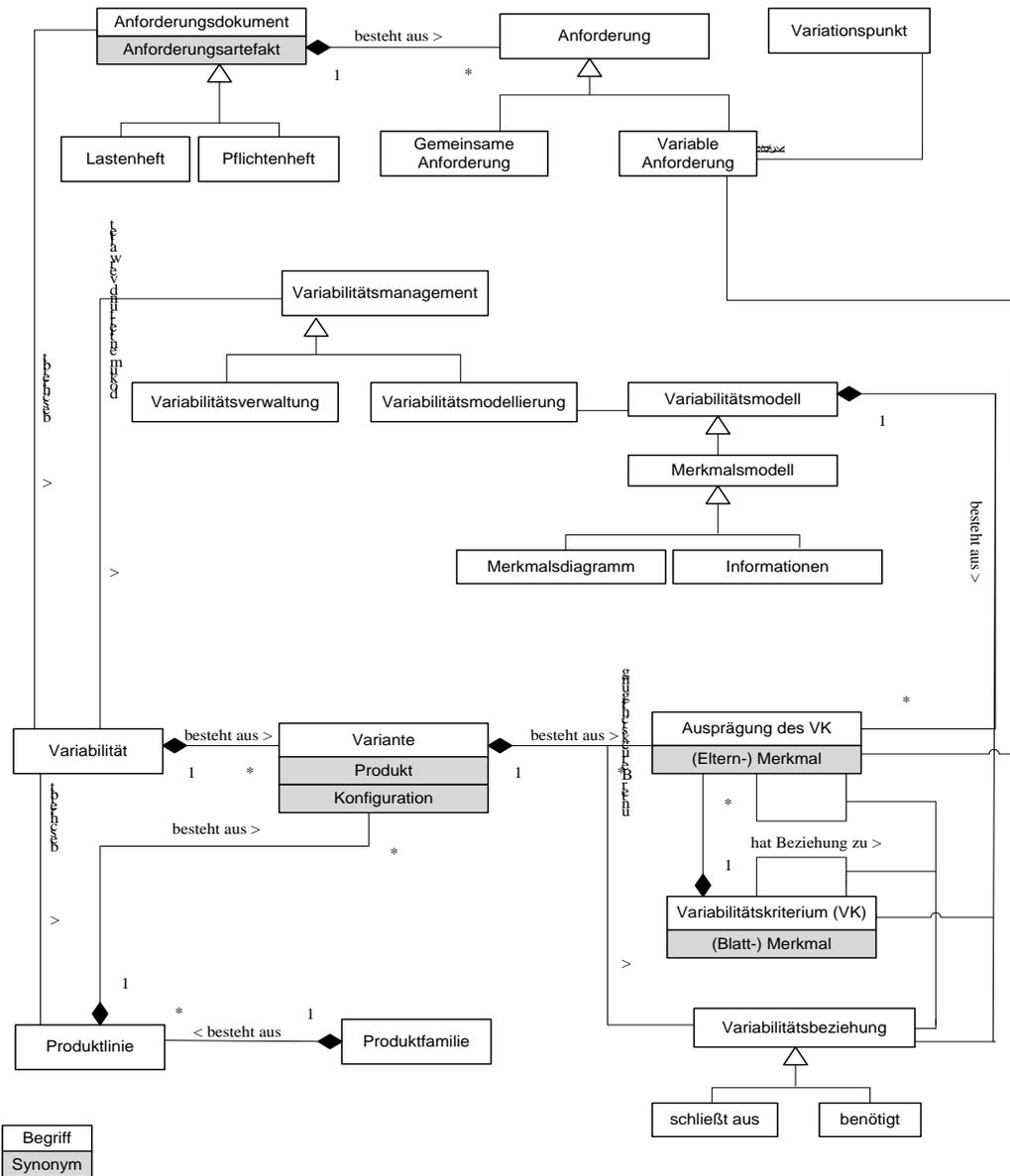


Abbildung 1: Begriffslandschaft

1.2.2 Produktlinienentwicklung

In der Motivation wurden schon die Begriffe Produktlinie und Produktfamilie verwendet. In diesem Abschnitt werden diese Begriffe definiert. Die Verwendung der Begriffe Produktfamilie und Produktlinie ist in der Literatur nicht einheitlich. In dieser Arbeit wird eine Definition des Begriffs Produktlinie benutzt, die auf den Definitionen von [CINo01] und [Reis08] basiert:

Definition 1-3: Produktlinie

Eine Produktlinie (PL) umfasst mehrere individuelle Produkte, die auf Basis einer gemeinsamen Produktlinienplattform in einer vordefinierten und dokumentierten Weise erstellt werden. Alle Produkte einer Produktlinie besitzen eine Reihe von Gemeinsamkeiten aber auch eine Reihe von Unterschieden.

Definition 1-4: Produkt (vgl. [Zele06])

Ein Produkt ist das Ergebnis eines vom Menschen bewirkten Transformationsprozesses, in dem Produktionsfaktoren unter Berücksichtigung von Wissen und unter Beachtung soziokultureller Nebenbedingungen in einen Output transformiert werden.

In Rahmen dieser Arbeit werden sowohl mechanische und elektronische Teile (engl. Hardware), als auch Software-Anteile eines Produktes als ein Produkt gekennzeichnet. Der Begriff Produktvariante wird in dieser Arbeit als Synonym zu dem Begriff Produkt verwendet.

Der Begriff *Produktfamilie* wird oft als Synonym zu Produktlinie verwendet, aber auch (wie in dieser Arbeit) als eine sehr komplexe, ggf. aus mehreren Produktlinien zusammengesetzte Produktlinie definiert.

Definition 1-5: Produktfamilie

Eine Produktfamilie umfasst mehrere unabhängige Produktlinien.

1.2.3 Variantenmanagement

Die variablen Eigenschaften eines Produktes werden im Rahmen dieser Arbeit die *Variabilitätskriterien* genannt.

Definition 1-6: Variabilitätskriterium

Ein Variabilitätskriterium ist eine Eigenschaft eines Bezugsgegenstandes, die variieren kann. Ein Variabilitätskriterium kann mehrere (mindestens aber zwei) zulässige Ausprägungen haben.

Beispielsweise hat die Komponente Sonnenblende eines Fahrzeugs folgende zwei Variabilitätskriterien:

- *Das erste Variabilitätskriterium ist Material und beschreibt aus welchem Material der Bezug der Komponente hergestellt werden kann. Dieses Variabilitätskriterium hat drei Ausprägungen: Stoff, PVC und Leder.*
- *Das zweite Variabilitätskriterium ist Beleuchtung und beschreibt, ob die Sonnenblende eine integrierte Beleuchtung besitzt. Dieses Variabilitätskriterium hat zwei Ausprägungen: mit Beleuchtung und ohne Beleuchtung.*

Die einzelnen Variabilitätskriterien und deren Ausprägungen können Beziehungen untereinander haben.

Definition 1-7: Variabilitätsbeziehungen

Die Variabilitätsbeziehungen beschreiben die Abhängigkeiten zwischen Variabilitätskriterien und/oder zwischen Ausprägungen der Variabilitätskriterien.

Die Beziehungen zwischen Variabilitätskriterien können sowohl technische als auch betriebswirtschaftliche Gründe haben. Ein Beispiel für die betriebswirtschaftliche Beziehung wäre, dass alle Sonnenblenden aus Leder auch eine Beleuchtung haben müssen. Für diese Beziehung existieren keine anderen Gründe außer einer Marketingentscheidung.

Basierend auf den Definitionen von Variabilitätskriterium (1-6) und Variabilitätsbeziehung (1-7) kann der Begriff *Variante* definiert werden:

Definition 1-8: Variante

Eine Variante ist ein Produkt, dessen Eigenschaften eine gültige Kombination der Ausprägungen der Variabilitätskriterien der Produktlinie unter Berücksichtigung der Beziehungen zwischen Eigenschaften darstellen. Dabei muss sich dieses Produkt von allen anderen Produktvarianten in mindestens einer Ausprägung des Variabilitätskriteriums unterscheiden.

Der Begriff *Konfiguration* wird in dieser Arbeit als Synonym zu den Begriffen Produkt und Variante verwendet. Am Beispiel der Komponente Sonnenblende existieren fünf gültige Varianten dieser Komponente:

- (V1) Sonnenblende aus Leder mit Beleuchtung;
- (V2) Sonnenblende aus Stoff mit Beleuchtung;
- (V3) Sonnenblende aus Stoff ohne Beleuchtung;
- (V4) Sonnenblende aus PVC mit Beleuchtung;
- (V5) Sonnenblende aus PVC ohne Beleuchtung.

Alle Produkte einer Produktlinie können mit dem Begriff *Variabilität* beschrieben werden (vgl. [BKPS04]):

Definition 1-9: Variabilität

Variabilität ist die Menge aller zulässigen Varianten einer Produktlinie.

Somit ist die Variabilität der Komponente Sonnenblende durch alle fünf Varianten (V1) – (V5) repräsentiert. Die Variabilität kann implizit (z.B. in der textuellen Beschreibung einer Komponente) oder explizit (z.B. durch Erstellen eines Variabilitätsmodells) dokumentiert werden. Die Vorteile der expliziten Dokumentation sind (vgl. [BKPS04]):

- die Unterstützung der Kommunikation zwischen verschiedenen Stakeholdern;
- die Entscheidungsunterstützung durch Stärkung der Bewusstheit der Variabilitätsexistenz sowie die Möglichkeiten die Gründe der Variabilität zu dokumentieren;
- die Unterstützung der Nachvollziehbarkeit zwischen Anforderungen und Varianten.

Die explizite Dokumentation von variablen Eigenschaften wird *Variabilitätsmodellierung* genannt.

Definition 1-10: Variabilitätsmodellierung

Die Variabilitätsmodellierung beschreibt die Variabilität eines Bezugsgegenstandes (z.B. eine Produktlinie).

Um die Gemeinsamkeiten und Unterschiede zwischen einzelnen Produkten einer Produktlinie verwalten zu können, wird *Variantenmanagement* (vgl. [Fran02]) eingesetzt.

Definition 1-11: Variantenmanagement

Das Variantenmanagement befasst sich sowohl mit der systematischen Ermittlung, Dokumentation und Bewertung von Varianten als auch mit der Erstellung von einzelnen Konfigurationen.

Zu den Aufgaben des Variantenmanagements können auch die Ermittlung der Kosten oder die Reduktion der Variantenzahl gehören. In Rahmen dieser Arbeit werden die Begriffe Varianten- und Variabilitätsmanagement als Synonyme verwendet.

1.3 Problembeschreibung

Im Bereich des Anforderungsmanagements fehlt oft der systematische Einsatz des Variantenmanagements. Die Notwendigkeit des systematischen Variantenmanagements in den Anforderungsdokumenten wurde schon seit mehreren Jahren von der Forschung und in der Industrie erkannt [HeHo03, BKPS04, Stre04, Tava06].

Es existiert eine Reihe von Forschungsarbeiten in diesem Bereich [Stre04, BLPW04, Tava06, Reis08]. Alle dieser Arbeiten haben unterschiedliche Lösungen für die Probleme, welche auch in dieser Arbeit betrachtet werden. Allerdings haben alle davor genannten Arbeiten den Neuanfang bei der Anforderungs- und Variabilitätsdokumentation als Ausgangspunkt für die Lösung der existierenden Probleme. Der Realität in der Praxis zeigt jedoch, dass die Produktlinien oft nicht geplant werden, sondern zuerst ein Produkt spezifiziert und erstellt wird, und erst danach die möglichen Varianten des Produktes abgeleitet werden. Hinzu kommt, dass ein Neuanfang für den Anforderungsprozess während der laufenden Entwicklung nicht möglich oder aus wirtschaftlichen Gründen nicht erwünscht ist. Dies führt dazu, dass der Variantenmanagementansatz auf bereits existierende Anforderungsdokumente anwendbar sein sollte.

Mehrere Ansätze für das Variantenmanagement basieren auf den Gedanken der Merkmalsmodellierung [ThHe02, BLPW04, Reis08]. Ein Merkmal ist nach Kang definiert als „*die entscheidende für den Benutzer sichtbare Leistung eines Systems*“ [KCHN90]. Im Rahmen dieser Arbeit wird folgende Definition (in Anlehnung an [Reis08]) benutzt.

Definition 1-12: Merkmal (engl. Feature)

Ein Merkmal ist eine Eigenschaft im weitesten Sinne, die die Produkte einer Produktlinie aufweisen können oder nicht.

Der Grund für die Verbreitung der Merkmalsmodellierung liegt in der einfache Notation und der intuitiven Vorgehensweise bei der Trennung von variablen und gemeinsamen Eigenschaften.

Um eine Variabilitätsmodellierungssprache für die Anforderungsdokumente auszuwählen müssen folgende zwei Aspekte berücksichtigt werden:

- Welchen Formalisierungsgrad hat die Anforderungsdokumentationssprache?
- In welcher Modellierungssprache ist die Variabilität in den Implementierungs- und Testartefakten dokumentiert?

Die meisten Anforderungsdokumente in der Industrie sind in natürlicher Sprache dokumentiert [NeLa03, Houd10, LeOt10]. Eine Möglichkeit die Variabilität der Anforderungsdokumente zu dokumentieren ist die Verwendung von Merkmalsmodellen. Die leicht erlernbare Notation der Merkmalsmodelle ermöglicht die schnelle Einführung und die effiziente Anwendung in natürlichsprachlichen Anforderungsdokumenten im Gegensatz zur Dokumentation von Variabilität mit Erweiterungen der UML-Sprache.

Trotz der Vorteile, ist die Nutzung von Merkmalsmodellierung in Anforderungsdokumenten in der Industrie eher eine Ausnahme. Die Gründe dafür bilden mehrere offene Problemfelder, welche die Nutzung von Merkmalsmodellen in Anforderungsdokumenten für die Industrie ineffizient bzw. unmöglich machen [Tava06, Reis08, RaGD10, ChBa10].

(P1) *Identifikation der Merkmale in den bestehenden Anforderungsdokumenten*

In den meisten Fällen müssen die Merkmalsmodelle auf Basis von existierenden Anforderungsdokumenten erstellt werden. Der Prozess der Identifikation von Merkmalsmodellelementen basiert aktuell auf manuellen Reviews [BoHo11]. Die Anzahl von Anforderungen in einem Anforderungsdokument in der Praxis liegt im Bereich von mehreren tausenden Anforderungen. Ein Anforderungsdokument für ein Motorsteuergerät bei BMW besteht zum Beispiel aus ca. 2500 Anforderungen [WoBr09]. Unter der Berücksichtigung des Umfangs der industriellen Anforderungsdokumente ist der Aufwand für die Identifikation der Merkmale in den bestehenden Anforderungsdokumenten mit Hilfe eines manuellen Reviews sehr groß [BoHo11].

(P2) *Anzahl der Variabilitätsmodelle*

Die bekannten und erprobten Ansätze des Variantenmanagements basieren auf einem zentralen Variabilitätsmodell. In großen und komplexen Organisationen sind solche zentralen Variabilitätsmodelle praktisch betrachtet nicht realistisch, sowohl hinsichtlich der Erstellung als auch der Pflege. Zum Beispiel, die Komponente *Player*, welche in einem Fahrzeug eingebaut ist, hat ca. 70 Merkmale. Wenn man beachtet, dass ein Fahrzeug aus mehreren tausenden Komponenten gebaut wird, wird die Dimension eines zentralen Variabilitätsmodells für ein hochkomplexes Produkt schnell klar. Als Alternative können mehrere unabhängige Variabilitätsmodelle erstellt werden. Diese Lösung führt oft zu Redundanzen und Konsolidierungsproblemen in der Phase der Konfigurationserstellung.

(P3) *Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten*

Die existierenden Ansätze für die Dokumentation der Variabilitätsinformationen in Anforderungsdokumenten haben entweder mehrere Defizite oder sind einfach in der Praxis nicht anwendbar (siehe Kapitel 3).

Unter Berücksichtigung des Umfangs von industriellen Anforderungsdokumenten (vgl. P1) ist leicht nachzuvollziehen, dass der zeitliche Aufwand für die Zuordnung von Anforderungen zu Merkmalen sehr hoch ist. Dazu kommt oft noch das Problem, dass die Merkmalsmodelle nicht durch Spezifikationsautoren erstellt werden und somit der zusätzliche Aufwand für die Einarbeitung des Variantenmanagers dazugezählt werden muss.

Ein wichtiger Unteraspekt ist die fehlende Werkzeugunterstützung für die Übernahme von Merkmalen in die Anforderungsmanagement-Tools. Es existieren zwar erste Werkzeuge, welche allerdings nicht alle Grundbedürfnisse der Industrie erfüllen. Daher ist ihr Einsatz nicht unproblematisch. Die aktuell verfügbaren Variantenmanagement-Werkzeuge werden in Kapitel 9 verglichen.

(P4) *Konfigurationserstellung*

Die Erstellung von Konfigurationen ist aus mehreren Gründen eine Herausforderung. Zum Beispiel ermöglichen die ersten Werkzeuge zur Merkmalsmodellierung zwar die Konfigurationserstellung, jedoch bieten sie keine Unterstützung für den Konfigurationserstellungsprozess, um den Benutzer mit unterstützenden Informationen zu begleiten und den zeitlichen Aufwand zu reduzieren. Somit ist der Konfigurationsprozess sehr zeitaufwändig.

(P5) *Einführungsstrategie*

Ein weiteres Problemfeld sind fehlende Empfehlungen und Beschreibungen, wie ein merkmalsbasiertes Variantenmanagement (MBVM) für Anforderungsartefakte in der Praxis eingeführt werden kann oder wie man von den bestehenden Ansätzen auf MBVM migrieren könnte.

(P6) *Evolution von Variabilitätsmodellen*

Die existierenden Ansätze zum Variantenmanagement beschäftigen sich nur sehr unzureichend mit der Frage der zeitlichen Veränderung der Variabilitätsmodelle und den daraus resultierenden Konsequenzen für den Entwicklungsprozess.

In dieser Arbeit werden die Problemfelder (P1) - (P5) angegangen und mögliche Lösungen vorgeschlagen. Das Problemfeld (P6) wurde im Rahmen dieser Arbeit nicht betrachtet, weil mögliche Lösungen für das Evolutionsproblem sehr umfangreich sind und somit den Rahmen der vorliegenden Dissertation sprengen würden. Als Übergangslösung kann die Versionierung von Variabilitätsmodellen verwendet werden.

1.4 Ziel und Beiträge der Arbeit

Basierend auf den identifizierten Bedürfnissen der Industrie und existierenden Problemfeldern wurde folgendes Ziel für diese Arbeit festgelegt:

Die Erarbeitung von methodischen Adaptionen, welche die Anwendung von merkmalsbasiertem Variantenmanagement in Anforderungsdokumenten bei hoch-komplexer und variantenreicher Produktentwicklung praxistauglich machen.

Die zentralen Beiträge diese Arbeit sind:

(B1) *Ein konfigurationsgekoppelter dezentraler Variabilitätsmodellierungsansatz zur Lösung des Komplexitätsproblems.*

Die bekannten und erprobten Ansätze des Variantenmanagements basieren auf einem zentralen Variabilitätsmodell. In großen und komplexen Organisationen sind solche zentralen Variabilitätsmodelle nicht realistisch. Bei dem konfigurationsgekoppelten dezentralen Variabilitätsmodellierungsansatz für Anforderungsdokumente wird vorgeschlagen, die Variabilität auf den unterschiedlichen Abstraktionsebenen von Anforderungen in dezentralen, über Konfigurationen gekoppelten, Variabilitätsmodellen, zu dokumentieren. Zusätzlich zu den existierenden Abstraktionsebenen, die im Abschnitt 2.1 vorgestellt werden, wird eine neue Ebene vorgeschlagen, welche die globalen Produkteigenschaften der Produktfamilie dokumentiert (siehe Kapitel 6). Um die Konsistenz der Variabilitätsmodelle innerhalb des Variabilitätsraums der Produktfamilie zu garantieren, werden Konsistenzregeln definiert.

(B2) *Ein semi-automatischer Ansatz für die Identifikation der Merkmale (MIA) in den bestehenden Anforderungsdokumenten.*

MIA basiert auf den Methoden der lexikalischen Analyse und generiert aus einem oder mehreren Anforderungsdokumenten die Kandidaten für die Variabilitätskriterien und deren Ausprägungen. Der Anwender bekommt die automatisierte Unterstützung und kann die automatisch generierte Merkmalskandidatenliste benutzen, um ein Variabilitätsmodell zu erstellen.

(B3) *Eine Dokumentationsform für die Variabilitätsinformationen in Anforderungsdokumenten.*

In Rahmen dieser Arbeit wird ein praktischer Requirements-to-Feature-Ansatz (R2F-Ansatz) vorgeschlagen. Dieser Ansatz basiert auf Ideen der Merkmalsmodellierung. Die variablen Anforderungen werden zu entsprechenden Ausprägungen der Variabilitätskriterien im Anforderungsdokument zugeordnet. Die Verwaltung von Variabilitätsinformationen sowie die Erstellung der Varianten werden in einem orthogonalen Variabilitätsmodell realisiert. Der R2F-Ansatz ermöglicht mit reduziertem Arbeitsaufwand die Unterstützung der Anforderungswiederverwendung.

(B4) *Ein Ansatz zur Unterstützung des Konfigurationsprozesses.*

Der Prozess der Konfigurationserstellung ist sowohl bei zentralen als auch bei dezentralen Variabilitätsmodellen sehr aufwändig. Bei dem zentralen Variabilitätsmodell ist dieser Prozess wegen der Menge an Merkmalen und Beziehungen sehr aufwändig. Bei den dezentralen Variabilitätsmodellen sind die Anzahl der Variabilitätsmodelle und die Beziehungen zwischen unterschiedlichen Variabilitätsmodellen die Ursache für den Aufwand. Für die Unterstützung des Konfigurationsprozesses bietet diese Arbeit einen Ansatz, der den Benutzer bei der Konfigurationserstellung durch die optimalen Fragenfolgen unterstützt.

(B5) *Empfehlungen für den Aufbau und die Einführung von merkmalsbasiertem Variantenmanagement in gewachsenen industriellen Umfeldern.*

In dieser Arbeit wird ein Kriterienkatalog für die Auswahl von benötigter Variantenmanagementunterstützung, ein gestuftes Einführungsmodell, sowie mögliche Migrationsvorgehen von existierenden Variabilitätsdokumentationsansätzen in der Praxis vorgeschlagen.

Die zentralen Beiträge (B1) - (B5) dieser Arbeit bieten jeweils Lösungen für die zuvor identifizierten Problemfelder.

Die erarbeiteten Beiträge ermöglichen die Anwendung von MBVM in Anforderungsdokumenten. Während für die Forschung die ersten zwei Ansätze (Konfigurationsgekoppelte dezentrale Variabilitätsmodellierung und der semi-automatische Ansatz für die Identifikation der Variabilitätsmodellelemente) von Bedeutung sind, spielen für die Praxis alle fünf Beiträge eine große Rolle. Die in dieser Arbeit vorgestellten Ansätze können sowohl zusammen als auch einzeln angewendet werden. Die sinnvollen Kombinationen sowie alternative Lösungen werden am Ende dieser Arbeit in Kapitel 11 dokumentiert und diskutiert.

1.5 Aufbau der Arbeit

Diese Arbeit ist entsprechend dem Ziel und erarbeitenden Lösungsbausteine folgendermaßen aufgebaut.

Kapitel 2 präsentiert den Stand der Forschung zum Thema Variantenmanagement in Anforderungsdokumenten. Es gibt einen Überblick über Anforderungsdokumentarten, Ansätze des Variantenmanagements allgemein und Ansätze für das Variantenmanagement in Anforderungsdokumenten.

Kapitel 3 präsentiert den Stand der Praxis zum Thema Variantenmanagement in Anforderungsdokumenten am Beispiel eines Automobilherstellers. Es gibt einen Überblick über die Landschaft an Anforderungsdokumenten, deren Eigenschaften und Besonderheiten. Nach Vorstellung der relevanten Szenarien für die Anforderungswiederverwendung in Anforderungsdokumenten, werden die Ergebnisse einer Studie zu den aktuellen in der Praxis eingesetzten Ansätzen für das Variantenmanagement in Anforderungsdokumenten präsentiert. Abschließend werden die identifizierten Herausforderungen für das Variantenmanagement in

Anforderungsdokumenten und die daraus abgeleiteten Anforderungen an den Variantenmanagementansatz präsentiert und diskutiert.

Kapitel 4 beschreibt einen semi-automatischen Ansatz für die Identifikation der Merkmale (MIA) in bestehenden Anforderungsdokumenten (B2) auf Basis einer lexikalischen Analyse.

Kapitel 5 führt eine formale Darstellung von Anforderungsdokumenten und den Merkmalsmodellen ein, die in dieser Arbeit verwendet werden. Diese formalen Darstellungen werden in weiteren Kapiteln dieser Arbeit verwendet, um die Ansätze und Prüfmethoden präzise beschreiben zu können.

Kapitel 6 beschreibt einen konfigurationsgekoppelten Variabilitätsmodellierungsansatz zur Lösung des Komplexitätsproblems (B1). Der erarbeitete Ansatz basiert auf den Ideen der dezentralen Variabilitätsmodellierung mit der Kopplung der einzelnen Variabilitätsmodelle über Konfigurationen.

Der Vorschlag für eine Dokumentationsform der Variabilitätsinformationen in den Anforderungsdokumenten (B3) wird im Kapitel 7 festgehalten. In diesem wird der R2F-Ansatz für das Variantenmanagement in Anforderungsdokumenten vorgestellt, welcher im Rahmen dieser Dissertation entwickelt wurde. Der Ansatz basiert auf den Ideen der Merkmalsmodellierung [Kang90] und ermöglicht die effiziente Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten. Außerdem präsentiert dieses Kapitel einen Algorithmus, welcher die Überprüfung der Korrektheit der erstellten Anforderungsdokumente für einzelne Produkte anhand des Anforderungsdokumentes einer Produktlinie ermöglicht.

Kapitel 8 beschreibt wie der Konfigurationserstellungsprozess optimiert werden kann (B4). Dafür wird eine Vorgehensweise zur Berechnung der optimalen Fragefolge zur Auswahl der Merkmale basierend auf den Beziehungen zwischen einzelnen Merkmalen in einem oder mehreren Merkmalsmodellen vorgestellt.

Kapitel 9 stellt einen Leitfaden zur Einführung des merkmalsbasierten Variantenmanagements in Anforderungsdokumenten inklusive Migrationsszenarien von bereits existierenden Ansätzen vor. In diesem Kapitel wird Beitrag B5 präsentiert.

Kapitel 10 präsentiert die Ergebnisse der Evaluation der Lösungsbausteine in zwei Projekten: in einer empirischen Studie an der Universität Ulm und in einem Projekt bei der Daimler AG.

Kapitel 11 bietet die Zusammenfassung der erarbeiteten Ansätze und skizziert die möglichen Anwendungsszenarien für diese Ansätze sowie eine Diskussion über alternative Lösungsmöglichkeiten. Am Ende dieses Kapitel wird ein Ausblick über potenzielle Erweiterungen dieser Arbeit gegeben.

Kapitel 2

Stand der Forschung

Das vorliegende Kapitel präsentiert den Stand der Forschung zum Thema Variantenmanagement in Anforderungsdokumenten. Abschnitt 2.1 gibt einen Überblick über Anforderungsdokumentarten, deren mögliche Strukturen und Qualitätskriterien sowie einen Überblick über Werkzeugunterstützung im Bereich der Anforderungsverwaltung. Abschnitt 2.2 präsentiert die Ansätze des Variantenmanagements im Allgemeinen und Abschnitt 2.3 zeigt Ansätze für Variantenmanagement in Anforderungsdokumenten im Spezifischen. Am Ende des Kapitels (Abschnitt 2.4) wird das Vorhaben dieser Arbeit von bereits existierenden Arbeiten auf dem Gebiet Variantenmanagement in Anforderungsdokumenten abgegrenzt.

2.1 Anforderungsdokumente

Die einzelnen Anforderungen an neue Produkte werden in Anforderungsdokumenten dokumentiert und verwaltet. Im deutschen Sprachraum haben sich zwei Typen von Anforderungsdokumenten etabliert: das Lastenheft und das Pflichtenheft. Im angelsächsischen Sprachraum ist diese Differenzierung nicht üblich [Pohl07].

Nach DIN 69905 enthält ein Lastenheft die Gesamtheit der Forderungen an die Lieferungen und Leistungen eines Auftragnehmers [DIN69905]. Die Erstellung eines Lastenhefts liegt im Regelfall in der Verantwortung des Auftraggebers. In dieser Arbeit wird folgende Definition (in Anlehnung an [Pohl07]) verwendet:

Definition 2-1: *Lastenheft*

Ein Lastenheft enthält eine Definition der Produktvariante, eine Beschreibung der wesentlichen Produktziele und benennt wichtige Kontextaspekte.

In dieser Arbeit wird zusätzlich der Begriff *Spezifikation* als Synonym zu dem Begriff *Lastenheft* verwendet.

Ein Pflichtenheft (nach DIN 69905) enthält die vom Auftragnehmer erarbeiteten Realisierungsvorgaben und beschreibt die Umsetzung des vom Auftraggeber vorgegebenen Lastenhefts [DIN69905]. Die Erstellung eines Pflichtenhefts liegt in der Verantwortung des Auftragnehmers. In dieser Arbeit wird folgende Definition (in Anlehnung an [Pohl07]) verwendet:

Definition 2-2: Pflichtenheft

Ein Pflichtenheft detailliert die im Lastenheft beschriebene Vision und die Systemziele sowie ggf. im Lastenheft definierte Rahmenbedingungen im Hinblick auf die angestrebte technische Umsetzung.

Die Anforderungen können in den Anforderungsdokumenten auf unterschiedliche Weise dokumentiert werden. Die möglichen Darstellungsformen sind: (1) textuelle Dokumentation in natürlicher Sprache, (2) Verhaltensmodelle, (3) Interaktionsmodelle, (4) formale Modelle und (5) unterschiedliche Kombinationen aus (1) - (5).

Die natürlichsprachliche Dokumentation ist die meist verbreitete Dokumentationsform für Anforderungen (vgl. [MiFN04, NeLa03]). Die Dokumentation in natürlicher Sprache hat sowohl Vorteile als auch Nachteile. Die wesentlichen Vorteile der natürlichsprachlichen Dokumentation [Pohl07, Part10] sind:

- *Universalität*: natürliche Sprache kann in beliebigen Domänen angewendet werden.
- *Flexibilität*: natürliche Sprache ermöglicht beliebige Abstraktionen und Detaillierungen.
- *Einfache Anwendung*: natürliche Sprache kann ohne Schulungen oder spezielle Werkzeuge angewendet werden.
- *Verbreitung*: natürliche Sprache ist das klassische Kommunikationsmittel zwischen Menschen.
- *Verständlichkeit*: natürliche Sprache kann ohne Schulungen von allen Menschen, welche die gleiche Sprache sprechen, verstanden werden.

Zu den Nachteilen der natürlichsprachlichen Dokumentation gehören die Mehrdeutigkeit und die Vagheit von Begriffen.

Um die Nachteile der natürlichsprachlichen Dokumentationsform zu reduzieren, können folgende Gegenmaßnahmen angewendet werden: die Verwendung von Glossaren, Normsprachen und syntaktische Anforderungsmuster für die Anforderungsformulierung sowie die Strukturierung von Anforderungsdokumenten und der kontrollierte Umgang mit Redundanz [Pohl07].

2.1.1 Qualität von Anforderungsdokumenten

Die Qualität der Anforderungsdokumente kann anhand der Erfüllung von Qualitätskriterien gemessen werden. Dabei wird die Qualität eines Anforderungsdokumentes durch die Qualität der einzelnen Anforderungen sowie durch die Qualität des gesamten Anforderungsdokumentes bestimmt. Im IEEE-Standard 830 werden die Qualitätskriterien für Anforderungen festgelegt [IEEE830]. Die wichtigsten dieser Qualitätskriterien sind [Pohl07, Part10]:

- *Eindeutigkeit*: Eine Anforderung ist eindeutig, wenn sie so formuliert ist, dass nur eine Interpretation möglich ist.
- *Konsistenz*: Eine Anforderung ist konsistent, wenn darin gemachte Aussagen nicht im Widerspruch miteinander stehen.
- *Korrektheit*: Eine Anforderung ist korrekt, wenn die relevanten Stakeholder die Korrektheit der vorliegenden Anforderung bestätigen und die Anforderung vollständig umgesetzt werden kann.
- *Nachvollziehbarkeit*: Eine Anforderung ist nachvollziehbar, wenn sowohl die Quelle der Anforderung, die Evolution der Anforderung und die Realisierung der Anforderung nachverfolgbar sind.

- *Vollständigkeit*: Eine Anforderung ist vollständig, wenn die Anforderung gemäß den festgelegten Kriterien dokumentiert ist und keine inhaltlichen Lücken aufweist.
- *Überprüfbarkeit*: Eine Anforderung ist überprüfbar, wenn ihre Erfüllung am erstellten Produkt überprüft werden kann.
- *Bewertung*: Eine Anforderung ist bewertet, wenn die Bedeutung ermittelt und dokumentiert ist.
- *Atomizität*: Eine Anforderung ist atomar, wenn die Anforderung einen einzelnen Sachverhalt beschreibt und nicht in mehrere Anforderungen unterteilt werden kann.
- *Realisierbarkeit*: Eine Anforderung ist realisierbar, wenn diese Anforderung in einem Artefakt (z.B. Software oder Hardware) erfüllt werden kann.

Für die Bestimmung der Qualität eines Anforderungsdokuments wurden in der Literatur verschiedene Qualitätskriterien vorgeschlagen. Die Qualitätskriterien für ein Anforderungsdokument bauen auf den Qualitätskriterien für die einzelnen Anforderungen (z.B. Eindeutigkeit, Nachvollziehbarkeit) auf. Die wichtigsten Kriterien für die Qualität eines Anforderungsdokumentes sind (vgl. [Part10, Pohl07]):

- *Änderbarkeit*: Ein Anforderungsdokument ist änderungsfreundlich, wenn dessen Struktur eine einfache, vollständige und konsistente Modifikation der Anforderungen ermöglicht.
- *Eindeutigkeit*: Ein Anforderungsdokument ist eindeutig, wenn nur eine Interpretation des Anforderungsdokumentes möglich ist.
- *Konsistenz*: Ein Anforderungsdokument ist konsistent, wenn jede einzelne Anforderung konsistent ist und wenn keine Konflikte zwischen Anforderungen existieren.
- *Vollständigkeit*: Ein Anforderungsdokument ist vollständig, wenn jede einzelne Anforderung vollständig spezifiziert ist und wenn alle relevanten Anforderungen im Dokument enthalten sind.
- *Nachvollziehbarkeit*: Ein Anforderungsdokument ist nachvollziehbar, wenn sowohl die Quelle der Anforderungen, die Evolution der Anforderungen und des Anforderungsdokumentes sowie die Implementierung des Anforderungsdokumentes nachvollziehbar sind.
- *Verifizierbarkeit*: Ein Anforderungsdokument ist verifizierbar, wenn überprüft werden kann, ob jede Anforderung darin erfüllt ist.
- *Angemessener Umfang und klare Struktur*.
- *Gemeinsame Zugreifbarkeit*.

2.1.2 Struktur von Anforderungsdokumenten

Für die Strukturierung eines Anforderungsdokumentes existieren keine festen Vorgaben. Es existieren aber mehrere Standards, die als Basis des Anforderungsdokumentes genommen werden können:

- [IEEE830] *Recommended Practice for Software Requirements Specifications* ist ein praxisnaher Standard für die Beschreibung und Definition von Softwareanforderungen.
- [IEEE1362] *Guide for Information Technology - System Definition* setzt sich mit den Anforderungen an den Betrieb eines Software-Systems auseinander.
- [IEEE1233] *Guideline for Developing of System Requirements Specifications* beschreibt die Entwicklung und Spezifikation von Anforderungen und deren Handhabung in der gesamten Produktentwicklung.

- [VDI/VDE3694] *Blatt 1 Vorgehensweise bei der Erstellung von Lasten-/Pflichtenheften* definiert was Lasten- und Pflichtenhefte sind, was sie enthalten sollen und wie sie erstellt werden sollten. Dieser Standard macht eine klare Trennung in der Aufgabenbeschreibung zwischen Benutzer- und Systemsicht.
- [VxT09] beschreibt alle Dokumente, welche im Laufe eines Entwicklungsprojektes entstehen. Für die Lasten- und Pflichtenhefte werden die Strukturen und Inhalte definiert. Dieser Standard macht eine klare Trennung in der Aufgabenbeschreibung zwischen Auftraggeber- und Auftragnehmer-Sicht.

Die IEEE Standards [IEEE830, IEEE1362] bilden zusammen eine solide Basis für die Entwicklung eines Anforderungsdokumentes aus Lieferanten-, Kunden- und Benutzersicht.

Der VDI/VDE Standard 3694 [VDI/VDE3694] wird häufig in Deutschland als Basis für Ausschreibungen benutzt. Für die Entwicklung von Produkten mit Softwareanteil kann dieser Standard mit den software-spezifischen Aspekten von [IEEE1362, IEEE830] angereichert werden.

Basierend auf diesen Standards können für den praktischen Einsatz optimierte Strukturen für die Anforderungsdokumente der Unternehmensverbände bzw. einzelner Unternehmen und Unternehmensbereiche entwickelt werden.

Die Anwendung von vordefinierten Strukturen für die Anforderungsdokumente bringt folgende Vorteile [Pohl07]:

- Wiederverwendung des erprobten Expertenwissens über Anforderungsstrukturierung;
- Referenz für die Vollständigkeit - die vorgegebene Struktur kann als Checkliste dienen;
- Konzentration der Spezifikationsautoren auf Inhalte;
- Einarbeitung für Lieferanten wird einfacher, da die gleichen Informationen immer am gleichen Ort dokumentiert werden.

2.1.3 Abstraktionsebenen

Um die Komplexität der Produktbeschreibung zu reduzieren werden Abstraktionsebenen für die Anforderungen verwendet [HaRH01]. Abstraktion hilft komplexere Probleme zu lösen, indem Details ignoriert bzw. reduziert werden.

Es existieren keine Vorgaben, wie viele und welche Abstraktionsebenen verwendet werden müssen. Die meisten Ansätze berücksichtigen drei oder vier Abstraktionsebenen, die allerdings von Ansatz zu Ansatz unterschiedlich definiert sind.

In dieser Arbeit werden folgende Abstraktionsebenen verwendet:

- Produkt-Ebene: Auf dieser Ebene werden die Anforderungen an das Gesamtprodukt aus Marketing- und Managementsicht beschrieben. Dabei wird dokumentiert aus welchen Systemen (Black-Box-Sicht) das Gesamtprodukt besteht.
- System-Ebene: Auf dieser Ebene werden die funktionalen und nicht-funktionalen Anforderungen an einzelne Systeme dokumentiert. Hier wird auch dokumentiert welche Komponenten die Beiträge für die Realisierung der Systemfunktionen liefern.
- Komponenten-Ebene: Auf dieser Ebene werden die funktionalen und nicht-funktionalen Anforderungen an einzelne Komponenten (Hard- und Software) dokumentiert.

In Tabelle 1 werden die eingeführten Ebenen mit den Abstraktionsebenen aus COSMOD-RE [PoSi07], RAM-Framework [GoWo06] und REMSES [KMPS08] in Beziehung gesetzt.

Tabelle 1: Abstraktionsebenen

	COSMOD-RE	RAM	REMSSES
Produkt-Ebene	System Layer	Product Level Feature Level	Gesamtsystem-Ebene
System-Ebene	Functional Components Layer	System Level	Funktionsgruppen-Ebene
Komponenten-Ebene	Hardware/Software Components Layer Software-Components-Mapping Layer	Component Level	Software/ Hardware-Ebene

In COSMOD-RE [PoSi07] wird eine Hierarchie aus folgenden vier Abstraktionsebenen vorgeschlagen. Dieses Modell kann an eine Domäne bzw. ein Projekt angepasst werden, indem zusätzliche Verfeinerungsebenen eingefügt oder die bestehenden Ebenen reduziert werden.

- Die oberste Ebene (System Layer) definiert die Black-Box-Sicht auf ein System. Auf dieser Ebene werden die Anforderungen an das Gesamtsystem dokumentiert.
- Auf der zweiten Ebene (Functional Components Layer) wird das System in die logischen Komponenten aus funktionaler Sicht aufgeteilt. Hier werden die Anforderungen an diese funktionalen Komponenten-Gruppen dokumentiert.
- Die nächste Ebene (Hardware/Software Components Layer) verfeinert die funktionalen Komponenten in die Hardware- und Software-Komponenten. Hier werden die Anforderungen an die einzelnen Hardware- und Software-Komponenten dokumentiert.
- Die vierte Ebene beschreibt die Zuordnung von Softwarekomponenten zu den Hardwarekomponenten.

Das RAM-Framework [GoWo06] verwendet auch vier Abstraktionsebenen:

- Die oberste Ebene (Product Level) beschreibt die Anforderungen an das Gesamtprodukt aus Marketing- und Managementsicht.
- In der zweiten Ebene (Feature Level) werden die Funktionen des Produktes (Features) aufgelistet. Dabei gehen die Anforderungen nicht ins Detail, wie eine bestimmte Funktionalität realisiert wird.
- Auf der nächsten Ebene werden die funktionalen und nicht-funktionalen Anforderungen an das System dokumentiert.
- Die unterste Ebene (Component-Level) dokumentiert die Anforderungen an die Komponenten, welche die Lösungswege beinhalten.

In Rahmen des Kooperationsprojektes REMSES wurden folgende drei Abstraktionsebenen definiert [KMPS08]:

- Gesamtsystem-Ebene: Auf dieser Ebene wird das System von außen betrachtet. Dabei werden die dem Systemnutzer zur Verfügung gestellten Interaktionen sowie die sichtbaren Funktionalitäten spezifiziert.
- Funktionsgruppen-Ebene: Die auf der Gesamtsystemebene definierten Anforderungen werden verfeinert und einzelnen Funktionsgruppen zugeordnet. Eine Funktionsgruppe ist eine logische Einheit, die eine Menge von Gesamtsystemfunktionen realisiert.
- Software/Hardware-Ebene: Auf dieser Ebene werden die Anforderungen von Funktionsgruppen an einzelne Software- und Hardware-Komponenten dokumentiert.

In der Literatur findet man auch verschiedene Konzepte aus der Industrie, wie die Anforderungen auf verschiedenen Abstraktionsebenen dokumentiert werden können (vgl. [Houd03, FHPR05, Allm07]). Dabei spiegelt das Ebenen-Modell oft die Organisationsstruktur des Unternehmens wieder. Dadurch entsteht eine Herausforderung für Abstraktionsmodelle von Herstellern und Zulieferern: Für die effiziente Zusammenarbeit müssen die Abstraktionsmodelle der Entwicklungspartner kompatibel sein.

2.1.4 Werkzeuge für Anforderungsmanagement

Die Verarbeitung von Anforderungen ist heutzutage ohne Werkzeugunterstützung kaum zu leisten. Selbst für einfache Komponenten (z.B. Sonnenblende im Fahrzeug) beinhaltet das Lastenheft mehrere hundert Anforderungen, die dokumentiert und gepflegt werden müssen. Für die Unterstützung des Anforderungsmanagements existieren auf dem Markt mehrere Werkzeuge. In der iX-Studie [HMRV07] wurden die 14 Werkzeuge (u.a. CaliberRM von Borland, IBM Rational RequisitePro von IBM, In-Step von microTOOL, MKS Requirements von MKS, Polarion von Polarion Software, CARE von SOPHIST, DOORS von IBM), die in Deutschland auf dem Markt verfügbar sind, analysiert und verglichen. Ziel dieser Studie war nicht die Bewertung einzelner Werkzeuge, sondern einen Überblick über die existierenden Werkzeuge zu geben sowie eine Unterstützung bei der Werkzeugauswahl zu bieten.

In den Bereichen Luftfahrt, Automobilindustrie und Bahntechnik dominiert das Werkzeug DOORS von IBM [HMRV07]. In dieser Arbeit wird das Werkzeug DOORS für die Realisierung der erarbeiteten Konzepte verwendet, deswegen wird im nächsten Abschnitt ein kurzer Überblick über DOORS gegeben.

DOORS® (Dynamic Object Oriented Requirements System) ist eine Software für das Anforderungsmanagement von der IBM. DOORS ermöglicht die strukturierte Aufzeichnung und Verwaltung von textuellen Anforderungen (Objekten), die mit Attributen versehen werden können, die ein Anwender frei definieren kann. Für die Verfolgung von Anforderungen während des Projektablaufs können die DOORS-Objekte miteinander verlinkt werden.

2.2 Variantenmanagement in Anforderungsdokumenten

Variantenmanagement ist ein breit diskutiertes Thema im Forschungsgebiet der Produktlinien-Entwicklung sowie der Software Produktlinien-Entwicklung (vgl. [ChAt07, SiDe07]). Das Variantenmanagement stellt eine Grundvoraussetzung für die effektive und effiziente Wiederverwendung von Anforderungen dar [ChAt07].

Die Gründe für die Variabilität in Anforderungsdokumenten sind vielfältig. Auf der einen Seite hängt die Variabilität in Anforderungsdokumenten von der geplanten Produktvariabili-

tät, auf der anderen Seite aber auch von der Variabilität in den Herstellungs-, Montage- und Ausschreibungsprozessen ab. Die Gründe für Produktvariabilität sind zum Beispiel:

- Unterschiedliche Kundenbedürfnisse;
- Unterschiede in der Funktionalität, in Anhängigkeit von der Produktausführung (z.B. Basismodell vs. Luxusmodell);
- Gesetze und Normen unterschiedlicher Länder;
- neue technische Standards und neue Kundenerwartungen [WeRe09].

2.2.1 Variabilitätsdokumentation

Die Dokumentation von Variabilität ist die Kerntätigkeit des Variabilitätsmanagements. Um die Variabilität eindeutig beschreiben zu können, muss zwischen drei Kriterien unterschieden werden:

- die Beschreibung dessen, was änderbar ist und welche Varianten möglich sind,
- die Lokalisierung der Variabilität, also in welchem Teil des Entwicklungsartefaktes die Variabilität auftreten kann,
- die Bedingungen, die an die Auswahl von variablen Elementen geknüpft sind.

In diesem Abschnitt werden zwei Schlüsselaspekte der Variabilitätsdokumentation erläutert:

- die Typisierung der Variabilitätsdokumentationsansätze;
- der Dokumentationsort von Variabilitätsmodellen.

Da sich Variabilität in allgemeiner Form nur sehr schwer beschreiben lässt, gibt es bereits Ansätze, die eine modellierende Darstellung der Variabilität ermöglichen. Das Ziel der Variabilitätsmodellierung ist die Dokumentation der Unterschiede von Produkten.

Bei der Beschreibung der Variabilitätsmodellierung wird oft das Metamodell von Pohl et al. [PoBL05] verwendet, welches in Abbildung 2 dargestellt ist. In dem Metamodell werden die Varianten mit den Variationspunkten in Verbindung gesetzt. Ein Variationspunkt in diesem Metamodell beschreibt *was variiert bzw. die variablen Elemente und Eigenschaften des Produkts* [PoBL05]. Diese Definition des Begriffs *Variationspunkt* entspricht dem Begriff *Variabilitätskriterium* (siehe Definition 1-10). Im weiteren Verlauf dieser Arbeit wird der Begriff *Variationspunkt* mit folgender Bedeutung verwendet:

Definition 2-3: Variationspunkt

Ein Variationspunkt markiert die variablen Stellen im Entwicklungsartefakt ohne genau anzugeben welche Eigenschaft des Entwicklungsartefaktes in dieser Stelle variiert.

Eine Variante in diesem Metamodell beschreibt das Variabilitätsobjekt, d.h. es beschreibt, wie der Variationspunkt variiert bzw. welche möglichen Ausprägungen der Variationspunkt hat.

Zum Beispiel der Variationspunkt „Autofarbe“ hat die Varianten „blau“, „rot“ und „grau“. Ein Variationspunkt kann in dem vorgestellten Metamodell mehrere Varianten haben. Eine Variante kann mehreren Variationspunkten zugeordnet werden. Der Variationspunkt und die Varianten können über eine Variationsbeziehung in Beziehung zueinander gesetzt werden. Der in dem Metamodell verwendete Begriff *Variante* entspricht dem in dieser Arbeit verwendeten Begriff *Ausprägung des Variabilitätskriteriums*.

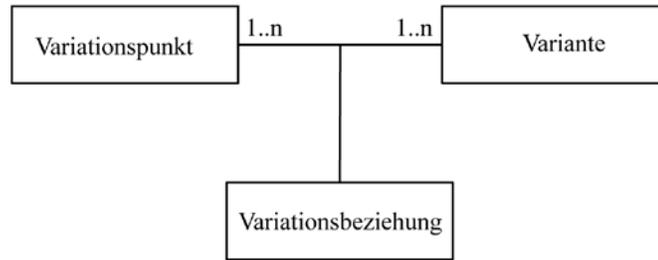


Abbildung 2: Metamodell der Variabilitätsmodellierung nach [PoBL05]

Da das Meta-Modell von [PoBL05] nicht die Bedürfnisse der Variabilitätsbeschreibung im Rahmen dieser Arbeit erfüllen könnte, wurde das Meta-Modell der Variabilitätsmodellierung angepasst. Die Abbildung 3 präsentiert das angepasste Metamodell der Variabilitätsmodellierung, welches im Rahmen dieser Arbeit verwendet wird:

- Zu einem Variabilitätskriterium (z.B. *Farbe des Produktes*) werden zwischen zwei und n Ausprägungen (z.B. *rot, grün, blau*) zugeordnet. Dabei gehört eine Ausprägung des Variabilitätskriteriums immer zu einem Variabilitätskriterium.
- Eine Variante besteht aus mindestens einer Ausprägung eines Variabilitätskriteriums. Eine Ausprägung eines Variabilitätskriteriums muss in mindestens einer Variante und kann in mehreren Varianten vorkommen. *Zum Beispiel ist eine Variante ein roter Kreis und die andere ein grünes Viereck.*
- Zwischen einzelnen Variabilitätskriterien bzw. einzelnen Ausprägungen von unterschiedlichen Variabilitätskriterien können Variabilitätsbeziehungen existieren. *Zum Beispiel ist die Ausprägung des Variabilitätskriteriums Farbe von der Ausprägung des Variabilitätskriteriums Form abhängig, sodass nur die roten Kreise bzw. nur grünen Dreiecke die gültigen Varianten sind.*

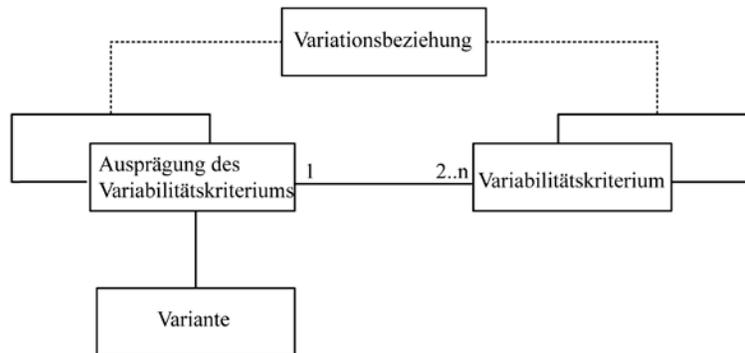


Abbildung 3: Metamodell der Variabilitätsmodellierung

Die Typisierung der Variabilitätsdokumentationsansätze

In der Literatur findet man mehrere Ansätze zur Variabilitätsmodellierung. Die existierenden Variabilitätsmodellierungstechniken wurden mehrmals miteinander verglichen und evaluiert

(vgl. [PoMe08, SiDe06]). Die existierenden Variabilitätsmodellierungstechniken können in drei Gruppen unterteilt werden:

1. *Merkmalsbasierte Modellierung*: Alle Ansätze in dieser Gruppe (z.B. FODA [KCHN90], featureRSEB [GrFA98], RequiLine [MaLi04], CBFM [CzHE05]) basieren auf der Idee der Merkmalsmodellierung, die im Rahmen des Feature Oriented Domain Analysis (FODA) Ansatzes entwickelt wurde [KCHN90]. Die Merkmalsmodelle werden als eine Abstraktionsebene verwendet, um die gemeinsamen und variablen Eigenschaften von Produkten einer Produktlinie zu dokumentieren und mit Stakeholdern zu diskutieren. Dabei wird eine Menge von Anforderungen, welche die gleiche Produkteigenschaft beschreiben, durch ein Merkmal ersetzt.
2. *UML-Erweiterungen*: Die Variabilitätsmodellierungsansätze in dieser Gruppe schlagen vor, UML-Modelle um Variabilitätsaspekte zu erweitern. Clauß [Clau01] hat vorgeschlagen, Variabilitätsaspekte durch neue Beziehungen und Stereotypen zu modellieren. Jacobsen et al. [JaGJ97] führten Variationspunkte in Use Case Modelle ein. Mehrere Arbeiten benutzen die Erweiterungen von Use Case Diagrammen für die Variabilitätsmodellierung (vgl. [MaLi02, HaPo03]). Eine besondere Untergruppe stellen die komponentenbasierten Methoden dar (RSEB [JaGJ97], Kobra [AtMB00, ABBK02], SPLIT [CoJB00]). Diese Ansätze benutzen nicht nur Erweiterungen von UML, sondern auch die Gedanken der Komponentenwiederverwendung.
3. *Eigenständige Variabilitätsmodellierungssprachen*: Die letzte Gruppe fasst alle Ansätze zusammen, die eine eigene Variabilitätsmodellierungssprache vorschlagen. Zum Beispiel ist Catalysis [Souz99] ein objektorientiertes, umfassendes Vorgehensmodell, das sich auf die komponentenorientierte Entwicklung konzentriert. Die eingesetzte Modellierungsnotation orientiert sich jedoch mehr an OOSE als an UML. Dabei werden teilweise die Bedeutungen von UML-Elementen verändert. Ein weiterer Ansatz mit der eigenständigen Variabilitätsmodellierungssprache ist COVAMOF [SDNB04]. In diesem Ansatz wird die Variabilität über unterschiedliche Typen der Variationspunkte und deren Status sowie die Beziehungen zwischen einzelnen Variabilitätspunkten modelliert. Weitere Beispiele für eigenständige Sprachen sind der ConIPF-Ansatz [HKWN06] sowie Koalish [AsSM04].

Für diese Arbeit wurde die merkmalsbasierte Modellierung als Basis der Variabilitätsmodellierung ausgewählt. Die Gründe dafür sind die leicht erlernbare Notation der Merkmalsmodelle und die intuitive Vorgehensweise bei der Trennung von variablen und gemeinsamen Eigenschaften, welche die schnelle Einführung und die effiziente Anwendung in natürlichsprachlichen Anforderungsdokumenten ermöglichen. Im Abschnitt 2.2.2 wird ein tieferer Einblick in grundlegende Ideen der Merkmalsmodellierung, gängige Notationsformen sowie die Vielfalt der Merkmalsansätze gegeben.

Der Dokumentationsort von Variabilitätsmodellen

Der andere wichtige Aspekt für die Dokumentation von Variabilität sind die unterschiedlichen Dokumentationsorte für die Variabilitätsmodelle. Es wird unterschieden zwischen *integrierten* und *orthogonalen* Variabilitätsmodellen.

Die *integrierten Variabilitätsmodelle* dokumentieren Variabilität in den bestehenden Entwicklungsartefakten, indem die bestehenden Informationen um die Variabilitätsinformationen erweitert werden. Am meisten sind die vorhandenen Anforderungs- oder Implementierungsmodelle betroffen. Die Modellierungssprachen werden erweitert oder die bestehenden Modellierungselemente von Modellierungssprachen werden durch Notizen oder Kommentare annotiert.

Die Dokumentation von Variabilität innerhalb von Modellen hat einige Vorteile. So können die notwendigen Entscheidungen zur Bindung der Variabilität relativ leicht erkannt werden. Ein Nachteil der integrierten Variabilitätsmodelle ist, dass oft die Variabilitätsinformationen über mehrere Modelle verteilt und ggf. sogar redundant dokumentiert werden. Dadurch wird es erschwert, die Variabilitätsinformationen konsistent zu halten [Pohl07].

Eine Alternative der integrierten Variabilitätsmodellierung ist die *orthogonale Variabilitätsmodellierung* [BGLN03, PoBL05, MeHP07]. Der Gedanke der orthogonalen Variabilitätsmodellierung basiert auf den Prinzipien der aspekt-orientierten Modellierung [KLMM97]. Die Variabilität wird getrennt von den anderen Entwicklungsartefakten dokumentiert. In einem Variabilitätsmodell werden nur Variabilitätsinformationen (z.B. Variabilitätskriterien und -ausprägungen, Varianten und Bedingungen) dokumentiert. Die Variabilitätsinformationen des orthogonalen Variabilitätsmodells werden mit ein oder mehreren Elementen der Entwicklungsartefakte verbunden.

Der Vorteil der orthogonalen Variabilitätsmodellierung ist, dass die Variabilitätsinformationen nicht über mehrere Modelle verteilt sind. Die orthogonalen Modelle unterstützen die Kommunikation der Variabilität gegenüber den Stakeholdern, da die Komplexität (gegenüber integrierten Variabilitätsmodellen) geringer ist. Die orthogonale Variabilitätsmodellierung unterstützt die Konsistenz der Modelle, da die Änderung nur an einer Stelle eingearbeitet werden muss. Zusätzlich unterstützt die orthogonale Variabilität die Nachvollziehbarkeit (vgl. [Pohl07]).

2.2.2 Merkmalsmodellierung

Die Merkmalsmodellierung wurde im Rahmen des FODA-Ansatzes 1990 am Software Engineering Institute (SEI) der Carnegie Mellon University entwickelt [KCHN90]. FODA liefert ein methodisches Vorgehen für die Domänenanalyse, indem Gemeinsamkeiten von verwandten Softwaresystemen analysiert und durch die generische Beschreibung von Anforderungen beschrieben werden. Dazu wurde erstmals die Notation der Merkmalsdiagramme verwendet, die eine Abstraktionsebene für die Anforderungen schaffen und die Abhängigkeiten zwischen diesen in grafischer Form veranschaulichen.

Die Merkmale stellen die strukturellen und funktionalen Eigenschaften der untersuchten Produktvarianten dar. Es wird dabei zwischen gemeinsamen und variablen Merkmalen unterschieden:

1. Ein gemeinsames Merkmal ist eine Eigenschaft, die alle Produkte einer Produktlinie aufweisen.
2. Ein variables Merkmal ist eine Eigenschaft, worin sich Produkte einer Produktlinie voneinander unterscheiden können.

Der Begriff *Variabilitätskriterium* (1-6) beschreibt nur die variablen Eigenschaften eines Bezugsgegenstandes. Der Begriff *Merkmal* wird im Rahmen dieser Arbeit zur Beschreibung aller Variabilitätskriterien und deren Ausprägungen benutzt.

Definition 2-4: Merkmal

Ein Merkmal ist sowohl eine variable Eigenschaft eines Bezugsgegenstandes als auch eine Ausprägung eines Variabilitätskriteriums.

Um die Variabilität einer Domäne bzw. einer Produktlinie darstellen zu können, wurden die Merkmalsmodelle entwickelt. Ein Merkmalsmodell besteht aus einem Merkmalsdiagramm sowie zusätzlichen Informationen.

Definition 2-5: Merkmalsmodell (engl. Feature Model)

Ein Merkmalsmodell umfasst die Merkmale von allen Produkten einer Produktlinie sowie die Beziehungen zwischen diesen Merkmalen. Es dient dazu Gemeinsamkeiten und Unterschiede verschiedener Produkte einer Produktlinie auf strukturierte und übersichtliche Weise zu dokumentieren.

Definition 2-6: Merkmalsdiagramm (engl. Feature Diagram)

Ein Merkmalsdiagramm ist eine grafische Darstellung aller Gemeinsamkeiten und aller Unterschiede der Produkte einer Produktlinie durch eine hierarchische Struktur.

Die Merkmalsdiagramme liefern eine abstrakte und prägnante Repräsentation von Variabilität, indem sie die gemeinsamen und variablen Merkmale von Instanzen einer Produktlinie wiedergeben.

Im weiteren Verlauf der Arbeit wird der Begriff Variabilitätsmodell zur Beschreibung der Merkmalsmodelle benutzt, welches nur die variablen Merkmale dokumentiert.

Definition 2-7: Variabilitätsmodell

Ein Variabilitätsmodell umfasst alle Variabilitätskriterien und deren Ausprägungen von allen Produkten einer Produktlinie sowie die Beziehungen zwischen Variabilitätskriterien und/oder deren Ausprägungen.

Die Merkmalsmodellierung verfolgt folgende Ziele:

- Funktionalitäten und qualitative Charakteristiken eines Produktes abstrakter modellieren, als es auf der konkreten Anforderungsebene möglich wäre [KCHN90].
- Gemeinsamkeiten und Variabilität einer Produktlinie zu modellieren sowie ein Begriffslexikon aufzubauen, welches die Kommunikation zwischen den Entwicklern und den beteiligten Domänenexperten bzw. Kunden unterstützt soll ([KKLK98, KCHN90, LeKL02]).
- Gemeinsamkeiten und Variabilität einer Produktlinie zu dokumentieren um potentielle Produkte zu identifizieren ([FeFB02, LeKL02, CzEi00, HSVM00]).

Im nächsten Abschnitt wird die Merkmalsmodellierung nach FODA detaillierter vorgestellt.

2.2.2.1 FODA

Alle gängigen Notationen der Merkmalsmodelle basieren auf den Ideen der FODA-Notation. Die Notation eines Merkmalsmodells nach FODA besteht aus vier Bestandteilen:

- *Merkmalsdiagramm* zur hierarchischen Zerlegung von Merkmalen;
- *Merkmalsdefinitionen* einschließlich Beschreibung und Bindezeit (engl. binding time);
- *Auswahlregeln* (engl. composition rules), um erlaubte und ungültige Kombinationen von Merkmalen zu bestimmen;

- *Gründe* (engl. rationale) für oder gegen die Auswahl eines Merkmals für eine Konfiguration.

Die Merkmalsdiagramme werden in FODA in Form von Baumdiagrammen (und-oder Bäume) aufgebaut, wobei die Knoten des Baumes den Merkmalen entsprechen. Abhängigkeiten werden durch Hierarchie und spezielle Typen für Knoten oder Kanten beschrieben. Es werden drei Typen von Merkmalen unterschieden:

- Ein *notwendiges Merkmal* (engl. mandatory feature) muss immer ausgewählt werden, sobald das übergeordnete Merkmal ausgewählt ist.
- Ein *optionales Merkmal* (engl. optional feature) kann ausgewählt werden, sobald das übergeordnete Merkmal ausgewählt ist.
- *Alternative Merkmale* (engl. alternative features) stellt eine Gruppe von Merkmalen dar, aus welcher nur ein Merkmal ausgewählt werden kann, sobald das übergeordnete Merkmal ausgewählt ist.

Grundsätzlich kann ein Merkmal nur ausgewählt werden, wenn dessen übergeordneter Knoten ebenfalls ausgewählt ist. Die Wurzel des Diagramms muss immer ausgewählt werden. Die Beziehungen zwischen Merkmalen, die nicht durch die Baumstruktur ausgedrückt sind, können durch zusätzliche Beziehungstypen beschrieben werden. In der Literatur existieren mehrere Arbeiten (z.B. [KCNP90, StRP03, LeKa04]) die sich mit der Typisierung der möglichen Beziehungen zwischen Merkmalen beschäftigen. In dieser Arbeit werden nur grundlegende Beziehungstypen betrachtet:

- *benötigt* (engl. requires): Falls Merkmal *a* das Merkmal *b* benötigt, dann kann das Merkmal *a* für eine Konfiguration des Merkmalsmodelles ausgewählt werden nur wenn auch das Merkmal *b* ausgewählt wird. Dieser Beziehungstyp ist unidirektional.
- *Schließt_aus* (engl. exclude): Falls Merkmal *a* das Merkmal *b* ausschließt, dann kann das Merkmal *a* für eine Konfiguration des Merkmalsmodelles ausgewählt werden nur wenn das Merkmal *b* nicht ausgewählt wird. Dieser Beziehungstyp ist bidirektional.

Die grafische Notation der Merkmalsmodellierung nach FODA stellt folgende Elemente zur Verfügung:

- Ein notwendiges Merkmal wird durch eine einfache Kante dargestellt. Das Merkmal *Karosserie* in Abbildung 4 ist ein notwendiges Merkmal.
- Ein optionales Merkmal wird durch eine einfache Kante und einen nicht ausgefüllten Kreis an dem zum Merkmal führenden Ende der Kante dargestellt. Das Merkmal *Anhängerkupplung* in Abbildung 4 ist ein optionales Merkmal.
- Eine Menge von alternativen Merkmalen wird durch einen nicht ausgefüllten Kreisbogen zwischen Kanten, die zu den Merkmalen führen, dokumentiert. Die Merkmale *Manuell und Automatik* in Abbildung 4 sind zwei alternative Merkmale.

Abbildung 4 präsentiert das vereinfachte Merkmalsmodell eines Autos. *Das Auto besteht aus der Karosserie, einem Motor, einem Getriebe und kann eine Anhängerkupplung haben. Der Motor muss entweder Benzin- oder Diesel-Motor sein und kann um einen Elektro-Motor erweitert werden(also ein Hybrid-Motor sein). Das Getriebe existiert in manueller und in automatischer Ausführung, wobei nur eines der beiden für ein Auto (im Gegensatz zum Motor) wählbar ist.*

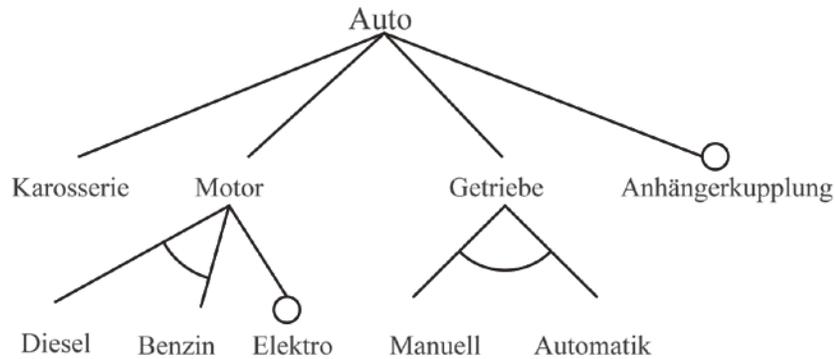


Abbildung 4: Beispiel des Merkmalsmodells mit FODA-Notation

2.2.2.2 Vielfalt von Merkmalsansätzen

Die Gründe für den Einsatz der Merkmalsmodellierung sind vielfältig und somit ist die Vielfalt von Merkmalsmodellierungsansätzen groß. Alle diese Ansätze erweitern das Modellierungspotenzial des ursprünglichen FODA-Ansatzes:

- [KKLK98, KCHN90] und [LeKL02] benutzen Merkmalsmodellierung für Domänenanalyse und -dokumentation.
- [FeFB02, LeKL02, CzEi00, HSVM00] setzen die Merkmalsmodellierung ein, um Produktlinien zu beschreiben und zu dokumentieren.
- [GrFA98] benutzen die Merkmalsmodellierung als unterstützende Methode zur Use Case Modellierung in RSEB [GrFA98].

Die unterschiedlichen Motivationen und Zielsetzungen spiegeln sich auch in den unterschiedlichen Modellierungselementen [LMNW03] wider. Die größten Änderungen finden sich vor allem in der Einführung von speziellen Abhängigkeitsbeziehungen und Merkmalsattributen:

- Für die grafische Darstellung der Beziehungen zwischen Merkmalen wurden explizite Modellierungselemente (z.B. gerichtete Pfeile) eingeführt [CzEi00].
- Für die Gruppierung von Merkmalen erlauben [FeFB02, GrFA98, HSVM00] frei wählbare, logische Gruppierungen. [LKKC00, LeKL02] schlagen vor, die Merkmale zu unterschiedlichen Sichten einzuordnen. Merkmale in der jeweiligen Sicht kennzeichnen Dienste in den Bereichen Capability, Domain Technology, Operating Environment und Implementation Technique. Neben dem Strukturierungselement der Gruppierung schlagen einige Ansätze vor (z.B. [FeFB02]), unterschiedliche Sichten auf die Merkmalsmodelle zu nutzen, um die Komplexität der Diagramme zu reduzieren.
- Des Weiteren bieten die Ansätze [FeFB02, GrFA98] die Möglichkeiten, die Merkmale zu attribuieren.

Anhand der Literaturrecherche wurde festgestellt, dass am häufigsten die Czarnecki-Eisenecker-Notation [CzEi00] verwendet wird. Die Czarnecki-Eisenecker-Notationsform [CzEi00] erweitert die Merkmalsmodellierungselemente von FODA durch die Einführung von *oder-Merkmalen*.

Bei den *oder-Merkmalen* muss bei der Selektion mindestens ein Merkmal ausgewählt werden, wenn das übergeordnete Merkmal ausgewählt wurde. Es können aber auch mehrere

Merkmale ausgewählt werden. Die grafische Notation wurde um die Darstellung von oder-Merkmalen erweitert. Die *oder-Beziehung* wird durch den ausgefüllten Kreisbogen dargestellt. Zusätzlich wurde vorgeschlagen die notwendigen Merkmale mithilfe von ausgefüllten Kreisen darzustellen.

Oft wird diese Notation mit einem weiteren Vorschlag von Czarnecki et al. [CzHE05] kombiniert, nämlich der Verwendung von Kardinalitäten. Die Merkmalsnotation wird um Kardinalitäten für einzelne Merkmale und Merkmalsgruppen erweitert. Die Kardinalität eines Merkmals bestimmt die minimale und maximale Anzahl von Ausprägungen, die von diesem Merkmal gewählt werden können.

Kardinalitäten für Gruppen bestimmen wie viele Merkmale in einer bestimmten Gruppe ausgewählt werden können. Die in FODA beschriebenen Merkmalstypen stellen in diesem Ansatz die Spezialfälle des allgemeinen Konzeptes dar:

- Die notwendigen Merkmale sind die Merkmale mit [1..1] Kardinalität;
- Die optionalen Merkmale die Merkmale mit [0..1] Kardinalität;
- Bei alternativen Merkmalen ist die Auswahlkardinalität [1,1] für das Merkmal selber und [0,1] für alle Kinder-Merkmale;
- Die Auswahlmerkmale werden mit [min, max] Kardinalität. Der min-Wert besagt wie viele Merkmale aus der Gruppe bei der Erstellung einer Konfiguration mindestens ausgewählt werden müssen, der max-Wert wie viele maximal.

Abbildung 5 präsentiert das von der Aussage her gleiche Beispiel wie Abbildung 4 jedoch in der kardinalitätsbasierten Notation. *Das Auto besteht aus der Karosserie, einem Motor, einem Getriebe und kann eine Anhängerkupplung haben. Der Motor muss entweder Benzin- oder Diesel-Motor und kann um ein Elektro-Motor erweitert werden(also ein Hybrid-Motor) sein. Die Information, dass ein Auto nicht gleichzeitig einen Benzin und Diesel-Motor haben darf, muss noch außerhalb des Modells dokumentiert werden. Das Getriebe existiert in manueller und in automatischer Ausführung, wobei nur eines der beiden für ein Auto (im Gegensatz zum Motor) wählbar ist.*

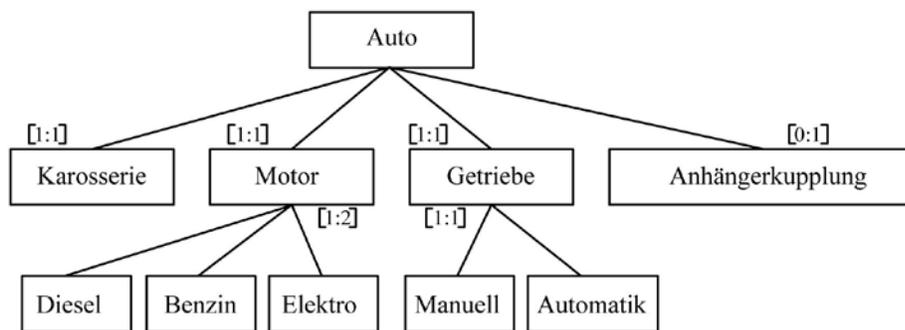


Abbildung 5: Beispiel für die kardinalitätsbasierte Notation eines Merkmalsmodells

Die genaue Beschreibung sowie der Vergleich von unterschiedlichen Merkmalsmodellierungsansätzen liegen nicht im Fokus dieser Arbeit und kann bei [LMMW03] oder [SiDi06] nachgelesen werden.

2.2.2.3 Merkmalsnotation in vorliegender Arbeit

In Rahmen der vorliegenden Arbeit wird die angepasste Czarnecki-Eisenecker-Notation [CzEi00] mit den Kardinalitätsangaben für auswählbare Merkmale verwendet. Für die Dokumentation der Elemente eines Variabilitätsmodells werden folgenden Darstellungen verwendet:

- Ein notwendiges Merkmal wird durch einen Merkmalsnamen am Ende einer einfachen Kante dargestellt. *Zum Beispiel das Merkmal Karosserie in Abbildung 6 ist ein notwendiges Merkmal.*
- Ein optionales Merkmal wird durch einen Merkmalsnamen am Ende einer einfachen Kante mit einem nicht ausgefüllten Kreis dargestellt. *Zum Beispiel das Merkmal Anhängerkupplung in Abbildung 6 ist ein optionales Merkmal.*
- Eine Menge von alternativen Merkmalen wird durch einen nicht ausgefüllten Kreisbogen zwischen Kanten, die zu den Merkmalen führen, dokumentiert. *Zum Beispiel die Merkmale Manuell und Automatik in Abbildung 6 sind zwei alternative Merkmale.*
- Eine Menge von auswählbaren Merkmalen wird durch einen ausgefüllten Kreisbogen zwischen Kanten, die zu den Merkmalen führen, dokumentiert. *Zusätzlich in der Nähe des ausgefüllten Kreisbogens wird die gültige Kardinalität für die Auswahl angegeben. Zum Beispiel die Merkmale Benzin, Diesel und Elektro in Abbildung 6 sind zwei auswählbare Merkmale, aus welchen minimal ein und maximal zwei in einer Konfiguration vorhanden sein können.*
- Die hierarchischen Beziehungen zwischen einzelnen Merkmalen werden mit Hilfe der einfachen Kanten dargestellt.
- Die „benötigt“ - und „schließt_aus“-Beziehungen werden durch gestrichelte, gerichtete Pfeile und Annotation des Beziehungstyps dokumentiert. *Zum Beispiel zwischen den Merkmalen Elektro und Anhängerkupplung in Abbildung 6 existiert eine „schließt aus“-Beziehung.*

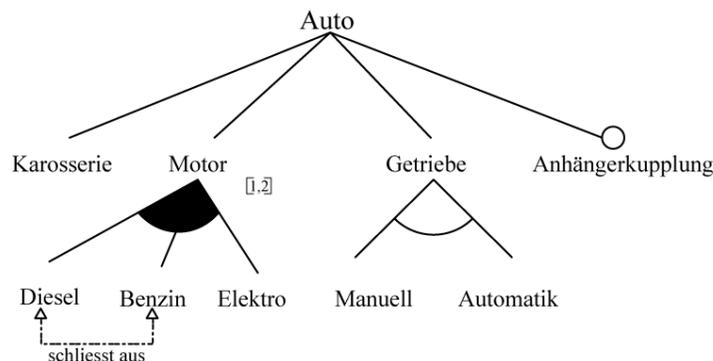


Abbildung 6: Beispiel des Merkmalsmodells mit FODA-Notation

2.2.2.4 Defizite der Merkmalsmodellierung

Trotz mehrerer Vorteile und großer Industrieakzeptanz in den Bereichen Produktmanagement und Software-Implementierung hat die Merkmalsmodellierung mehrere Defizite. Die Probleme und Nachteile der Merkmalsmodellierung wurden bereits in mehreren Arbeiten diskutiert (vgl. [BüLP04, MaLi04, Tava06]).

Die grundlegenden Defizite (vgl. [MaLi04], [ChBa10]) der Merkmalsmodellierung verursachen die Probleme, die den Einsatz von Merkmalsmodellierung insbesondere in dem Bereich Anforderungsmanagement nicht attraktiv bzw. in manchen Situationen sogar unmöglich machen.

- *Komplexität der Merkmalsmodelle:* Die Merkmalsmodelle können die Kommunikation unterstützen und sind übersichtlich, jedoch nur solange die Anzahl der Merkmale und Beziehungen zwischen den Merkmalen relativ klein ist. Merkmalsmodelle mit mehreren tausenden Merkmalen dokumentieren zwar die Variabilität, können aber nicht effektiv für die Kommunikation oder zur Entscheidungsfindung benutzt werden.
- *Abstraktionsebenen:* Bei der Erstellung von Merkmalsmodellen werden in einem Modell oft die Merkmale dokumentiert, welche die variablen Eigenschaften eines Artefaktes auf den unterschiedlichen Abstraktionsebenen repräsentieren. Dies kann dazu führen, dass Elemente als Merkmale aufgenommen werden, die eigentlich Konfigurationen darstellen, und somit die Redundanz von Information begünstigt wird.
- *Redundanz:* Bei sehr komplexen Merkmalsmodellen können Merkmale existieren, die gleiche Produkteigenschaften auf unterschiedliche Weisen dokumentieren. Der Modellierungsstil kann auch zu redundanten Informationen führen. Abbildung 7 stellt ein Beispiel einer redundanten Modellierung dar. Beide Modelle dokumentieren die gleiche Variabilität. In dem rechten Merkmalsmodell ist das Merkmal Europa zweimal vorhanden, im linken Merkmalsmodell jedoch nur einmal.
- *Änderbarkeit von Merkmalsmodellen:* Die Merkmalsmodelle unterliegen auch zeitlichen Veränderungen. Bei relativ kleinen Modellen können die Änderungen in den meisten Fällen problemlos gepflegt werden. Bei Merkmalsmodellen mit mehreren tausenden Merkmalen ist das Änderungsmanagement sehr aufwändig und kontinuierlich. Aber auch die Änderbarkeit von hundert dezentralen Variabilitätsmodellen ist bereits problematisch.
- *Inkonsistenz:* Die Kombination aus Redundanz und Änderbarkeit kann zu Inkonsistenzen in den Modellen oder Produktkonfigurationen führen. Je größer das Merkmalsmodell ist, desto höher ist die Wahrscheinlichkeit für das Auftreten von Inkonsistenzen. Auch die reine dezentrale Variabilitätsmodellierung kann zu Inkonsistenzen führen.

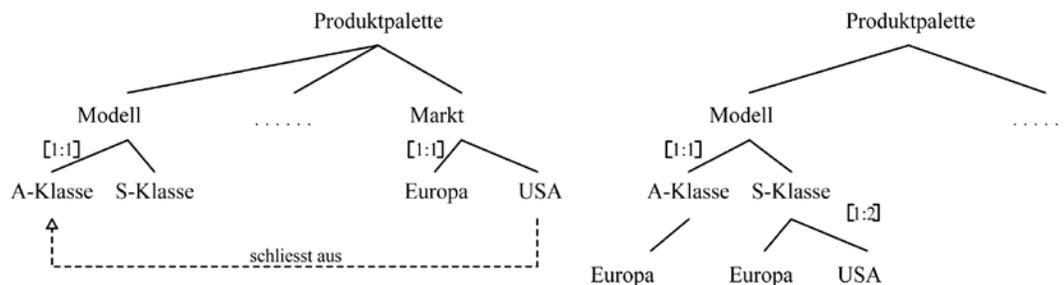


Abbildung 7: Redundante Modellierung

Der konfigurationsgekoppelte dezentrale Variabilitätsansatz, der in dieser Arbeit vorgestellt wird, bietet eine Lösung für die Beherrschung der Komplexität, sowie der Vermeidung von

Redundanz und Inkonsistenzen in den Merkmalsmodellen, in denen mehrere tausend Merkmale verwaltet werden.

2.3 Variabilitätsdokumentation in Anforderungsdokumenten

Für die Dokumentation von Variabilität in Anforderungsartefakten können sowohl integrierte als auch orthogonale Variabilitätsmodelle verwendet werden. Eine zentrale Herausforderung ist der Aufwand für die Zuordnung von Anforderungen zu Variabilitätsinformationen. Der am meisten verbreitete Ansatz für diese Zuordnung sowohl in der Forschung als auch in der Industrie ist der Matrix-Ansatz (vgl. [Pohl07, WeRe09]).

In dem *Matrix-Ansatz* wird eine Matrix erstellt. Die Zeilen der Matrix sind die Anforderungen an die gesamte Produktlinie. Die Spalten der Matrix repräsentieren die geplanten Produktvarianten. Falls eine Anforderung für eine Produktvariante gültig ist, wird eine direkte Zuordnung (z.B. durch „x“) erstellt. Ein Beispiel für diesen Ansatz sowie die Nachteile für den Einsatz in der Industrie werden im Abschnitt 3.2 vorgestellt.

Weber und Reiser [WeRe09] haben mehrere Ansätze zum Variabilitätsmanagement in Anforderungsdokumenten am Beispiel der Spezifikationen der Automobilhersteller identifiziert und bewertet.

Objekt-Hierarchie-Ansatz: In diesem Ansatz werden in der Spezifikation neue Spezifikationsobjekte hinzugefügt, welche die Variationspunkte für die zu spezifizierende Produktlinie darstellen. Die Definition eines Variationspunktes in diesem Ansatz entspricht der Definition von [PoBL05]. Alle variablen Anforderungen, die einen Variationspunkt beschreiben, werden als Kinder-Objekte von diesem Variationspunkt in der Spezifikation dokumentiert. Ein Lastenheft für ein Produkt wird durch Auswahl der benötigten Variationspunkte erstellt. Die variablen Anforderungen werden nur dann ins Lastenheft übernommen, wenn der zugehörige Variationspunkt für dieses Produkt ausgewählt wurde.

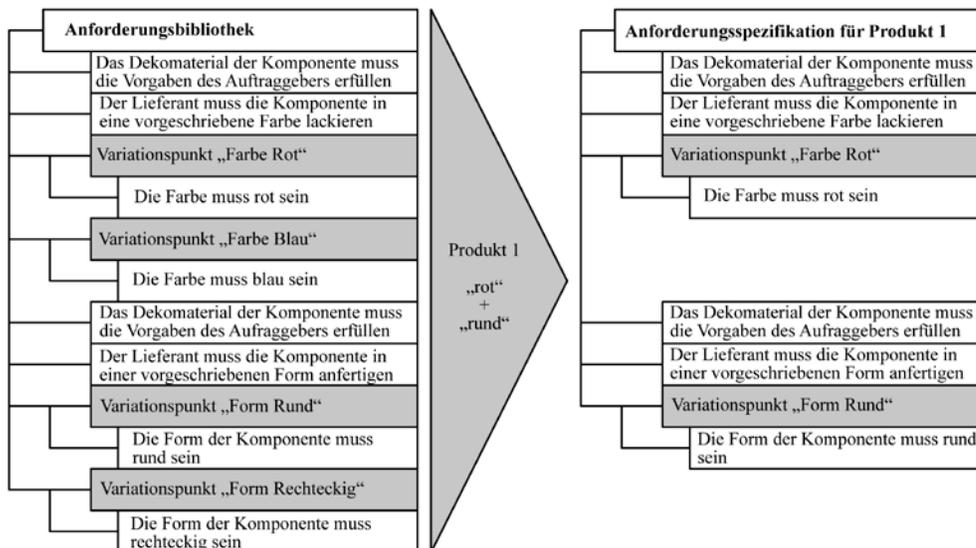


Abbildung 8: Objekt-Hierarchie-Ansatz

Abbildung 8 präsentiert ein Beispiel der Anwendung des Objekt-Hierarchie-Ansatzes. In der Anforderungsbibliothek sind die Anforderungen an mehrere Produkte einer Produktlinie dokumentiert. Die Produkte werden durch die Auswahl von folgenden Variationspunkten be-

stimmt: „Farbe Rot“, „Farbe Blau“, „Form rund“, „Form rechteckig“. Produkt 1 muss rot sein und eine runde Form haben. Die Spezifikation für Produkt 1 beinhaltet nur die variablen Anforderungen, die als Kinderobjekte von den ausgewählten Variationspunkten dokumentiert wurden.

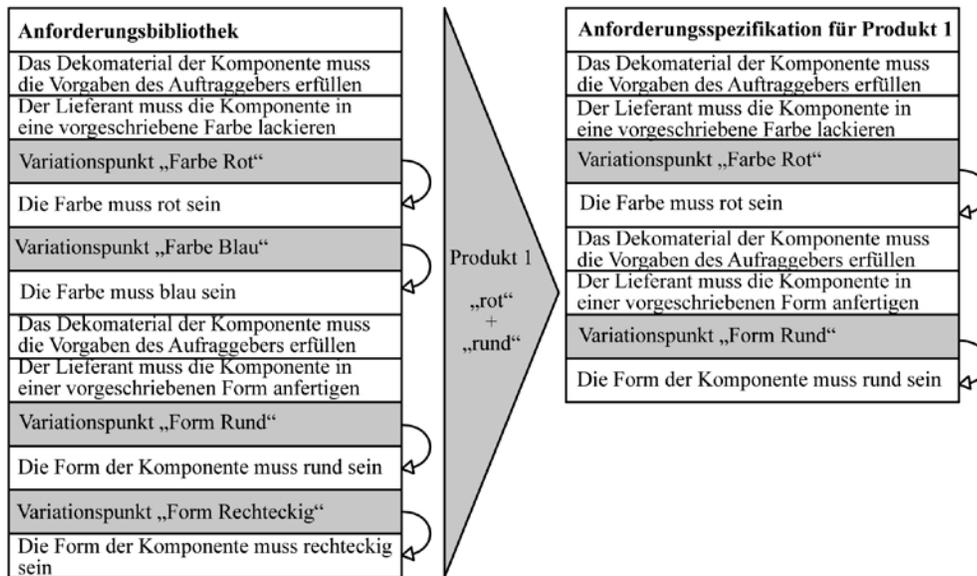


Abbildung 9: Nachvollziehbarkeitslinks

Nachvollziehbarkeitslinks: In diesem Ansatz werden die variablen Anforderungen mit Variationspunkten mithilfe von Nachvollziehbarkeitslinks verbunden. Abbildung 9 präsentiert das gleiche Beispiel wie Abbildung 8, jedoch mit Nachvollziehbarkeitslinks.

Sequentielle Reihenfolge von Spezifikationsobjekten: In diesem Ansatz werden in der Spezifikation neue Spezifikationsobjekte hinzugefügt, welche die Variationspunkte für die zu spezifizierende Produktlinie darstellen. Jeder Variationspunkt besteht aus zwei Spezifikationsobjekten, die den Anfang und das Ende des variablen Teils der Spezifikation markieren. Alle variablen Anforderungen, die einen Variationspunkt beschreiben, werden zwischen diesen zwei Variationspunkten dokumentiert. Die variablen Anforderungen werden nur dann ins Lastenheft übernommen, wenn der zugehörige Variationspunkt für dieses Produkt ausgewählt wurde.

Attributierung von Anforderungen: In diesem Fall bekommen die Anforderungen zusätzliche Attribute, in welchen dokumentiert wird, für welche variable Eigenschaft eine Anforderung relevant ist. Dabei wird eine positive Dokumentationsform benutzt. Bei dieser Dokumentationsform werden alle Anforderungen mit allen gültigen Werten von allen Attributen dokumentiert. Abbildung 10 präsentiert das gleiche Beispiel wie Abbildung 8, jedoch mittels Attributen bei Anforderungen umgesetzt.

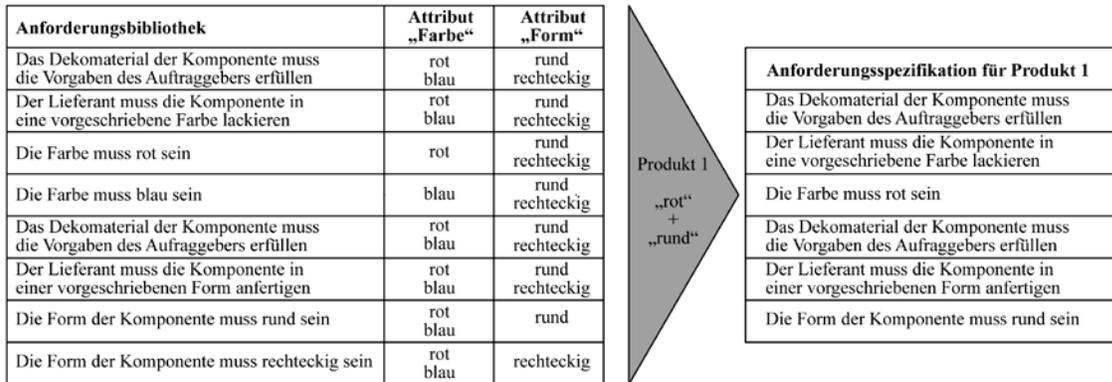


Abbildung 10: Attribuierung von Anforderungen

Weber und Reiser [WeRe09] haben in ihrer Arbeit die oben vorgestellten Ansätze evaluiert und festgestellt, dass keiner dieser Ansätze die Bedürfnisse der Industrie abdeckt. Der größte Nachteil dieser Ansätze ist der manuelle Aufwand für die Zuordnung von Anforderungen zu Varianten. Dieser Aspekt wird im Kapitel 3.2 am Beispiel einer Spezifikation der Daimler AG verdeutlicht.

2.4 Diskussion

Bei der Literaturrecherche wurde eine Reihe Forschungsarbeiten identifiziert, die sich mit dem Thema Variantenmanagement in Anforderungsdokumenten beschäftigen. In den nächsten Abschnitten werden drei Arbeiten zu diesem Thema vorgestellt, die auch im Kontext der Automobilindustrie entstanden sind. Der Grund, dass die Automobilindustrie besonderes Interesse in diesem Forschungsgebiet hat, liegt sowohl an der kontinuierlich steigenden Anzahl von Anforderungen und der steigenden Komplexität von Anforderungsdokumenten (vgl. [WeWe03, Houd10]) als auch an der kontinuierlich steigenden Anzahl von Produktvarianten [DLPW08].

2.4.1 Alternative Ansätze

In diesem Abschnitt werden alternative Ansätze zum Thema Variabilitätsmanagement in Anforderungsdokumenten und deren Vor- und Nachteile präsentiert.

Bühne et al. [BüLP04] haben sich mit dem Thema der Komplexität und Redundanz von Informationen in Merkmalsmodellen beschäftigt. Sie haben vorgeschlagen ein fahrzeugübergreifendes Merkmalsmodell für alle Fahrzeugmodelle zu erstellen. Dieses fahrzeugübergreifende Merkmalsmodell sollte nur die obligatorischen und optionalen Merkmale beinhalten. Die optionalen Merkmale müssen laut Bühne et al. zu den Fahrzeugmodellen zugeordnet werden. Diese Beziehungen zwischen Fahrzeugmodellen und Merkmalen können obligatorisch, optional oder alternativ sein. Diese Verbindung zwischen Merkmalen und Fahrzeugmodellen soll es ermöglichen, eine Sicht auf ein Merkmalsmodell für ein einzelnes Fahrzeugmodell zu erstellen. Als Erweiterung dieses Ansatzes wurde vorgeschlagen, einzelne Merkmale nicht nur zu Fahrzeugmodellen, sondern auch zu weiteren Variabilitätskriterien (z.B. Aufbauform der Karosserie, Absatzmärkte) zuzuordnen.

Die Idee, ein Merkmalsmodell, welches fahrzeugübergreifende Variabilität beinhaltet, zu erstellen ist nicht nur wegen der Anzahl von Merkmalen, sondern auch aus der Sicht des Änderungsmanagements unrealistisch. Um die Größe von einem fahrzeugübergreifenden Merkmalsmodell andeuten zu können, werden folgende drei Beispiele verwendet [Bout10a]:

- Das Merkmalsmodell einer Sonnenblende beinhaltet 20 Merkmale.
- Das Merkmalsmodell einer Säulenverkleidung beinhaltet circa 70 Merkmale.
- Das Merkmalsmodell eines Navigationsgeräts beinhaltet mehr als 150 Merkmale.

Auch die Zuweisung von einzelnen Merkmalen zu Variabilitätskriterien ist wegen der hohen Anzahl von Merkmalen in einem fahrzeugübergreifenden Merkmalsmodell unrealistisch. Ein weiterer Nachteil dieses Ansatzes ist, dass keine Möglichkeiten der Verbindung des Variabilitätsmodells mit Anforderungsartefakten vorgeschlagen wurden.

Tavakoli Kolagari [Tava06] hat in seiner Dissertation die Variabilitätsmodellierung für Anforderungsdokumente auf Basis der Kategorientheorie entwickelt. Er führt den Begriff Anforderungsbibliothek ein. Den Hauptbestandteil der Anforderungsbibliothek bilden die abstrakten und technischen Anforderungen ohne Designentscheidungen. Die abstrakten Anforderungen beschreiben die Zielfunktionalität sowie die benötigten Parameter und Signale. Die technischen Anforderungen erweitern die abstrakten Anforderungen mit technischen Informationen (z.B. Verhalten im Fehlerfall oder Werte und Kodierung von Signalen und Parametern). Die Anforderungen werden zu Sub-Variationspunkten zugeordnet, und die Sub-Variationspunkte gehören zu Variationspunkten. Diese Unterscheidung zwischen Sub-Variationspunkten und Variationspunkten war notwendig, um die Zugehörigkeit von Anforderungen zu Baureihen beschreiben zu können. Die Sub-Variationspunkte beschreiben die Zuordnung von Anforderungen zu Baureihen.

In seiner Arbeit hat Tavakoli Kolagari die zwei Begriffe *baureihen-spezifische Variabilität* und *baureihen-übergreifende Variabilität* eingeführt. Unter einer Baureihe wird in der Automobilindustrie eine Version eines Modells verstanden. Zum Beispiel die E-Klasse von Mercedes-Benz aus dem Jahr 2002 ist die Baureihe 211 und die Version aus dem Jahr 2010 ist die Baureihe 212. Die *baureihen-spezifische Variabilität* beschreibt variable Aspekte einer Baureihe (z.B. Ausstattung, Sonderfahrzeuge, Marktunterschiede). Die *baureihen-übergreifende Variabilität* beschreibt die Variabilität zwischen Baureihen und kann funktionale als auch baureihen-spezifische Variabilität beinhalten.

Der Fokus der Arbeit von [Tava06] lag in der Grundlagenforschung. Ein Nachteil seiner Arbeit liegt in der fehlenden Instanziierung seines Ansatzes und der fehlenden Repräsentationsmöglichkeit der Variabilität in Anforderungsartefakten.

Reiser [Reis08] hat sich in seiner Dissertation auch mit der Frage der Komplexität von Merkmalsmodellen beschäftigt. Zur Lösung des Komplexitätsproblems hat Reiser den Multi-Level-Feature-Tree-Ansatz (MLFT-Ansatz) und den Configuration-Links-Ansatz vorgeschlagen.

Der MLFT-Ansatz ist die Weiterentwicklung der Arbeit von Bühne et al. [BüLP04] und stellt einen Mittelweg zwischen einem zentralen Merkmalsmodell und mehreren dezentralen Merkmalsmodellen dar. Ein MLFT ist ein Baumdiagramm, in dem jedes Blatt ein eigenständiges Merkmalsmodell ist. Dabei stellt ein Eltern-Merkmal (mit Ausnahme von der Baumwurzel) immer das Referenzmodell für die Kinder-Merkmale dar. Auf der Basis von einem Referenz-Merkmalsmodell können mehrere Merkmalsmodelle (z.B. für unterschiedliche Fahrzeugmodelle) als Instanziierungen des Referenz-Merkmalsmodells erstellt werden. Wichtig ist, dass ein referenziertes Modell nur ein Referenz-Modell haben kann.

In der Realität sind oft die Produktkomponenten vorhanden, welche die Funktionen für mehrere Systeme realisieren. Zum Beispiel realisiert die Komponente *Blinker* die Funktion „Richtungswechsel anzeigen“ für das System *Außenlicht* und die Funktion „Schließung Feedback“ für das System *Schließung*. In diesen Fällen ist der MLFT-Ansatz wegen der Bedingung, dass jedes referenzierte Modell nur von einer Referenz abgeleitet werden kann, nicht anwendbar.

In der Arbeit von [Reis08] wird nicht dargestellt, wie Variabilitätsinformationen in einem Anforderungsartefakt dokumentiert werden können. Es wird vorgeschlagen die Anforderungen mit den Auswahlkriterien zu annotieren. Diese Auswahlkriterien entsprechen den logischen Merkmalskombinationen. Diese Vorgehensweise hat mehrere Nachteile, welche im Kapitel 6 diskutiert werden.

Der Configuration-Links-Ansatz beschäftigt sich mit der Erstellung von Konfigurationen mehrerer abhängiger Merkmalsmodelle. Dieser Ansatz wird detailliert im Kapitel 6 vorgestellt und diskutiert.

2.4.2 Abgrenzung zur alternativen Ansätzen

Die vorliegende Arbeit beschäftigt sich auch mit den Fragen Komplexität und Redundanz in Merkmalsmodellen. Dabei werden die Herausforderungen von industriellen Anforderungsdokumenten (z.B. Anzahl von Variabilitätskriterien und Ausprägungen sowie die Gesamtanzahl von Anforderungen) berücksichtigt.

Als Lösung des Komplexitätsproblems bietet diese Arbeit den konfigurationsgekoppelten dezentralen Variabilitätsmodellierungsansatz. Dieser Ansatz wird im Kapitel 6 vorgestellt. Im Kapitel 7 wird eine praktische Realisierung des Merkmalsmodellierungsansatzes für Anforderungsdokumente sowie die Darstellung von Merkmalen in Anforderungsdokumenten vorgestellt.

Im Gegensatz zur Arbeit von [Tava06] liegt der Fokus dieser Arbeit nicht in der Grundlagenforschung, sondern auf der Adaption und Erweiterung bereits existierender Forschungsansätze, in Richtung der Praxistauglichkeit. Dafür werden methodische Adapter entwickelt, die es ermöglichen etablierte Forschungsarbeiten zur Merkmalsmodellierung in industriellen Anforderungsdokumenten anzuwenden.

Im Gegensatz zur Arbeit von [Reis08] beschäftigt sich diese Arbeit auch mit Fällen, in denen der MLFT-Ansatz nicht anwendbar ist. Außerdem schließt die vorliegende Arbeit die Lücke der Arbeit von [Reis08], indem ein Ansatz für die Dokumentation von Variabilitätsinformationen in einem Anforderungsartefakt präsentiert wird (Kapitel 7).

Kapitel 3

Stand der Praxis

In diesem Kapitel wird der Stand der Praxis zum Thema Variantenmanagement in Anforderungsdokumenten in der Industrie präsentiert. Abschnitt 3.1 gibt einen Überblick über die Landschaft der Anforderungsdokumente. Dabei werden die Eigenschaften und Besonderheiten der Anforderungsdokumente in der Praxis vorgestellt und diskutiert. Nach der Vorstellung der relevanten Szenarien für die Anforderungswiederverwendung im Abschnitt 3.2, werden die Ergebnisse der Studie zum aktuellen Stand des Variantenmanagements in Anforderungsdokumenten am Beispiel eines Automobilherstellers präsentiert. Abschließend werden im Abschnitt 3.3 die identifizierten Herausforderungen für das Variantenmanagement in Anforderungsdokumenten sowie daraus folgende Anforderungen an den Variantenmanagementansatz für Anforderungsdokumente präsentiert und diskutiert.

3.1 Anforderungsdokumente in der Praxis

In den letzten Jahren ist die Bedeutung von Anforderungsdokumenten bei der Entwicklung von hoch-komplexen technischen Produkten (z.B. Fahrzeuge, Flugzeuge, Roboter) rapide gestiegen (vgl. [Grim03, WeWe03, HeHo04, Houd10]). Die Gründe dafür sind zwar vielfältig, aber es ist möglich die zwei gewichtigsten Gründe zu nennen: (1) die steigende Komplexität von Produkten (z.B. Fahrzeugen) sowie (2) die ständige Herausforderung, Entwicklungskosten und -zeit zu reduzieren.

Hoch-qualitative Anforderungsdokumente sind die Voraussetzung für erfolgreiche Zusammenarbeit mit Zulieferern weltweit. Deshalb wird Anforderungsmanagement-Kompetenz bei allen Produktherstellern ständig aufgebaut (vgl. [Monz08, WoBr09, Houd10]). Besonders hoch ist das Bedürfnis nach einem systematischen, tool-gestützten Anforderungsmanagement für software-basierte Komponenten [Houd10].

Die Qualität der Anforderungsdokumente hat entscheidenden Einfluss auf die Qualität der gelieferten Komponenten und die anfallenden Aufwände für Abnahme und Systemintegration [HeHo04]. Deswegen spielt Anforderungsmanagement in Industrieprojekten eine besonders große Rolle, in denen die einzelnen Komponenten des Gesamtproduktes (z.B. Fahrzeug, Flugzeug) von unterschiedlichen Lieferanten erstellt werden.

Das Anforderungsmanagement in der Automobilindustrie spielt seit fast 10 Jahren eine wichtige Rolle ([Houd03, WeWe03]) und ist heutzutage ein unabdingbarer Teil der Fahrzeugentwicklung. Anforderungsdokumente in der Automobilindustrie sind durch folgende Eigenschaften charakterisiert:

- Die hohe Anzahl von Anforderungen. Ein Lastenheft für eine Komponente beinhaltet oft mehrere tausend Anforderungen. Wie bereits bei Problembeschreibung erwähnt wurde, besteht ein Lastenheft für ein Motorsteuergerät bei BMW aus ca. 2500 Anforderungen [WoBr09] und ein ausgedrucktes Lastenheft für ein Kombiinstrument bei der Daimler AG besteht aus ca. 6000 Seiten [Bout10].
- Die großen Anteile der Lösungsbeschreibungen, um sicher zu sein, dass die unterschiedlichen Komponenten (von verschiedenen Lieferanten) zusammen funktionieren werden [Houd10].
- Die Berücksichtigung mehrerer Lösungsvarianten innerhalb einer Spezifikation [AIOp08].
- Die Dokumentationsstruktur, weil die Anforderungsdokumente oft die Organisationsstruktur widerspiegeln ([Houd03, Allm07]).
- Die Dokumentation von Anforderungen auf mehreren Abstraktionsebenen [Bout10].

3.1.1 Anforderungsdokumentenlandschaft

Die Anforderungen an hoch-komplexe Produkte werden oft auf verschiedenen Abstraktionsebenen dokumentiert, um die Komplexität der Produkte durch die gezielte Teilung zu beherrschen [HaRH01]. So werden zum Beispiel bei der Daimler AG die Anforderungen an neue Fahrzeuge und deren Komponenten auf drei Abstraktionsebenen dokumentiert: auf der Fahrzeug-, System- und Komponenten-Ebene [Bout10a]. Auf jeder Abstraktionsebene werden die Anforderungen in den entsprechenden Anforderungsdokumenten festgehalten:

- Das *Fahrzeuganforderungsdokument (FAD)* enthält eine umfangreiche Beschreibung des neuen Fahrzeugs. Im FAD werden zum Beispiel die Marketingaspekte (z.B. Zielgruppen, Produktpositionierung auf den Märkten), Designanforderungen und technischen Konzepte des neuen Fahrzeuges festgehalten. Ein FAD wird immer für eine neue Baureihe erstellt und dokumentiert die Anforderungen an alle Varianten einer Baureihe (z.B. Limousine, Cabrio, aber auch USA-Ausführung oder Taxi-Ausführung).
- Das *Systemanforderungsdokument (SAD)* bildet die Brücke zwischen den Anforderungen an das Gesamtfahrzeug und den Anforderungen an die einzelnen Komponenten. Ein System wird definiert als eine Menge logisch zusammenhängender Komponenten, die eine Gruppe von kundenerlebbaren Funktionen realisieren. Beispiele für ein System sind Außenlicht oder Sitzheizung. Die Notwendigkeit des SAD entsteht aus der Problematik, die Komponentenanforderungen direkt aus Fahrzeuganforderungen vollständig und widerspruchsfrei abzuleiten. Das SAD enthält alle Informationen zum entsprechenden System und die Definition von Schnittstellen zu anderen Systemen. Durch die Zuordnung von Komponenten zu Funktionen und Eigenschaften lässt sich nachvollziehen, welche Komponentenanforderungen aus welchen Fahrzeuganforderungen entstanden sind. Das SAD ist auch die Basis für Systemtest, FMEA und Systemdokumentation. Darüber hinaus liefert es strukturiert die Informationen für die Dokumentation der Systeme im Bereich After Sales. Das SAD ist eine interne Spezifikation und wird in der Regel nicht an Lieferanten weiter gegeben.

- Das *Komponentenanforderungsdokument (KAD)* dokumentiert die Anforderungen an eine einzelne Komponente (z.B. eine Sonnenblende, einen Stoßfänger). In einem KAD werden sowohl technische als auch nichttechnische (z.B. prozessuale) Anforderungen an die Komponentenentwicklung und spätere Serienproduktion dokumentiert. Die Anforderungen beschreiben alle beschlossenen Varianten (Farbvarianten, Ausstattungsvarianten, Ländervarianten etc.). Auf der Basis eines KAD wird ein Komponentenvergelastenheft erstellt. Ein Vergelastenheft besteht aus mindestens einem Lastenheft, kann aber auch mehrere Lastenhefte für mehrere Komponentenvarianten beinhalten. Somit können in einem Vergelastenheft widersprüchliche Anforderungen beinhaltet sein. Ein Vergelastenheft bildet die Basis für die Ausschreibung und den Vertrag mit dem Lieferanten. Deswegen ist es sehr wichtig, dass der Lieferant jederzeit nachvollziehen kann, zu welcher Komponentenvariante eine Anforderung gehört.

Die Beziehungen zwischen einzelnen Typen von Anforderungsdokumenten sind nicht 1:1 (siehe Abbildung 11). Die Gründe dafür liegen in der Verwendung von gleichen Komponenten für die Realisierung unterschiedlicher Funktionen in mehreren Systemen und die Wiederverwendung von Systemen in mehreren Fahrzeugmodellen. Die Komponente Türschloss realisiert zum Beispiel sowohl Funktionen des Systems Schließung als auch Funktionen des Systems Außenlicht. Beide Systeme werden in mehreren Fahrzeugmodellen (Baureihen) eingebaut.

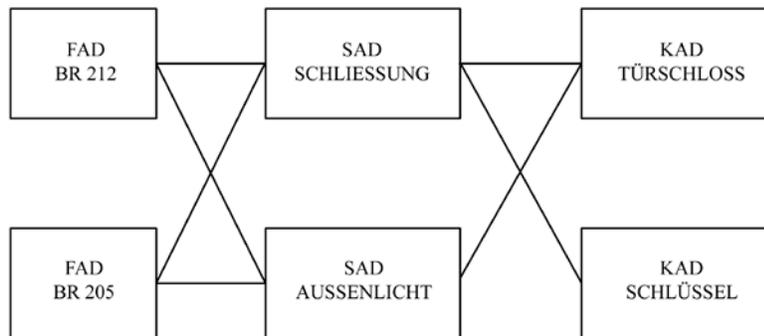


Abbildung 11: Informationsfluss zwischen unterschiedlichen Anforderungsdokumententypen

Für eine neue Baureihe werden ein Fahrzeuganforderungsdokument, ca. 90 Systemlastenhefte und ca. 400 Komponentenlastenhefte bzw. Komponenten-Vergelastenhefte erstellt [Houd10].

3.1.2 Eigenschaften von Anforderungsdokumenten

Für jeden Anforderungsdokumententyp existieren entsprechende Standardvorlagen, die auf Basis des VDI/VDE Standards 3694 [VDI/VDE3694] entwickelt wurden. Eine Standardvorlage unterstützt den Entwicklungs- und Spezifikationsprozess durch die Verwendung standardisierter Inhalte und Anforderungspakete. Die aktuelle Version der KAD-Standardvorlage der Daimler AG besteht zum Beispiel zurzeit aus circa 200 Seiten technischer Anforderungen und circa 200 Seiten mit Projektanforderungen. Die KAD-Standardvorlage beinhaltet Standardanforderungen, die nicht gelöscht oder verändert werden dürfen, aber auch Kapitel, deren

Befüllung der Entwickler komponentenspezifisch vornehmen muss (z.B. Verantwortlichkeiten, Termine, Gewichte).

Die ständige Weiterentwicklung der Standardvorlage berücksichtigt die Erfahrungen aus abgeschlossenen Projekten sowie Verbesserungen zu identifizierten Defiziten in der existierenden Vorlage. In einigen Fällen kommt es auch zu Strukturänderungen der Standardvorlage. In einem solchem Fall müssen alle komponenten-spezifischen Anforderungen neu angeordnet werden. Derartige Umstrukturierungen von mehreren tausenden Anforderungen in mehreren hundert Anforderungsdokumenten sind sehr kostenintensiv. Deshalb wurde ein Konzept zur Trennung von standardisierten und spezifischen Inhalten ausgearbeitet.

Die Grundidee ist die Teilung der Anforderungen in zwei Gruppen: die Anforderungen an eine Komponente oder ein System (spezifische Anforderungen) und die Projekt-, Produktions- und Logistik-Anforderungen. Die spezifischen Anforderungen einer Komponente oder eines Systems werden in einer Anforderungsbibliothek verwaltet. Eine Anforderungsbibliothek beinhaltet die Anforderungen an eine Komponente bzw. ein System für alle Baureihen sowie alle definierten Varianten. Die Anzahl von Objekten in einer Anforderungsbibliothek kann von 100 bis zu 50.000 variieren (vgl. [LeOt10, Houd10]). Ein Objekt kann sowohl eine Anforderung als auch eine Überschrift oder eine Information sein. Entsprechend der Klassifikation von Regnell et al. [ReSW08] können die meisten Anforderungsdokumente in die Klasse "large-scale RE" (ab 1.000 Anforderungen) bzw. in die Klasse "very large-scale RE" (mehr als 10.000 Anforderungen) eingeordnet werden [LeOt10].

Bei der Erstellung eines KAD werden die relevanten entsprechenden Anforderungen aus der Anforderungsbibliothek in die Standardvorlage eingefügt. Dieses Konzept bringt eine Reihe Vorteile mit sich:

- (1) Die Anforderungen an eine Komponente werden in einer Anforderungsbibliothek über den gesamten Produktentstehungsprozess dokumentiert;
- (2) Durch die Speicherung der Anforderungen in einer Anforderungsbibliothek wird die Wiederverwendung von Anforderungen unterstützt.

Die Herausforderungen für dieses Konzept stellen das Varianten- und Änderungsmanagement dar.

3.1.3 Szenarien für Anforderungswiederverwendung

In diesem Abschnitt werden die typischen Szenarien für die Wiederverwendung von Anforderungen am Beispiel der Daimler AG vorgestellt.

Die Entwicklung einer Komponente bzw. eines Systems geschieht in den meisten Fällen nicht im Rahmen der Entwicklung eines neuen Fahrzeugs. Diese Entkoppelung erlaubt auf der einen Seite die Wiederverwendung von Komponenten und Systemen in den unterschiedlichen Baureihen, auf der anderen Seite führt es zu divergierenden Lebenszyklen von Fahrzeugen, Systemen und Komponenten. So kommt es oft vor, dass innerhalb einer Baureihe unterschiedliche Versionen eines Systems bzw. einer Komponente eingesetzt werden [WeRe09].

Bis zu 80% der Anforderungen einer Baureihe werden als Basis für eine neue Baureihe wiederverwendet [Houd10]. Eine Art der Wiederverwendung, die in der Praxis noch oft Anwendung findet, besteht darin, dass die Anforderungsdokumente von einer Baureihe kopiert und angepasst werden. Zum Beispiel können die Anforderungen für den Fensterheber einer E-Klasse für den Fensterheber der C-Klasse kopiert und an wenigen Stellen (z.B. Geometriedaten) angepasst werden.

Eine andere Möglichkeit ist die Wiederverwendung von Anforderungen durch die Dokumentation von mehreren Varianten einer Komponente (bzw. eines Systems) in einer Anforderungsbibliothek. Die Voraussetzung für die Anwendbarkeit dieses Szenarios ist, dass die Zuordnung von jeder einzelnen Anforderung in der Anforderungsbibliothek zu den relevanten Baureihen-Varianten bzw. Komponenten-Varianten jederzeit nachvollziehbar ist. Diese Nachvollziehbarkeit wird benötigt, um die Lastenhefte für einzelne Varianten erstellen zu können.

Ein weiteres Ziel-Szenario ist die Wiederverwendung von Anforderungen und Stücklisteninformationen. Das Wissen über die Zuordnung von Anforderungen zu Varianten kann die Verbindung zwischen Anforderungen und Stücklisten herstellen. Eine Stückliste ist eine strukturierte Auflistung von Komponenten eines Produktes. Diese Information über die Verbindung zwischen Anforderungen und Stücklisten erlaubt im Fall eines Fehlers, der während der Produktion oder Anwendung auftritt, anhand der Komponenten-Teilenummer alle Anforderungen der fehlerhaften Komponente zu identifizieren. Danach wäre es möglich die Korrektheit der Anforderung zu überprüfen und bei Bedarf Gegenmaßnahmen durchzuführen. Außerdem wäre es möglich, anhand dieser Zuordnungsinformationen alle anderen Varianten dieser Komponenten, die die gleichen Anforderungen implementieren, zu identifizieren.

Die nachvollziehbare Zuordnung von Anforderungen zu Varianten unterstützt auch das Änderungsmanagement. Sollte eine Anforderung während des Entwicklungsprozesses geändert werden, so wäre es immer möglich, alle betroffenen Baureihen und Komponenten- bzw. System-Varianten zu identifizieren.

3.2 Variantenmanagement in Anforderungsdokumenten

Für die Identifikation und Analyse der existierenden Ansätze des Variantenmanagements bei der Daimler AG, wurden mehrere Komponenten- und System-Anforderungsbibliotheken betrachtet und mehrere Domänen-Experten befragt. Während der Befragungen wurden die Bewertungskriterien für Variantenmanagementansätze und insbesondere die Dokumentationsform von Variabilitätsinformationen in den Anforderungsdokumenten diskutiert und festgelegt [Bout08].

Es konnten vier Bedarfswelder identifiziert werden:

- (1) Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten;
- (2) Dokumentation von Variabilitätsinformationen in einem Variabilitätsmodell;
- (3) Erstellung von varianten-spezifischen Anforderungsdokumenten;
- (4) Pflege der Änderungen von Variabilitätsinformationen.

Im Review einer Analyse von Anforderungsdokumenten und der Befragung von Experten, wurde festgestellt, dass aktuell nur die Aspekte (1) und (3) in der Praxis in den unterschiedlichsten Ausprägungen existieren. Die Ansätze zu den Aspekten (2) und (4) im Umfeld der Anforderungsdokumente fehlen.

3.2.1 Variabilitätsdokumentationsansätze

Während der Analyse von Anforderungsdokumenten wurden fünf Variabilitätsdokumentationsansätze identifiziert, welche in den nächsten Abschnitten näher vorgestellt und anhand folgender Kriterien (K1) - (K8) bewertet werden [Bout10a]:

- (K1) Gleiche Anforderungen werden nur einmal dokumentiert.
- (K2) Die Zuordnung von Anforderungen zu Varianten ist erkennbar. Dies bedeutet, dass es klar sein muss, für welche Produkte die vorliegende Anforderung gültig ist.
- (K3) Die Zuordnung von Anforderungen zu Varianten ist ohne spezielle Vorkenntnisse verständlich.
- (K4) Die initiale Zuordnung von Anforderungen beim Hinzufügen neuer Varianten (z.B. neue Baureihe) ist erkennbar.
- (K5) Der Aufwand für die Zuordnung von Anforderungen zu Varianten muss im Vergleich zur aktuellen Situation reduziert werden.
- (K6) Die Zuordnung von variablen Anforderungen zu den variablen Eigenschaften muss toolgestützt prüfbar sein.
- (K7) Die Beziehungen zwischen variablen Eigenschaften sind erkennbar. Dies bedeutet, dass es klar sein muss, welche variable Eigenschaften nicht in einem Produkt vorkommen dürfen oder welche variable Eigenschaften in einem Produkt vorkommen müssen.
- (K8) Die Einhaltung der Beziehungen zwischen variablen Eigenschaften in Anforderungsdokumenten muss toolgestützt prüfbar sein.

In den nächsten Abschnitten werden die bei der Daimler AG identifizierten Variabilitätsdokumentationsansätze für Variantenmanagement in Anforderungsdokumenten vorgestellt und anhand eines Beispiels, welches bei der Vorstellung des ersten Ansatzes detailliert beschrieben wird, erläutert. Jeder Ansatz wird anhand der Kriterien (K1) – (K8) bewertet [Bout09]:

Variantendokumentation im Anforderungstext: Bei diesem Ansatz wird manuell in jeder Anforderung dokumentiert, für welche Variante diese Anforderung gültig ist. Tabelle 2 präsentiert ein Beispiel für diesen Ansatz. Diese Notation bedeutet, dass die erste Anforderung nur für die E- und S-Klasse relevant ist. Die zweite Anforderung ist nur für die G-Klasse relevant. Die dritte Anforderung ist für alle drei Klassen relevant.

Wie dieser Ansatz die Kriterien erfüllt, ist in der Tabelle 6 dokumentiert. Der Zeitaufwand für die Zuordnung der Anforderungen zu den gültigen Produkten unter Berücksichtigung der Anforderungsanzahl in einer Anforderungsbibliothek ist sehr hoch. Somit ist eine effektive und effiziente Arbeit mit diesem Ansatz nicht möglich. Außerdem ist die Fehlerwahrscheinlichkeit bei manueller Zuordnung und fehlenden automatisierten Prüfmöglichkeiten sehr hoch.

Tabelle 2: Textuelle Ansatz

Anforderungen
Das Dekormaterial der Komponente muss aus Leder sein. [E-Klasse, S-Klasse]
Das Dekormaterial der Komponente muss aus PVC sein. [G-Klasse]
Das Dekormaterial der Komponente muss aus Stoff sein.

Variantendokumentation in Spalten: Dieser Ansatz basiert auf der Dokumentation der Variantenausprägungen mit Hilfe von Anforderungsattributen. Dieser Ansatz kann auf zwei unterschiedliche Weisen benutzt werden.

Die einfache Version dieses Ansatzes entspricht dem *Matrix-Ansatz* aus Kapitel 2. Für jede Variante (z.B. E-Klasse, G-Klasse, S-Klasse) wird ein Attribut angelegt. Für jedes Attribut wird im Anforderungsdokument eine eigene Spalte erstellt. Für die Zuordnung der Anforderungen zu Varianten werden zwei Werte verwendet: „X“ (falls die Anforderung für diese Produktvariante gültig ist) und „-“ (falls die Anforderung für diese Variante nicht relevant ist). Tabelle 3 präsentiert die Variantendokumentation in Spalten.

Tabelle 3: Matrix-Ansatz

Anforderungen	E-Klasse	S-Klasse	G-Klasse
Das Dekormaterial der Komponente muss aus Leder sein.	X	X	-
Das Dekormaterial der Komponente muss aus PVC sein.	-	-	X
Das Dekormaterial der Komponente muss aus Stoff sein.	X	X	X

Als eine Erweiterung dieses Ansatzes kann ein zusätzliches Attribut eingeführt werden, welches die initialen Werte der Variabilität dokumentiert (genannt Common-Spalte). In dieser Spalte werden alle Anforderungen markiert, die für alle existierenden Varianten gültig sind. Diese Zuordnung wird beim Anlegen neuer Varianten-Spalten verwendet, um den Aufwand der Zuordnung von Anforderungen zu neuen Varianten zu minimieren.

Eine tool-spezifische Ausprägung des Matrix-Ansatzes ist die Verwendung von Multi-Enumeration-Attributen in DOORS. Diese Variante des Spalten-Ansatzes ermöglicht die Anzahl von Spalten zu reduzieren. Tabelle 4 präsentiert die Variantendokumentation in Spalten mit Hilfe des Multi-Enumeration-Attributs *Klasse*.

Die erweiterte Variante des Spalten-Ansatzes dokumentiert die Zuordnung von Anforderungen zu mehreren Variabilitätskriterien. In diesem Fall steigt die Anzahl der Spalten rapide, was zu Unübersichtlichkeit führt. Da die Variabilitätsinformationen sonst nirgendwo dokumentiert sind, fehlt bei diesem Ansatz die Möglichkeit die Beziehungen zwischen einzelnen Variabilitätskriterien zu dokumentieren und somit überprüfen zu können. Die Multi-Enumeration-Attribute können im Fall mehrerer Variabilitätskriterien nicht berücksichtigt werden, weil es zu falschen Aussagen führen kann.

Tabelle 4: Spalten-Ansatz

Anforderungen	Klasse
Das Dekormaterial der Komponente muss aus Leder sein.	E-Klasse
	S-Klasse
Das Dekormaterial der Komponente muss aus PVC sein.	G-Klasse
Das Dekormaterial der Komponente muss aus Stoff sein.	E-Klasse
	G-Klasse
	S-Klasse

Wie der Spalten-Ansatz die Bewertungskriterien (K1)-(K8) erfüllt, ist in der Tabelle 6 dokumentiert. Die effektive Arbeit mittels dieses Ansatzes ist möglich, wenn die Anzahl von Varianten relativ niedrig ist.

Variantendokumentation durch Kapitelstruktur: Dieser Ansatz basiert auf der expliziten Trennung von variantenübergreifenden und variantenspezifischen Anforderungen. Die variantenübergreifenden Anforderungen werden in einem eigenen Kapitel beschrieben, das in jedes Lastenheft übernommen wird. Falls die variantenspezifischen Anforderungen vom gemeinsamen Kern abweichen, werden diese Abweichungen in einem separaten Kapitel beschrieben.

Die Vorteile dieses Ansatzes sind die Transparenz der Beziehungen zwischen Anforderungen und Produktvarianten, sowie die erkennbaren initialen Werte. Ein großer Nachteil dieses Ansatzes ist die fehlende Übersichtlichkeit der Anforderungsrelevanz: es kann vorkommen, dass im gemeinsamen Teil eine Anforderung existiert, die in einer Produktvariante nicht gültig ist. Wie dieser Ansatz die Bewertungskriterien (K1)-(K8) erfüllt, kann aus Tabelle 6 entnommen werden.

Variantendokumentation in unterschiedlichen Anforderungsdokumenten: Dieser Ansatz ähnelt dem vorherigen Ansatz. Auch hier werden die variantenübergreifenden und variantenspezifischen Anforderungen getrennt verwaltet. Die variantenübergreifenden Anforderungen werden in einem eigenen Anforderungsdokument beschrieben. Die variantenspezifischen Anforderungen werden in einem anderen Anforderungsdokument beschrieben. Ein Lastenheft wird auf Basis von mehreren Anforderungsdokumenten erstellt. Die Vor- und Nachteile dieses Ansatzes sind deckungsgleich mit denen des Kapitel-Ansatzes. Die Erfüllung der Bewertungskriterien von diesem Ansatz ist in der Tabelle 6 dokumentiert.

Code-Regel-Ansatz: Beim Code-Regel-Ansatz werden die Varianten mithilfe eines Codes dokumentiert. Der Code entspricht dem Code im Stücklistenverwaltungssystem. Dabei wird zwischen Dokument- und Systemvariabilität unterschieden. Die Dokument-Variabilität bezieht sich auf die Baureihen, für die eine Anforderung gültig ist, und wird in einer Spalte durch ein Attribut-Auswahlfeld dokumentiert. Die Systemvariabilität wird durch eine Code-Liste und boolesche Operatoren definiert. Die geerbte oder per Link eingebundene Variabilität wird über eine UND-Verknüpfung mit der restlichen Variabilität der Elemente vereinigt. Die Gesamt-Variabilität wird als Summe (UND-Verknüpfung) der Dokumenten- und System-Variabilität gebildet.

Die Variabilität für die erste Anforderung ist [212 UND 221 UND 002]. Die Notation bedeutet, dass die erste Anforderung nur für die E-(212) und S-Klasse (221) mit Bezug aus Leder (002) relevant ist.

Der größte Vorteil von diesem Ansatz ist die mögliche Kopplung mit Stücklisten. Zu den Nachteilen dieses Ansatzes gehört unter anderem, dass für eine eindeutige Zuordnung die Anforderungen mehrmals dokumentiert werden müssen. Außerdem kann die Gesamtvariabilität nur mit Hilfe der Code-Beschreibung nachvollzogen werden.

Tabelle 5: Code-Regel-Ansatz

Anforderungen	Klasse	Bezug
Das Dekormaterial der Komponente muss aus Leder sein.	212 221	002
Das Dekormaterial der Komponente muss aus PVC sein.	463	001
Das Dekormaterial der Komponente muss aus Stoff sein.	212 221 463	003

Tabelle 6 fasst die Ergebnisse der Bewertung der vorgestellten Ansätze zusammen. Der Wert „X“ bedeutet, dass ein Ansatz das Kriterium erfüllt. Der Wert „-“ bedeutet, dass ein Ansatz das Kriterium nicht erfüllt. Der Wert „*“ bedeutet, dass ein Ansatz das Kriterium unter bestimmten Bedingungen erfüllt. Zum Beispiel können alle vorgestellten Ansätze effektiv und effizient sein, wenn ein System bzw. eine Komponente sich nur in einem Variabilitätskriterium unterscheiden und die Anzahl der Anforderungen relativ gering ist. In den meisten Fällen unterscheiden sich die Systeme bzw. die Komponenten in mehreren Variabilitätskriterien. Die Beziehungen zwischen einzelnen Variabilitätskriterien können beim Code-Regel-Ansatz dokumentiert werden. Keiner der vorgestellten Ansätze bietet die Überprüfung der Zuordnung von Anforderungen zu den Variabilitätsinformationen (K6).

Tabelle 6: Bewertung existierender Ansätze

Ansatz	K1	K2	K3	K4	K6	K7	K8
Textueller Ansatz	X	X	X	-	-	-	-
Spalten-Ansatz	X	X	X	-	*	-	-
Spalten-Ansatz mit Common-Wert	X	X	X	X	*	-	-
Attribut-Spalten-Ansatz	X	X	X	-	*	-	-
Kapitel-Ansatz	X	X	X	X	-	-	-
Ansatz der verteilten Anforderungsdokumente	X	X	X	X	-	X	X
Code-Regel-Ansatz	-	X	-	-	-	-	-

3.2.2 Herausforderungen für Variantenmanagementansätze

Nach Analyse der existierenden Ansätze in der Praxis am Beispiel der Daimler AG und grundlegender Literaturrecherche nach Berichten aus anderen Industriedomänen, wurde iden-

tifiziert, dass die Anwendung der theoretischen Konzepte für das Variantenmanagement (z.B. merkmalsbasiertes Variantenmanagement) in den industriellen Anforderungsdokumenten wegen einer Reihe von Herausforderungen nicht möglich ist. Dabei kann zwischen allgemeinen Herausforderungen und Herausforderungen, die für die Daimler AG von Bedeutung sind, unterschieden werden.

Die allgemeinen Herausforderungen sind:

- *Spezifikationsprache*: Fast alle Anforderungsdokumente werden in natürlicher Sprache spezifiziert [NeLa03].
- *Umfang und Heterogenität der Anforderungsdokumente*: Im Kapitel 3.1.2 wurden die typischen Umfänge von Lastenheften vorgestellt. Bei der Beschreibung der Wiederverwendungsszenarien wurde betont, dass die erkennbare Zuordnung von Anforderungen zu einzelnen Varianten die grundlegende Voraussetzung ist. Somit ist es verständlich, dass der Aufwand für die Zuordnung von einzelnen Anforderungen zu Varianten enorm ist und dieses Zuordnungsproblem die größte Herausforderung für das Variabilitätsmanagement in Lastenheften darstellt.
- *Komplexität von Variabilitätsmodellen*: Die Anzahl von Varianten in den Anforderungsdokumenten variiert sehr stark. In Anforderungsdokumenten für mechanische Komponenten werden oft nur wenige Varianten dokumentiert. Bei Komponenten aus dem Elektrik-Elektronik-Bereich ist die Anzahl der Varianten sehr groß. Während die Komponente *Sonnenblende* zum Beispiel nur vier Variabilitätskriterien hat, hat die Komponente *Säulenverkleidung* 15 und die Elektrik-Elektronik-Komponente *Player* ca. 70 Variabilitätskriterien². Wenn man beachtet, dass ein Fahrzeug je nach Modell aus mehreren tausenden Komponenten aufgebaut wird, werden die Variabilitätsdimensionen schnell klar.
- *Dokumentation von Anforderungen auf mehreren Abstraktionsebenen*: Der Variantenmanagement-Ansatz muss die bestehenden Abstraktionsebenen berücksichtigen.
- *Die vorgegebene Struktur eines Anforderungsdokumentes*: Der Variantenmanagement-Ansatz darf die existierende Struktur der Anforderungsdokumente nicht ändern.

Zusätzlich wurden bei der Daimler AG folgende Herausforderungen identifiziert:

- *Menschlicher Faktor*: Die Komponenten- und System-Anforderungen werden bei der Daimler AG von den Entwicklern spezifiziert. Als Experten im Bereich der Produktentwicklung sind sie entsprechend als Maschinenbauer oder Ingenieure ausgebildet. Der Großteil dieser Spezialisten hat jedoch kein umfangreiches Know-how in den Informatik- und Anforderungsmanagement-Bereichen. Die Hauptaufgabe eines solchen Spezialisten besteht in der Entwicklung von Systemen und System-Komponenten. Für die Erstellung eines Lastenheftes haben die Entwickler zwei bis sechs Monate zur Verfügung. Nach der Erstellungsphase sind die Entwickler „nur“ für die Änderungspflege in ihren Lastenheften zuständig. Die Zeitspanne zwischen der Erstellung neuer Lastenhefte beträgt drei bis fünf Jahre. Daraus entsteht eine weitere Anforderung an den Variantenmanagement-Ansatz: Er muss die Fähigkeiten der zukünftigen Anwender berücksichtigen. Er muss einfach und intuitiv bedienbar sein. Die Software-Unterstützung muss es ermöglichen, das Variabilitätsmodell einfach grafisch zu visualisieren. Die Anwendung von Prädikaten-Logik oder programmierähnlichen Sprachen wird von den potenziellen Anwendern nicht akzeptiert.

²Die Anzahl von Variabilitätskriterien und die Anzahl von Merkmalen unterscheidet sich wegen der Unterschiede in der Begriffsdefinition.

- *IT-Infrastruktur*: DOORS ist ein Standard-Tool für das Anforderungsmanagement auf der System- und Komponenten-Ebene bei der Daimler AG. Daraus folgt, dass die Kompatibilität des Variantenmanagementansatzes mit DOORS eine notwendige Bedingung darstellt. Andererseits muss der Variantenmanagementansatz eine tool-unabhängige Basis haben, um auch in Zukunft mit einem potenziell anderen Anforderungsmanagement-Tool funktionsfähig zu bleiben.

3.3 Diskussion

Basierend auf den Erkenntnissen aus Forschung und Praxis wurden die drei Grundbausteine eines Variantenmanagementansatzes in Anforderungsdokumenten identifiziert:

- (1) Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten,
- (2) Dokumentation von Variabilitätsinformationen im Variabilitätsmodell,
- (3) Management von Variabilitätsinformationen.

Die Analyse der Variantenmanagementansätze bei der Daimler AG hat gezeigt, dass in der Praxis in erster Linie nur nach einer Lösung für den ersten Grundbaustein gesucht wird. Um die Wiederverwendung von Anforderungen langfristig unterstützen und verwalten zu können, müssen jedoch alle drei Grundbausteine des Variantenmanagements in einem Variantenmanagementansatz berücksichtigt werden.

Nach der Analyse der Herausforderungen an einen Variantenmanagementansatz wurden folgende Anforderungen an einen Variantenmanagementansatz identifiziert [Bout10a]:

- (A1) Der Variantenmanagementansatz muss fähig sein, die Variabilität in natürlichsprachlich-formulierten Anforderungen zu identifizieren, dokumentieren und verwalten.
- (A2) Der neue Variantenmanagementansatz muss skalierbar sein. Der Ansatz muss kleine und große Variabilität im gleichen Maße unterstützen: von nur einigen bis hin zu mehreren hundert Variabilitätskriterien.
- (A3) Variabilitätsinformationen müssen auf Basis bestehender Anforderungsdokumente identifizierbar sein.

Folgende Anforderungen an die Dokumentation von Variabilitätsinformationen wurden während mehreren Interviews mit Komponenten- und Systemverantwortlichen bei der Daimler AG festgelegt:

- (A4) Die Dokumentation von Variabilitätsinformationen muss leicht erlernbar sein.
- (A5) Die Beziehungen zwischen einzelnen Variabilitätskriterien müssen dokumentierbar und prüfbar sein.
- (A6) Die Dokumentation von Variabilitätsinformationen muss durch ein Werkzeug unterstützt werden. Wichtig ist, dass das Variantenmanagementwerkzeug mit den Anforderungsmanagementwerkzeugen kompatibel ist.

An die Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten wurden folgende Anforderungen, auf Basis der Bedürfnisse der Komponenten- und Systemverantwortlichen bei der Daimler AG, festgelegt:

- (A7) Der Dokumentationsansatz in Anforderungsdokumenten muss sowohl für die Anforderungsdokumente mit mehreren 10.000 Anforderungen als auch für die Anforderungsdokumente mit einigen 100 Anforderungen anwendbar sein.
- (A8) Gleiche Anforderungen dürfen nur einmal dokumentiert werden.
- (A9) Der Dokumentationsansatz darf zu keinen Strukturänderungen in den bestehenden Anforderungsdokumenten führen.
- (A10) Der Dokumentationsansatz darf die Änderung von Variabilitätsmodellen aus Anforderungsdokumenten heraus nicht ermöglichen.
- (A11) Der Dokumentationsansatz muss den Aufwand für die Zuordnung von Anforderungen zu den Variabilitätskriterien minimieren.
- (A12) Die Zuordnung von Anforderungen zu Variabilitätskriterien muss leicht erkennbar ohne spezielles Wissen im Bereich des Variantenmanagements sein.
- (A13) Die Zuordnung von Anforderungen zu Varianten muss automatisch bzw. semi-automatisch überprüfbar sein.

Basierend auf den identifizierten Anforderungen wurden im Rahmen dieser Arbeit folgende Lösungsbausteine definiert, die in den nächsten Kapiteln detailliert vorgestellt werden:

- Kapitel 4: Semi-automatischer Ansatz zur Identifikation der Merkmale in den bestehenden Anforderungsdokumenten (MIA). Durch den MIA-Ansatz werden die Anforderungen A1 und A3 erfüllt.
- Kapitel 6: Konfigurationsgekoppelte dezentrale Variabilitätsmodellierung (KDVM) zur Lösung des Komplexitätsproblems bei Erstellung und Management der Variabilität. Der KDVM-Ansatz unterstützt die Erfüllung folgenden Anforderungen A2, A5, A7, A8 erfüllt.
- Kapitel 7: Ansatz zur Dokumentation von Variabilitätsinformationen in den bestehenden Anforderungsdokumenten (R2F) inklusive die Überprüfung der Zuordnung. Durch den R2F-Ansatz werden die Anforderungen A4 - A13 erfüllt.

Kapitel 4

Merkmalsidentifizierungsansatz (MIA)

Dieses Kapitel präsentiert den Merkmalsidentifizierungsansatz MIA, welcher einen Variantenmanager oder einen Anforderungsautor bei der Identifikation der Merkmale in bestehenden Anforderungsdokumenten unterstützt. Zuerst wird der aktuell eingesetzte Prozess zur Identifikation der Merkmale in Anforderungsdokumenten in der Industrie vorgestellt. Danach wird der neue semi-automatische Ansatz zu Merkmalsidentifizierung vorgestellt und das Werkzeug MIA, welches den vorgestellten Algorithmus implementiert, präsentiert. Abschließend wird der neue Ansatz anhand von vier industriellen Anforderungsdokumenten evaluiert. Am Ende des Kapitels werden die alternativen Ansätze für die Gewinnung von Informationen aus Textdokumenten vorgestellt und mit MIA verglichen.

4.1 Manuelle Identifizierungen von Merkmalen

Die aktuelle Vorgehensweise zur Identifikation der Merkmale in einem Anforderungsdokument basiert auf einem manuellen Review durch Domänenexperten. Während des Reviews analysiert das Experten-Team variable Anforderungen und versucht die Merkmale zu extrahieren. Dabei wird nach Eigenschaften gesucht, die für ein Produkt relevant und für andere nicht relevant sind. Das können sowohl funktionale Eigenschaften als auch technische oder gestalterische Aspekte sein.

Um die Merkmale auf Basis der gefundenen Phänomene zu identifizieren, suchen die Domänenexperten nach den Gründen für die Existenz dieser Phänomene. Dieses Vorgehen kann anhand des folgenden Beispiels demonstriert werden.

Variable Anforderung A: *“Das Zentralsteuergerät muss die Signale IND_FRONT und IND_REAR spätestens nach 200ms, nachdem es das Signal DOORS_LOCK_SIGNAL identifiziert hat, senden.”*

Variable Anforderung B: *“Das Zentralsteuergerät muss die Signale IND_FRONT and IND_REAR und IND_MIRROR spätestens nach 200ms, nachdem es das Signal DOORS_LOCK_SIGNAL identifiziert hat, senden.”*

Auf den ersten Blick kann festgestellt werden, dass der Unterschied zwischen beiden Anforderungen in der Aufforderung das Signal IND_MIRROR zu schicken liegt. Die Anforderung B beinhaltet das Signal IND_MIRROR und die Anforderung A nicht. Wird nach dem Grund für diese Aufforderung gesucht, so wird festgestellt, dass die Anforderung

zung B nur für die Produktvarianten relevant ist, welche das Merkmal „Spiegelsensor“ in der Variantenkonfiguration besitzen.

Berücksichtigt man die verteilte Landschaft von Anforderungsdokumenten sowie deren Anzahl und Umfang (Kapitel 3.1), so ist es verständlich, dass dieser Prozess sehr zeitaufwändig und kostenintensiv ist. Die Teil-Automatisierung des Merkmalidentifikationsprozesses kann den Zeitaufwand und somit die Kosten reduzieren.

4.2 Semi-automatischer Ansatz

Die Basis für die Automatisierung des Merkmalsidentifikationsprozesses können die Ansätze aus dem Bereich der lexikalischen Analyse bereitstellen.

Definition 3-1: *Lexikalische Analyse (engl. lexical analysis)[FrBY92]*

Lexikalische Analyse ist die Zerlegung eines Dokuments in Wörter.

Im Kapitel 1.2 wurde der Begriff „Merkmal“ (1-12) festgelegt. Der Merkmalsname muss ein prägnantes Wort sein, das die Produkteigenschaft beschreibt. In den meisten Merkmalsmodellen, die im Rahmen der Literaturrecherche analysiert wurden, ist der Merkmalsname ein Substantiv oder eine Kombination aus einem Adjektiv und Substantiv.

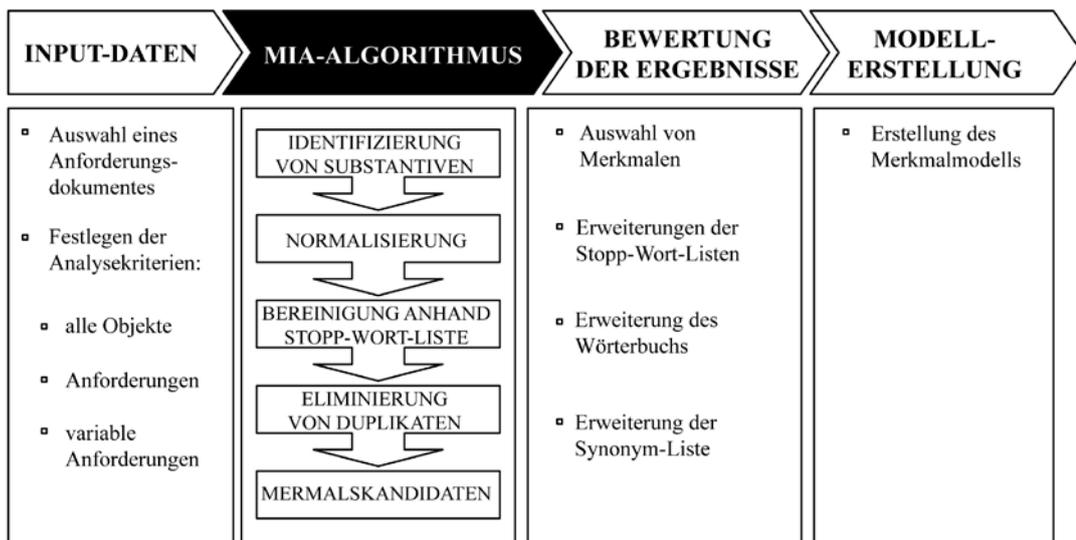


Abbildung 12: Merkmalsidentifikationsansatz (MIA)

Im Rahmen dieser Arbeit wurde ein Ansatz für die semi-automatische Identifikation von Merkmalen entwickelt: MIA. Dieser Ansatz ist in Abbildung 12 dargestellt. MIA besteht aus vier Schritten:

1. *Input-Daten:* In diesem Schritt wird festgelegt, in welchem Anforderungsdokument die Merkmale identifiziert werden sollen.
2. *Automatische Generierung der Merkmalskandidaten (MIA-Algorithmus):* In diesem Schritt werden in dem vorgegebenen Anforderungsdokument die Merkmalskandidaten identifiziert und in Form einer Liste dem Benutzer zur Verfügung gestellt.
3. *Bewertung der Ergebnisse:* Der Benutzer muss aus der erstellten Merkmalskandidatenliste die tatsächlichen Merkmale identifizieren. Zusätzlich können neue Wörter in das Wörterbuch und in die Stoppwort-Liste eingefügt werden.

4. *Variabilitätsmodellerstellung*: Auf Basis der identifizierten Merkmale kann ein Merkmalsmodell erstellt werden. Dabei muss beachtet werden, dass die generierte Merkmalskandidatenliste nur eine Teilmenge der tatsächlichen expliziten Merkmale darstellt. Die impliziten Merkmale (oft sind es Eltern-Merkmale) können nicht identifiziert werden, weil sie in den Anforderungsdokumenten nicht vorkommen. Die impliziten Merkmale müssen vom Benutzer identifiziert werden. *Beispiel: MIA identifiziert die beiden expliziten Merkmale „A-Säule“ und „B-Säule“. Das Elternmerkmal „Säulenposition“ ist im Anforderungsdokument nicht dokumentiert und kann somit nicht durch MIA identifiziert werden.*

Die Schritte 1, 3 und 4 sind manuelle Tätigkeiten, die von einem Domänenexperten durchgeführt werden müssen. Der Schritt 2 ist automatisiert. Die Generierung der Merkmalskandidatenliste kann mit Hilfe des Tools MIA, welches im Rahmen dieser Arbeit entwickelt wurde, durchgeführt werden. Die Aufgaben des MIA-Algorithmus sind: (i) Identifikation von Substantiven, (ii) Normalisierung von Substantiven, (iii) Entfernen von Stoppwörtern und (iv) Entfernen von Duplikaten. Die einzelnen Aufgaben des MIA-Algorithmus werden in den nächsten Abschnitten detailliert vorgestellt.

4.2.1 Identifikation von Substantiven

Die erste Aufgabe des Algorithmus ist die Identifikation aller Substantive in dem vom Benutzer ausgewählten Anforderungsdokument. Für diese Aufgabe wurde ein Part-of-Speech-Tagger benutzt.

Für die Identifikation von Substantiven wird das Part-of-Speech-Tagging-Verfahren (PoST) angewendet. Mit Hilfe dieses Verfahrens können die einzelnen Wörter eines Dokuments zu den Wortarten (Substantive, Adjektive, Verben) zugeordnet werden. Es existieren zwei PoST-Verfahrenstypen: regelbasierte und statistische. Bei den regelbasierten Ansätzen werden zuerst die Regeln definiert, welche am Text abgearbeitet werden. Bei den statistischen (stochastischen) Ansätzen lernt das Programm mithilfe von Trainingsdaten.

Die meisten PoST wurden für die englische Sprache entwickelt und verwenden Penn Treebank [MaSM93] als Trainingsdatensatz. Giesbrecht und Evert [GrEv09] haben für die in dem deutschsprachigen Raum existierenden PoST eine Evaluierung durchgeführt. Im Rahmen dieser Evaluierung wurden die folgenden fünf statistischen Tagger bewertet: TreeTager [Schm95], Wortart-Tagging Trigrams'n'Tags [Bran00], SVMtagger [GiMá04], Stanford Log-linear Part-Of-Speech Tagger [ToKM03], Apache UIMA Tagger [MZMB08], [NRWF08]. Nach den Ergebnissen der Evaluierung von Giesbrecht und Evert [GrEv09] liefert der Stanford Tagger die besten Ergebnisse bei der Datenbankbearbeitung. Deswegen wird dieser Tagger im Rahmen dieser Arbeit für die Identifikation von Substantiven in RE-Artefakten verwendet.

Im Rahmen der Arbeit von [LeOt10] wurde getestet wie viele Substantive der Stanford Tagger tatsächlich findet. Die Analyse von 100 zufälligen Anforderungen aus einem Anforderungsdokument hat gezeigt, dass in 84 Anforderungen alle Substantive korrekt identifiziert wurden. In acht Anforderungen wurden andere Wortarten als Substantive gekennzeichnet. In acht weiteren Anforderungen wurden nicht alle vorhandenen Substantive identifiziert. Die Fehler wurden vor allem in den Anforderungen festgestellt, welche in zwei Sprachen (Englisch und Deutsch) geschrieben sind. Abbildung 13 stellt die Ergebnisse der Evaluierung des Stanford Tagger grafisch dar.

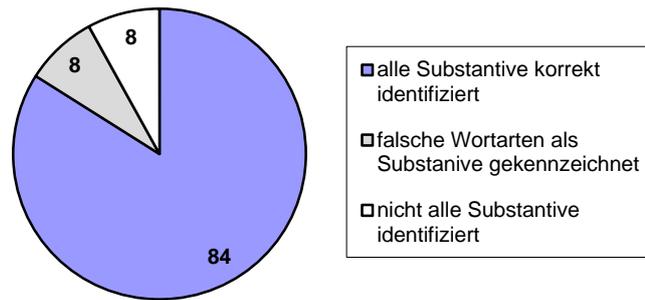


Abbildung 13: Evaluation von Stanford Tagger

4.2.2 Normalisierung von Substantiven

In dem zweiten Schritt des MIA-Algorithmus müssen die von PoST identifizierten Substantive normalisiert werden. Die Normalisierung von Substantiven ist die Reduzierung eines Wortes auf seine Grundform (für Substantive: Nominativ, Singular). Die Normalisierung von Substantiven kann mit Hilfe des linguistischen Verfahrens der Lemmatisierung durchgeführt werden.

Definition 3-2: Lemmatisierung

Lemmatisierung ist eine lexikographische Reduktion der Flexionsformen eines Wortes auf eine Grundform, aber auch die Zuordnung (oder auch Rückführung) einer Vollform zum entsprechenden Lemma.

Das Ziel der Lemmatisierung ist, die Wortformen, die durch die Flexion der Sprache gebildet werden, auf die Grundform abzubilden. Zum Beispiel: die Wörter Haus, Hauses, Häuser, Häusern werden zu dem Wort Haus normalisiert. Dabei werden zusammengesetzte Wörter (z.B. Briefträger), Bindestrichkomposita (z.B. Schiff-Fahrt) oder Ableitungen (z.B. wie farbig von Farbe) als eigenständige Lemmata geführt.

Es existieren zwei Ansätze für die Lemmatisierungsverfahren: regelbasierte Wortformnormierung und wörterbuchbasierte Wortformnormierung.

Die regelbasierte Lemmatisierung versucht mit Hilfe einer Regelbasis die Grundform eines Wortes zu bilden. Im Allgemeinen bestehen die Regeln aus einfachen String-Ersetzungsregeln. Beispiel: "en" -> "e", "e" -> " " mit der Wirkung: "Blumen" -> "Blume" "Tische" -> "Tisch". Der Nachteil von diesem Ansatz ist, dass die Regelbasis nicht alle Fälle der Grundformreduzierung abdecken kann. Die daraus resultierenden Fehler können zu den zwei bekannten Fehlerklassen, Overstemming und Understemming, zugeordnet werden [FrBY92].

Definition 3-3: Overstemming

Bei dem Overstemming wird bei der Grundformreduzierung nicht nur die Flexionsendung sondern auch ein Teil des Stamms abgeschnitten.

Definition 3-4: Understemming

Understemming: Bei dieser Fehlerklasse wird die Flexionsendung nicht vollständig abgeschnitten.

Regelbasierte Verfahren können bei Sprachen mit geringer morphologischer Komplexität (z.B. Englisch) mit großem Erfolg eingesetzt werden. Bei komplexeren Sprachen wie z.B.

der flexionsreichen und kompositumsträchtigen deutschen Sprache, wurden keine erfolgreichen Ansätze entwickelt [Leiz96].

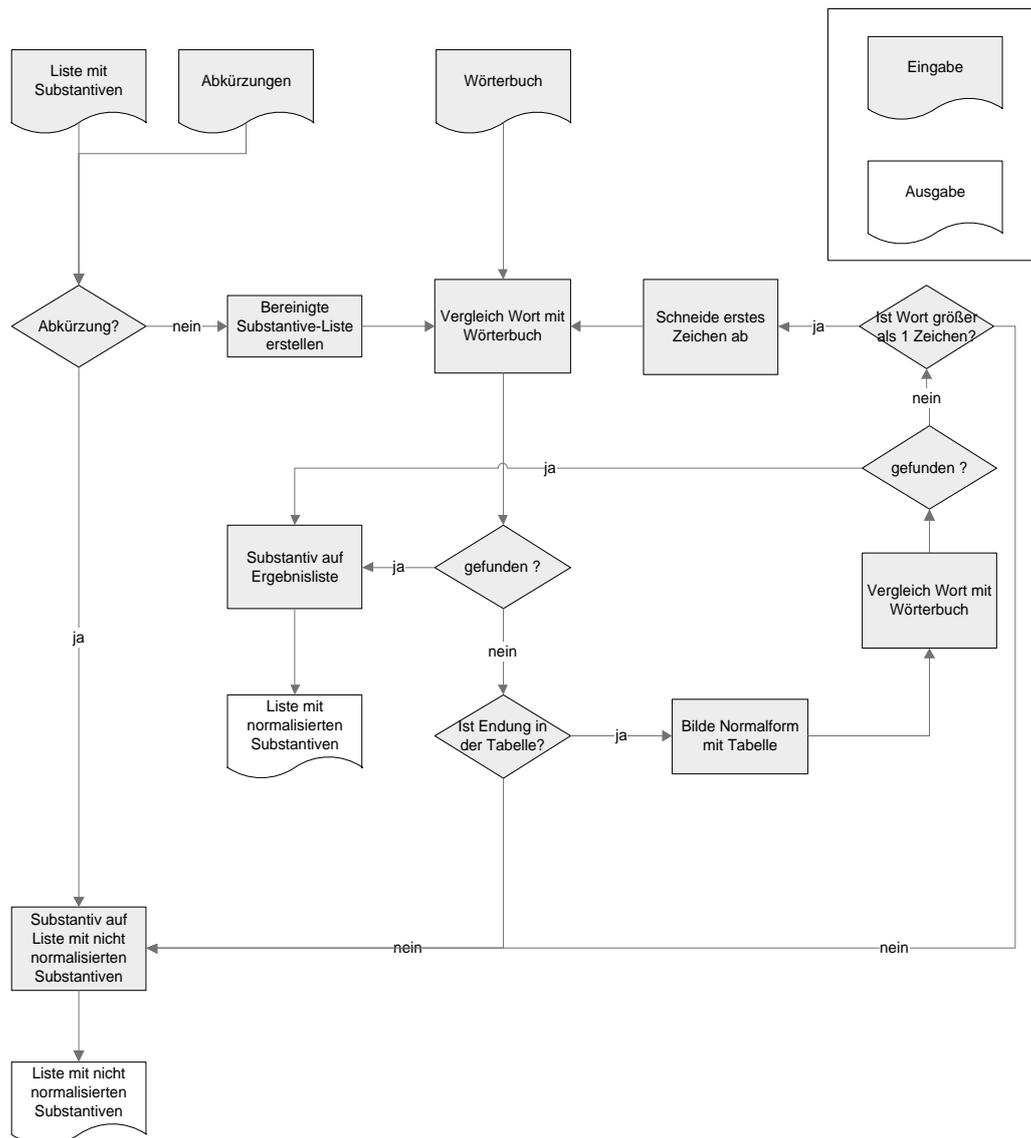


Abbildung 14: Normalisierung in MIA

Die wörterbuchbasierten Verfahren beruhen auf Einzelfalllösungen, wodurch sie gegenüber den regelbasierten zuverlässiger sind. Der Nachteil von dieser Verfahrensart ist die Notwendigkeit einer kontinuierlichen Wortdatenbankpflege. Bei den wörterbuchbasierten Verfahren liegen unterschiedliche Ansätze vor, die sich folgendermaßen einteilen lassen:

1. Vollform-Wörterbuch: in diesem Wörterbuch wird jede Wortform dokumentiert.
2. Stammform-Wörterbuch: in diesem Wörterbuch wird nur die Stammform dokumentiert.

Der manuelle Aufbau einer Wortformen-Datenbasis des deutschen Grundwortschatzes würde den Rahmen dieser Arbeit sprengen. Deswegen wurde nach Wortdatenbanken für die deutsche Sprache recherchiert, aus denen Daten extrahiert werden und welche als Basis dienen können. Hierbei wurden mehrere Datenbanken gefunden. Als Basis für diese Arbeit wurde das Morphy-Wörterbuch, welches im Rahmen einer Examensarbeit 1994 an der Universität-Gesamthochschule Paderborn von Leizus entwickelt wurde, gewählt

[Leiz96]. Die Datenbasis beinhaltet nach der Installation 17.380 Substantive und 1.409 Eigennamen. Hierbei ist zu berücksichtigen, dass es sich hier um Grundformen handelt, hinter denen sich abhängig von der Flexionsklasse mehrere Wortformen verbergen. Dieses exportierte Wörterbuch wurde um die Substantive erweitert, die bei der Analyse von mehreren Anforderungsdokumenten bei der Daimler AG identifiziert wurden. Nach der Anwendung von MIA auf vier Datensätze beinhaltet das Wörterbuch 26.523 Einträge.

Abbildung 14 präsentiert den Normalisierungsalgorithmus. Zuerst werden die Abkürzungen (z.B. Abteilungsnahmen) identifiziert und aus der Menge der Substantive, die normalisiert werden müssen, entfernt. Die verbliebenen Wörter werden mit dem Wörterbuch verglichen.

Wird bei diesem Vergleich ein Substantiv aus der Liste in dem Wörterbuch gefunden, so wird es in der Liste mit den normalisierten Substantiven eingetragen. Wenn das Substantiv nicht im Wörterbuch gefunden wird, versucht MIA anhand der Endungstabelle [Leiz96] die Endung zu entfernen. Die Endungstabelle ist im Anhang I dokumentiert. Falls die Endung des Substantivs nicht in der Tabelle vorhanden ist, so wird das Substantiv in der Liste mit nicht normalisierten Substantiven aufgenommen. Falls die Endung des Substantivs in der Tabelle vorhanden ist, wird das Substantiv normalisiert und danach mit dem Wörterbuch verglichen. Wird das Word im Wörterbuch nicht gefunden, dann wird (falls das Word aus mehr als drei Buchstaben besteht) der erste Buchstabe entfernt, um der Wortstamm zu finden. Nach der Entfernung des ersten Buchstabens wird der neue Vergleich mit dem Wörterbuch gemacht. Falls die Normalisierung nicht möglich ist, wird das Substantiv in die Liste der nicht normalisierten Substantive eingetragen.

Die Ergebnisse von diesem Schritt sind zwei Listen mit Substantiven, die in dem ausgewählten Anforderungsdokument identifiziert wurden und entweder normalisiert wurden oder nicht.

4.2.3 Bereinigung der Kandidatenliste

Nachdem alle Substantive in dem ausgewählten Anforderungsdokument identifiziert und normalisiert sind, müssen die Duplikate und die Stoppwörter entfernt werden. Allgemein werden solche Wörter als Stoppwörter klassifiziert, welche in einem Dokument zu häufig vorkommen und somit keinen nutzbaren Index-Term darstellen [FrBY92]. In dieser Arbeit werden die Stoppwörter anders als in der Literatur definiert.

Definition 3-5: Stoppwort

Ein Stoppwort ist ein Substantiv, welches in keinem Anforderungsdokument der Domäne ein Merkmal beschreiben kann.

Die Ausgangsbasis für die Erstellung der Stoppwortliste kann die Spezifikationsvorlage sein. In solchen Spezifikationsvorlagen sind viele Substantive beinhaltet, die den Rahmen (z.B. den Firmennamen, die Rollenbezeichnung) und das Projekt (die Projektphasen, Projektschritte) und nicht das Produkt beschreiben. Die initiale Stoppwortliste wurde aus Spezifikationsvorlagen mit Hilfe der ersten drei Schritte (Identifizierung von Substantiven, Lemmatisierung und Entfernung von Duplikaten) und anschließendem Review erstellt. Eine Alternative wäre die sukzessive Identifizierung der Stoppwörter und die Erstellung von Stoppwörterlisten anhand der Erfahrungen aus Projekten. Die initiale Stoppwortliste beinhaltete 2.701 Substantive und Akronyme.

Außerdem werden mit Hilfe regulärer Ausdrücke (engl. Regular Expression) alle Abteilungsnamen und Email-Adressen aus der initialen Kandidatenliste entfernt.

Nach der Anwendung von MIA auf vier Datensätze (siehe 5.4) beinhaltet die Stoppwortliste 3.749 Substantive und 463 Abkürzungen.

Als Ergebnis der Bereinigung wird eine bereinigte Merkmalskandidatenliste erstellt, die alle potenziellen Merkmale eines Anforderungsdokumentes beinhaltet. Diese Merkmalskandidatenliste kann für die Merkmalsmodellerstellung oder als Grundlage für ein Experten-Review benutzt werden.

4.3 MIA-Tool

Der vorgestellte Ansatz wurde im Rahmen dieser Dissertation in einem Werkzeug prototypisch implementiert. Das MIA-Tool wurde in der Programmiersprache Java entwickelt.

Das Wörterbuch, die Stoppwort-Liste und die Synonym-Liste wurden in Excel erstellt und verwaltet. Das MIA-Tool kann Daten aus der DOORS-Datenbank importieren³ und analysieren.

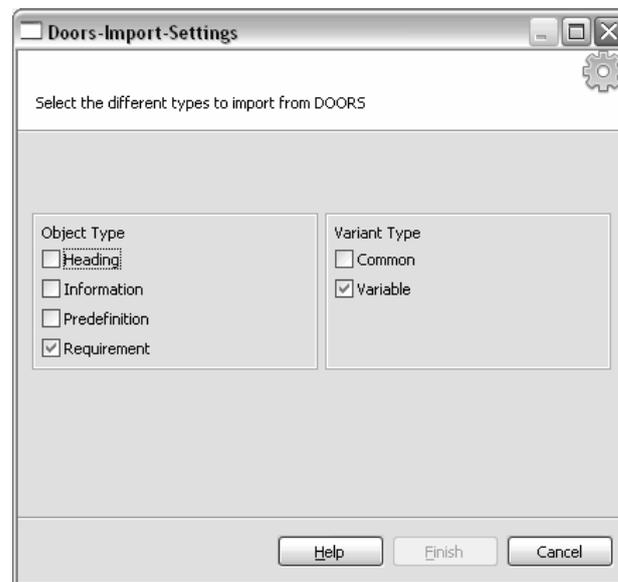


Abbildung 15: MIA-Tool – Festlegung des Imports

Bei der Import-Funktion kann der Benutzer auswählen, welche Objekte in einem Anforderungsdokument analysiert werden müssen. Es können sowohl alle Objekte (Überschriften, Informationen und Anforderungen) als auch nur Anforderungen importiert werden. Des Weiteren kann der Benutzer bestimmen, ob alle Anforderungen oder nur variable Anforderungen importiert werden müssen. Falls diese Unterscheidung (variable oder gemeinsame Anforderungen) in dem Anforderungsdokument nicht vorhanden ist, dann wird es dem Benutzer mitgeteilt und alle Anforderungen werden importiert. Abbildung 15 präsentiert dieses Auswahlfenster im MIA-Tool.

Für die Analyse-Funktion stehen dem Benutzer die folgenden Funktionen zur Verfügung (siehe Abbildung 16):

1. *Grammar Check*: Diese Funktion überprüft die Rechtschreibung in den importierten Anforderungen.
2. *Combine Synonyms*: Diese Funktion sucht in der Merkmalskandidaten-Liste nach Synonymen und präsentiert sie dem Benutzer als eine Wort-Gruppe.

Des Weiteren kann der Benutzer folgende Filtereinstellungen aktivieren:

³ Für den Import der Daten von DOORS wird die entsprechende Funktion von TraceTool [Leus10] benutzt.

3. *Abbreviation*: Mit der Aktivierung dieses Filters werden alle Abkürzungen, die in einer Abkürzungsliste verwaltet werden, aus der initialen Kandidatenliste entfernt.
4. *Short words*: Mit der Aktivierung dieses Filters werden alle Wörter, welche aus weniger als vier Buchstaben bestehen, aus der initialen Kandidatenliste entfernt.
5. *Email Adress and Department Names*: Mit der Aktivierung dieses Filters werden alle Abteilungsnamen und Email-Adressen aus der initialen Kandidatenliste entfernt.

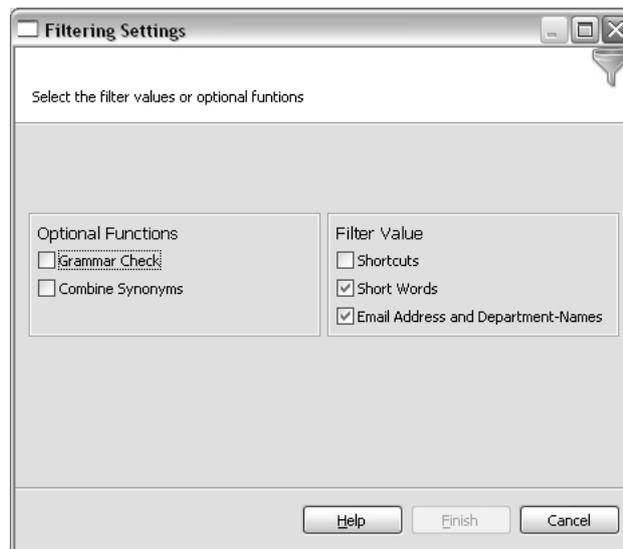


Abbildung 16: MIA-Tool – Festlegung der Filterwerte

Die Ergebnisse der Merkmalsidentifizierung werden dem Benutzer in einem Fenster präsentiert, welches aus folgenden drei Teilen besteht (siehe Abbildung 17):

- (A) *Information*: In diesem Abschnitt wird dem Benutzer angezeigt wie viele Merkmalskandidaten identifiziert wurden und wie viele er davon ausgewählt hat. Zusätzlich werden die Informationen über die Anzahl der normalisierten Wörter, Abkürzungen, Synonyme angezeigt.
- (B) *Feature Candidates*: In diesem Abschnitt werden alle identifizierten Merkmalskandidaten aufgelistet und zwar zuerst die, die normalisiert wurden und danach diejenigen, die nicht normalisiert werden konnten.
- (C) *Features*: In diesem Abschnitt werden alle vom Benutzer ausgewählten Merkmale dokumentiert.

Der Benutzer kann die Merkmale aus der Merkmalskandidaten-Liste einfach in der Merkmalsliste übernehmen oder bei Bedarf zuerst die Rechtschreibfehler korrigieren und dann übernehmen. Die falschen Übernahme-Entscheidungen können jederzeit gelöscht werden. Falls in der Merkmalskandidaten-Liste der Benutzer die neuen Einträge für das Wörterbuch, die Stoppwort- oder Synonymliste findet, kann er diese Einträge jederzeit in das Wörterbuch, die Stoppwort- oder Synonymliste übernehmen. Die vom Benutzer identifizierten Merkmale können dann mit der Export-Funktion in eine Excel-Datei übertragen werden.

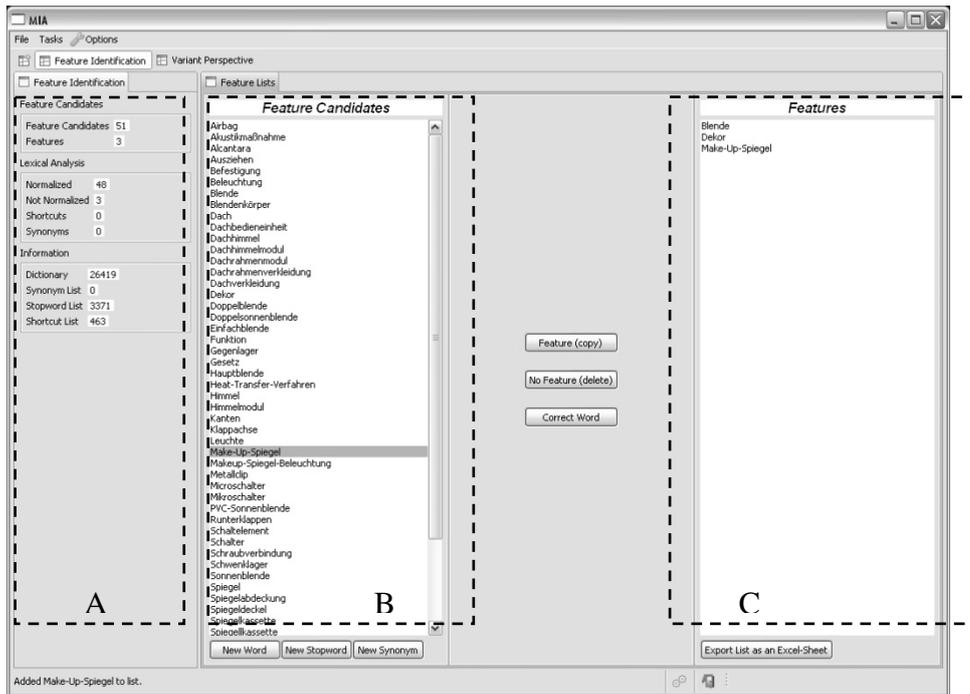


Abbildung 17: MIA-Tool

4.4 Evaluation des Algorithmus

Die Evaluierung von MIA wurde anhand mehrerer Anforderungsdokumente der Daimler AG durchgeführt. Für diese Arbeit wurden die Evaluationsergebnisse von vier Datensätzen ausgewertet und vorbereitet. Tabelle 7 stellt diese vier Datensätze vor.

Tabelle 7: Evaluierungsdatensätze

Anzahl von	DS1	DS2	DS3	DS4
DOORS-Objekten	1397	727	1329	501
Anforderungen	1124	543	712	274
Variable Anforderungen	1001	37	-	475

Der erste Datensatz (DS1) repräsentiert das Anforderungsdokument für die Komponenten der Säulenverkleidung. Dieses Anforderungsdokument dokumentiert sowohl die Anforderungen an unterschiedliche Säulen (z.B. A-Säule, B-Säule) und deren Varianten (z.B. B-Säule mit Kleiderhaken und B-Säule ohne Kleiderhaken) als auch die Anforderungen an die verschiedenen Baureihen. Das Anforderungsdokument besteht aus 1397 DOORS-Objekten, davon sind 1124 Anforderungen an die Komponente.

Der zweite Datensatz (DS2) stellt das Anforderungsdokument für die Komponente Sonnenblende dar. Dieses Anforderungsdokument dokumentiert die Anforderungen an die unterschiedlichen Varianten der Komponente für die verschiedenen Baureihen. Das Anforderungsdokument besteht aus 727 DOORS-Objekten, davon sind 543 Anforderungen an die Komponente.

Der dritte Datensatz (DS3) stellt das Anforderungsdokument für die Komponente Federung dar. Dieses Anforderungsdokument dokumentiert die Anforderungen an die unterschiedlichen Varianten der Komponente für die verschiedenen Baureihen. Das Anforderungsdokument besteht aus 1329 DOORS-Objekten, davon sind 712 Anforderungen an die Komponente.

Der vierte Datensatz (DS4) bildet das Anforderungsdokument für die Werkstoffe im Fahrzeuginterieur ab. Dieses Anforderungsdokument dokumentiert die Werkstoff-Anforderungen sowohl an unterschiedliche Komponenten (z.B. Hutablage, Handschuhfach) und deren Varianten (z.B. die Serienausstattung oder die Ausstattungslinie Avantgard) als auch an die verschiedenen Baureihen. Das Anforderungsdokument besteht aus 501 DOORS-Objekten, davon sind 274 Anforderungen.

Die Anforderungen haben im Ausgangszustand in keinem der Datensätze eine Kennzeichnung, ob es eine variable oder eine gemeinsame Anforderung ist. Anhand der Zuordnung zu den einzelnen Komponenten und Baureihen wurden die variablen Anforderungen mit Hilfe eines manuellen Reviews in den Datensätzen DS1, DS2 und DS4 identifiziert.

Außerdem wurde für jeden Datensatz von einem Domänenexperten ein Variabilitätsmodell erstellt, das im Rahmen dieser Arbeit als Referenz für die Beurteilung der Ergebnisse verwendet wird. Es wurden nur die Variabilitätskriterien (variablen Merkmale) dokumentiert. Die variablen Merkmale sind optionale, alternative oder auch obligatorische Merkmale, deren Kinder-Merkmale optionale oder alternative Merkmale sind. Die Anzahl der Merkmale in dem entsprechenden Merkmalsmodell für jeden Datensatz sind in der Tabelle 8 dokumentiert.

Für die Bewertung der Ergebnisse von MIA werden die folgenden zwei Metriken [BYRN06, CHSR07] verwendet:

Definition 3-6: *Trefferquote (engl. Recall)*

Trefferquote gibt an wie hoch der Anteil der zurückgelieferten relevanten Merkmale im Vergleich zu der Gesamtanzahl der relevanten Merkmale ist.

$$R = \frac{\text{Anzahl der Merkmalskandidaten} \cap \text{Anzahl der Merkmale}}{\text{Anzahl der Merkmale}}$$

Definition 3-6: *Genauigkeit (engl. Precision)*

Genauigkeit gibt an wie hoch der Anteil relevanter Merkmale im Vergleich zu der Gesamtanzahl der zurückgegebenen Merkmale ist.

$$P = \frac{\text{Anzahl der Merkmalskandidaten} \cap \text{Anzahl der Merkmale}}{\text{Anzahl der Merkmalskandidaten}}$$

4.4.1 Phase I

In der ersten Phase wurden die zuvor vorgestellten vier Datensätze mit Hilfe des MIA-Algorithmus analysiert. Während der Analyse wurden das anfangs erstellte Wörterbuch und die Stoppwortliste verwendet. Es wurden zwei Durchläufe durchgeführt. Beim ersten Durchlauf wurden alle Anforderungen der Datensätze analysiert. Im zweiten Durchlauf wurden nur die variablen Anforderungen aus den Datensätzen analysiert.

4.4.1.1 Erster Durchlauf

Im ersten Schritt wurde für alle Datensätze DS1 - DS4 die Merkmalskandidaten-Liste anhand der Analyse aller Anforderungen des Datensatzes erstellt. Tabelle 8 präsentiert die Ergebnisse für die Datensätze DS1 und DS2. In dem Anhang II-A sind die Ergebnisse für alle vier Datensätze dokumentiert.

Tabelle 8: Erster Durchlauf in der Phase I

	DS1				DS2			
	Man.	MIA	R	P	Man.	MIA	R	P
Anzahl Anforderungen	1124	1124			543	543		
Dauer in Minuten		25				36		
Anzahl von Merkmalskandidaten		241				380		
Anzahl von variablen Merkmalen	98	44	0,45	0,18	9	8	0,89	0,02
Anzahl von Variabilitätskriterien	15	8	0,53	0,03	4	3	0,75	0,01
Anzahl von AVK	83	36	0,43	0,15	5	5	1,00	0,01

Die beste Trefferquote (89%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (19%) war für DS4. Die höchste Genauigkeit (18%) erzielte MIA-Algorithmus bei DS1, bei den drei anderen Datensätzen lag die Genauigkeit nur bei 2%.

Unterschiedlich sind auch die Ergebnisse für die Variabilitätskriterien (VK) und für die Variabilitätskriterien-Ausprägungen (AVK):

- VK: Die beste Trefferquote (75%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (31%) war für DS1. Die höchste Genauigkeit von 3% erzielte MIA-Algorithmus für DS1, bei den drei anderen Datensätzen lag die Genauigkeit nur bei 1%.
- AVK: Die beste Trefferquote (100%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (4%) war für DS3. Die höchste Genauigkeit (15%) erzielte MIA-Algorithmus für DS1, bei den drei anderen Datensätzen lag die Genauigkeit nur bei 1%.

4.4.1.2 Zweiter Durchlauf

Die ersten Ergebnisse haben gezeigt, dass die Trefferquote groß, und die Genauigkeit sehr niedrig sind. Deswegen wurde entschieden, dass beim zweiten Durchlauf nur die variablen Anforderungen analysiert werden. Es wurden drei Datensätze, DS1, DS2 und DS4, analysiert. Der Datensatz DS3 wurde nicht analysiert, da das Wissen über den Typ der Anforderung (variabel oder gemeinsam) fehlte. Diese Ergebnisse sind vollständig im Anhang II-B dokumentiert. Tabelle 9 bietet einen Ausschnitt davon.

Die beste Trefferquote (89%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (19%) war für DS4. Die höchste Genauigkeit (22%) erzielte MIA-Algorithmus für DS1. Die niedrigste Genauigkeit von 2% erzielte MIA für DS4.

Auch in diesem Fall sind die Ergebnisse für die VK und AVK unterschiedlich:

- VK: Die beste Trefferquote (75%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (21%) war für DS4. Die höchste Genauigkeit von 3% erzielte MIA-Algorithmus für DS1, bei den drei anderen Datensätzen lag die Genauigkeit nur bei 1%.

- AVK: Die beste Trefferquote (100%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (18%) war für DS4. Die höchste Genauigkeit (19%) erzielte MIA-Algorithmus für DS1, für DS4 lag die Genauigkeit nur bei 2%.

Tabelle 9: Zweiter Durchlauf in der Phase I

	DS1*				DS2*			
	Man.	MIA	R	P	Man.	MIA	R	P
Anzahl Anforderungen	1012	1012			56	56		
Dauer in Minuten		16				1		
Anzahl von Merkmalskandidaten		178				48		
Anzahl von variablen Merkmalen	98	40	0,41	0,22	9	8	0,89	0,17
Anzahl von VK	15	6	0,40	0,15	4	3	0,75	0,06
Anzahl von AVK	83	34	0,41	0,19	5	5	1,00	0,10

4.4.1.3 Diskussion der Ergebnisse der Phase I

Die Ergebnisse von beiden Durchläufen wurden miteinander verglichen und analysiert.

Bei DS1 sank die Trefferquote im zweiten Durchlauf um 4%. Das zeigt, dass einige Merkmale in den gemeinsamen Anforderungen dokumentiert sind. Die Genauigkeit stieg um 4%. Die Werte für die Merkmale mit und ohne Kinder zeigten gleiches Verhalten.

Bei DS2 änderte sich die Trefferquote nicht. Die Genauigkeit stieg um 15%. Die Werte für die Merkmale mit und ohne Kinder zeigten gleiches Verhalten.

Bei DS4 haben sich die beiden Werte nicht geändert.

Es wurde analysiert, welche Merkmale nicht gefunden wurden:

- In erster Linie können die impliziten Merkmale nicht gefunden werden. Die impliziten Merkmale sind in den meisten Fällen die Eltern-Merkmale und werden oft nicht in den Anforderungen dokumentiert, weil die Anforderungen die einzelnen Merkmale (z.B. einfache und doppelte Sonnenblende) beschreiben und nicht die Eltern-Merkmale (z.B. Ausführung).
- Auch wurden die Merkmale nicht gefunden, welche durch Abkürzungen und/oder Zahlen dokumentiert sind. Zum Beispiel sind in DS1 viele Blatt-Merkmale durch die Stücklisten-Nummern im Anforderungsdokument dokumentiert.

Im zweiten Schritt wurden die Ergebnisse der Metrik Genauigkeit analysiert. Dabei wurden folgende Gründe für die hohe Anzahl von Merkmalskandidaten identifiziert:

- In manchen Datensätzen wurde sowohl die neue als auch die alte Rechtschreibung verwendet.
- In den Anforderungsdokumenten werden viele Synonyme verwendet.
- In den Merkmalskandidaten wurden viele Stopp-Wörter identifiziert.
- Außerdem werden auch immer die notwendigen Merkmale identifiziert, welche allerdings im Rahmen des R2F-Ansatzes nicht benötigt werden und somit nicht als gefundene Merkmale in die Evaluationsstatistik aufgenommen wurden.

4.4.2 Phase 2

Als Vorbereitung zur Durchführung der zweiten Phase wurden folgende Optimierungen durchgeführt:

- Alle Datensätze wurden auf die neue Rechtschreibungsform umgestellt.
- Das Wörterbuch und die Stopwort-Liste wurden erweitert.
- In dem Werkzeug MIA wurde die Funktion „Synonyme zusammenfassen“ implementiert. Diese Funktion fasst alle Merkmalskandidaten zusammen, welche in der Synonym-Liste als Synonyme dokumentiert sind.

4.4.2.1 Erster Durchlauf

In dem ersten Durchlauf der zweiten Phase wurden analog zu dem ersten Durchlauf der ersten Phase alle Anforderungen in den Datensätzen analysiert. Die Ergebnisse von diesem Durchlauf sind im Anhang II-C dokumentiert. Tabelle 10 stellt die Ergebnisse für die Datensätze DS1 und DS2 dar.

Tabelle 10: Erster Durchlauf in der Phase II

	DS1				DS2			
	Man.	MIA	R	P	Man.	MIA	R	P
Anzahl Anforderungen	1124	1124			543	543		
Dauer in Minuten		20				35		
Anzahl von Merkmalskandidaten		201				205		
Anzahl von variablen Merkmalen	98	44	0,45	0,21	9	8	0,89	0,04
Anzahl von VK	15	8	0,53	0,04	4	3	0,75	0,01
Anzahl von AVK	83	36	0,43	0,17	5	5	1,00	0,02

Die beste Trefferquote (89%) erzielte der MIA-Algorithmus für DS2, die schlechteste Trefferquote (31%) war für DS4. Die höchste Genauigkeit (22%) erzielte MIA-Algorithmus für DS1. Die niedrigste Genauigkeit von 3% erzielte MIA für DS4.

Die Ergebnisse für die VK und AVK sind Folgende:

- VK: Die beste Trefferquote (75%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (21%) war für DS4. Die höchste Genauigkeit von 4% erzielte MIA-Algorithmus für DS1, bei den drei anderen Datensätzen lag die Genauigkeit bei 1%.
- AVK: Die beste Trefferquote (100%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (36%) war für DS4. Die höchste Genauigkeit (17%) erzielte MIA-Algorithmus für DS1, während bei DS2 und DS4 lag die Genauigkeit nur bei 2%.

4.4.2.2 Zweiter Durchlauf

Im zweiten Durchlauf der zweiten Phase wurden identisch zu dem zweiten Durchlauf der ersten Phase nur die variablen Anforderungen in den Datensätzen DS1, DS2 und DS4 analysiert. Die Ergebnisse von diesem Durchlauf sind in dem Anhang II-D dokumentiert.

Tabelle 11 stellt die Ergebnisse für Datensätze DS1 und DS2 dar. Die beste Trefferquote (89%) erzielte MIA-Algorithmus für DS2, die schlechteste Trefferquote (31%) war für

DS4. Die höchste Genauigkeit (30%) erzielte MIA-Algorithmus für DS2. Die niedrigste Genauigkeit von 6% erzielte MIA für DS4.

Tabelle 11: Zweiter Durchlauf in der Phase II

	DS1*				DS2*			
	Manuell	MIA	R	P	Manuell	MIA	R	P
Anzahl Anforderungen	1012	1012			56	56		
Dauer in Minuten		17				1		
Anzahl von Merkmalskandidaten		164				40		
Anzahl von variablen Merkmalen	98	50	0,51	0,30	9	8	0,89	0,20
Anzahl von VK	15	11	0,73	0,22	4	3	0,75	0,08
Anzahl von AVK	83	39	0,47	0,24	5	5	1,00	0,13

Die Ergebnisse für die VK und AVK sind folgende:

- Merkmale mit Kindern: Die beste Trefferquote (75%) erzielte der MIA-Algorithmus für DS2, die schlechteste Trefferquote (21%) war für DS4. Die höchste Genauigkeit von 22% erzielte MIA-Algorithmus für DS1, für DS4 lag die Genauigkeit bei 1%.
- Merkmale ohne Kindern: Die beste Trefferquote (100%) erzielte der MIA-Algorithmus für DS2, die schlechteste Trefferquote (36%) war für DS4. Die höchste Genauigkeit (24%) erzielte der MIA-Algorithmus für DS1, für DS4 lag die Genauigkeit nur bei 5%.

4.4.2.3 Diskussion der Ergebnisse der Phase II

Die Ergebnisse der zweiten Phase zeigen, dass durch die Optimierungen eine Verbesserung der Genauigkeit erreicht werden kann, ohne dass dabei die Trefferquote negativ beeinflusst wird.

Die besten Ergebnisse zeigt MIA bei der Anwendung auf ausschließlich variable Anforderungen. In diesem Fall liefert MIA die besten Werte der Metrik Genauigkeit. Um die Trefferquote zu steigern ist es empfehlenswert, auch die Überschriften des Anforderungsdokumentes in die Analyse einzubeziehen. Der Grund dafür ist, dass die impliziten Merkmale, falls sie dokumentiert sind, typischerweise in den Überschriften zu finden sind.

Bei der Analyse von ausschließlich variablen Anforderungen ist der niedrigste Wert der Trefferquote 31% für DS3, der höchste Wert liegt bei 89% für DS2. Der niedrigste Wert der Genauigkeit liegt bei 6% für DS4 und der höchste bei 30% für DS2.

Hayes et al. [HaDS05] haben eine Klassifikation für die Trefferquote und Genauigkeit von semi-automatischen Ansätzen vorgestellt. Tabelle 12 stellt diese Klassifikation dar. Nach dieser Klassifikation gehört MIA bei der Trefferquote in die Gruppen von "akzeptabel" bis "exzellent". Bei der Genauigkeit kann MIA nicht immer in die Gruppe "akzeptabel" eingestuft werden.

Die Grundvoraussetzung für die Anwendung des vorgestellten Ansatzes ist, dass der Entwicklungs- und Spezifikationsprozess nicht auf den Konzepten der Produktlinien-Entwicklung aufbaut. Weitere Bedingungen für die Anwendung von MIA sind:

- Die Anforderungen müssen in einer natürlichen Sprache dokumentiert werden.

- Die Variabilität in den Anforderungen muss durch Wörter und nicht durch Zahlen (z.B. 10mm, 15mm) beschrieben werden.

Tabelle 12: Klassifikation für die Trefferquote und Genauigkeit nach [HaDS05]

Metrik	Akzeptabel	Gut	Exzellent
Trefferquote	> 60%	> 70%	> 80%
Genauigkeit	> 20%	> 30%	> 50%

Die Dokumentation des Typs der Anforderung (gemeinsam oder variabel) kann die Anzahl von Merkmalskandidaten reduzieren, weil nur die variablen Anforderungen die variablen Eigenschaften im Text beinhalten können.

Unabhängig von allen Verbesserungen, die noch im Rahmen des MIA-Algorithmus möglich sind (z.B. Gruppierung von Wörtern mit dem gleichen Stamm), werden immer mehr Kandidaten vorgeschlagen werden als es tatsächliche Merkmale gibt. Ein Grund dafür sind die notwendigen Merkmale, welchen in den Anforderungen dokumentiert sind aber im Rahmen der Variabilitätsmodellierung nach dem R2F-Ansatz nicht betrachtet werden.

Ein größeres Problem stellen die fehlenden Merkmale in der Merkmalskandidatenliste dar. Es können implizite Merkmale aber auch Merkmale sein, die durch die Zahlen, Abkürzungen, Adjektive oder Kombinationen aus Adjektiven und Substantiven sein. Eine mögliche Verbesserung ist die Berücksichtigung von numerischen Werten (z.B. Stücklistennummern).

4.5 Diskussion

Während der Literaturrecherche wurde eine Reihe Forschungsarbeiten gefunden, die sich mit dem Thema der Identifikation von Abstraktionstermen in natürlichsprachlichen Dokumenten beschäftigt haben. Im nächsten Abschnitt werden diese Arbeiten kurz vorgestellt. Im abschließenden Kapitel werden die Vor- und Nachteile des vorgestellten MIA-Ansatzes diskutiert.

4.5.1 Alternative Ansätze

Es existieren mehrere Ansätze, welche die Methoden der lexikalischen Analyse benutzen um Anforderungsdokumente zu analysieren bzw. zu strukturieren. Diese Ansätze können in drei Gruppen unterteilt werden:

1. Identifizierung von Abstraktionstermen in Anforderungsdokumenten,
2. Identifizierung von Abstraktionstermen in Produktbeschreibungen und Dokumentationen,
3. Identifikation von Abstraktionstermen und Beziehungen in einem natürlichsprachlichen Dokument.

Identifizierung von Abstraktionstermen in Anforderungsdokumenten

Eine Reihe von Ansätzen benutzt die lexikalischen Methoden, um die Abstraktionsterme in den Anforderungsdokumenten zu identifizieren. Die Gemeinsamkeit von allen Ansät-

zen aus dieser Gruppe ist, dass sie alle ein Anforderungsdokument als Menge von Zeichen oder Wörter betrachten.

Die ersten Ansätze Berry et al. [BeYY87] und Smeaton [Smea95] haben grammatikalische Parser benutzt um bestimmte Wortarten zu identifizieren.

Aguilera [Agui87, AgBe90] hat vorgeschlagen solche Phrasen als Abstraktionsterme zu identifizieren, die am häufigsten im Text vorkommen.

Der Ansatz von Maarek [MaBe88] benutzt die Methode lexikalische Ähnlichkeit und identifiziert solche Wort-Paare als Abstraktionsterme, welche am häufigsten zusammen in den Sätzen vorkommen.

Palmer und Liang [PaLi92] haben sich während der Erarbeitung eines Ansatzes für die automatisierte Anforderungsanalyse mit Indexierung und Clustering von Anforderungen beschäftigt. Für die Indexierung von Anforderungen haben sie die Verben und spezifische Substantive als Indikatoren für Anforderungsklassifikation vorgeschlagen. Sie haben ein Verben-Lexikon erstellt. Für die Klassifizierung haben sie die Information Retrieval (IR)-Methoden (z.B. Zip's Gesetz) angepasst.

Goldberg und Berry [GoBe94] betrachten mit AbstFinder jeden Satz als einfache Zeichenfolge. Alle Zeichenfolgen werden paarweise verglichen um identische Teil-Zeichenfolgen zu finden. Die gefundenen Teil-Zeichenfolgen repräsentieren die Abstraktionsterme für das analysierte Anforderungsdokument. Der Vorteil von diesem Ansatz ist, dass nicht die Terme aus einzelnen Wörtern, sondern auch die Terme aus mehreren Wörtern gefunden werden könnten. Der Nachteil von diesem Ansatz ist, dass in der deutschen Sprache (ohne Normalisierung) viele Terme nicht als identische Terme identifiziert werden wegen der Flexion.

Der Ansatz von Lecoeuche [Leco00] vergleicht die Häufigkeit des Vorkommens eines Abstraktionsbegriffs in einem Basisdokument mit der Häufigkeit des Vorkommens im analysierten Anforderungsdokument. Danach werden beide Werte in Verbindung gesetzt. Der berechnete Korrelationskoeffizient zeigt, ob es sich um einen Abstraktionsbegriff für das Anforderungsdokument handelt oder nicht.

Gacitua et al. [GaSC10] haben einen Ansatz zur Identifikation der Abstraktionsterme (aus einem oder mehreren Wörtern) aus Anforderungsartefakten präsentiert. Dieser Ansatz berücksichtigt die Häufigkeit des Vorkommens der Wörter in einem Korpus. Zusätzlich werden syntaktische Muster angewendet um auch die Terme aus mehreren Wörtern zu identifizieren. Dieser Ansatz wurde für die englische Sprache konzipiert und anhand des normativen Korpus für die englische Sprache evaluiert. Für die deutsche Sprache existiert ein entsprechendes Korpus zurzeit nicht.

Identifikation von Abstraktionstermen in Produktbeschreibungen und Dokumentationen

Eine Reihe der Arbeiten beschäftigt sich mit der Identifizierung von Produktfeatures in Produktbeschreibungen oder Kundenreviews im Bereich E-Commerce.

Hu und Liu [HuLi04] benutzen den NL Processor Linguistic Parser um die Produktmerkmale in Produktbeschreibungen zu identifizieren. Unter Produktmerkmalen werden dabei die am häufigsten verwendeten Substantive verstanden.

Popescu und Etzioni [PoEt05] benutzen eine ähnliche Vorgehensweise um die Produktmerkmale mithilfe von KnowItAll Web Information Extraction System zu identifizieren. Das KnowItAll Web Information Extraction System befasst sich mit der Herausforderung der Automatisierung der Suche im Internet.

Identifikation von Abstraktionstermen und Beziehungen in einem natürlichsprachlichen Dokument

Der Ansatz [Abbo83] wurde entwickelt um die Datentypen, Objekte, Operatoren und Kontrollstrategien aus einem englischen Text abzuleiten. Im Rahmen dieses Ansatzes wurde eine Anleitung erstellt, wie man ein Anforderungsdokument manuell mit Hilfe der Methoden der lexikalischen Analyse in einen Quellcode transformieren kann.

Chen [Chen83] hat einen Ansatz erarbeitet, um natürlichsprachlichen Text in ein Entity-Relationship-Diagramm umzuwandeln.

Chen et.al. [ChZM05] haben einen semi-automatischen Ansatz zur Erstellung von Merkmalsmodellen auf Basis der existierenden Anforderungsdokumente vorgestellt. Der Ansatz basiert auf den Techniken des Text-Clusterings. Die Vorgehensweise beschreibt wie ausgehend von mehreren Produktspezifikationen zuerst die Produkt-Merkmalsmodelle und danach Produktlinien-Merkmalsmodelle erstellt werden könnten. Der vorgestellte Ansatz besteht aus drei Schritten: (i) Identifikation der funktionalen Anforderungen, (ii) Modellierung der einzelnen Anforderungen und Beziehungen zwischen diesen in einem ungerichteten Graphen, (iii) Identifikation von Merkmalen und Gruppierung von Anforderungen zu den Merkmalen. Die Grundlage für den Clustering-Ansatz ist die Gewichtung einzelner Beziehungen zwischen einzelnen Anforderungen. Dieser Ansatz kann bei großen industriellen Spezifikationen nicht angewendet werden, weil die Voraussetzung (die Beziehungen zwischen einzelnen Anforderungen müssen dokumentiert werden) nicht gegeben ist.

4.5.2 Abgrenzung zu alternativen Ansätzen

In der vorliegenden Arbeit wurden die Methoden der lexikalischen Analysis verwendet um die Merkmalskandidaten aus natürlichsprachlichen Dokumenten zu identifizieren, welche in der deutschen Sprache erstellt wurden.

Dieser Ansatz wurde entwickelt für die einmalige Identifikation von Merkmalen aus bestehenden natürlichsprachlichen Anforderungsdokumenten. Wie bereits in 4.4.2.3 erwähnt wurde, sind die ersten Ergebnisse der Merkmalsidentifikation von MIA nach [HaDS05] im Bereich "akzeptabel" bis "exzellent" für die Trefferquote und nicht immer im Bereich "akzeptabel" bei der Genauigkeit.

Im Vergleich mit vorgestellten alternativen Ansätzen sind die Ergebnisse der Trefferquote von MIA mindestens so gut oder sogar besser als von Gacitua et al. [GaSC10]. Die Ergebnisse der Genauigkeit können durch die genannten Weiterentwicklungsmaßnahmen noch weiter verbessert werden.

Der Nachteil des vorgestellten Ansatzes ist, dass im aktuellen Stand nur Abstraktionsterme aus einem Wort identifiziert werden können. Es wäre möglich, die syntaktischen Muster zu benutzen um Abstraktionsterme aus mehreren Wörtern zu identifizieren. Allerdings muss damit gerechnet werden, dass die Ergebnisse der Genauigkeit noch weiter verschlechtert werden ohne große Verbesserungen bei der Trefferquote zu gewinnen. Der Grund für den Rückgang der Genauigkeit liegt in der steigenden Anzahl der betrachteten Merkmalskandidaten durch die Berücksichtigung von Kombinationen aus Adjektiven und Substantiven. Ob dabei die Genauigkeit steigt ist fraglich, weil in den meisten betrachteten Merkmalmodellen Merkmale durch ein Substantiv beschrieben wurden.

Im Kapitel 10.1 werden die Ergebnisse der Evaluation von MIA-Anwendung in der praxisnahen Fallstudien vorgestellt und diskutiert.

Kapitel 5

Formalisierung

Das vorliegende Kapitel präsentiert die formale Darstellung der Anforderungsdokumente (5.2) und die formale Darstellung der Merkmalsmodelle (5.3). Diese formalen Darstellungen sind notwendig, um die Ansätze und Prüfmethoden, die in den weiteren Kapiteln vorgestellt werden, beschreiben und evaluieren zu können. Im abschließenden Teil dieses Kapitels werden alternative formale Darstellungen für die Merkmalsmodelle diskutiert.

5.1 Vorteile der formalen Darstellung

Die grafische und textuelle Darstellung von Merkmalsmodellen wurde im Kapitel 2.2.2 vorgestellt. Diese Darstellung reicht nicht aus, um die in dieser Arbeit entwickelten Ansätze korrekt und eindeutig beschreiben zu können. Deswegen werden in diesem Kapitel sowohl die Anforderungsdokumente als auch die Merkmalsmodelle formalisiert.

Die formale Darstellung von Anforderungsdokumenten existiert in der Forschung auf der Ebene der formalen Dokumentation von einzelnen Anforderungen. Im Rahmen dieser Arbeit wird die formale Darstellung von einem gesamten Anforderungsdokument benötigt, unabhängig davon welchen Formalisierungsgrad einzelne Anforderungen haben. Deswegen wird im Rahmen dieses Kapitels ein Vorschlag zur Formalisierung der natürlichsprachlichen Anforderungsdokumente gemacht (5.2).

Die Formalisierung von Merkmalsmodellen ist notwendig zur eindeutigen Beschreibung von Variabilitätsinformationen und zur Sicherstellung, dass nur die zulässigen Konfigurationen möglich sind. In der Forschung existieren mehrere Formalisierungsansätze für Merkmalsmodelle. Diese Formalisierungsansätze unterscheiden sich sowohl in den mathematischen Theorien (z.B. logische Operatoren, aussagenlogische Formeln, formale Sprachen), als auch in den Zwecken, wofür die Merkmalsmodelle formalisiert wurden.

Eine der ersten Formalisierungen für Merkmalsmodelle wurde von Batory [Bato05] vorgestellt. Seine Formalisierung basiert auf einer iterativen Grammatik und Formeln der Aussagenlogik. Dabei werden für die Verbindung von einzelnen Variablen die folgenden fünf Konnektoren benutzt: die Konjunktion, die Disjunktion, die Negation, die Implikation und die Äquivalenz.

Schobbens et al. [SHTB07] haben in ihrer Arbeit eine generische Semantik für Merkmalsmodelle erarbeitet. Dafür haben sie sowohl die originalen FODA-Merkmalsmodelle,

als auch mehrere Weiterentwicklungen (FORM [KKLK98], FeatuRSEB [GrFA98], van-Grup et al. [vGBS01], Generative Programming [CzEi00], Riebisch et al. [RiSP04], PLUSS [ErBB05]) formalisiert. Das Ziel ihrer Arbeit war, die unterschiedlichen Merkmalsprachen vergleichbar zu machen, um die Auswahl eines Ansatzes zu unterstützen. Für jeden Modellierungsansatz wurden folgende Informationen ausgewertet und formalisiert: (i) Typ des Merkmalsdiagramms (Baum vs. Graph), (ii) Typen der Merkmalsknoten, (iii) grafische Abhängigkeitsbeziehungen und (iv) textuelle Bedingungen.

Reiser [Reis08] hat im Rahmen seiner Dissertation eine weitere Formalisierung (Io/VM) von Merkmalsmodellen vorgeschlagen, um seine zwei Ansätze beschreiben und evaluieren zu können. Er hat sowohl die FODA-Merkmalsmodelle als auch die Erweiterungen (z.B. Kardinalität von Merkmalen, Parametrisierung von Merkmalen) formalisiert.

Für die vorliegende Arbeit wird ein Formalisierungsansatz benötigt, welcher die Merkmalsmodelle und Anforderungsdokumente sowie die Beziehungen zwischen diesen zwei Artefakt-Typen formal beschreiben lässt. Da keiner der analysierten Ansätze zur formalen Darstellung von Merkmalsmodellen dies erlaubt, wurde im Rahmen dieser Arbeit eine eigenständige Formalisierung der Merkmalsmodelle auf Basis der Arbeit von Reiser [Reis08] und Anforderungsdokumenten erarbeitet.

5.2 Formale Darstellung der Anforderungsdokumente

In diesem Abschnitt werden die Anforderungsdokumente und deren Bestandteile formal spezifiziert.

Eine Anforderung a besteht aus mehreren Wörtern w . Natürlich ist die Reihenfolge der Wörter in einer Anforderung wichtig. Aber für die Konzepte, welche im Rahmen dieser Arbeit präsentiert werden, spielt sowohl die Reihenfolge als auch die Anzahl der mehrmals vorkommenden Wörter keine Rolle. Deswegen wird eine Anforderung a als eine Wortmenge definiert.

$$a := \{w_i \mid i = 1, \dots, n\} \quad (5-1)$$

Eine Anforderung a kann eine allgemeine Anforderung oder eine variable Anforderung sein. Eine allgemeine Anforderung a^{com} ist eine Anforderung, die für alle Produkte gültig ist. Eine variable Anforderung a^{var} ist eine Anforderung, die für ein oder mehrere Produkte (aber nicht für alle Produkte) gültig ist.

Alle allgemeinen Anforderungen a^{com} aus einer Anforderungsbibliothek $ABib$ bilden eine Teilmenge A^{com} von Anforderungen, welche in jeder Konfiguration dieser Anforderungsbibliothek vorkommt.

$$A^{com} := \{a^{com} \mid a^{com} \text{ ist allgemeine Anforderung} \wedge a^{com} \in ABib\} \quad (5-2)$$

Alle variablen Anforderungen a^{var} aus der Anforderungsbibliothek $ABib$ bilden eine Teilmenge A^{var} von Anforderungen, welche in einer oder mehrerer Konfigurationen dieser Anforderungsbibliothek vorkommen können.

$$A^{var} := \{a^{var} \mid a^{var} \text{ ist variable Anforderung} \wedge a^{var} \in ABib\} \quad (5-3)$$

Alle Anforderungen an alle Produkte einer Produktlinie werden in einer Anforderungsbibliothek $ABib$ dokumentiert und verwaltet. Eine Anforderung in der Anforderungsbibliothek ist entweder eine allgemeine oder eine variable Anforderung:

$$ABib := A^{com} \cup A^{var} \quad (5-4)$$

$$A^{com} \cap A^{var} = \emptyset \quad (5-5)$$

Eine Konfiguration der Anforderungsbibliothek $Conf(ABib)$ für ein Produkt P beinhaltet alle gemeinsamen Anforderungen und eine Auswahl der variablen Anforderungen, welche für das ausgewählte Produkt gültig sind.

$$Conf(ABib)^P := A^{com} \cup \{a \mid a \in A^{var} \wedge a \text{ gültig für } P\} \quad (5-6)$$

5.3 Formale Darstellung eines Variabilitätsmodells

In Kapitel 2.2.3 wurden die Bestandteile und die Notationsregeln für die Merkmalsmodelle für die vorliegende Arbeit festgelegt. In Tabelle 13 sind alle Elemente eines Merkmalsmodells nochmal zusammengefasst.

Tabelle 13: Elemente eines Merkmalsmodells

Element Bezeichnung	Beschreibung
Wurzel	Die Wurzel des Merkmalsmodells ist das oberste Element in dem Merkmalsmodell.
notwendiges Merkmal	Ein notwendiges Merkmal muss immer ausgewählt werden, sobald das übergeordnete Merkmal ausgewählt ist.
optionales Merkmal	Ein optionales Merkmal kann ausgewählt werden, sobald das übergeordnete Merkmal ausgewählt ist.
auswählbare Merkmale	Auswählbare Merkmale sind eine Gruppe von Merkmalen, aus welcher eine durch Kardinalität bestimmte Anzahl von Merkmalen ausgewählt werden kann, sobald das übergeordnete Merkmal ausgewählt ist.
Kardinalität	Die Kardinalität bestimmt die minimale und die maximale Anzahl der auswählbaren Merkmale, welche aus der Gruppe von auswählbaren Merkmalen ausgewählt werden kann, sobald das übergeordnete Merkmal ausgewählt ist.
alternative Merkmale	Alternative Merkmale sind eine Gruppe von Merkmalen, aus welcher nur ein Merkmal ausgewählt werden kann, sobald das übergeordnete Merkmal ausgewählt ist.
hierarchische Beziehung	Die hierarchische Beziehung zwischen zwei Merkmalen beschreibt die Eltern-Kind-Beziehung. Das Kind-Merkmal kann nur ausgewählt werden, wenn sein Eltern-Merkmal ausgewählt wurde.
„benötigt“-Beziehung	a „benötigt“ b besagt, dass Merkmal a nur ausgewählt werden kann, wenn das „benötigte“ Merkmal b ausgewählt wurde.
„schließt_aus“-Beziehung	„schließt_aus“-Beziehung beschreibt die bidirektionalen Beziehungen zwischen Merkmalen, die sich gegenseitig ausschließen, und deshalb nicht gleichzeitig für eine Konfiguration ausgewählt werden können.

Wie bereits definiert, besteht ein merkmalsbasiertes Variabilitätsmodell VM aus einer Wurzel w , einer Menge M von Merkmalen und einer Menge R von Beziehungen zwischen Merkmalen.

$$VM := (w, M, R) \quad (5-7)$$

Die Merkmale m in der Menge M können vom Typ notwendig m^m , optional m^o , alternativ m^{xor} oder auswählbar m^{or} sein. Die zulässige Anzahl der auswählbaren Merkmale eines Variabilitätsmodells wird mithilfe von Kardinalität $Card^{or}$ dokumentiert.

$$Card^{or} = [min, max] \quad (5-8)$$

mit $min \geq 0, max \geq 1, max \geq min, min, max \in \mathbb{N}$

Jede Beziehung r kann unabhängig von ihrem jeweiligen Typ durch das Anfangsmerkmal $\alpha(r)$ und Endmerkmal $\omega(r)$ eindeutig beschrieben werden.

$$r \subseteq M \times M \quad (5-9)$$

$$(m_i, m_j) \in r \Leftrightarrow \alpha(r) = m_i \wedge \omega(r) = m_j, \text{ wobei } m_i, m_j \in M \ i, j \in \mathbb{N} \quad (5-10)$$

Die Menge der Beziehungen R in einem Variabilitätsmodell VM besteht aus folgenden Beziehungstypen:

$$R := \{r^{kind}, r^{eltern}, r^{needs}, r^{exclude}\} \quad (5-11)$$

Die hierarchischen Beziehungen r^{kind}, r^{eltern} in dem Variabilitätsmodell beschreiben die Eltern-Kind-Beziehung zwischen einzelnen Merkmalen.

Seien m_i, m_j zwei Merkmale aus M , welche in einer hierarchischen Beziehung zu einander stehen:

Die Beziehung r^{kind} kann folgendermaßen dokumentiert werden:

$$kind_von(m_i) := \{m_j \in M: (m_i, m_j) \in r^{kind}\} \quad (5-12)$$

Dabei ist diese Beziehung r^{kind} irreflexiv:

$$\forall m_i \in M: (m_i, m_i) \notin r^{kind} \quad (5-13)$$

Außerdem ist diese Beziehung r^{kind} assymetrisch: Wenn das Merkmal m_j ein Kind-Merkmal von dem Merkmal m_i ist, dann kann das Merkmal m_i nicht das Kind-Merkmal von dem Merkmal m_j sein.

$$\forall m_i, m_j \in M: (m_i, m_j) \in r^{kind} \Rightarrow (m_j, m_i) \notin r^{kind} \quad (5-14)$$

Die Beziehung r^{eltern} kann folgendermaßen dokumentiert werden:

$$eltern_von(m_j) := \{m_i \in M: m_j \in kind_von(m_i)\} \quad (5-15)$$

Ein Merkmal m in dem Variabilitätsmodell VM kann nur ein Eltern-Merkmal haben.

$$\forall m_j \in M: |eltern_von(m_j)| = 1 \quad (5-16)$$

Ein Merkmal ohne Kinder-Merkmale wird ein Blatt-Merkmal m^{blatt} genannt und ist charakterisiert durch:

$$\nexists m \in M: m \in kind_von(m^{blatt}) \quad (5-17)$$

Zusätzlich zu den hierarchischen Beziehungen können in einem Variabilitätsmodell zwei andere Typen von Beziehungen „benötigt“ und „schließt_aus“ existieren.

Die Beziehung „benötigt“ r^{needs} beschreibt, dass die Auswahl eines Merkmals m_i die Auswahl eines anderen Merkmals m_j mit $m_i \neq m_j$ benötigt. Diese Beziehung ist assymetrisch und irreflexiv.

$$r^{needs} \subseteq M \times M \quad (5-18)$$

$$\forall i, j \text{ mit } i \neq j: (m_i, m_j) \in r^{needs} \Rightarrow (m_j, m_i) \notin r^{needs} \quad (5-19)$$

$$\forall m_i \in M: (m_i, m_i) \notin r^{needs} \quad (5-20)$$

Die Beziehung „schließt_aus“ $r^{exclude}$ beschreibt, dass die Auswahl eines Merkmals m_i die Auswahl eines anderen Merkmals m_j mit $m_i \neq m_j$ ausschließt. Diese Beziehung ist symmetrisch.

$$r^{exclude} \subseteq M \times M \quad (5-21)$$

$$\forall i, j \text{ mit } i \neq j: (m_i, m_j) \in r^{exclude} \Rightarrow (m_j, m_i) \in r^{exclude} \quad (5-22)$$

$$\forall m_i \in M: (m_i, m_i) \notin r^{exclude} \quad (5-23)$$

Eine Konfiguration $Var(VM)$ eines Variabilitätsmodells wird definiert als eine Teilmenge von Merkmalen und Beziehungen des Variabilitätsmodells VM .

$$Var(VM) := \{(w, M', R') \mid (w, M, R) = VM \wedge M' \subseteq M, R' \subseteq R\} \quad (5-24)$$

Um die Konsistenz eines Variabilitätsmodells zu garantieren, müssen folgende notwendigen Bedingungen erfüllt werden:

- Jedes Merkmal m innerhalb eines Variabilitätsmodells VM hat einen eindeutigen Namen.

$$\forall VM = (w, M, R), m_i, m_j \in M \text{ gilt} \\ m_i.Name \neq m_j.Name, \text{ falls } i \neq j \quad (5-25)$$

- Ein Merkmal darf nur ein Eltern-Merkmal haben (5-14).
- Beziehungen zwischen mehreren Merkmalen müssen sich gegenseitig ausschließen.

$$r^{needs} \cap r^{exclude} = \emptyset \quad (5-26)$$

Das bedeutet, dass falls zwischen zwei Merkmalen eine „benötigt“-Beziehung existiert, darf zwischen diesen zwei Merkmalen keine „schließt_aus“-Beziehung existieren und umgekehrt.

Die in diesem Kapitel vorgestellten Formalisierungen der Anforderungsdokumente und Variabilitätsmodelle werden in den nächsten Kapiteln zur Präzisierung der erarbeiteten Ansätze verwendet.

Kapitel 6

Konfigurationsgekoppelte dezentrale Variabilitätsmodellierung

Dieses Kapitel stellt einen neuen konfigurationsgekoppelten Variabilitätsmodellierungsansatz für die Dokumentation von hochkomplexen Variabilitätsmodellen vor. Dieser Ansatz basiert auf dem Merkmalsmodellierungsansatz und bietet eine Lösung zur Bewältigung des Komplexitätsproblems unter Berücksichtigung der Variabilitätskriterien auf den unterschiedlichen Abstraktionsebenen.

6.1 Zentrale vs. dezentrale Variabilitätsmodellierung

Bereits in Kapitel 2 wurde gezeigt, dass die Variabilitätsmodellierung ein wichtiger Teil des Variabilitätsmanagements ist. Die zentrale Frage der Variabilitätsmodellierung ist wie viele Variabilitätsmodelle müssen erstellt werden, um die Variabilität in Anforderungsdokumenten auf allen Abstraktionsebenen zu dokumentieren.

Alle Ansätze in der Forschung können in zwei Gruppen unterteilt werden:

- Eine Gruppe empfiehlt den Aufbau eines zentralen Variabilitätsmodells. Die Vor- und Nachteile der zentralen Variabilitätsmodellierung werden im nächsten Abschnitt gezeigt.
- Die andere Gruppe benutzt mehrere dezentrale Variabilitätsmodelle, um die Variabilität mehrerer abhängiger Produktlinien zu verwalten. Diese Vorgehensweise wird im Abschnitt 6.1.3 erläutert.

6.1.1 Zentrales Variabilitätsmodell

Die meisten Merkmalsmodellierungsansätze haben als Basis ein zentrales Variabilitätsmodell, welches alle Merkmale einer Produktlinie abbildet ([BHPW04], [CzEi00]). Die Erstellung eines solchen Modells erfolgt dann nach dem Top-down-Prinzip.

Eine wichtige Frage bei der Entwicklung von hochkomplexen Produkten (z.B. Fahrzeuge oder Flugzeuge) ist: Was ist eine Produktlinie? Muss das Variabilitätsmodell die Variabi-

lität des Gesamtproduktes oder die einzelnen Komponenten und Systeme, welche auch in Produktlinienansätzen entwickelt werden, beinhalten?

Die Erstellung eines zentralen Variabilitätsmodells für ein hochkomplexes Gesamtprodukt ist zwar theoretisch möglich, aber der praktische Nutzen dieses Modells wäre fast gleich Null. Die Gründe dafür sind Folgende:

- *Größe und Komplexität eines zentralen Variabilitätsmodells:* Diese zwei Aspekte wurden bereits im Kapitel 1.3 bei der Erläuterung der Problembereiche vorgestellt. Ein zentrales Variabilitätsmodell für ein hoch-komplexes Produkt wie ein Fahrzeug wäre so komplex, dass Änderungen und Pflege auch mit einer Werkzeugunterstützung nicht möglich wären. Die mögliche Größe eines zentralen Variabilitätsmodells für ein Fahrzeug kann anhand des bereits in vorherigen Kapiteln erwähnten Beispiels verdeutlicht werden. *Das Variabilitätsmodell der Fahrzeugkomponente Sonnenblende beinhaltet ca. 20 Merkmale. Das Variabilitätsmodell eines CD-Players, welcher in ein Fahrzeug eingebaut wird, beinhaltet schon ca. 70 Merkmale. Diese zwei Beispiele zeigen nur die Merkmalanzahl für zwei Fahrzeugkomponenten, während ein Fahrzeug heutzutage aus ca. 100 Systemen und tausenden Komponenten besteht [Houd10].* Auch wenn in dem Variabilitätsmodell nur die Variabilitätskriterien (VK) und deren Ausprägungen (AVK) dokumentiert werden, wird die Anzahl von Variabilitätskriterien im Bereich von mehreren Tausenden liegen.
- *Erstellungsprozess:* Ein weiteres Problem ist die Prozessausrichtung bei der Erstellung des zentralen Merkmalsmodells. Eine Top-down-Vorgehensweise funktioniert hier nicht, weil mehrere Merkmale auf der Fahrzeug- und Systemebene nicht vor der Komponentenvariabilität definiert werden können. Eine Bottom-Up-Vorgehensweise funktioniert auch nicht, weil viele Komponenten und Systeme unabhängig von dem Gesamtfahrzeugprojekt entwickelt werden.
- *Aktualität:* Viele Komponenten und Systeme haben divergierende Lebenszyklen [TaRe07]. Für das Variabilitätsmodell bedeutet dies, dass es permanent an die neuen Änderungen angepasst werden muss.

Außerdem ist zu beachten, dass es bei komplexen Produkten, die aus mehreren Systemen und Komponenten bestehen, wichtig ist, nicht nur die Konfigurationen des Gesamtproduktes sondern auch die Konfigurationen von einzelnen Systemen und Komponenten erstellen zu können. Die Konfigurationen von Systemen und Komponenten sind die Grundlage für die Erstellung der variantenspezifischen Spezifikationen, welche ihrerseits die Grundlage für Verträge mit Zulieferern bilden.

6.1.2 Dezentrale Variabilitätsmodelle

Eine Alternative zur zentralen Variabilitätsmodellierung ist die dezentrale Variabilitätsmodellierung. Im Fall der Fahrzeugentwicklung wäre es notwendig, für jede einzelne Komponente und jedes einzelne System ein separates Variabilitätsmodell zu erstellen. Diese Lösungsvariante beinhaltet folgende Herausforderungen:

- Die einzelnen Variabilitätsmodelle müssen redundanzfrei sein.
- Die Konsistenz einzelner Variabilitätsmodelle sowie der gesamten Endproduktkonfiguration muss sichergestellt sein.

Eine Analyse der existierenden Ansätze im Bereich der dezentralen Variabilitätsmodellierung [Schm10] hat gezeigt, dass bisher nur wenige Ansätze in der Literatur existieren. Keiner dieser Ansätze wurde in der Praxis evaluiert.

Die zentrale, sowie die dezentrale Variabilitätsmodellierung haben mehrere unterschiedliche Vor- und Nachteile und sind aus unterschiedlichen Gründen nicht praxistauglich. Deswegen wurde im Rahmen dieser Arbeit ein konfigurationsgekoppelter dezentraler

Variabilitätsmodellierungsansatz (KDVM-Ansatz) erarbeitet, welcher die Vorteile sowohl der zentralen als auch der dezentralen Variabilitätsmodellierung beinhaltet. Die Nachteile einzelner Vorgehensweisen werden durch die Kombination der Vorgehensweisen und die Regeln des KDVM-Ansatzes minimiert.

Im Abschnitt 6.2 wird das Konzept der konfigurationsgekoppelten dezentralen Variabilitätsmodellierung (KDVM) präsentiert. Im Abschnitt 6.2.1 wird die Dokumentationsmöglichkeit für die variablen Eigenschaften, im 6.2.2 die Dokumentationsmöglichkeit für die Beziehungen zwischen den variablen Eigenschaften dargestellt. Besondere Aufmerksamkeit wird auf den Prozess der Identifikation der Beziehungen zwischen einzelnen Variabilitätskriterien und deren Ausprägungen gelegt. Abschließend wird im Abschnitt 6.3 der Konfigurationsprozess von KDVM beschrieben.

6.2 Der KDVM-Ansatz

Der Grundgedanke des KDVM-Ansatzes besteht darin, mehrere (kleinere) dezentrale Variabilitätsmodelle für einzelne Elemente (z.B. System, Komponente) auf den entsprechenden Abstraktionsebenen zu erstellen. Dabei spiegeln die Variabilitätskriterien (VK) auf den einzelnen Ebenen die unterschiedlichen Sichten eines Produktes bzw. seiner Bestandteile wider. So werden auf der Systemebene die VK aus funktionaler Sicht erfasst, während auf der Komponentenebene die VK aus technischer Sicht behandelt werden. Die Besonderheit von KDVM besteht darin, dass die VK auf den höheren Ebenen die Konfigurationen von Variabilitätsmodellen in den tieferen Ebenen sind. Dabei können zwischen einzelnen VK oder AVK sowohl auf einer Ebene, als auch auf den unterschiedlichen Ebenen die „benötigt“- bzw. „schließt_aus“-Beziehungen existieren. *Zum Beispiel, ist die Ausprägung Luxus des Variabilitätskriteriums (AVK) Sitzheizung im Variabilitätsmodell auf Produkt-Ebene die Konfiguration des Variabilitätsmodells Sitzheizung (VK Sitzheizung und Sitzlüftung müssen für diese Konfiguration ausgewählt werden) auf der System-Ebene.* Dieses Beispiel wird detailliert im nächsten Abschnitt vorgestellt. Diese Verknüpfung einzelner Variantenmodelle über Konfigurationen erhält die Vorteile der zentralen Variabilitätsmodellierung.

Für die Dokumentation der Variabilitätsmodelle wird die in Kapitel 2.2 vorgestellte Dokumentationsform benutzt. Dabei müssen die Merkmalseigenschaften um einen weiteren Aspekt erweitert werden: Jedes Merkmal benötigt ein zusätzliches Attribut „Ebene“ (*m. Ebene*), welches die Kompatibilität der Modelle auf allen Abstraktionsebenen unterstützen wird.

Zusätzlich zu den bereits existierenden Abstraktionsebenen wird beim KDVM-Ansatz eine zusätzliche Portfolio-Ebene eingeführt, um die Variabilität von mehreren Produktlinien verwalten zu können. Das Variabilitätsmodell auf Portfolio-Ebene dokumentiert die globalen Variabilitätskriterien.

Definition 6-1: Globales Merkmal

Ein globales Merkmal ist ein Variabilitätskriterium, welches Einfluss auf die Konfigurationsentscheidungen auf jeder anderen Abstraktionsebene haben kann.

Die Konfiguration des Variabilitätsmodells auf der Portfolioebene hat Einfluss auf mehrere Modelle auf allen anderen Ebenen. Neue Produkte (Fahrzeuge) werden als die Konfigurationen globaler VK definiert.

Am Beispiel der Mercedes-Benz Car Group wurden folgende globale Merkmale identifiziert: die Modellklasse (z.B. A-, B-, E-Klasse), die Karosserieform (z.B. Cabrio, Limousine) und der Markt. Beispielsweise existieren zwei Varianten der Tachometeranzeige: die Geschwindigkeit wird in km/h oder in mph angezeigt. Das entsprechende Merkmal auf der Komponenten-Ebene ist Geschwindigkeitsanzeige. Die Auswahl des globalen Varia-

ibilitätskriteriums (z.B. der Markt) beeinflusst die Auswahl der Ausprägung der Geschwindigkeitsanzeige.

Eine ähnliche Vorgehensweise (Product-Sets-Ansatz) wurde von Reiser und Weber in [ReWe05] vorgeschlagen, um Produktvariabilität zu beschreiben.

In den nächsten Abschnitten wird anhand eines Beispiels detailliert erläutert, wie die VK und deren Beziehungen auf unterschiedlichen Abstraktionsebenen der Variabilitätsmodelle dokumentiert und verwaltet werden können.

6.2.1 Dokumentation von Variabilitätskriterien

Bevor die Dokumentation von Merkmalen im KDVM-Ansatz präsentiert wird, werden zunächst alle Erkenntnisse zur Anforderungsdokumentation hochkomplexer Produkte zusammengefasst:

- Die Anforderungen an komplexe Produkte werden oft auf unterschiedlichen Abstraktionsebenen (siehe 2.1.3) identifiziert und dokumentiert.
- Die variablen Eigenschaften auf unterschiedlichen Abstraktionsebenen spiegeln die unterschiedlichen Sichten (Abstraktionsebenen) auf ein Produkt bzw. seine Bestandteile wider. In dieser Arbeit wird zwischen drei Abstraktionsebenen (Produkt, System, Komponente), sowohl für die Anforderungsdokumente als auch für die Dokumentation der Variabilitätsmodelle, unterschieden.

Abbildung 18 stellt ein Beispiel für die Dokumentation der Variabilitätsmodelle mit dem KDVM-Ansatz anhand vereinfachter Daten für den Automobilhersteller Mercedes-Benz dar.

Das Variabilitätsmodell auf der *Portfolioebene (PKW)* besteht aus zwei obligatorischen VK: *Fahrzeugmodell* (mit zwei alternativen Ausprägungen: *S-Klasse* und *C-Klasse*) und *Markt* (mit zwei optionalen Ausprägungen: *Europa* und *USA*). Jede Konfiguration, dieses Variabilitätsmodells muss diese beiden VK beinhalten. Jede Konfiguration dieses Variabilitätsmodells darf nur eine Ausprägung des Fahrzeugmodells (entweder *C-* oder *S-Klasse*) beinhalten und kann auf einem oder auf beiden Märkten (d.h. in *Europa* oder in den *USA* oder sowohl in den *USA* als auch in *Europa*) angeboten werden. Dieses Modell kann als erstes anhand der vorhandenen Daten des Marketings erstellt werden.

Das Variabilitätsmodell auf der *Produkt-Ebene* kann in den meisten Fällen erst nach der Erstellung der Variabilitätsmodelle auf der *System-* und *Komponenten-Ebene* erstellt werden. Grund dafür ist, dass die AVK im Variabilitätsmodell auf der *Produkt-Ebene* den Konfigurationen bzw. Teilkonfigurationen aus den zwei unteren Ebenen entsprechen. Das Variabilitätsmodell auf dieser Ebene besteht in diesem Beispiel aus den obligatorischen VK *Sitzheizung* und *Klimatisierung*. Das System *Sitzheizung* steht in zwei alternativen Ausprägungen zur Verfügung: *Basis* und *Luxus*.

Das Variabilitätsmodell auf der *System-Ebene* beschreibt die funktionale Variabilität und die Möglichkeiten der technischen Realisierung eines Systems. In dem vorgestellten Beispiel besteht das Variabilitätsmodell des Systems *Sitzheizung* aus einem obligatorischen VK *Heizung*, welches in zwei alternativen Formen realisiert werden kann: *2-Stufen* oder *4-Stufen*, und einem optionalen VK *Lüftung*.

Das Variabilitätsmodell auf der *Komponenten-Ebene* beschreibt die technische und gestalterische Variabilität der Komponenten. Beispielsweise besteht das Variabilitätsmodell der Komponente *Sitz* aus einem obligatorischen VK *Heizelement* mit zwei alternativen Realisierungen *2-Pole* oder *4-Pole*, und einem optionalen VK *Lüfter*.

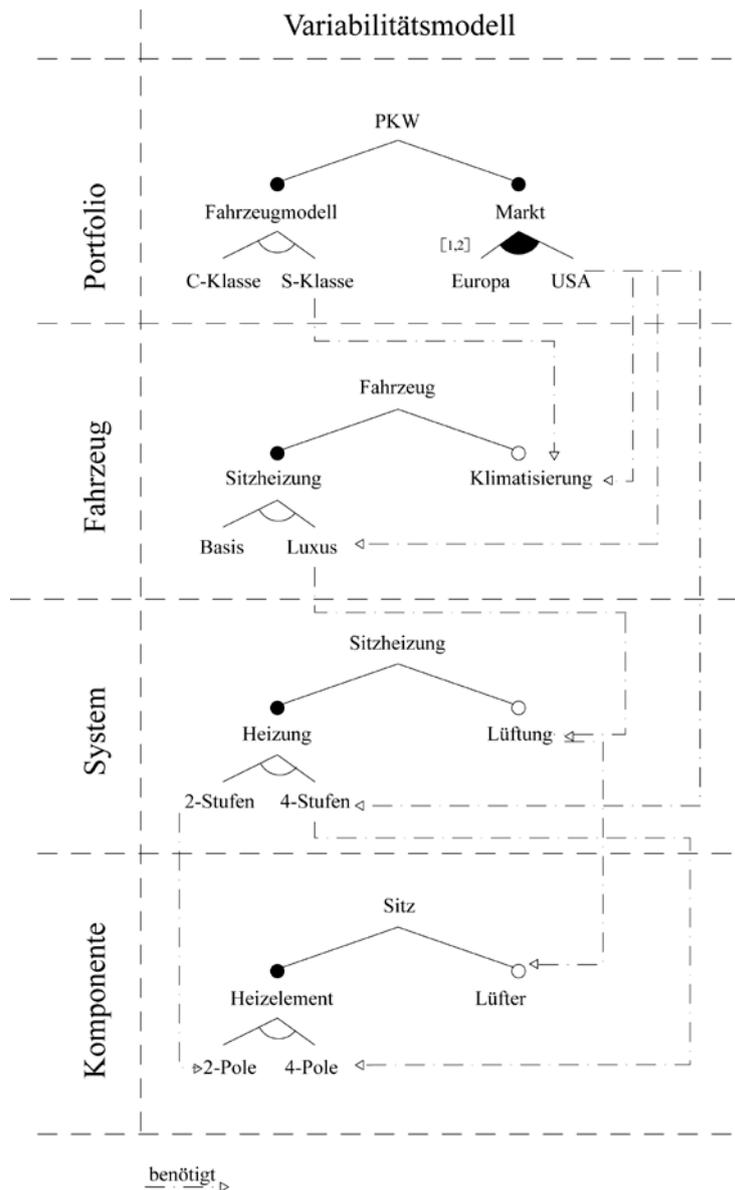


Abbildung 18: Beispiel für den KDVM-Ansatz

6.2.2 Beziehungen zwischen Variabilitätskriterien

Zwischen einzelnen Variabilitätskriterien sowohl innerhalb eines Variabilitätsmodells als auch zwischen unterschiedlichen Variabilitätsmodellen können Abhängigkeitsbeziehungen existieren. In Abbildung 18 wurde ein Beispiel für mögliche Beziehungen gegeben:

- Die AVK *S-Klasse* im Portfolio-Variabilitätsmodell *benötigt die VK Klimatisierung* im Variabilitätsmodell auf der Produkt-Ebene.
- Die AVK *USA* im Portfolio-Variabilitätsmodell benötigt das VK *Klimatisierung* und die AVK *Luxus* im Variabilitätsmodell auf der Produktebene.
- Die AVK *USA* im Portfolio-Variabilitätsmodell benötigt die AVK *4-Stufen* im Variabilitätsmodell auf der System-Ebene.
- Die AVK *Luxus* im Variabilitätsmodell auf der Produkt-Ebene benötigt die AVK *Lüftung* im Variabilitätsmodell auf der System-Ebene.

- Die AVK *Lüftung* im Variabilitätsmodell auf der System-Ebene benötigt die AVK *Lüfter* im Variabilitätsmodell auf der Komponenten-Ebene.
- Die AVK *2-Stufen* im Variabilitätsmodell auf der System-Ebene benötigt die AVK *2-Pole* im Variabilitätsmodell auf der Komponenten-Ebene.
- Die AVK *4-Stufen* im Variabilitätsmodell auf der System-Ebene benötigt die AVK *4-Pole* im Variabilitätsmodell auf der Komponenten-Ebene.

In dem Thema Beziehungen zwischen einzelnen Merkmalen existieren zwei Herausforderungen:

- (1) Wie identifiziert man die Beziehungen zwischen Merkmalen in den unterschiedlichen Variabilitätsmodellen?
- (2) Wie dokumentiert man die identifizierten Beziehungen zwischen Merkmalen aus den unterschiedlichen Variabilitätsmodellen?

Die Lösung der ersten Herausforderung wird im nächsten Abschnitt diskutiert und die Möglichkeiten der Dokumentation der identifizierten Beziehungen in 6.2.2.2 präsentiert.

6.2.2.1 Identifikation der Beziehungen zwischen Merkmalen

Die Beziehungen zwischen Merkmalen beschreiben, welchen Einfluss die Auswahl eines Merkmals auf die Auswahlmöglichkeiten der anderen Merkmale hat. Diese Herausforderung der Identifikation von Beziehungen besteht sowohl bei den komplexen zentralen Variabilitätsmodellen als auch bei der dezentralen Variabilitätsmodellierung. Während im ersten Fall die Beziehungen innerhalb eines Modells identifiziert und gepflegt werden müssen, werden im zweiten Fall zusätzlich die Beziehungen zwischen den einzelnen Modellen betrachtet. In der Literatur existieren zwar mehrere Arbeiten (z.B. [KCNP90], [StRP03], [LeKa04]) über die Typisierung der möglichen Beziehungen, aber keiner dieser Ansätze hat sich mit der Frage der Identifikation von Beziehungen beschäftigt.

Die vorliegende Arbeit bietet einen neuartigen Ansatz, mit dessen Hilfe die Beziehungen zwischen Merkmalen eines Modells sowie zwischen Merkmalen unterschiedlicher Variabilitätsmodelle auf unterschiedlichen Abstraktionsstufen identifiziert werden können.

Um die Beziehungen identifizieren zu können, müssen die einzelnen Variabilitätsmodelle in der Form von Verträglichkeitsmatrizen dokumentiert werden.

Definition 6-2: *Verträglichkeitsmatrix*

Eine Verträglichkeitsmatrix ist die Erweiterung des morphologischen Kastens zur Definition der Kombinierbarkeit von Teillösungen.

[Birk80]

Definition 6-3: *Morphologischer Kasten*

Der morphologische Kasten ist eine mehrdimensionale Matrix und bildet das Kernstück der morphologischen Analyse.

[Zwic89]

Definition 6-4: *Morphologische Analyse*

Die morphologische Analyse ist eine kreative heuristische Methode, um komplexe Problembereiche vollständig zu erfassen und alle möglichen Lösungen vorurteilslos zu betrachten.

[ScZu02]

Die Methode des morphologischen Kastens wurde auf der Grundlage der Morphologie mit dem Ziel entwickelt zu gegebenen Problemen auf vorurteilsfreiem Weg eine Lösungsableitung zu ermöglichen [Zwic89]. Für die Variabilitätsmodellierung werden die Variabilitätskriterien anstelle der Teilfunktionen und die Ausprägungen anstelle der Teillösungen dokumentiert. Abbildung 19 präsentiert ein Beispiel für einen morphologischen Kasten. *Dieser Kasten beschreibt die Variabilität eines Fahrzeugs. Ein Fahrzeug in diesem Beispiel hat drei variable Komponenten: einen Motor, ein Getriebe und eine Anhängerkupplung. Der Motor kann in einer der drei Varianten gebaut werden: Benzin, Diesel oder Elektro. Das Getriebe kann entweder manuell oder automatisch sein. Die Anhängerkupplung kann vorhanden oder nicht vorhanden sein.*

Komponente	Lösungsvarianten		
	1	2	3
Motor	Benzin	Diesel	Elektro
Getriebe	manuell	automatisch	-
Anhängerkupplung	mit	ohne	-

Abbildung 19: Morphologischer Kasten

Abbildung 20 präsentiert ein Beispiel für eine Verträglichkeitsmatrix. Die Komponente *Motor* hat dabei drei Ausprägungen: *Benzin*, *Diesel*, *Elektro*. Die Komponente *Anhängerkupplung* ist optional. In der Verträglichkeitsmatrix haben alle zulässige Kombinationen den Wert „1“ und alle unzulässigen Kombinationen den Wert „-1“.

		Anhängerkupplung	
		Mit	Ohne
Motor	Benzin	1	1
	Diesel	1	1
	Elektro	-1	1

Abbildung 20: Beispiel einer Verträglichkeitsmatrix.

Für den vorgeschlagenen KDVM-Ansatz ist es notwendig, folgende fünf Verträglichkeitsmatrizen zu erstellen, welche die für die Produktdefinition notwendigen Verträglichkeitsbeziehungen dokumentieren:

- (3) *Komponente-zu-Komponente* zur Darstellung der Abhängigkeiten zwischen den Merkmalen einzelner Komponenten;
- (4) *System-zu-System* zur Darstellung der Abhängigkeiten zwischen den Merkmalen einzelner Systeme;
- (5) *Komponente-zu-Produkt* zur Darstellung der Abhängigkeiten zwischen den Merkmalen mehrerer Komponenten eines Produktes;
- (6) *Portfolio-zu-System* zur Darstellung der Abhängigkeiten zwischen den globalen Merkmalen und den Merkmalen der Systeme;
- (7) *Portfolio-zu-Komponente* zur Darstellung der Abhängigkeiten zwischen den globalen Merkmalen und den Merkmalen der Komponenten.

Diese Matrizen können als Grundlage für Interviews mit Komponenten- und Systemverantwortlichen benutzt werden.

Zur Unterstützung der Erstellung von Verträglichkeitsmatrizen können Konfigurationsmatrizen erstellt werden.

Definition 6-3: Konfigurationsmatrix

Eine Konfigurationsmatrix beschreibt die Verbindung zwischen zwei unterschiedlichen Sichten (z.B. zwischen Kundensicht und technischer Sicht) auf ein (Teil-) Produkt. In Anlehnung an [Puls02]

Abbildung 21 präsentiert ein Beispiel für eine Konfigurationsmatrix für ein Fahrzeug. Die Variabilitätsausprägungen aus der Verträglichkeitsmatrix werden bei der Zuordnung zu einer Konfiguration mit dem Wert „1“ dokumentiert und alle anderen mit dem Wert „0“. Das Fahrzeugmodell „Basis 1“ hat einen Benzin-Motor und ein manuelles Getriebe. Das Fahrzeugmodell „Basis 2“ hat einen Diesel-Motor und ein automatisches Getriebe. Das Fahrzeugmodell „Umwelt“ hat einen Elektro-Motor und ein automatisches Getriebe.

		Fahrzeugmodell		
		Basis 1	Basis 2	Umwelt
Motor	Benzin	1	0	0
	Diesel	0	1	0
	Elektro	0	0	1
Getriebe	Manuell	1		0
	Automatik	0	1	1

Abbildung 21: Konfigurationsmatrix

Für die Dokumentation der Produkte mit dem vorgeschlagenen KDVM-Ansatz müssen folgende zwei Konfigurationsmatrizen erstellt werden:

- *System-zu-Komponente*, um die geplanten Konfigurationen eines Systems zu dokumentieren.
- *Produkt-zu-System*, um die geplanten Konfigurationen eines Produktes zu dokumentieren.

Der Beziehungsidentifikationsprozess besteht aus folgenden Schritten:

Schritt 1: Voraussetzung für den ersten Schritt ist die Verfügbarkeit von Informationen über das neue Produkt. Diese Informationen können aus der Firmen-Portfolio-Planung entnommen werden. Wird zum Beispiel ein neues Fahrzeug geplant, so werden zuerst vom Vertrieb die Modellreihe (z.B. S-Klasse) und die Vertriebsmärkte (z.B. Europa und USA) festgelegt. Im ersten Schritt können folgende Verträglichkeitsmatrizen analysiert und ausgefüllt werden: Verträglichkeitsmatrix *Portfolio-zu-Systeme* und Verträglichkeitsmatrix *Portfolio-zu-Komponenten*.

Schritt 2: Im zweiten Schritt können die Verträglichkeitsmatrizen *Komponente-zu-Komponente* und *Produkt-zu-Komponente* erstellt werden. Diese Matrizen können als Grundlage für Interviews mit den Komponentenverantwortlichen benutzt werden, um so die vorhandenen Beziehungen zu identifizieren.

Schritt 3: Im dritten Schritt können die Konfigurationsmatrizen *System-zu-Komponente* und *Produkt-zu-System* erstellt werden, um die nachfolgenden Arbeitsschritte zu unterstützen.

Schritt 4: Am Ende des Beziehungsidentifikationsprozesses kann die Verträglichkeitsmatrix *System-zu-System* ausgefüllt werden, die dann als Grundlage für die Interviews mit Systemverantwortlichen benutzt wird, um hier die vorhandenen Beziehungen zu identifizieren.

Schritt 5: Nachdem alle Matrizen fertiggestellt sind, müssen die Verträglichkeitsmatrizen auf Konsistenz (z.B. das Vorhandensein von Ringbeziehungen und sich gegenseitig ausschließende Beziehungen) überprüft und bei Bedarf angepasst werden. Falls eine Anpassung stattgefunden hat, müssen auch die korrespondierenden Konfigurationsmatrizen angepasst werden.

Der größte Vorteil der Anwendung von Verträglichkeitsmatrizen besteht in ihrer Einfachheit und Übersichtlichkeit. Der große Nachteil ist jedoch, dass wegen der zweidimensionalen Matrizendarstellung nur Zweierkombinationen von Abhängigkeiten dargestellt werden können, und somit die Verträglichkeitsmatrizen nicht für die endgültige Dokumentation aller Beziehungen benutzt werden können.

6.2.2.2 Dokumentation der Beziehungen zwischen Merkmalen

Bei der Literaturrecherche zu dieser Arbeit wurden fünf Basisansätze zur Dokumentation der Beziehungen zwischen Merkmalen gefunden:

- (1) Attribuierung von Merkmalen mit logischen Ausdrücken [CzEi00],
- (2) Nachvollziehbarkeitslinks (engl. Traceability Links) zwischen einzelnen Modellen [Reis08],
- (3) Unterschiedliche Kombinationen aus den Ansätzen (1) und (2),
- (4) Matrizen mit Teilkonfigurationen [BigL11],
- (5) Verwendung von „Configuration Links“ [Reis08].

In der Arbeit von Reiser [Reis08] werden alle diese genannten Ansätze detailliert vorgestellt und verglichen. Sein Configuration-Links-Ansatz bietet die beste Methode für die Dokumentation der Beziehungen zwischen mehreren Merkmalsmodellen auf unterschiedlichen Abstraktionsebenen.

Die Beziehungen werden in so genannten „Decision Tables“ (Entscheidungstabellen) dokumentiert. Abbildung 22 bietet ein Beispiel für eine derartige „Decision Table“.

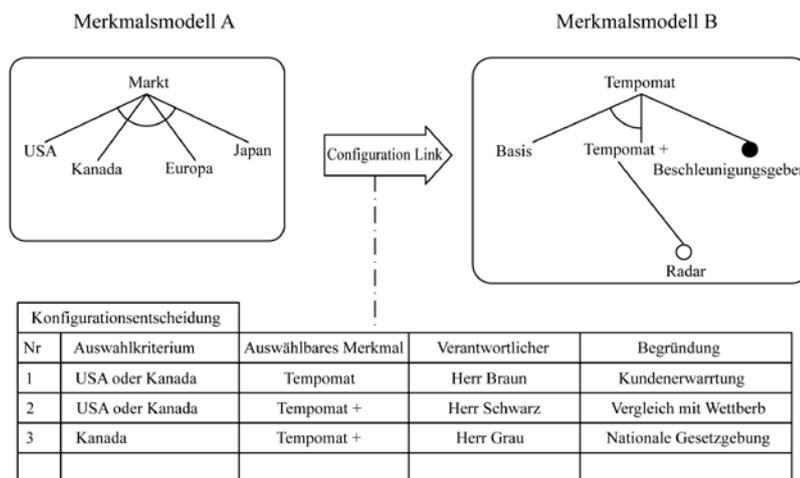


Abbildung 22: Beispiel zur Verwendung von „Configuration Links“ [Reis08]

6.2.3 Konfigurationserstellung

Der Zweck der Variabilitätsmodellierung ist nicht nur die Dokumentation der Variabilität, sondern auch die Unterstützung der Konfigurationserstellung.

Bei der Konfigurationserstellung für KDVM besteht eine der Herausforderungen darin, sowohl den Unterschied der Konfigurationzwecke (z.B. Fahrzeugkonfiguration vs. Komponentenkonfiguration) als auch das Zusammenspiel der Konfigurationen auf den unterschiedlichen Ebenen zu berücksichtigen. So entspricht eine Konfiguration auf Produkt-Ebene einem Produkt bzw. einer vorkonfigurierten Produktlinie, während eine Konfiguration auf der System-Ebene eine Systemvariante repräsentiert, die in ein Produkt eingebaut werden kann. Eine auf der Komponentenebene erstellte Konfiguration stellt eine Komponentenvariante dar, die eine bestimmte Funktion einer Systemvariante realisiert oder die in ein Produkt eingebaut werden kann.

Bei der Konfigurationserstellung müssen die Variabilitätsmodelle eine bestimmte Qualität aufweisen. In diesem Kapitel werden die dafür notwendigen Bedingungen für die Variabilitätsmodelle beschrieben.

Um die Konsistenz der Modelle auf allen Ebenen zu garantieren, müssen zuerst die Konsistenzregeln der einzelnen Variabilitätsmodelle (Formeln 5-25, 5-26) erweitert werden:

1. Der Name des Variabilitätsmodells darf nicht mehrmals in einer Ebene vorkommen. Dabei wird der Name eines Variabilitätsmodells VM über den Namen der Wurzel w dokumentiert:

$$\forall VM_i, VM_j \in \text{Ebene } P, i, j \in \mathbb{N}: w_i. \text{Name} = w_j. \text{Name} \Rightarrow i \neq j \quad (6-1)$$

Auf unterschiedlichen Ebenen können die Namen der Variabilitätsmodelle mehrmals vorkommen:

$$\begin{aligned} \forall VM_i \in \text{Ebene } P, VM_j \in \text{Ebene } S, i, j \in \mathbb{N}: \\ w_i. \text{Name} = w_j. \text{Name} \Rightarrow P \neq S \end{aligned} \quad (6-2)$$

2. Die Beziehungen zwischen mehreren Merkmalen aus unterschiedlichen Modellen dürfen sich nicht gegenseitig ausschließen (5-26). Das bedeutet, fall zwischen dem Merkmal m_i aus dem Variabilitätsmodell VM_i und dem Merkmal m_j aus dem Variabilitätsmodell VM_j eine „benötigt“-Beziehung existiert, darf nicht zwischen diesen Merkmalen eine „schließt_aus“-Beziehung existieren.

$$\begin{aligned} \forall m_i \in VM_i, m_j \in VM_j \text{ mit } i \neq j: \\ \exists (m_i, m_j) \in r^{needs} \Rightarrow (m_j, m_i) \notin r^{exclude} \end{aligned} \quad (6-3)$$

$$\begin{aligned} \forall m_i \in VM_i, m_j \in VM_j \text{ mit } i \neq j: \\ (m_i, m_j) \in r^{exclude} \Rightarrow (m_j, m_i) \notin r^{needs} \end{aligned} \quad (6-4)$$

Für die Dokumentation der Konfigurationsentscheidungen existieren unterschiedliche Ansätze: die „Stage Configuration“ [CzHE05], der Matrix-Ansatz [Pohl08], die „Configuration Criteria“ [CzEi00], „Links“ und „Configuration Links“ [Reis08]. Diese verschiedenen Ansätze berücksichtigen zwar die Konfigurationsentscheidungen eines Merkmalsmodells, die die Konfigurationsentscheidungen in einem anderen Merkmalsmodell beeinflussen können, jedoch vermag keiner dieser Ansätze Konfigurationsentscheidungen über mehrere Ebenen zu unterstützen. Auch die Frage der effizienten Gestaltung des Konfigurationsprozesses wird in keinem dieser Ansätze betrachtet.

Rabiser et al. [RaGD10] haben eine umfangreiche Literaturrecherche zum Thema Konfigurationsansätze durchgeführt. Es wurde von ihnen gezeigt, dass zwar viele unterschied-

liche Ansätze und Werkzeuge existierten, aber alle diese Ansätze und Werkzeuge als Insellösungen entwickelt wurden, um die Probleme und Herausforderungen einzelner Domänen zu lösen.

Ein wichtiger Aspekt bei der Konfigurationserstellung ist außerdem die Richtung von Konfigurationsentscheidungen. Die klassische Vorgehensweise in den Konfigurationsansätzen ist „top-down“ (von der Portfolio-Ebene zu den Komponenten-Konfigurationen) [CzHE05]. Diese Vorgehensweise funktioniert in der Praxis oft nicht. Hingegen erlaubt der vorgestellte KDVM-Ansatz sowohl die klassische Top-Down-Vorgehensweise als auch eine Kombination aus Top-Down- und Bottom-Up-Vorgehensweisen, welche an der Produktstrategie des Unternehmens ausgerichtet werden kann.

In Rahmen dieser Arbeit, wurde eine Vorgehensweise erarbeitet, wie man den Konfigurationsprozess anhand der optimalen Reihenfolge von Konfigurationsentscheidungen effizient gestalten kann. Diese Vorgehensweise wird im Detail im Kapitel 8 vorgestellt werden.

6.3 Diskussion

Wie bereits diskutiert, existieren in der Literatur nur wenige Ansätze, die sich mit dem Thema der Merkmalsmodellierung für hochkomplexe Produktlinien beschäftigen ([BLPW04], [Reis08] [Tava06], [Reis08]). Noch weniger Ansätze existieren im Bereich der dezentralen Variabilitätsmodellierung für hochkomplexe Produktlinien [Schm10].

In der Arbeit von Reiser [Reis08] wurde ein „Multi-Level-Feature-Tree“-Ansatz (MLFT-Ansatz) vorgestellt, in dem ein Variantenbaum aus mehreren Produktlinien-Variantenbäumen erstellt wird. Dieser Ansatz kann als Mittelweg zwischen der zentralen und der dezentralen Variabilitätsmodellierung angesehen werden. Der MLFT-Ansatz basiert auf der Annahme, dass der Umfang von Produktlinien auf der unteren Abstraktionsebene eine Teilmenge des Umfangs von Produktlinien der oberen Abstraktionsebene sein muss. Basierend auf dieser Annahme wurde der Einsatz von Referenzmodellen vorgeschlagen. Jedes referenzierte Modell darf in diesem Ansatz nur ein Referenzmodell aufweisen.

Da die Komponenten oft mehrere Funktionen in unterschiedlichen Systemen realisieren, kann der MLFT-Ansatz nicht durchgängig auf allen Abstraktionsebenen realisiert werden. Der MLFT-Ansatz kann zwar auf der Produkt- und System-Ebene angewendet werden, aber nicht auf der Komponenten-Ebene. So kann ein Fahrzeug-Referenzmodell auf Produkt-Ebene erstellt werden, das alle Merkmale beinhaltet, die ein Kunde bei der Fahrzeugbestellung zur Auswahl hat. Auf Systemebene können dann, ausgehend von einem Systemmerkmalsmodell (Referenzmodell), mehrere fahrzeugmodellabhängige Systemvarianten (referenzierte Modelle) erstellt werden. Die Komponenten-Ebene wurde in diesem Ansatz nicht betrachtet. Die Erstellung von fahrzeugmodellabhängigen System- bzw. Komponentenvarianten-Merkmalsmodellen ist jedoch überflüssig, da hier nicht die Notwendigkeit besteht, die Variabilität einzelner System- und Komponentenvarianten für ein Fahrzeugmodell in einem eigenen Variantenmodell zu verwalten. Umso mehr besteht der Bedarf, alle System- bzw. Komponentenvarianten in fahrzeugmodellunabhängigen Merkmalsmodellen zu verwalten, um die Wiederverwendung von Anforderungen zu unterstützen.

Der vorgestellte KDVM-Ansatz bietet die Unterstützung der Variabilitätsmodellierung durch die Erstellung von kleineren aber durch die Konfigurationen verbundenen Variabilitätsmodellen. Diese Vorgehensweise ermöglicht eine effektive Variabilitätsmodellierung in den Anforderungsdokumenten von hochkomplexen Produkten. Die Anwendung des KDVM-Ansatzes bringt gegenüber der Anwendung von rein zentralen oder dezentralen Variabilitätsmanagementansätzen folgende Vorteile:

- Die Komplexität ist für die einzelnen Anwender überschaubar aufgrund der Anwendung dezentraler Variabilitätsmodelle auf den unterschiedlichen Abstraktionsebenen.
- Die vorgeschlagene Kopplung zwischen den Abstraktionsebenen über die Konfigurationen ermöglicht es, die Vorteile der zentralen Variabilitätsmodellierung beizubehalten.
- Der KDVM-Ansatz ist sofort anwendbar. Es ist nicht notwendig, zuerst ein riesiges Variabilitätsmodell zu erstellen, um das effektive und effiziente Wiederverwenden von Anforderungen zu ermöglichen. Der Ansatz bietet die Möglichkeit, die Variabilitätsmodelle für einzelne Anforderungsdokumente zu erstellen und erst danach zu verbinden.

Kapitel 7

R2F-Ansatz

Das vorliegende Kapitel stellt den neuen Ansatz für die Dokumentation der Variabilitätsinformationen in Anforderungsdokumenten vor. Basierend auf diesem Dokumentationsansatz wurde im Rahmen dieser Arbeit ein Variabilitätsmanagementansatz inklusive Prozessbeschreibung und Rollenverteilung erarbeitet, welcher in diesem Kapitel präsentiert wird. Des Weiteren werden in diesem Kapitel zwei semi-automatische Ansätze vorgestellt, die die Benutzer bei der Dokumentation von Variabilitätsinformationen in den Anforderungsdokumenten unterstützen können. Abschließend werden die Vor- und Nachteile des neuen Ansatzes diskutiert.

7.1 Variabilität in Anforderungsdokumenten

Im Kapitel 3 wurden die unterschiedlichen Dokumentationsmöglichkeiten der Variabilitätsinformationen diskutiert. Der Requirements-to-Features-Ansatz (R2F-Ansatz) benutzt den Gedanken der dezentralen Variabilitätsmodellierung aus dem vorherigen Kapitel und schlägt vor, die für ein Anforderungsdokument relevanten Variabilitätsinformationen in einem separaten Variabilitätsmodell zu dokumentieren und die Anforderungen lediglich den relevanten Variabilitätskriterien zuzuordnen. Das Variabilitätsmodell und das Anforderungsdokument müssen dazu in Beziehung gesetzt werden. Dabei kann ein Variabilitätsmodell für mehrere Anforderungsdokumente relevant sein. Ebenso ist es möglich, dass ein Anforderungsdokument Beziehungen zu mehreren Variabilitätsmodellen haben kann.

Für die Dokumentation des Variabilitätsmodells können die kommerziell erhältlichen Variantenmanagementwerkzeuge benutzt werden. Die im Variabilitätsmodell dokumentierten Variabilitätskriterien und deren Ausprägungen müssen in das Anforderungsdokument importiert werden. Deswegen besteht eine Vorbedingung für den Einsatz eines Variantenmanagementtools darin, dass es mit dem Anforderungsmanagementtool kompatibel ist und den Austausch von Daten ermöglicht. In Kapitel 9 wird dargestellt, wie die Funktionalität des Anforderungsmanagementtools DOORS erweitert werden kann, um damit die Variabilität dokumentieren zu können.

Im Rahmen des R2F-Ansatzes werden die Anforderungen nur den Variabilitätskriterien und deren Ausprägungen aus dem zugehörigen Variabilitätsmodell und nicht den Produkten zugeordnet.

Für die Dokumentation der Zuordnung einer Anforderung zu einem Merkmal wird der Operator „zugeordnet“, als Symbol, \mathbb{D} eingeführt.

$$\mathbb{D}: ABib \times M \rightarrow \mathbb{B} \quad (7-1)$$

Mit der Hilfe des neuen Operators \mathbb{D} kann die Zuordnung einer Anforderung a zu einem Merkmal m auch formal folgendermaßen dokumentiert werden:

$$\forall a \in ABib, m \in M: a \mathbb{D} m = true$$

bedeutet a ist m zugeordnet (7-2)

Wichtig ist, dass eine Anforderung a nur zu einem Merkmal m bzw. mehreren Merkmalen und nicht zu einer Konfiguration des Variabilitätsmodells $Var(VM)$ zugeordnet wird.

$$\forall a \in ABib, (w, M', R') \in Var(VM): a \mathbb{D} M' = false$$

bedeutet a ist nicht $Var(VM)$ zugeordnet (7-3)

Diese Festlegung bringt große Vorteile bei der Wiederverwendung von Anforderungen. Die Anforderungen müssen nicht mehr jedem neuem Produkt (einer neuen Konfiguration des Variabilitätsmodells) zugeordnet werden, sondern alle Anforderungen für ein neues Produkt $Conf(ABib)^P$ können jederzeit automatisiert ermittelt werden, durch die Auswahl der Anforderungen, welche zu Merkmalen aus der entsprechenden Konfiguration des Variabilitätsmodells $Var(VM)$ gehören.

Ein Lastenheft wird durch Auswahl der gültigen Anforderungen für die festgelegte Konfiguration des Variabilitätsmodells erstellt. Die Regeln für die Auswahl einer Anforderung sind folgende:

- (1) Die Anforderung ist dem Merkmal zugeordnet, welches Teil der aktuellen Konfiguration ist, oder
- (2) sie ist eine gemeinsame Anforderung für alle Produkte.

Generell kann man zwischen folgenden drei Notationsformen für die Dokumentation der variablen Eigenschaften in den Anforderungsdokumenten unterscheiden:

- *Spezifische Dokumentation:* Nur die variablen Anforderungen werden den gültigen Ausprägungen der Variabilitätskriterien zugeordnet. Nicht gekennzeichnete Anforderungen sind für alle Produktvarianten gültig.
- *Positive Dokumentation:* Jede Anforderung wird mit allen für diese Anforderung gültigen Ausprägungen der Variabilitätskriterien gekennzeichnet. Das bedeutet, dass eine Anforderung, welche für alle Produkte gültig ist, auch mit allen Ausprägungen von Variabilitätskriterien in Verbindung gesetzt wird.
- *Negative Dokumentation:* Bei den variablen Anforderungen werden die Ausprägungen der Variabilitätskriterien, die für diese Anforderung nicht gültig sind, explizit ausgeschlossen.

Tabelle 14: Notationsformen

Anforderung	Spezifische	Positive	Negative
Anforderung A		Limousine Cabrio Offroad	
Anforderung B	Limousine Cabrio	Limousine Cabrio	Offroad
Anforderung C	Offroad	Offroad	Limousine Cabrio

Tabelle 14 präsentiert ein Beispiel für die vorgestellten Dokumentationsformen. *Die Anforderung A ist gültig für alle drei möglichen Karosserieformen (Limousine, Cabrio, Offroad). Die Anforderung B ist nur für die zwei Karosserieformen (Limousine, Cabrio), nicht aber für die Variante Offroad gültig. Die Anforderung C ist nur für die Variante Offroad gültig.*

Die am häufigsten verwendete Notation ist die positive Notation. Unter Berücksichtigung der Anforderungsvielzahl ist es aber nicht immer sinnvoll diese Notationsform zu benutzen. Es ist vorteilhafter die spezifische Notation zu benutzen. Bei der spezifischen Notation werden nur die variablen Anforderungen mit den gültigen Ausprägungen von Variabilitätskriterien in Verbindung gesetzt. Deswegen wird im R2F-Ansatz vorgeschlagen, nur die spezifische Notationsform zu benutzen. Somit kann die Formel 7-2 erweitert werden:

$$a^{var} \text{ \textcircled{=} } m \text{ mit } a^{var} \in ABib, m \in M \quad (7-4)$$

Zur Erläuterung der Vorteile der Dokumentation der Variabilitätsinformationen in den Anforderungsdokumenten nach dem R2F-Ansatz kann das Beispiel der Komponente *Sonnenblende* benutzt werden. *Das Anforderungsdokument beinhaltet 543 Anforderungen. Bisher wurden alle diese Anforderungen jedem neuen Produkt zugeordnet. Für eine neue Baureihe werden typischerweise mehrere Ausstattungsvarianten geplant. Angenommen, dass für die neue Baureihe drei Varianten der Komponente Sonnenblende geplant sind, müsste der Entwickler 1629 (3x543) Entscheidungen bezüglich der Zuordnung von Anforderungen zu den neuen Sonnenblendenvarianten fällen. Mit dem R2F-Ansatz wird hingegen zuerst das Variabilitätsmodell der Komponente Sonnenblende erstellt. Ein Auszug dieses Modells ist in Abbildung 23 vorgestellt und beinhaltet neun Merkmale. Danach müssen 37 variable Anforderungen mit den gültigen Ausprägungen der vier Variabilitätskriterien in Verbindung gesetzt werden. Solange sich das Variabilitätsmodell nicht ändert, ist der Aufwand für diese Zuordnung einmalig. Die genaue Vorgehensweise wird im nächsten Kapitel vorgestellt. Für jede neue Variante der Sonnenblende muss zuerst die neue Konfiguration des Variabilitätsmodells erstellt werden. Diese Konfiguration wird danach als Filter für das Anforderungsdokument benutzt. Das Anwenden dieses Filter ermöglicht es, ein Lastenheft zu erstellen. In Kapitel 10 werden die im Rahmen der durchgeführten Evaluation ermittelten zeitlichen Vorteile durch den R2F-Ansatz präsentiert und diskutiert.*

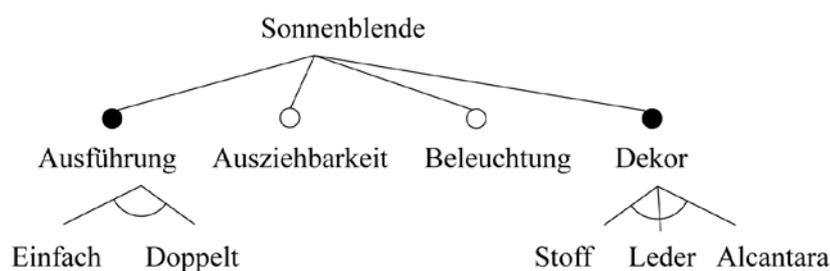


Abbildung 23: Auszug des Variabilitätsmodells der Komponente *Sonnenblende*

7.2 Variantenmanagementprozess

In diesem Kapitel wird der Variantenmanagementprozess nach dem R2F-Ansatz präsentiert. Dieser Prozess besteht aus fünf Schritten, die in Abbildung 24 in einer Übersicht zusammengefasst sind:

- (1) *Variabilitätskriterien (VK) und deren Ausprägungen (AVK) identifizieren:* Die Tätigkeit im Rahmen des ersten Schrittes ist abhängig davon, ob bereits ein Anforderungsdokument existiert oder nicht. Falls bereits ein Anforderungsdokument existiert und nur die Variabilitätskriterien und deren Ausprägungen in dem existierenden Anforderungsdokument identifiziert werden müssen, kann dies mit MIA oder durch ein manuelles Review gemacht werden. Falls das Anforderungsdokument noch nicht existiert, können die variablen Eigenschaften anhand der Produktidee identifiziert werden.
- (2) *Variabilitätsmodell erstellen:* Im zweiten Schritt werden die Variabilitätsmodelle erstellt, die die identifizierten Variabilitätskriterien und Ausprägungen in eine Beziehung zu einander stellt. Dabei müssen nicht nur die hierarchischen Beziehungen zwischen den Merkmalen innerhalb eines Variabilitätsmodells, sondern auch die Beziehungen zwischen Merkmalen in unterschiedlichen Modellen (z.B. zu den globalen Merkmalen) dokumentiert werden. Dafür kann die Vorgehensweise aus Kapitel 6 benutzt werden.

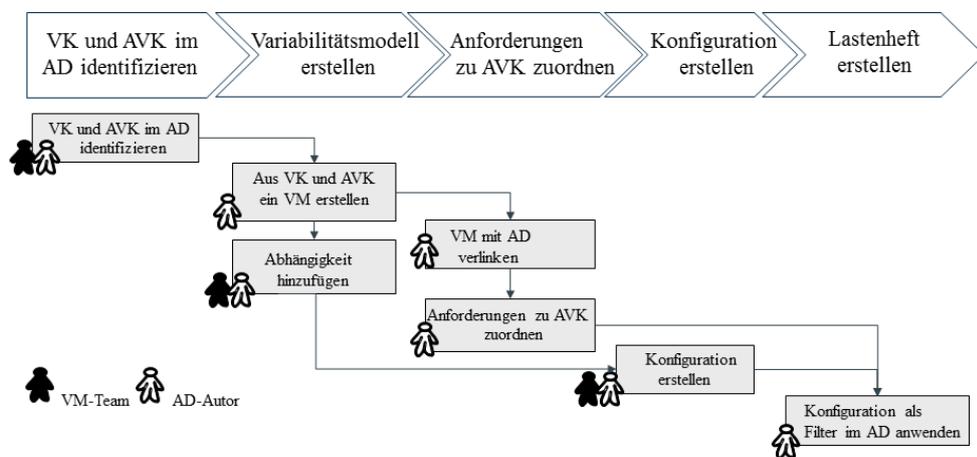


Abbildung 24: R2F-Prozess bei der Einführung des R2F-Ansatzes

- (3) *Variable Anforderungen den AVK zuordnen:* Nachdem das Variabilitätsmodell erstellt wurde, müssen die Merkmale in das Anforderungsdokument als Anforderungsattribute importiert werden. Danach müssen die variablen Anforderungen den Merkmalen zugeordnet werden. Dafür kann entweder ein Review durchgeführt oder der R2F-Zuordnungs-Algorithmus (ZV-Algorithmus) verwendet werden.
- (4) *Konfigurationen erstellen:* Anhand des Variabilitätsmodells können nun die Konfigurationen erstellt werden, die die neuen Produkte beschreiben. Bei der Erstellung einer neuen Konfiguration wird entschieden, welche variablen Eigenschaften das neue Produkt haben wird und welche nicht.
- (5) *Lastenheft erstellen:* Basierend auf der erstellten Zuordnung der Anforderungen zu den Variabilitätskriterien und den erstellten Konfigurationen können jetzt die Lastenhefte für die einzelne Produkte erstellt werden. Die Konfigurationenwerte (welche Merkmale sind Teile der Konfiguration und welche nicht) werden als Filter-Werte für das Anforderungsdokument verwendet.

Es ist empfehlenswert, für den vorgestellten Prozess zwei Rollen zu unterscheiden: den Autor des Anforderungsdokuments und den Variabilitätsmanager. Der Autor dokumentiert und verwaltet die Anforderungen in den Anforderungsdokumenten. Der Variabilitätsmanager erstellt und pflegt die Variabilitätsmodelle.

Den ersten und vierten Schritt im R2F-Prozess müssen der Autor und der Variabilitätsmanager gemeinsam durchführen. Der zweite Schritt sowie der Import von Merkmalen in

das Lastenheft gehören zu den Aufgaben des Variabilitätsmanagers. Der dritte und fünfte Schritt kann vom Autor allein durchgeführt werden.

Wird ein Lastenheft für ein neues Produkt benötigt, müssen nur die Schritte 4 und 5 ausgeführt werden (Abbildung 25). Voraussetzung dafür ist, dass ein neues Produkt keine neuen produktspezifischen Anforderungen besitzt. Ist das nicht der Fall, können die produktspezifischen Anforderungen als variable Anforderungen mit ihrer Zuordnung zu den Merkmalen in das Anforderungsdokument aufgenommen werden.

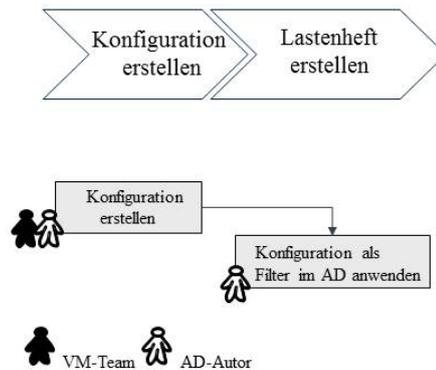


Abbildung 25: R2F-Prozess bei der Lastenhefterstellung

Eine besondere Form produktspezifischer Anforderungen stellen neue Anforderungen dar, die die Innovationsvariabilität beschreiben. Diese Anforderungen können zu Änderungen oder Erweiterungen im Variabilitätsmodell führen. In diesem Fall müssen alle fünf Prozessschritte neu durchgeführt werden.

7.3 Semi-automatische Tool-Unterstützung

In diesem Kapitel werden zwei Algorithmen vorgestellt, die den Benutzer bei der Anwendung des R2F-Prozesses unterstützen können.

Der erste Algorithmus generiert einen Zuordnungsvorschlag (ZV) der variablen Anforderungen zu den Merkmalen. Der zweite Algorithmus (ZÜ) überprüft die durch den Benutzer erstellte Zuordnung der Anforderung zu Merkmalen und gibt dem Benutzer eine Rückmeldung über die getroffenen Zuordnungen. Die grafischen Darstellungen von beiden Algorithmen sind im Anhang III-A und III-C dokumentiert.

7.3.1 Zuordnungsvorschlag (ZV)

Der ZV-Algorithmus sucht in jeder variablen Anforderung im Anforderungsdokument nach Merkmalsnamen aus dem zugehörigen Variabilitätsmodell. Wenn der ZV-Algorithmus einen Merkmalsnamen im Anforderungstext findet, dann wird ein Merkmalsvorschlag generiert.

Falls keine Unterscheidung zwischen gemeinsamen und variablen Anforderungen im Anforderungsdokument existiert, können alle Anforderungen betrachtet werden. Jedoch müssen die Ergebnisse im manuellen Review nachgebessert werden.

Formal aufgeschrieben sieht der ZV-Algorithmus folgendermaßen aus:

$$\forall a \in ABib:$$

$$(\forall m_k \in M: m_k.Name \in a \Rightarrow a \text{ \textcircled{=} } m_k = true) \wedge \\ \left(\forall m_t \in M: m_t.Name \notin a \Rightarrow a \text{ \textcircled{=} } m_t = false \right) \text{ mit } m_k \neq m_t \quad (7-5)$$

7.3.2 Zuordnungsüberprüfung (ZÜ)

Der ZÜ-Algorithmus prüft in dem Anforderungsdokument die existierende, durch das manuelle Review entstandene Zuordnung der Anforderungen zu den Merkmalen des zugehörigen Variabilitätsmodells.

Der ZÜ-Algorithmus überprüft für jede existierende Zuordnung $a \text{ \textcircled{=} } m_k = true$, ob der Name des zugeordneten Merkmals m_k und ob die Namen von anderen nicht zugeordneten Merkmalen m_t in dem Anforderungstext a vorkommen.

$$\forall a \in ABib, m_k, m_t \in M \text{ mit } m_k \neq m_t \quad (7-6)$$

Falls der Merkmalsname $m_k.Name$ in der zugeordneten Anforderung a vorkommt, das heißt $m_k.Name \in a$, dann müssen folgende zwei Fälle betrachtet werden:

- In der zugeordneten Anforderung a kommt der Name von keinem weiteren Merkmal aus dem Variabilitätsmodell vor. In diesem Fall wird die Zuordnung der Anforderung a zum Merkmal m_k als richtig interpretiert, und es werden dem Benutzer keine Meldungen angezeigt.

$$m_t.Name \notin a \Rightarrow a \text{ \textcircled{=} } m_k = true \quad (7-7)$$

- In der zugeordneten Anforderung a kommt der Name von mindestens einem anderen nicht zugeordneten Merkmal m_t aus dem Variabilitätsmodell vor. In diesem Fall wird die Zuordnung der Anforderung a zum Merkmal m_k als richtig interpretiert und es wird zusätzlich ein Zuordnungsvorschlag für die Anforderung a zum Merkmal m_t generiert und dem Benutzer angezeigt.

$$m_t.Name \in a \Rightarrow a \text{ \textcircled{=} } m_k = true \wedge a \text{ \textcircled{=} } m_t = true \quad (7-8)$$

Falls der Merkmalsname $m_k.Name$ in der zugeordneten Anforderung a nicht vorkommt, das heißt $m_k.Name \notin a$, dann müssen zwei weitere Fälle betrachtet werden:

- In dem Anforderungstext wird mindestens ein Name von einem anderem nicht zugeordneten Merkmal aus dem Variabilitätsmodell identifiziert. In diesem Fall wird die Zuordnung der Anforderung a zum Merkmal m_k als falsch interpretiert, und dem Benutzer wird eine Fehlermeldung angezeigt. Zusätzlich wird ein Zuordnungsvorschlag für die Anforderung a zum Merkmal m_t generiert und dem Benutzer angezeigt.

$$m_t.Name \in a \Rightarrow a \text{ \textcircled{=} } m_k = false \wedge a \text{ \textcircled{=} } m_t = true \quad (7-9)$$

- In dem Anforderungstext kommt kein Merkmalsname vor. In diesem Fall wird die Zuordnung der Anforderung a zum Merkmal m_k als falsch interpretiert und dem Benutzer wird eine Fehlermeldung angezeigt.

$$m_t.Name \notin a \Rightarrow a \text{ \textcircled{=} } m_k = false \quad (7-10)$$

7.3.3 Evaluierung der R2F-Algorithmen

Die beiden vorgestellten Algorithmen wurden als Erweiterung der Funktionalität des Anforderungsmanagementtools DOORS umgesetzt.

Die Evaluierung der R2F-Algorithmen wurde anhand von vier Anforderungsdokumenten durchgeführt. Die ersten zwei Datensätze DS1 und DS2 sind identisch mit den Datensätzen aus der Evaluation des MIA-Algorithmus. Für diese zwei Datensätze wurden im Rahmen eines Projektes bei der Daimler AG die Anforderungen zu Merkmalen zugeordnet. Für die Datensätze DS3 und DS4 wurde die Zuordnung zum Zeitpunkt der Evaluation der R2F-Algorithmen noch nicht durchgeführt. Deswegen wurde auf zwei weitere Datensätze DS5 und DS6 zurückgegriffen, welche im Rahmen eines Praktikums an der Universität Ulm (siehe Kapitel 10) von Studenten erstellt wurden.

Tabelle 15: Evaluierungsdatensätze

Anzahl von	DS1	DS2	DS5	DS6
DOORS-Objekten	1397	727	195	234
Anforderungen	1124	543	112	178
Variable Anforderungen	1001	37	-	-

In Tabelle 15 werden die vier Datensätze kurz vorgestellt. In den Datensätzen DS1 und DS2 wurden die variablen Anforderungen gekennzeichnet. In den Datensätzen DS5 und DS6 wurden die variablen Anforderungen nicht gekennzeichnet.

7.3.3.1 Evaluierung des ZV-Algorithmus

Für die Evaluierung des ZV-Algorithmus wurden zuerst die Referenzdatensätze erstellt, in denen die Zuordnung manuell im Rahmen eines Reviews erstellt wurde. Danach wurde die Zuordnung mit Hilfe des ZV-Algorithmus durchgeführt. Zusätzlich wurden zwei alternative Lösungen evaluiert:

- (1) Die erste Variante ist die Dokumentation des Variabilitätsmodells in einem separaten DOORS-Varianten-Modul (R2F-VM). Das DOORS-Varianten-Modul wird im Kapitel 9 vorgestellt.
- (2) Die zweite Variante ist die Dokumentation des Variabilitätsmodells in Form einer Variantenkriterien- und Variantenausprägungen-Liste in einer .txt Datei (R2F-L).

Bei der Evaluierung wurden folgende Werte analysiert:

- *Anzahl der Merkmale:* Wie viele Merkmale aus dem entsprechenden Variabilitätsmodell wurden zugeordnet?
- *Anzahl der Zuordnungen:* Wie viele Zuordnungen von variablen Anforderungen zu Merkmalen wurden erstellt?
- *Anzahl der richtigen Zuordnungen:* Wie viele variablen Anforderungen wurden zu richtigen Merkmalen, welche die Variabilität der Anforderung repräsentieren, zugeordnet?
- *100% korrekte Zuordnungen:* Wie viele variablen Anforderungen wurden korrekt zu allen richtigen Merkmalen zugeordnet?
- *Teilweise korrekte Zuordnungen:* Wie viele variablen Anforderungen wurden zum Teil den richtigen Merkmalen zugeordnet?

- *Anzahl falsche Zuordnungen*: Wie viele variablen Anforderungen wurden falschen Merkmalen zugeordnet?
- *Anzahl fehlende Zuordnungen*: Wie viele Zuordnungen von variablen Anforderungen zu entsprechenden Merkmalen fehlen?

Für die Auswertung der Ergebnisse werden analog zu Kapitel 5 zwei Metriken benutzt: *Trefferquote (R)* und *Genauigkeit (P)*.

Die Ergebnisse der manuellen und automatischen Zuordnung sind in Anhang III-B dokumentiert. Tabelle 16 gibt einen Überblick am Beispiel des Datensatzes DS2. Es werden die Zuordnungsergebnisse aus zwei automatisierten Zuordnungen und einer manuellen Zuordnung gegenübergestellt.

Tabelle 16: Evaluierung des R2F-Zuordnungsalgorithmus für DS2

DS2	Manuell	R2F_VM			R2F_L		
		#	R	P	#	R	P
Anzahl Merkmale	9	4	0,44		4	0,44	
Dauer	01:00	00:01			00:01		
Anzahl Zuordnungen	48	20			7		
Anzahl korrekte Zuordnungen	48	13	0,65	0,27	5	0,71	0,10
- 100%korrekte Zuordnungen	48	8	0,40	0,01	1	0,14	0,02
- teilweise korrekte Zuordnungen	0	5	0,25	0,1	4	0,57	0,08
Anzahl falsche Zuordnungen	0	7	0,35		2	0,29	
Anzahl fehlende Zuordnungen	0	35		0,73	43		0,89

Für DS2 wurden mit R2F_VM und R2F_L jeweils vier aus neun Variabilitätskriterien gefunden:

- Mit R2F_VM wurden 20 Zuordnungen erstellt, davon waren 65% der erstellten Zuordnungen ganz oder teilweise richtig. „Teilweise richtige Zuordnung“ bedeutet, dass nicht alle relevanten Variabilitätskriterien/ Ausprägungen, aber mindestens ein Variabilitätskriterium richtig zugeordnet wurde. 35% der erstellten Zuordnungen waren falsch. Der Vergleich der Ergebnisse des manuellen Reviews mit denen des ZV-Algorithmus zeigte, dass 73% der Zuordnungen nicht erkannt wurden.
- Mit R2F_L wurden nur sieben Zuordnungen erstellt, davon waren nur eine Zuordnung richtig, und vier Zuordnungen teilweise richtig. Außerdem wurden zwei falsche Zuordnungen generiert. Bei dem Vergleich der Ergebnisse des manuellen Reviews mit denen des ZV-Algorithmus wurde festgestellt, dass 89% der Zuordnungen nicht erkannt wurden.

Die Evaluation hat gezeigt, dass die Ergebnisse bei der Anwendung von R2F_VM mindestens zweimal besser sind als bei der Anwendung von R2F_L aber deutlich schlechter als manueller Zuordnung. Der Vorteil der automatisierten Zuordnung liegt in der Bearbeitungszeit.

7.3.3.2 Evaluierung des ZÜ-Algorithmus

Bei der Evaluierung des ZÜ-Algorithmus wurden die Referenz-Datensätze aus der Evaluierung des ZV-Algorithmus benutzt. Bei der Evaluierung des ZÜ-Algorithmus wurden folgende Werte analysiert:

- *Anzahl der Fehlermeldungen:* Wie viele fehlerhafte Zuordnungen von variablen Anforderungen zu falschen Merkmalen, welche die Variabilität der Anforderung nicht repräsentieren, wurden identifiziert?
- *Anzahl richtiger Fehler:* Wie viele Fehlermeldungen sind tatsächlich fehlerhafte Zuordnungen?
- *Anzahl der Zuordnungsvorschläge:* Wie viele zusätzliche Zuordnungsvorschläge von variablen Anforderungen zu richtigen Merkmalen wurden generiert?
- *Richtige Zuordnungsvorschläge:* Wie viele der zusätzlichen Zuordnungsvorschläge sind tatsächlich richtig?

Das ZÜ-Algorithmus wurde anhand der Datensätze DS1, DS2, DS5 und DS6 evaluiert. Die Ergebnisse dieser Evaluierung sind in Tabelle 17 dokumentiert und können folgendermaßen zusammengefasst werden:

- Die Anzahl der Fehlermeldungen im Vergleich zu der Anzahl tatsächlicher Fehler lag beim schlechtesten Ergebnis bei 23 zu 1.
- Es wurde ein Fehler gefunden, welcher während der Vorbereitung der Datensätze durch den manuellen Review Prozess nicht gefunden wurde.
- Außerdem wurden zwischen 1 und 34 korrekte Zuordnungsvorschläge generiert.

Tabelle 17: Evaluierung des ZÜ-Algorithmus.

	DS1	DS2	DS5	DS6
Anzahl Fehlermeldungen	23	1	0	4
Anzahl richtige Fehler	1	0	0	0
Anzahl Zuordnungsvorschläge	251	10	6	7
Anzahl richtige Zuordnungsvorschläge	34	1	2	4

7.3.3.3 Ergebnisse der Evaluierung

Für die Zuordnung der variablen Anforderungen zu den Merkmalen existiert zurzeit sowohl in der Forschung als auch in der Praxis nur eine Lösung – das manuelle Review der Anforderungsdokumente.

Die vorliegende Arbeit bietet eine neuartige pragmatische Lösung der semi-automatischen Unterstützung sowohl der Zuordnung der variablen Anforderungen zu den zuvor identifizierten Merkmalen als auch der Überprüfung der existierenden Zuordnung mit Hilfe einer Text-Analyse.

Die Ergebnisse der Evaluierung beider Algorithmen haben gezeigt, dass diese zwar zur Unterstützung bei der Zuordnungserstellung bzw. bei der Zuordnungsüberprüfung benutzt werden können, aber die Ergebnisse nicht ungeprüft übernommen werden, sondern lediglich als Startpunkt für einen manuellen Review-Prozesses benutzt werden.

Als Alternative zu den vorgestellten R2F-Algorithmen können Gruppierungsansätze (eng. Clustering) verwendet werden. Die Gruppierungsansätze fassen die Elemente einer Menge nach bestimmten Kriterien zu Gruppen (engl. Cluster) zusammen. In [Wigg97] werden vier Kategorien von Gruppierungsansätze vorgestellt: graphentheoretische, Konstruktions-, Optimierungs- und hierarchische Algorithmen. Diese vier Kategorien bilden die Basis für die meisten konkreten Algorithmen, die jeweils Hybrid-Lösungen dieser Basis-Kategorien darstellen. Für Zuordnungsaufgaben können allerdings nur die Algorithmen angewendet werden, die nicht auf der Häufigkeit des Auftretens eines Begriffs im analysierten Dokument oder Korpus aufbauen (z.B. TextQuest [IIEO2001] oder Terminology Driven Literature Mining [NeSA03]). Der Grund dafür ist, dass die Häufigkeit des Auf-

treten eines Terms in einem Anforderungsdokument oder in einem Referenzdokument keine Aussagen über die Zugehörigkeit zu den variablen Eigenschaften dieses Terms erlaubt.

7.4 Diskussion

In diesem Unterkapitel wird der vorgestellte R2F-Ansatz anhand der im Kapitel 3.2.1 festgelegten Kriterien auf seine Praxistauglichkeit bewertet. Anschließend wird die Erfüllung der Anforderungen an einen Variabilitätsmanagementansatz, die im Kapitel 3.3 festgelegt wurden, analysiert. Parallel dazu wird der R2F-Ansatz mit den alternativen Ansätzen für die Dokumentation der Variabilitätsinformationen in den Anforderungsdokumenten, die im Kapitel 2.3 vorgestellt wurden, bezüglich der Erfüllung der Anforderungen verglichen.

Kriterien aus der Praxis

Die Bewertung der Praxistauglichkeit des R2F-Ansatzes erfolgt nach folgenden Kriterien aus Kapitel 3:

(K1) Gleiche Anforderungen werden nur einmal dokumentiert: Dieses Kriterium wird durch den R2F-Ansatz nicht verletzt, da es nicht notwendig ist, gleiche Anforderungen mehrmals zu dokumentieren.

(K2) Die Zuordnung der Anforderungen zu den Varianten ist erkennbar: Bei dem R2F-Ansatz werden nur die variablen Anforderungen den variablen Produkteigenschaften zugeordnet. Somit ist klar, dass die nicht zugeordneten Anforderungen invariabel und somit für alle Produkte gültig sind. Die zugeordneten Anforderungen sind nur für die Produkte gültig, die die ausgewählten variablen Produkteigenschaften besitzen.

(K3) Die Zuordnung der Anforderungen zu den Varianten ist ohne spezielle Vorkenntnisse verständlich: Die Dokumentation der variablen Anforderungen zu den variablen Eigenschaften geschieht durch die Auswahl der Merkmalsnamen, die in natürlicher Sprache dokumentiert sind. Somit ist dieses Kriterium auch erfüllt.

(K4) Die initiale Zuordnung der Anforderungen beim Hinzufügen neuer Varianten (z.B. neue Baureihe) ist erkennbar: Die neuen Produkte werden nicht mehr im Anforderungsdokument, sondern durch Auswahl der variablen Eigenschaften in dem zugehörigen Variabilitätsmodell definiert. Somit sind für ein neues Produkt sofort alle gültigen Anforderungen für die bereits dokumentierten variablen Eigenschaften bekannt.

(K5) Der Aufwand für die Zuordnung von Anforderungen zu Varianten wird reduziert: Durch die ausgewählte Notationsform (Zuordnung nur der variablen Anforderungen zu Variabilitätskriterien und deren Ausprägungen) wird der Aufwand für die Zuordnung deutlich reduziert.

(K6) Die Beziehungen zwischen variablen Eigenschaften sind erkennbar: Diese Informationen werden im Variabilitätsmodell und nicht im Anforderungsdokument festgehalten. Im Variabilitätsmodell wird anhand von Konfigurationsmechanismen sichergestellt, dass nur die gültigen Kombinationen der Produkteigenschaften in einem Produkt vorkommen können. Mit Hilfe der Beziehungen zwischen dem Variabilitätsmodell und dem Anforderungsdokument wird sichergestellt, dass nur die gültigen Kombinationen von Anforderungen in dem entsprechenden Lastenheft vorkommen.

(K7) Die Beziehungen zwischen variablen Eigenschaften sind prüfbar: Die Beziehung zwischen einzelnen variablen Eigenschaften können im Variabilitätsmodell jederzeit formal überprüft werden.

Somit erfüllt der vorgestellte R2F-Ansatz alle im Kapitel 3.2.1 festgelegten Kriterien.

Erfüllung der Anforderungen an einen VM-Ansatz

Im nächsten Schritt wird gezeigt, dass der R2F-Ansatz auch alle Anforderungen zum Variantenmanagement (VM) gemäß Abschnitt 3.3 erfüllt.

An den R2F-Ansatz wurden die Anforderungen A1, A2, A3, A6 und A10 gestellt. Die alternativen Dokumentationsansätze, die im Kapitel 2.3 vorgestellt wurden, liefern nur die Angaben zur Dokumentation der Variabilität in Anforderungsdokumenten. Bei alternativen Ansätzen wurde nicht vorgestellt, wie die Variabilität verwaltet werden soll. Somit ist die Beurteilung der Erfüllung dieser fünf Anforderungen durch die alternativen Ansätze nicht gegeben.

Die Bewertung der Anforderungen des VM-Ansatzes erfolgt nach folgenden Kriterien:

(A1) *Der Variantenmanagementansatz muss fähig sein, die Variabilität in natürlichsprachlich formulierten Anforderungen zu identifizieren, zu dokumentieren und zu verwalten:* Diese Anforderung ist durch den R2F-Ansatz komplett erfüllt.

(A2) *Der Variantenmanagementansatz muss skalierbar sein, d.h. der Ansatz muss sowohl eine kleine als auch eine große Variabilität in gleichem Maße unterstützen (von nur einigen bis hin zu mehreren hundert Variabilitätskriterien):* Um diese Anforderung zu erfüllen, muss der konfigurationsgekoppelte Variabilitätsmodellierungsansatz aus Kapitel 6 verwendet werden.

(A3) *Variabilitätsinformationen müssen auf Basis bestehender Anforderungsdokumente identifizierbar sein:* Diese Anforderung wird durch den R2F-Ansatz erfüllt.

(A6) *Die Dokumentation von Variabilitätsinformationen muss durch ein Werkzeug unterstützt werden. Wichtig ist dabei, dass das Variantenmanagementwerkzeug mit den Anforderungsmanagementwerkzeugen kompatibel ist:* Wie der R2F-Ansatz durch ein Tool unterstützt werden kann, wird im Kapitel 8 detailliert vorgestellt.

(A10) *Der Dokumentationsansatz darf die Änderung von Variabilitätsmodellen aus den Anforderungsdokumenten heraus nicht ermöglichen:* Der R2F-Ansatz erfüllt diese Anforderung. Außerdem wurde diese Anforderung bei der Definition der Anforderungen und Szenarien für das VM-Tool berücksichtigt.

Für alle weiteren Anforderungen aus Kapitel 3.3 ist ein Vergleich der Anforderungserfüllung zwischen den unterschiedlichen Ansätzen möglich. Es werden folgende fünf Dokumentationsansätze mit dem R2F-Ansatz verglichen und bewertet:

- Matrix-Ansatz;
- Ansatz Sequentielle Reihenfolge (SR);
- Ansatz Objekt Hierarchie (OH);
- Ansatz Nachvollziehbarkeitslink (N-Link);
- Attribut-Ansatz.

Die Ergebnisse des Vergleichs werden in Tabelle 18 in einer Übersicht zusammengefasst. Dabei bedeutet „+“: Anforderung ist erfüllt, „-“: Anforderung ist nicht erfüllt, „0“: die Anforderung kann zwar erfüllt werden, aber nur in bestimmtem Kontext (z.B. als Teil von eines VM-Ansatzes).

(A4) *Die Dokumentation von Variabilitätsinformationen muss leicht erlernbar sein:* Diese Anforderung wird von allen vorgestellten Dokumentationsansätzen erfüllt.

(A5) *Die Beziehungen zwischen einzelnen Variabilitätskriterien müssen dokumentierbar und prüfbar sein:* Diese Anforderungen werden vom R2F-Ansatz durch die Einbindung des orthogonalen Variabilitätsmodells erfüllt. Auch die Attribut- und N-Link-Ansätze können diese Anforderung erfüllen, wenn sie um die Variabilitätsmodelle erweitert werden.

Tabelle 18: Erfüllung der Anforderungen an VM-Ansatz

	R2F	Matrix	OH	SR	N-Links	Attribut
A4	+	+	+	+	+	+
A5	+	-	-	-	0	0
A7	+	-	-	-	-	0
A8	+	+	0	0	+	+
A9	+	+	-	-	0	+
A11	+	-	-	-	0	0
A12	+	+	+	+	-	+
A13	+	-	-	-	-	-

(A7) *Der Dokumentationsansatz in Anforderungsdokumenten muss sowohl für Anforderungsdokumente mit mehreren 10.000 Anforderungen als auch für Anforderungsdokumente mit einigen 100 Anforderungen anwendbar sein:* Der R2F-Ansatz bietet den effizientesten Weg, die Anforderungen mit einer Variabilität zu verbinden. Alle anderen vorgestellten Ansätze erfordern deutlich mehr Aufwand für die gleiche Verknüpfung der Anforderungen mit variablen Informationen.

(A8) *Gleiche Anforderungen dürfen nur einmal dokumentiert werden:* Diese Anforderung wird von den R2F-, Matrix-, Attribut- und N-Link-Ansätzen erfüllt. Die zwei anderen Ansätze erfüllen diese Anforderung nur gelegentlich.

(A9) *Der Dokumentationsansatz darf zu keinen Strukturänderungen in den bestehenden Anforderungsdokumenten führen:* Diese Anforderung wird von den R2F-, Matrix-, N-Link und Attribut-Ansätzen erfüllt. Die OH- und SR-Ansätze erfüllen diese Anforderung nicht, weil zusätzliche Hierarchieobjekte (Variationspunkte) eingeführt werden.

(A11) *Der Dokumentationsansatz muss den Aufwand für die Zuordnung der Anforderungen zu den Variabilitätskriterien minimieren:* Nur der R2F-Ansatz bietet eine semi-automatische Unterstützung der Zuordnung. Außerdem wird durch die ausgewählte Notationsform die Anzahl der benötigten Zuordnungen minimal gehalten.

(A12) *Die Zuordnung der Anforderungen zu den Variabilitätskriterien muss ohne spezielles Wissen im Bereich des Variantenmanagements leicht erkennbar sein:* Diese Anforderung wird von allen Ansätzen mit Ausnahme vom N-Link-Ansatz erfüllt.

(A13) *Die Zuordnung der Anforderungen zu den Varianten muss automatisch bzw. semi-automatisch überprüfbar sein:* Nur der R2F-Ansatz bietet die semi-automatische Überprüfung der Zuordnung.

Aus dem vorgestellten Vergleich der Erfüllung der Anforderungen ist ersichtlich, dass nur der R2F-Ansatz als einziger Ansatz alle Anforderungen erfüllt.

Kapitel 8

Unterstützung der Konfigurationserstellung

Kapitel 8 beschreibt wie der Konfigurationserstellungsprozess unterstützt werden kann. Bei der Konfigurationserstellung muss der Produktmanager bzw. Spezifikationsautor eine Menge Entscheidungen treffen, um die Eigenschaften eines neuen Produktes auszuwählen. Bei hoch-komplexen Produkten kann die Anzahl von Entscheidungen im Bereich von 10.000 liegen. In diesem Kapitel wird die Vorgehensweise zur Berechnung der optimalen Reihenfolge von den Entscheidungsfragen basierend auf den Beziehungen zwischen einzelnen Merkmalen in einem oder mehreren Variabilitätsmodellen vorgestellt.

8.1 Unterstützung durch optimale Fragenfolge

Der Konfigurationsprozess eines Variabilitätsmodells basiert auf einer Menge Entscheidungen, um die Eigenschaften eines Produktes bzw. die Spezifikation zu bestimmen. Viele Entscheidungen beim Konfigurationsprozess sind nicht unabhängig. Oft verkleinert eine getroffene Entscheidung die Anzahl der restlichen, noch zu treffenden Entscheidungen. Zum Beispiel, in dem Variabilitätsmodell ist festgehalten, dass das Merkmal A das Merkmal B ausschließt. Wird in einer Konfiguration das Merkmal A ausgewählt, dann muss keine Entscheidung über das Merkmal B getroffen werden, weil es implizit abgewählt wird.

Die Ergebnisse der Studie von Rabiser et al. [RaGD09] zeigen, dass viele Befragte sich eine Unterstützung während des Konfigurationsprozesses wünschen. Diese Unterstützung kann sowohl die Bereitstellung der für die Entscheidung notwendigen Informationen sein als auch die Minimierung der Anzahl von Entscheidungen für die Konfigurationserstellung durch die geführte Fragestellung.

In der vorliegenden Arbeit wurde ein Ansatz zur Minimierung der notwendigen Entscheidungen bei der Erstellung einer Konfiguration eines Variabilitätsmodells erarbeitet. Diese minimale Anzahl der Entscheidungen wird in der vorliegenden Arbeit die *optimale Entscheidungsfolge* genannt.

Definition 8-1: Optimale Entscheidungsfolge

Die optimale Entscheidungsfolge erlaubt mit minimaler Anzahl an Entscheidungen eine gewünschte gültige Konfiguration des Variabilitätsmodells zu erstellen.

Der erarbeitete Ansatz zur Berechnung der optimalen Entscheidungsfolgen kann sowohl für die Merkmalsmodelle nach FODA als auch für die Variabilitätsmodelle, welche im Kapitel 2 definiert wurden, verwendet werden. Deswegen werden in nächsten Kapiteln die Begriffe Merkmal und Merkmalsmodell entsprechend den Definitionen (1-12) und (2-5) benutzt.

Um die optimale Entscheidungsfolge zu bestimmen, werden die Abhängigkeiten zwischen einzelnen Merkmalen und die Auswahlwahrscheinlichkeit für jedes Merkmal in dem Merkmalsmodell berechnet. Die genaue Vorgehensweise wird im Kapitel 8.2 detailliert vorgestellt.

8.2 Bestimmung der optimalen Entscheidungsfolge

Für die Berechnung der optimalen Entscheidungsfolge eines Merkmalsmodells MM müssen für jedes Merkmal m folgende Werte berechnet werden:

- (1) Die Anzahl $R^e(m)$ der von m ausgehenden „schließt_aus“-Beziehungen $r^{exclude}$

$$R^e(m) = |\{(m' \in MM) | (m, m') \in r^{exclude}\}| \quad (8-1)$$

- (2) Die Anzahl $R^n(m)$ der in m eingehenden „benötigt“-Beziehungen r^{needs} :

$$R^n(m) = |\{(m' \in MM) | (m', m) \in r^{needs}\}| \quad (8-2)$$

- (3) Die Auswahl-Wahrscheinlichkeit $p(m)$.

Eine optimale Entscheidungsfolge kann nur in Merkmalsmodellen vorliegen, welche außer hierarchischen auch Abhängigkeitsbeziehungen haben. Falls in einem Merkmalsmodell keine Abhängigkeitsbeziehungen vorkommen, so ist jede Entscheidungsfolge optimal.

Definition 8-2: Einstiegspunktmerkmal

Ein Einstiegspunktmerkmal ist ein Merkmal in dem Variabilitätsmodell, welches (i) ein Vorfahre des betrachteten Merkmals und (ii) erstes Merkmal in dem relevanten Teilbaum ist, über welches eine Entscheidung getroffen werden muss. Die beiden Voraussetzungen (i) und (ii) sind notwendig, weil ein Merkmal nur ausgewählt werden kann, wenn alle Vorfahren dieses Merkmals bereits ausgewählt wurden.

Definition 8-3: Voraussetzungsmerkmal

Ein Voraussetzungsmerkmal ist ein Merkmal, welches die Senke der „benötigt“-Beziehungen ist. Der Grund dafür ist, dass das Merkmal m nur ausgewählt werden kann, wenn alle Voraussetzungsmerkmale ausgewählt wurden.

Um die optimale Entscheidungsfolge in einem Merkmalsmodell mit vorhandenen Abhängigkeitsbeziehungen zu bestimmen, müssen folgende Schritte durchgeführt werden:

Schritt 1: In dem Merkmalsmodells MM wird nach einem Merkmal m gesucht, welches an der maximalen Anzahl der „schließt_aus“-Beziehungen beteiligt ist. Der Grund dafür

ist, dass die Auswahl eines Merkmals, welches „schließt_aus“-Beziehungen zu anderen Merkmalen hat, die Anzahl der benötigten Entscheidungen bezüglich der abhängigen Merkmale reduziert.

- Falls so ein Merkmal m existiert, so muss zuerst überprüft werden, ob Merkmal m die „benötigt“-Beziehungen zu anderen Merkmalen hat:
Falls solche „benötigt“-Beziehungen existieren, so muss zuerst die Entscheidung bezüglich der Voraussetzungsmerkmale getroffen werden. Dabei muss die erste Frage nach dem Einstiegspunktmerkmal für ein Voraussetzungsmerkmal sein.
Falls keine Voraussetzungsmerkmale existieren oder alle Voraussetzungsmerkmale ausgewählt wurden, dann wird nach dem Merkmal gefragt, welches als Einstiegspunktmerkmal bestimmt wird.
- Falls keine Merkmale mit „schließt_aus“-Beziehungen vorhanden sind, so wird zum nächsten Schritt übergegangen.
- Sind in dem betrachteten Merkmalsmodell mehrere Merkmale mit der gleichen Anzahl „schließt_aus“-Beziehungen vorhanden, so muss für alle diese Merkmale die Analyse aus dem Schritt 2 gemacht werden.

Schritt 2: In dem Merkmalsmodell MM wird nach einem Merkmal m gesucht, in welches die maximale Anzahl der „benötigt“-Beziehungen eingehen.

- Falls so ein Merkmal m mit der maximalen Anzahl der eingehenden „benötigt“-Beziehungen existiert, so muss zuerst überprüft werden, ob dieses Merkmal auch die ausgehenden „benötigt“-Beziehungen hat:
Falls ja, so muss zuerst die Entscheidung bezüglich der Voraussetzungsmerkmale des Merkmals m getroffen werden. Dabei muss die erste Entscheidung bezüglich des Einstiegspunktmerkmals für ein Voraussetzungsmerkmal sein.
Falls keine Voraussetzungsmerkmale existieren oder alle Voraussetzungsmerkmale ausgewählt wurden, dann wird nach dem Merkmal gefragt, welches als Einstiegspunktmerkmal bestimmt wird.
- Falls mehrere Merkmale mit der gleichen Anzahl von „benötigt“-Beziehungen vorhanden sind, so muss für alle dieser Merkmale die Auswahl-Wahrscheinlichkeit berechnet werden (Schritt 3).
- Falls es keine Merkmale weder mit „schließt_aus“- noch mit „benötigt“-Beziehungen gibt, so ist jede Entscheidungsfolge optimal.

Schritt 3: Zwischen den Merkmalen mit gleicher Anzahl den „schließt_aus“- und/oder „benötigt“-Beziehungen wird nach dem Merkmal m gesucht, welches die größte Auswahlwahrscheinlichkeit hat. Um die Auswahlwahrscheinlichkeit eines Merkmals zu berechnen, muss zuerst die Anzahl aller möglichen Kombinationen (ohne Berücksichtigung der Abhängigkeitsbeziehungen) für das Merkmalsmodell berechnet werden (8.2.1). Danach kann die Auswahl-Wahrscheinlichkeit für jedes Merkmal in dem betrachteten Merkmalsmodell berechnet werden (8.2.2).

- Falls so ein Merkmal m existiert, so muss zuerst überprüft werden, ob dieses Merkmal die Voraussetzungsmerkmale hat. Falls Voraussetzungsmerkmale existieren, dann wird als erstes nach dem Merkmal gefragt, welches als Einstiegspunktmerkmal für das Voraussetzungsmerkmal bestimmt wird.
- Falls keine Voraussetzungsmerkmale existieren oder alle Voraussetzungsmerkmale ausgewählt wurden, dann wird nach dem Merkmal gefragt, welches als Einstiegspunkt bestimmt wird.
- Falls mehrere Merkmale mit der gleichen Auswahlwahrscheinlichkeit existieren, so kann nach einem entscheidungsrelevanten Vorfahren eines dieser Merkmale gefragt werden.

In den nächsten Abschnitten wird vorgestellt, wie die Berechnungen für ein Merkmalsmodell durchgeführt werden können. Die davor skizzierte Vorgehensweise ist im Anhang IV-A in Form einer Entscheidungstabelle vorgestellt.

8.2.1 Berechnung der lokalen Konfigurationsanzahl

Um die Auswahlwahrscheinlichkeit für jedes Merkmal in einem Merkmalsmodell MM zu berechnen, muss zuerst die lokale Konfigurationsanzahl für alle Eltern-Merkmale und der Wurzel berechnet werden.

Definition 8-4: Lokale Konfigurationsanzahl

Die lokale Konfigurationsanzahl eines Merkmals in einem Merkmalsmodell beschreibt in wie vielen verschiedenen Ausprägungen des Teil-Modells dieses Merkmal vorhanden ist.

Die Berechnung der lokalen Konfigurationsanzahl hängt von der Anzahl der Hierarchieebenen des betrachteten Merkmals ab. Die Hierarchieebenen in einem Merkmalsmodell werden von der Wurzel an (Ebene 1) bis zur letzten Ebene (Ebene N) durchnummeriert. Die Abbildung 26 verdeutlicht die Benennung von Hierarchieebenen in einem Merkmalsmodell.

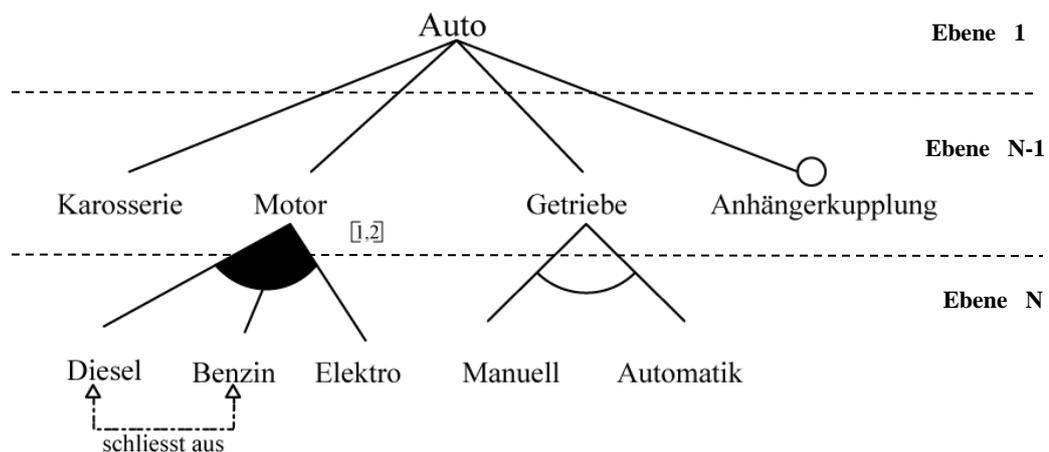


Abbildung 26: Hierarchieebenen in einem Merkmalmodell

Die erste Ebene, auf welcher die lokale Konfigurationsanzahl berechnet werden kann, ist die vorletzte Ebene (N-1). Auf dieser Ebene können folgende vier Basisfälle vorkommen, weil es vier Merkmalstypen gibt:

- (F1) Alle Kinder-Merkmale des Merkmals m sind obligatorische Merkmale km^m .
- (F2) Alle Kinder-Merkmale des Merkmals m sind optionale Merkmale km^o .
- (F3) Alle Kinder-Merkmale des Merkmals m sind alternative Merkmale km^{xor} .
- (F4) Alle Kinder-Merkmale des Merkmals m sind auswählbare Merkmale km^{or} . Dabei ist die Kardinalität der Auswahl $Card^{or}$ bekannt.

Zusätzlich zu den vorgestellten Basisfällen können die Kinder-Merkmale des Merkmals m von mehreren Typen ($km^m, km^o, km^{or}, km^{xor}$) sein. Dabei sind folgende 11 Kombinationen möglich:

- A Die Kombinationen von jeweils zwei verschiedenen Merkmalstypen:
 $(km^m, km^o), (km^m, km^{or}), (km^m, km^{xor}), (km^o, km^{or}),$
 $(km^o, km^{xor}), (km^{or}, km^{xor}), .$

- B Die Kombinationen von jeweils drei verschiedenen Merkmalstypen:
 $(km^m, km^o, km^{or}), (km^m, km^o, km^{xor}), (km^m, km^{or}, km^{xor}),$
 $(km^o, km^{or}, km^{xor})$.
- C Die Kombination von allen vier verschiedenen Merkmalstypen
 $(km^m, km^o, km^{or}, km^{xor})$.

Um die lokale Konfigurationsanzahl für ein Merkmal m zu berechnen, bei welchem die Kinder-Merkmale von den unterschiedlichen Typen sind, werden Hilfsmerkmale hm pro Kinder-Merkmalstyp eingeführt. Die Einführung von Hilfsmerkmalen ermöglicht die Berechnung der lokalen Konfigurationsanzahl anhand der vier Basisfälle F1-F4.

Die Formeln zur Berechnung der lokalen Konfigurationsanzahl auf der vorletzten Ebene (N-1) anhand dieser vier Basisfälle werden im Abschnitt 8.2.1.1 vorgestellt. Um die Formeln zur Berechnung der Konfigurationsanzahl zu verdeutlichen wird ein abstraktes Beispiel benutzt, welches in der Abbildung 27 präsentiert ist.

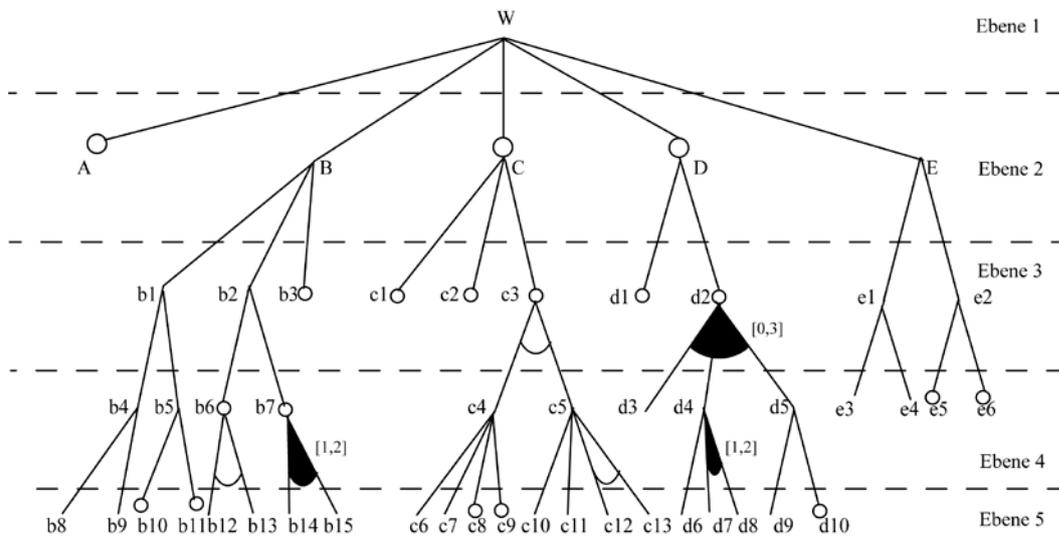


Abbildung 27: Beispielsmerkmalsmodell

Um die lokale Konfigurationsanzahl für ein Merkmal m auf einer höheren Hierarchieebene zu berechnen, müssen die Formeln der Ebene N-1 angepasst werden. Die notwendige Anpassungen und die zugehörige Formeln werden im Abschnitt 8.2.1.2 vorgestellt. Nachdem die lokale Konfigurationsanzahl für jeden Teilbaum des Merkmalsmodells MM bestimmt wurde, kann die Anzahl der Konfigurationen für ein komplettes Merkmalsmodell berechnet werden.

8.2.1.1 Berechnung der lokalen Konfigurationsanzahl auf Ebene N-1

Für die Berechnung der Basisfälle (F1-F4) können folgende Formeln verwendet werden.

Fall 1: Die lokale Konfigurationsanzahl $\#var(m)$ in dem Fall, dass alle n Kinder-Merkmale km_i^m des Eltern-Merkmals m obligatorisch sind, ist gleich 1.

$$\forall m, km_i^m \in M \wedge (m, km_i^m) \in r^{kind} : \#var(m) = 1$$

mit $1 \leq i \leq n, n \in \mathbb{N}$ (8-3)

In dem Beispiel aus der Abbildung 27 ist dieser Fall für das Merkmal $b4$ relevant. Die lokale Konfigurationsanzahl ist gleich 1:

$$\#var(b4) = 1$$

Fall 2: Die lokale Konfigurationsanzahl $\#var(m)$ in dem Fall, dass alle n Kinder-Merkmale km_i^o des Eltern-Merkmals m optional sind, kann anhand folgender Formel (Menge aller Kombinationen) berechnet werden:

$$\forall m, km_i^o \in M \wedge (m, km_i^o) \in r^{kind}: \#var(m) = \sum_{i=0}^n \frac{n!}{i!(n-i)!}$$

mit $1 \leq i \leq n, n \in \mathbb{N}$ (8-4)

In dem Beispiel ist dieser Fall für das Merkmal $b5$ relevant. Die lokale Konfigurationsanzahl $\#var(b5)$ wird folgendermaßen berechnet:

$$\#var(b5) = \sum_{i=0}^2 \frac{2!}{i!(2-i)!} = \frac{2!}{0!(2-0)!} + \frac{2!}{1!(2-1)!} + \frac{2!}{2!(2-2)!} = 4$$

Fall 3: Die lokale Konfigurationsanzahl $\#var(m)$ in dem Fall, dass alle n Kinder-Merkmale km_i^{or} des Eltern-Merkmals m alternative Merkmale sind, entspricht der Anzahl der alternativen Kinder-Merkmale:

$$\forall m, km_i^{xor} \in M \wedge (m, km_i^{xor}) \in r^{kind}: \#var(m) = n$$

mit $1 \leq i \leq n, n \in \mathbb{N}$ (8-5)

In der Abbildung 27 ist dieser Fall für das Merkmal $b6$ relevant. Die lokale Konfigurationsanzahl für das Merkmal $b6$ ist gleich der Anzahl der Kinder-Merkmale:

$$\#var(b6) = 2$$

Fall 4: Die lokale Konfigurationsanzahl $\#var(m)$ in dem Fall, dass alle Kinder Merkmale km_i^{or} des Eltern-Merkmals m auswählbare Merkmale mit der angegebenen Auswahlkardinalität $Card^{or} = [min, max]$ sind, wird folgendermaßen berechnet:

$$\forall m, km_i^{or} \in M \wedge (m, km_i^{or}) \in r^{kind} : \#var(m) = \sum_{i=min}^{max} \frac{n!}{i!(n-i)!}$$

mit $[min, max] = Card^{or} \ 0 \leq min \leq max \leq n, 1 \leq i \leq n, n \in \mathbb{N}$ (8-6)

In dem Beispiel ist dieser Fall für das Merkmal $b7$ mit der Auswahlkardinalität $Card^{or}(b7) = [1,2]$ relevant. Die lokale Konfigurationsanzahl $\#var(b7)$ wird folgendermaßen berechnet:

$$\#var(b7) = \sum_{i=1}^2 \frac{2!}{i!(2-i)!} = \frac{2!}{1!(2-1)!} + \frac{2!}{2!(2-2)!} = 3$$

Fall 5: Zur Berechnung der lokalen Konfigurationsanzahl $\#var(m)$ für ein Merkmal m , dessen Kinder-Merkmale von verschiedenen Typen sind, werden die Hilfsmerkmale hm eingeführt.

$$\forall m, km_i^m, km_j^o, km_k^{or}, km_l^{xor} \in M:$$

$$(m, km_i^m) \in r^{kind} \wedge (m, km_j^o) \in r^{kind} \wedge (m, km_k^{or}) \in r^{kind} \wedge (m, km_l^{xor}) \in r^{kind}$$

mit $0 \leq p, q, s, t$ und $p + q + s + t = n$
 $1 \leq i \leq p, 1 \leq j \leq q, 1 \leq k \leq s, 1 \leq l \leq t$

$$\exists km_i^m \in M \Rightarrow \exists hm^m \in M: (m, hm^m) \in r^{kind} \wedge (hm^m, km_i^m) \in r^{kind}$$

$$\begin{aligned} \exists km_j^o \in M &\Rightarrow \exists hm^o \in M: (m, hm^o) \in r^{kind} \wedge (hm^o, km_j^o) \in r^{kind} \\ \exists km_k^{or} \in M &\Rightarrow \exists hm^{or} \in M: (m, hm^{or}) \in r^{kind} \wedge (hm^{or}, km_k^{or}) \in r^{kind} \\ \exists km_l^{xor} \in M &\Rightarrow \exists hm^{xor} \in M: (m, hm^{xor}) \in r^{kind} \wedge (hm^{xor}, km_l^{xor}) \in r^{kind} \end{aligned} \quad (8-7)$$

Ein Hilfsmerkmal hm ist immer ein obligatorisches Merkmal und wird zwischen dem Merkmal m und dessen Kinder-Merkmalen km hinzugefügt, sodass ein Hilfsmerkmal hm ein neues Kinder-Merkmal von m und ein neues Eltern-Merkmal von allen Kinder-Merkmalen km eines gleichen Typs wird.

Die Abbildung 28 verdeutlicht diese Erweiterung des Beispiel- Merkmalsmodells um die Hilfsmerkmale für die Merkmale $c4$ und $c5$.

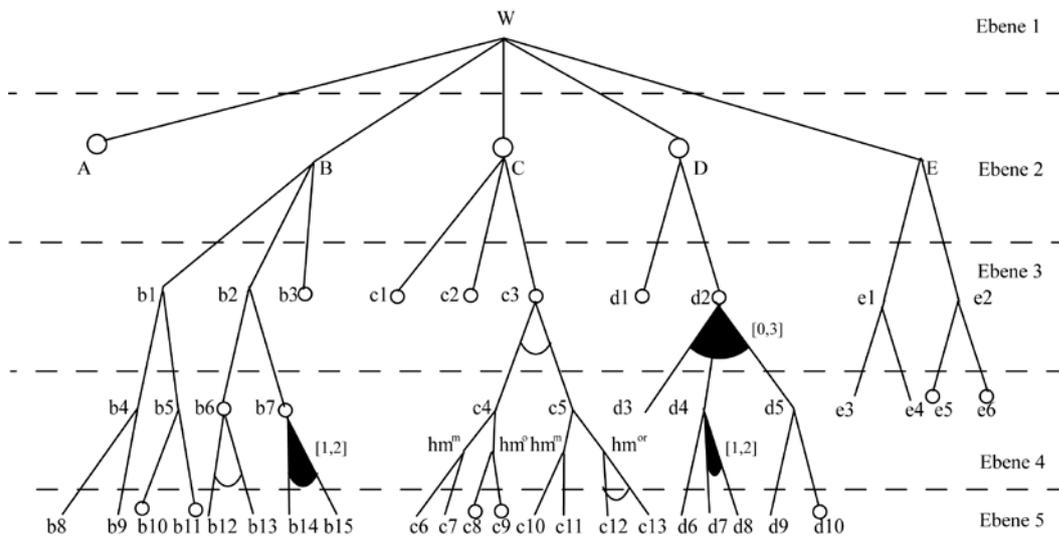


Abbildung 28: Erweiterung des Beispiels um Hilfsmerkmale

Zur Berechnung der lokalen Konfigurationsanzahl für das Merkmal m muss zuerst die Anzahl der lokalen Konfigurationen pro Hilfsmerkmal $\#var(hm)$ berechnet werden:

1. Für das Hilfsmerkmal hm^m , welches die obligatorischen Kinder-Merkmale km^m gruppiert, entspricht die Anzahl der lokalen Konfigurationen 1:

$$\#var(hm^m) = 1 \quad (8-8)$$

2. Für das Hilfsmerkmal hm^o , welches die q optionalen Kinder-Merkmale km^o gruppiert, kann die Anzahl der lokalen Konfigurationen $\#var(hm)$ anhand der angepassten Formel 8-4 berechnet werden:

$$\#var(hm^o) = \sum_{j=0}^q \frac{q!}{j! (q-j)!} \quad (8-9)$$

3. Für das Hilfsmerkmal hm^{xor} , welches die s alternativen Kinder-Merkmale km^{xor} gruppiert, entspricht die Anzahl der lokalen Konfigurationen $\#var(hm)$ der Anzahl von alternativen Merkmalen:

$$\#var(hm^{xor}) = s \quad (8-10)$$

4. Für das Hilfsmerkmal hm^{or} , welches die t auswählbaren Kinder-Merkmale km^{or} gruppiert, kann die Anzahl der lokalen Konfigurationen $\#var(hm)$ anhand der angepassten Formel 8-5 berechnet werden.

$$\#var(hm^{or}) = \sum_{l=\min}^{\max} \frac{t!}{l!(t-l)!} \quad (8-11)$$

Die lokale Konfigurationsanzahl für das Merkmal $\#var(m)$ kann unter Berücksichtigung der Anzahl der lokalen Konfigurationen für alle eingeführten Hilfsmerkmale $\#var(hm)$ berechnet werden:

$$\exists hm^m, hm^o, hm^{or}, hm^{xor} \in M:$$

$$\#var(m) = \#var(hm^m) * \#var(hm^o) * \#var(hm^{or}) * \#var(hm^{xor}) \quad (8-12)$$

In dem Beispiel werden für das Merkmal $c4$ zwei Hilfsmerkmale hm^m und hm^o eingeführt (Abbildung 28). Zuerst wird die lokale Konfigurationsanzahl für das jeweilige Hilfsmerkmal berechnet:

$$\#var(hm^m(c4)) = 1$$

$$\#var(hm^o(c4)) = \sum_{i=0}^2 \frac{2!}{i!(2-i)!} = 4$$

Abschließend kann die lokale Konfigurationsanzahl für das Merkmal $c4$ berechnet werden:

$$\#var(c4) = \#var(hm^m(c4)) * \#var(hm^o(c4)) = 1 * 4 = 4$$

8.2.1.2 Berechnung auf höheren Modellebenen

Nachdem die lokale Konfigurationsanzahl für jedes Eltern-Merkmal auf der Ebene N-1 berechnet wurde, kann auch die lokale Konfigurationsanzahl auf den höheren Ebenen (bis Ebene 2) berechnet werden.

Falls alle Kinder-Merkmale des Merkmals m Blätter (siehe Definition 5-16) km^{blatt} sind, kann die lokale Konfigurationsanzahl $\#var(m)$ in Abhängigkeit vom Typ der Kinder-Merkmale anhand der Formeln aus dem vorherigen Kapitel berechnet werden.

Falls alle Kinder-Merkmale des Merkmals m obligatorische Blatt-Merkmale sind $km^{m+blatt}$, dann kann die lokale Konfigurationsanzahl $\#var(m)$ anhand der Formel 8-3 berechnet werden.

Im Beispiel aus der Abbildung 27 ist dieser Fall für das Merkmal $e1$ relevant. $\#var(e1) = 1$

Sind alle Kinder-Merkmale des Merkmals m optionale Blatt-Merkmale $km^{o+blatt}$, dann kann die lokale Konfigurationsanzahl $\#var(m)$ anhand der Formel 8-4 berechnet werden.

In dem Beispiel ist dieser Fall für das Merkmal $e2$ relevant, welches zwei optionale Kinder-Merkmale $e5$ und $e6$ hat. Somit wird die lokale Konfigurationsanzahl $\#var(e2)$ folgendermaßen berechnet:

$$\#var(e2) = 2 * n = 2 * 2 = 4$$

Falls alle Kinder-Merkmale des Merkmals m alternative Blatt-Merkmale $km^{or+blatt}$ sind, dann kann die lokale Konfigurationsanzahl $\#var(m)$ anhand der Formel 8-5 berechnet werden.

Falls alle Kinder-Merkmale des Merkmals m auswählbare Blatt-Merkmale $km^{or+blatt}$ sind, kann die lokale Konfigurationsanzahl $\#var(m)$ anhand der Formel 8-6 berechnet werden.

Bei Bedarf können auch Hilfsmerkmale eingeführt werden. Die Berechnungen können anhand der Formeln aus dem vorherigen Kapitel durchgeführt werden.

Des Weiteren müssen für Eltern-Merkmale ab der Ebene N-2 folgende Fälle F6-F13 berücksichtigt werden:

Fall 6: Alle Kinder-Merkmale des Merkmals m sind obligatorische Merkmale km^m und haben eigene Kinder kkm . In diesem Fall wird die lokale Anzahl der Konfigurationen $\#var(m)$ folgendermaßen berechnet:

$$\forall m, km_i^m, kkm_j \in M \wedge (m, km_i^m) \in r^{kind} \wedge (km_i^m, kkm_j) \in r^{kind}:$$

$$\#var(m) = \prod_{i=1}^n \#var(km_i^m)$$

mit $1 \leq i \leq n$, mit $1 \leq j \leq n$, $n \in \mathbb{N}$ (8-13)

In dem Beispiel ist dieser Fall für das Merkmal b1 relevant. Die lokale Konfigurationsanzahl für das Merkmal b1 ist:

$$\#var(b1) = \#var(b4) * \#var(b5) = 1 * 4 = 4$$

Fall 7: Alle Kinder-Merkmale des Merkmals m sind optionale Merkmale km^o und haben eigene Kinder kkm . In diesem Fall wird die lokale Anzahl der Konfigurationen $\#var(m)$ folgendermaßen berechnet:

$$\forall m, km_i^o, kkm_j \in M \wedge (m, km_i^o) \in r^{kind} \wedge (km_i^o, kkm_j) \in r^{kind}:$$

$$\#var(m) = \prod_{i=1}^n (\#var(km_i^o) + 1)$$

mit $1 \leq i \leq n$, mit $1 \leq j \leq n$, $n \in \mathbb{N}$ (8-14)

In dem Beispiel aus der Abbildung 27 ist dieser Fall für das Merkmal b2 relevant. Die lokale Konfigurationsanzahl für das Merkmal b2 ist:

$$\#var(b2) = (\#var(b6) + 1) * (\#var(b7) + 1) = (2 + 1) * (3 + 1) = 12$$

Fall 8: Die Kinder-Merkmale des Merkmals m sind alternative Merkmale km^{xor} und haben eigene Kinder kkm . In diesem Fall wird die lokale Anzahl der Konfigurationen für das Merkmal m folgendermaßen berechnet:

$$\forall m, km_i^{xor}, kkm_j \in M \wedge (m, km_i^{xor}) \in r^{kind} \wedge (km_i^{xor}, kkm_j) \in r^{kind}:$$

$$\#var(m) = \sum_{i=1}^n \#var(km_i^{xor})$$

mit $1 \leq i \leq n$, mit $1 \leq j \leq n$, $n \in \mathbb{N}$ (8-15)

In dem Beispiel ist dieser Fall für das Merkmal c3 relevant. Die lokale Konfigurationsanzahl für das Merkmal c3 ist gleich:

$$\#var(c3) = \#var(c4) + \#var(c5) = 4 + 2 = 6$$

Fall 9: Die Kinder-Merkmale des Merkmals m sind auswählbare Merkmale km^{or} mit der Auswahlkardinalität $Card^{or} = [min, max]$ und haben eigene Kinder kkm .

$$\forall m, km_i^{or}, kkm_j \in M \wedge (m, km_i^{or}) \in r^{kind} \wedge (km_i^{or}, kkm_j) \in r^{kind};$$

mit $1 \leq i \leq n$, mit $1 \leq j \leq n$, $n \in \mathbb{N}$ (8-18)

In diesem Fall muss die lokale Anzahl der Konfigurationen für das Merkmal m in fünf Schritten berechnet werden. Dieses Vorgehen wird anhand des Merkmals $d2$ aus dem Beispiel (Abbildung 27) verdeutlicht.

1. Zuerst wird die lokale Anzahl der Konfigurationen für das Merkmal m ohne Berücksichtigung der Kinder-Merkmale kkm der Kinder-Merkmale km^{or} anhand der Formel 8-6 berechnet:

$$\#var'(m) = \sum_{i=min}^{max} \frac{n!}{i!(n-i)!}$$

$$\#var'(d2) = \sum_{i=0}^3 \frac{3!}{i!(3-i)!} = \frac{3!}{0!3!} + \frac{3!}{1!2!} + \frac{3!}{2!1!} + \frac{3!}{2!1!} = 8$$

2. Danach wird für jedes Kinder-Merkmal km^{or} , welches eigene Kinder-Merkmale kkm hat, bestimmt, in wie vielen Konfigurationen des Merkmals m es vorkommen kann. Dabei ist der minimale Wert für die Variable i mindestens 1, weil das Merkmal m mindestens in einer Konfiguration vorkommt.

$$\#km^{or} = \sum_{i=1}^{max} \frac{n!}{i!(n-i)!} * \frac{i}{n} = \sum_{i=1}^{max} \frac{(n-1)!}{(i-1)!(n-i)!}$$

(8-19)

In dem Beispiel sind es die Merkmale $d4$ und $d5$:

$$\#(d4) = \#(d5) = \sum_{i=1}^3 \frac{(3-1)!}{(i-1)!(3-i)!} = \frac{2!}{0!2!} + \frac{2!}{1!1!} + \frac{2!}{2!0!} = 4$$

3. Falls mehrere Kinder-Merkmale km^{or} des Merkmals m eigene Kinder haben, dann muss bestimmt werden, in wie vielen Kombinationen alle diese Merkmale beteiligt sind. Bei zwei Kinder-Merkmalen km_1^{or} und km_2^{or} wird die Formel 8-20 verwendet:

$$\#km_2^{or} | km_1^{or} = \sum_{i=2}^{max} \frac{n!}{i!(n-i)!} * \frac{i}{n} * \frac{i-1}{n-1} = \sum_{i=2}^{max} \frac{(n-2)!}{(i-2)!(n-i)!}$$

(8-20)

$$\#d4 | d5 = \sum_{i=2}^3 \frac{(3-2)!}{(i-2)!(3-i)!} = \frac{1!}{0!(3-2)!} + \frac{1!}{1!(0)!} = 2$$

Bei drei Kinder-Merkmalen km_1^{or} , km_2^{or} und km_3^{or} mit eigenen Kinder-Merkmalen werden die Formeln 8-21 und 8-22 verwendet:

$$\#km_2^{or}|km_1^{or} = \#km_3^{or}|km_1^{or} = \#km_3^{or}|km_2^{or} = \frac{n!}{2!(n-2)!} * \frac{2}{n} * \frac{1}{n-1} \quad (8-21)$$

$$\#km_3^{or}|km_2^{or}|km_1^{or} = \frac{n!}{3!(n-3)!} * \frac{3}{n} * \frac{2}{n-1} * \frac{1}{n-2} \quad (8-22)$$

Bei mehr als drei Kinder-Merkmalen mit eigenen Kinder-Merkmalen müssen die Formeln 8-21 und 8-22 entsprechend erweitert werden, um die entsprechenden Kombinationen mit der Beteiligung von allen (vier oder mehr Merkmale) zu berücksichtigen.

4. Jetzt kann bestimmt werden, in wie vielen Konfigurationen des Merkmals m nur Blatt-Kinder-Merkmale vorkommen. Dafür wird von der Anzahl der lokalen Konfigurationen jeweils die Anzahl der Konfigurationen, in welchen die Kinder-Merkmale eigene Kinder-Merkmale haben, abgezogen. Anschließend muss dieser Wert, um die Anzahl der Konfigurationen in welchen mehrere Kinder-Merkmale mit eigenen Kindern gleichzeitig vorkommen, korrigiert werden.

$$\#var''(m) = \#var'(m) - \sum_{i=k}^l \#km_i^{or} + \sum \#km_i^{or}|km_j^{or} \quad (8-23)$$

$$\#var''(d2) = \#var'(d2) - \#d4 - \#d5 + \#d4|d5 = 8 - 4 - 4 + 2 = 2$$

5. Anschließend kann die lokale Anzahl der Konfigurationen für das Merkmal m mit Berücksichtigung der Kinder-Merkmale km mit Kindern berechnet werden. Die lokale Konfigurationsanzahl des Merkmals $\#var(m)$ besteht aus:
 - der Konfigurationsanzahl des Merkmals m mit Berücksichtigung nur der Blatt-Kinder-Merkmale $\#var''(m)$,
 - der Konfigurationsanzahl für jedes Kinder-Merkmal km^{or} mit eigenen Kindern ohne die Konfigurationen, in welchen mehrere Kinder-Merkmale km^{or} gleichzeitig vorkommen $(\#km_1^{or} - \#km_1^{or}|km_2^{or}) * \#var(km_1^{or})$,
 - der Anzahl der Konfigurationen für Kinder-Merkmale km^{or} mit eigenen Kindern, in welchen mehrere Kinder-Merkmale km^{or} gleichzeitig vorkommen $(\#km_1^{or}|km_2^{or}) * \#var(km_1^{or}) * \#var(km_2^{or})$.

Bei zwei Kinder-Merkmalen km_1^{or} und km_2^{or} mit eigenen Kinder-Merkmalen wird die Formel 8-24 verwendet:

$$\begin{aligned} \#var(m) = & \#var''(m) + \\ & (\#km_1^{or} - \#km_1^{or}|km_2^{or}) * \#var(km_1^{or}) + (\#km_2^{or} - \#km_1^{or}|km_2^{or}) * \\ & \#var(km_2^{or}) + (\#km_1^{or}|km_2^{or}) * \#var(km_1^{or}) * \#var(km_2^{or}) \end{aligned} \quad (8-24)$$

In dem Beispiel wird die Anzahl der Konfigurationen für das Merkmal $d2$ folgendermaßen berechnet:

$$\begin{aligned} var(d2) = & \#var''(d2) + (\#d4 - \#d4|\#d5) * \#var(d4) + \\ & (\#d5 - \#d4|d5) * \#var(d5) + (\#d4|d5) * \#var(d4) * \#var(d5) = \\ & 2 + (4 - 2) * 3 + (4 - 2) * 2 + 2 * 3 * 2 = 24 \end{aligned}$$

Bei drei Kinder-Merkmalen km_1^{or} , km_2^{or} und km_3^{or} mit eigenen Kindern wird die Formel 8-25 verwendet:

$$\begin{aligned}
 \#var(m) &= \#var''(m) + \\
 &(\#km_1^{or} - \#km_1^{or} | km_2^{or} - \#km_1^{or} | km_3^{or}) * \#var(km_1^{or}) + \\
 &(\#km_2^{or} - \#km_2^{or} | km_1^{or} - \#km_2^{or} | km_3^{or}) * \#var(km_2^{or}) + \\
 &(\#km_3^{or} - \#km_3^{or} | km_1^{or} - \#km_2^{or} | km_3^{or}) * \#var(km_3^{or}) + \\
 &(\#km_1^{or} | km_2^{or} | km_3^{or}) * \#var(km_1^{or}) * \#var(km_2^{or}) * \#var(km_3^{or})
 \end{aligned} \tag{8-25}$$

Bei mehr als drei Kinder-Merkmalen mit eigenen Kinder-Merkmalen muss die Formel 8-25 entsprechend erweitert werden.

Fall 13: Falls die Kinder-Merkmalen des Merkmals m von unterschiedlichen Typen sind, werden die Hilfsmerkmale wie im Kapitel 8.2.1.1 eingeführt.

In dem Beispiel aus der Abbildung 27 ist dieser Fall für das Merkmal B relevant:

$$\begin{aligned}
 \#var(B) &= hm^m(B) * hm^o(B) = \\
 &(\#var(b1) * \#var(b2)) * \#var(b3) = 4 * 12 * 2 = 96
 \end{aligned}$$

Die Anzahl der möglichen Konfigurationen für ein Merkmalsmodell MM mit der Berücksichtigung nur der hierarchischen Beziehungen kann anhand der Formeln aus den Abschnitten 8.2.1.1 und 8.2.1.2 erarbeitet werden. Dafür muss zuerst die Anzahlen der lokalen Konfigurationen für jeden Teilbaum berechnet werden.

Die durchgeführte Analyse von mehreren hunderten Merkmalsmodellen hat gezeigt, dass auf der zweiten Ebene eines Merkmalsmodells nur optionale und obligatorische Merkmale dokumentiert werden. Somit werden in der Formel nur diese zwei Merkmalstypen berücksichtigt. Bei der Berechnung der Konfigurationen für ein komplettes Merkmalsmodell muss berücksichtigt werden, ob es sich bei optionalen Merkmalen um ein Blatt-Merkmal handelt oder nicht. Die optionalen Blatt-Merkmale auf der zweiten Ebene können nur in zwei lokalen Konfigurationen des Modells vorkommen. Dieser Aspekt wird in Formel 8-26 berücksichtigt.

Die Anzahl der Konfigurationen $Var(MM)$ für ein Merkmalsmodell MM kann folgendermaßen berechnet werden:

$$\begin{aligned}
 &\forall m_i^{m+blatt}, m_j^{o+blatt}, m_k^m, m_l^o \in M \wedge w, M \in VM \wedge \\
 &(w, m_i^{m+blatt}) \in r^{kind} \wedge (w, m_j^{o+blatt}) \in r^{kind}, (w, m_k^m) \in r^{kind}, (w, m_l^o) \in r^{kind} : \\
 &\#Var(MM) = 2 * q * \prod_{k=1}^s \#var(m_k^m) * \prod_{l=1}^t (\#var(m_l^o) + 1)
 \end{aligned}$$

$$\text{mit } 1 \leq i \leq p, 1 \leq j \leq q, 1 \leq k \leq s, 1 \leq l \leq t, i + j + k + l \leq n \tag{8-26}$$

Für das Merkmalsmodell aus dem Beispiel wird die Anzahl der Konfigurationen folgendermaßen berechnet:

$$\begin{aligned}
 \#Var(MM) &= 2 * 1 * \#var(B) * \#var(E) * (\#var(C) + 1) * (\#var(D) + 1) \\
 &= 2 * 96 * 1 * (24 + 1) * (48 + 1) = 235200
 \end{aligned}$$

8.2.2 Berechnung der Auswahl-Wahrscheinlichkeit

Nachdem die lokalen Anzahlen der Konfigurationen für jedes Merkmal und die Gesamtanzahl der Konfigurationen für das gesamte Merkmalsmodell berechnet wurden, kann die Wahrscheinlichkeit der Auswahl eines Merkmals für eine Konfiguration $p(m)$ berechnet werden.

8.2.2.1 Auswahl-Wahrscheinlichkeit auf der zweiten Ebene

Die Auswahl-Wahrscheinlichkeit $p(m)$ für Merkmale auf der zweiten Hierarchieebene (die Merkmale auf dieser Ebene sind die Kinder-Merkmale der Wurzel des Merkmalsmodells) kann in Abhängigkeit von dem Merkmalstyp anhand folgender drei Fälle berechnet werden:

1. Die Auswahl-Wahrscheinlichkeit für ein obligatorisches Merkmals m^m ist gleich 1, da ein obligatorisches Merkmal auf zweiter Hierarchieebene in jeder Konfiguration des Variabilitätsmodells vorkommt.

$$\forall m^m \in M \wedge w, M \in VM \wedge (w, m^m) \in r^{kind} : p(m^m) = 1 \quad (8-27)$$

In dem Beispiel aus der Abbildung 27 kann nach Formel 8-21 die Auswahlwahrscheinlichkeit für die Merkmale B und E berechnet werden:

$$p(B) = 1$$

$$p(E) = 1$$

2. Die Auswahl-Wahrscheinlichkeit für ein optionales Merkmals ohne Kinder-Merkmale $m^{o+blatt}$ ist gleich 0,5. Der Grund dafür ist, dass ein optionales Merkmal auf zweiter Hierarchieebene in der Hälfte aller Konfiguration des Variabilitätsmodells vorhanden ist.

$$\forall m^{o+blatt} \in M \wedge w, M \in VM \wedge (w, m^{o+blatt}) \in r^{kind} : p(m^o) = 0,5 \quad (8-28)$$

In dem Beispiel ist die Auswahlwahrscheinlichkeit für das Merkmal A gleich 0,5:

$$p(A) = 0,5$$

3. Die Auswahl-Wahrscheinlichkeit für ein optionales Merkmal m^o mit Kinder-Merkmalen kann anhand der Relation von der lokalen Konfigurationsanzahl $\#var(m^o)$ zu dem korrigierten Wert der Konfigurationsanzahl berechnet werden. Der korrigierte Wert berücksichtigt den optionalen Typ des Merkmals m^o und entspricht der lokalen Konfigurationsanzahl $\#var(m^o)$ erweitert um eine weitere Konfigurationsmöglichkeit, falls das Merkmal m^o nicht in der Konfiguration des Merkmalsmodells vorhanden ist.

$$\forall m^o \in M \wedge w, M \in VM \wedge (w, m^o) \in r^{kind} : \\ p(m^o) = \frac{\#var(m^o)}{(\#var(m^o) + 1)} \quad (8-29)$$

In dem Beispiel kann nach der Formel 8-29 die Auswahlwahrscheinlichkeit für das Merkmal C berechnet werden:

$$p(C) = \frac{\#var(C)}{(\#var(C) + 1)} = 0,96$$

8.2.2.2 Auswahl-Wahrscheinlichkeit auf den weiteren Ebenen

Um die Auswahl-Wahrscheinlichkeit $p(m)$ für Merkmale auf weiteren Hierarchieebenen zu berechnen, muss die Auswahlkardinalität berücksichtigt werden. In der vorliegenden Arbeit wurde entschieden, nur die Auswahlkardinalität von auswählbaren Merkmalen

explizit zu dokumentieren. Für optionale Merkmale ist die Auswahlkardinalität [0,1]. Bei alternativen Merkmalen ist die Auswahlkardinalität [0,1] für alle Kinder-Merkmale.

4. Die Auswahl-Wahrscheinlichkeit für ein obligatorisches Merkmal m^m ist gleich der Auswahl-Wahrscheinlichkeit des Eltern-Merkmals $eltern_von(m)$:

$$p(m) = p(eltern_von(m)) \quad (8-30)$$

In dem Beispiel kann nach Formel 8-30 die Auswahlwahrscheinlichkeit für die Merkmale $b1$ und $b2$ bestimmt werden:

$$p(b1) = p(b2) = p(B) = 1$$

5. Die Auswahlwahrscheinlichkeit für ein optionales Merkmal m^o kann anhand der Relation der lokalen Konfigurationsanzahl zu der lokalen Konfigurationsanzahl des Eltern-Merkmals berechnet werden. Diese Relation muss anschließend mit der Wahrscheinlichkeit der Auswahl des Eltern-Merkmals multipliziert werden, da die Auswahlwahrscheinlichkeit für ein Merkmal m abhängt von der Auswahlwahrscheinlichkeit des Eltern-Merkmals $p(eltern_von(m))$.

$$p(m) = \frac{\#var(m)}{\#var(eltern_von(m))} * p(eltern_von(m)) \quad (8-31)$$

In dem Beispiel kann nach Formel 8-31 die Auswahlwahrscheinlichkeit für das Merkmal $b6$ bestimmt werden:

$$p(b6) = \frac{\#var(b6)}{\#var(b2)} * p(b2) = \frac{2}{12} * 1 \approx 0,17$$

6. Die Auswahlwahrscheinlichkeit für ein alternatives oder auswählbares Merkmal m kann anhand der Relation der Konfigurationen unter Beteiligung des Merkmals m zu der lokalen Konfigurationsanzahl des Eltern-Merkmals berechnet werden. Diese Relation muss anschließend mit der Wahrscheinlichkeit der Auswahl des Eltern-Merkmals multipliziert werden, da die Auswahlwahrscheinlichkeit für ein Merkmal m von der Auswahl-Wahrscheinlichkeit des Eltern-Merkmals $p(eltern_von(m))$.

$$p(m) = \frac{2^{max-min} * \#var(m)}{\#var(eltern_von(m))} * p(eltern_von(m))$$

mit $min, max \in Card(m)$

(8-32)

7. Die Auswahl-Wahrscheinlichkeit für ein Blatt-Merkmal m^{blatt} kann anhand der Relation von der lokalen Konfigurationsanzahl des Blatt-Merkmals $\#var(m^{blatt})$ zu der lokalen Konfigurationsanzahl des Eltern-Merkmals $\#var(eltern_von(m^{blatt}))$ berechnet werden. Diese Relation muss anschließend mit der Wahrscheinlichkeit der Auswahl des Eltern-Merkmals multipliziert werden, da die Auswahl-Wahrscheinlichkeit für ein Blatt-Merkmal m^{blatt} abhängt von der Auswahl-Wahrscheinlichkeit des Eltern-Merkmals $p(eltern_von(m^{blatt}))$ ab.

$$p(m^{blatt}) = \frac{\#var(m^{blatt})}{\#var(elter_n_von(m^{blatt}))} * p(elter_n_von(m^{blatt})) \quad (8-33)$$

Die lokale Konfigurationsanzahl für ein Blatt-Merkmal kann anhand der angepassten Formel 8-19 berechnet werden:

$$\#m^{blatt} = \sum_{i=min}^{max} \frac{n!}{i!(n-i)!} * \frac{i}{n} \quad (8-34)$$

Somit sieht die Formel zur Berechnung der Auswahl-Wahrscheinlichkeit für ein Blatt-Merkmal m^{blatt} folgendermaßen aus:

$$p(m^{blatt}) = \frac{\sum_{i=min}^{max} \frac{n!}{i!(n-i)!} * \frac{i}{n}}{\#var(elter_n_von(m^{blatt}))} * p(elter_n_von(m^{blatt})) \quad (8-35)$$

In dem Beispiel kann nach Formel 8-35 die Auswahlwahrscheinlichkeit für die Merkmale $b12$ und $b13$ bestimmt werden:

$$p(b12) = p(b13) = \frac{\sum_{i=0}^1 \frac{2!}{i!(2-i)!} * \frac{i}{2}}{\#var(b6)} * p(b6) = \frac{1}{2} * 0,33 \approx 0,17$$

8.2.3 Berechnung der optimalen Entscheidungsfolge

Zur Verdeutlichung des vorgestellten Algorithmus zur Bestimmung der optimalen Entscheidungsfolge wird in diesem Kapitel eine optimale Entscheidungsfolge für das Beispiel-Merkmalmodell aus der Abbildung 27 berechnet. Die Abbildung 29 präsentiert das Beispiel-Merkmalmodell, welches um die folgenden 14 Abhängigkeitsbeziehungen erweitert wurde:

1. $(b6, c8) \in r^{exclude}$
2. $(b6, d7) \in r^{exclude}$
3. $(b6, d10) \in r^{exclude}$
4. $(b7, c9) \in r^{exclude}$
5. $(b7, d1) \in r^{exclude}$
6. $(c1, d7) \in r^{exclude}$
7. $(c1, e6) \in r^{exclude}$
8. $(c1, A) \in r^{needs}$
9. $(c1, d1) \in r^{needs}$
10. $(c1, e5) \in r^{needs}$
11. $(D, b3) \in r^{needs}$
12. $(e6, b3) \in r^{needs}$
13. $(b10, d1) \in r^{needs}$

In dem nächsten Schritt wurde überprüft, ob die Abhängigkeiten die Anzahl der benötigten Entscheidungen in diesem Merkmalsmodell reduzieren. In der Tabelle 20 wurden die Ergebnisse dokumentiert. Für jede Entscheidung wurden auch die Auswirkungen durch die Abhängigkeitsbeziehungen (also welche Merkmale sind für diese Entscheidung notwendig oder welche Merkmale werden durch diese Entscheidung abgewählt) dokumentiert. Um die Voraussetzungen durch hierarchische als auch durch „benötigt“-Beziehungen einzuhalten, musste man die Reihenfolge der Fragen anpassen. Durch die Berücksichtigung der Abhängigkeiten zwischen einzelnen Merkmalen reduzierte sich die Anzahl der notwendigen Entscheidungen auf 18.

Tabelle 20: Entscheidungsfolge mit Beziehungsberücksichtigung

Nr neu	Nr alt	Entscheidung	benötigt	schließt_aus	Konfiguration
1	1	A			A
2	17	D	b3		D
3	18	d1	D	b7	-
	2	b10	d1		-
4	3	b11			b11
5	4	b6		d7, d10, c8	b6
6	5	b12 xor b13	b6		b12
7	6	b7		c9, d1	b7
8	7	b14 or b15	b7		b14,b15
9	8	b3			b3
10	9	C			C
	10	e1	A, d1, e5,C	d7, e6	-
11	11	c2	C		c2
12	12	c3	C		c3
13	13	c4 or c5	C, c3		c4
	14	e8	C,c4	b6	-
	15	e9	C, c4	b7	-
	16	e12 xor e13	d1, c5		-
14	19	d2	D		d2
15	20	d3 or d4 or d5	D,d2		d3,d4,d5
16	21*	d7 or d8	D,d2,d4	d7 schließt_aus b6, c1	d7
	22	d10	D,d2,d5	b6	-
17	23	e5			e5
18	24	e6	b3	c1	e6

In den nächsten Abschnitten wird die Anwendung des vorgestellten Algorithmus zur Bestimmung einer optimalen Reihenfolge in dem betrachteten Beispiel detailliert vorgestellt.

Phase 1: In dem Merkmalsmodell wird nach einem Merkmal m gesucht, welches an der maximalen Anzahl der „schließt_aus“-Beziehungen beteiligt ist. Dafür wird zuerst für alle Merkmale in dem Beispiel-Merkmalsmodell, welche in den „schließt_aus“-Beziehungen beteiligt sind, die Anzahl der „schließt_aus“-Beziehungen bestimmt:

- $R^e(b6)=3$
- $R^e(d7)=2$
- $R^e(b7)=2$
- $R^e(c1)=2$
- $R^e(c8)=1$
- $R^e(d10)=1$
- $R^e(c9)=1$
- $R^e(d1)=1$
- $R^e(e6)=1$

Das Merkmal *b6* ist in der maximalen Anzahl der „schließt_aus“-Beziehungen beteiligt und hat keine Voraussetzungsmerkmale. Somit wird die erste Entscheidung in den optimalen Folgen bezüglich des Merkmals *b6* sein.

In der Tabelle 21 ist das Ergebnis der ersten Phase dokumentiert. Da das Merkmal *b6* in der gesuchten Konfiguration vorkommt, sind durch die „schließt_aus“-Beziehungen die Merkmale *d7*, *d10* und *c8* ausgeschlossen

Tabelle 21: Optimale Entscheidungsfolge nach Phase 1

Nr	Entscheidung	benötigt	schließt_aus	Konfiguration
1	<i>b6</i>		<i>d7, d10, c8</i>	<i>b6</i>
	<i>d7</i>		<i>b6</i>	-
	<i>d10</i>		<i>b6</i>	-
	<i>c8</i>		<i>b6</i>	-

Phase 2: In dem Merkmalsmodell wird wieder nach einem Merkmal *m* gesucht, welches an der maximalen Anzahl der „schließt_aus“-Beziehungen beteiligt ist.

Die Merkmale *d7*, *b7* und *c1* sind jeweils in zwei schließt „schließt_aus“-Beziehungen beteiligt. Das Merkmal *d7* ist durch die Auswahl des Merkmals *b6* in dem vorherigen Schritt ausgeschlossen. Das Merkmal *c1* wird durch die Auswahl des Merkmals *b7* nicht auswählbar sein, da das Merkmal *c1* das Merkmal *d1* benötigt, welches eine „schließt_aus“-Beziehung mit dem Merkmal *b7* hat.

In der Tabelle 22 ist das Ergebnis der zweiten Phase dokumentiert. Das Merkmal *b7* gehört in die gesuchte Konfiguration, somit durch die „schließt_aus“-Beziehungen die Merkmale *c9*, *d1*, *b10* und *c1* müssen nicht weiter betrachtet werden.

Tabelle 22: Optimale Entscheidungsfolge nach Phase 2

Nr	Entscheidung	benötigt	schließt_aus	Konfiguration
1	<i>b6</i>		<i>d7, d10, c8</i>	<i>b6</i>
	<i>d7</i>		<i>b6</i>	-
	<i>d10</i>		<i>b6</i>	-
	<i>c8</i>		<i>b6</i>	-
2	<i>b7</i>		<i>c9, d1</i>	<i>b7</i>
	<i>c9</i>		<i>b7</i>	-
	<i>d1</i>		<i>b7</i>	-
	<i>b10</i>	<i>d1</i>		-
	<i>c1</i>	<i>d1</i>		-

Phase 3: In dem Merkmalsmodell wird wieder nach einem Merkmal *m* gesucht, welches an der maximalen Anzahl der „schließt_aus“-Beziehungen beteiligt ist.

Tabelle 23: Optimale Entscheidungsfolge nach Phase 3

Nr	Entscheidung	benötigt	schließt_aus	Konfiguration
1	<i>b6</i>		<i>d7, d10, c8</i>	<i>b6</i>
	<i>d7</i>		<i>b6</i>	-
	<i>d10</i>		<i>b6</i>	-
	<i>c8</i>		<i>b6</i>	-
2	<i>b7</i>		<i>c9, d1</i>	<i>b7</i>
	<i>c9</i>		<i>b7</i>	-
	<i>d1</i>		<i>b7</i>	-
	<i>b10</i>	<i>d1</i>		-
	<i>c1</i>	<i>d1</i>		-
3	<i>b3</i>			<i>b3</i>
4	<i>e6</i>	<i>b3</i>	<i>c1</i>	<i>e6</i>

Die Merkmale *c8*, *c9*, *d1*, *d10* und *e6* sind jeweils in einer „schließt_aus“-Beziehung beteiligt. Die Merkmale *c8*, *c9*, *d1* und *d10* sind durch die Auswahl der Merkmale

b6 und *b7* in den vorherigen Schritten ausgeschlossen. Das Merkmal *e6* benötigt das Merkmal *b3*. Somit wird zuerst nach dem Merkmal *b3* gefragt und anschließend nach *e6*.

In der Tabelle 23 ist das Ergebnis der dritten Phase dokumentiert. Die Merkmale *b3* und *e6* gehören in die gesuchte Konfiguration.

Phase 4: In dem Merkmalsmodells wird wieder nach einem Merkmal *m* gesucht, welches an der maximalen Anzahl der „schließt_aus“-Beziehungen beteiligt ist. Da in dem Beispielsmodell kein weiteres Merkmal vorhanden ist, welches an einer „schließt_aus“-Beziehung beteiligt ist, wird im nächsten Schritt nach einem Merkmal *m* gesucht, in welches der maximalen Anzahl der „benötigt“-Beziehungen eingehen. In dem Beispielsmodell sind folgende Merkmale die Senken der „benötigt“-Beziehungen:

- $R^n(A)=1$
- $R^n(d1)=3$
- $R^n(e5)=1$
- $R^n(b3)=2$

Die Entscheidung bezüglich des Merkmals *b3* wurde bereits in der Phase 4 getroffen. Das Merkmal *d1* ist durch die Auswahl des Merkmals *b7* ausgeschlossen. Die Merkmale *A* und *e5* sind jeweils die Senke von einer „benötigt“-Beziehung, somit muss die Auswahlwahrscheinlichkeit für die berechnet werden:

$$p(A)=0,5$$

$$p(e5)=0,67$$

Die Auswahlwahrscheinlichkeit des Merkmals *e5* ist höher als bei dem Merkmal *A*, somit wird zuerst nach dem Merkmal *e5* und danach nach dem Merkmal *A* gefragt werden.

In der Tabelle 24 ist das Ergebnis der vierten Phase dokumentiert.

Tabelle 24: Optimale Entscheidungsfolge nach Phase 4

Nr	Entscheidung	benötigt	schließt_aus	Konfiguration
1	<i>b6</i>		<i>d7, d10, c8</i>	<i>b6</i>
	<i>d7</i>		<i>b6</i>	-
	<i>d10</i>		<i>b6</i>	-
	<i>c8</i>		<i>b6</i>	-
2	<i>b7</i>		<i>c9, d1</i>	<i>b7</i>
	<i>c9</i>		<i>b7</i>	-
	<i>d1</i>		<i>b7</i>	-
	<i>b10</i>	<i>d1</i>		-
	<i>c1</i>	<i>d1</i>		-
3	<i>b3</i>			<i>b3</i>
4	<i>e6</i>	<i>b3</i>	<i>c1</i>	<i>e6</i>
5	<i>e5</i>			<i>e5</i>
6	<i>A</i>			<i>A</i>

Phase 5: In dem Beispiel-Merkmalsmodell sind keine Merkmale geblieben, welche entweder an einer „schließt_aus“-Beziehung beteiligt sind oder die Senke einer „benötigt“-Beziehung sind. Somit kann die verbliebene Fragenfolge nicht optimiert werden. Die Tabelle 25 dokumentiert die optimale Fragenfolge für das Beispielsmodell.

Durch die Verwendung des erarbeiteten Algorithmus konnte die Fragenfolge von 18 Fragen (siehe Tabelle 20) auf 15 Fragen (siehe Tabelle 25) reduziert werden. In dem betrachteten Beispiel-Merkmalsmodell waren nur 24 Merkmale variabel, in einem Merkmalsmodell aus der realen Welt ist die Anzahl der variablen Merkmalen viel höher, somit ist auch das Potenzial für die Optimierung viel größer.

8.2.4 Optimale Entscheidungsfolge bei KDVM

Die in den vorherigen Abschnitten vorgestellte Vorgehensweise kann sowohl für zentrale Variabilitätsmodelle als auch dezentrale Variabilitätsmodelle verwendet werden.

Um das Konzept der Bestimmung der optimalen Entscheidungsfolge bei der KDVM anwenden können, muss folgende Anpassung gemacht werden:

- Zuerst müssen alle Entscheidungen bezüglich der Merkmale gemacht werden, welche Beziehungen zu den anderen Modellen haben.
- Anschließend müssen alle Entscheidungen innerhalb der einzelnen Modelle getroffen werden.

Dabei kann die folgende Vorgehensweise verwendet werden:

- (1) Treffe alle Entscheidungen bezüglich der Merkmale, welche Beziehungen zu anderen Modellen auf anderen Abstraktionsebenen haben.
- (2) Treffe alle Entscheidungen bezüglich der Merkmale, welche Beziehungen zu anderen Modellen auf der gleichen Abstraktionsebene haben.
- (3) Bestimme die optimale Entscheidungsfolge in dem betroffenen Merkmalsmodell unter Berücksichtigung der Entscheidungen aus (1) und (2).

Tabelle 25: Optimale Entscheidungsfolge

Nr	Entscheidung	benötigt	schließt_aus	Konfiguration
1	<i>b6</i>		<i>d7, d10, c8</i>	<i>b6</i>
	<i>d7</i>		<i>b6</i>	-
	<i>d10</i>		<i>b6</i>	-
	<i>c8</i>		<i>b6</i>	-
2	<i>b7</i>		<i>c9, d1</i>	<i>b7</i>
	<i>c9</i>		<i>b7</i>	-
	<i>d1</i>		<i>b7</i>	-
	<i>b10</i>	<i>d1</i>		-
	<i>c1</i>	<i>d1</i>		-
3	<i>b3</i>			<i>b3</i>
4	<i>e6</i>	<i>b3</i>	<i>c1</i>	<i>e6</i>
5	<i>e5</i>			<i>e5</i>
6	<i>A</i>			<i>A</i>
7	<i>D</i>	<i>b3</i>		<i>D</i>
8	<i>d2</i>	<i>D</i>		
9	<i>d3 or d4 or d5</i>	<i>D, d2</i>		<i>d3, d4, d5</i>
8	<i>d8</i>	<i>D, d2, d4</i>	-	<i>d8</i>
9	<i>C</i>			<i>C</i>
10	<i>c2</i>	<i>C</i>		<i>c2</i>
11	<i>c3</i>	<i>C</i>		<i>c3</i>
12	<i>c4 or c5</i>	<i>C, c3</i>		<i>c4</i>
	<i>c12 xor c13</i>	<i>d1, c5</i>		-
13	<i>b11</i>			<i>b11</i>
14	<i>b12 xor b13</i>	<i>b6</i>		<i>b12</i>
15	<i>b14 or b15</i>	<i>b7</i>		<i>b14, b15</i>

8.3 Diskussion

In der Forschung existieren mehrere Ansätze zur Lösung unterschiedlicher Optimierungsprobleme. Zum Beispiel existieren Lösungen zur Optimierung der Entscheidungen in den Decision Support Systems [WKHH95], allerdings orientieren sich die Lösungen an der Lösung der domain-spezifischen Probleme mit dem Ziel die Qualität der Entscheidungen zu verbessern. Ein weiterer Bereich der Optimierungsprobleme ist der Dialog-Design [Inou04]. Das Ziel der Ansätze in diesem Bereich ist die Dialoglänge zu minimie-

ren. Dabei wird nach einer kleinsten Reihenfolge der Dialogfragen gesucht mit der höchsten Beantwortungswahrscheinlichkeit.

Für das Thema der Konfigurationserstellung von Variabilitätsmodellen existieren zwar einige Ansätze ([DhGR11], [CzHE05], [LiYT10]). Aber keiner von diesen Ansätzen bietet die flexible Unterstützung des Benutzers ohne die Eingabe der zusätzlichen Informationen.

Das Thema der Optimierung des Konfigurationsprozesses von Variabilitätsmodellen ist ganz neu. Zum Zeitpunkt der Erstellung dieser Arbeit wurden nur zwei Ansätze gefunden, welche sich mit diesem Thema beschäftigen:

- Der erste Ansatz ist in der Arbeit von Nöhler und Egver [NöEg11] vorgestellt. Dieser Ansatz bietet eine automatisierte Unterstützung der Entscheidungen, welche anhand eines Entscheidungsbaums oder eines Variabilitätsmodells gemacht werden müssen. Das Ziel von diesem Ansatz entspricht dem Ziel des Ansatzes aus der vorliegenden Arbeit: Die Anzahl der Entscheidungen, welche durch den Benutzer gemacht werden müssen, zu reduzieren. Der Ansatz von Nöhler und Egver basiert auf der paarweisen Betrachtung der Beziehungen zwischen einzelnen Entscheidungsfragen bzw. Variabilitätsaspekten. Die Anzahl der Konfigurationen des Variabilitätsmodells wird dabei nicht berechnet, sondern nur geschätzt. Der Anfang des Konfigurationsprozesses kann sowohl durch den vorgestellten Algorithmus, als auch durch den Benutzer bestimmt werden.
- Der zweite Ansatz wurde in der Arbeit von Chen und Erwig [ChEr11] präsentiert. Dieser Ansatz beschäftigt sich mit der Optimierung des Konfigurationsprozesses von Merkmalsmodellen. Um den optimalen Konfigurationsprozess zu bestimmen wird ein Merkmalsmodell in eine algebraische Notation (Choice Description) transformiert und danach wird für jedes Merkmal das *Auswahlpotenzial* (eng. Selectivity) berechnet. Das Auswahlpotenzial eines Merkmals wird durch zwei Faktoren bestimmt: (i) Anzahl aller Konfigurationen mit diesem Merkmal und (ii) Anzahl der Merkmale, welche automatisch ausgewählt bzw. nicht relevant werden bei Auswahl des betrachteten Merkmals. Um die optimale Reihenfolge der Konfiguration zu bestimmen, wird zuerst das Merkmal mit dem größten Auswahlpotenzial betrachtet. Danach wird für jedes weiteres Merkmal in dem Merkmalsmodell das Auswahlpotenzial neu berechnet.

Im Gegensatz zu dem ersten Ansatz betrachtet der Ansatz aus der vorliegenden Arbeit alle Beziehungen zwischen allen Merkmalen und bietet die Möglichkeit zur Berechnung der Gesamtanzahl der Konfigurationen sowohl für das ganze Variabilitätsmodell als auch für einzelne Merkmale. Der Unterschied zu dem zweiten Ansatz liegt darin, dass nicht alle Merkmale in dem Variabilitätsmodell, sondern nur die Merkmale mit den Beziehungen betrachtet werden. Der vorgestellte Ansatz zur Bestimmung der optimalen Fragefolge wird die Dauer der Konfigurationserstellung nie verlängern, weil die Dauer des gestützten Konfigurationsprozesses im schlimmsten Fall der Dauer des manuellen Konfigurationsprozesses entsprechen wird.

Zur Berechnung der Anzahl der möglichen Konfigurationen eines Variabilitätsmodells existieren mehrere unterschiedliche Vorgehensweisen:

- Czarnecki und Kim [CzKi05] transformieren ein Merkmalsmodell in logische Ausdrücke und benutzen die SAT-Solver um die Anzahl der Konfigurationen zu berechnen.
- Benavides et al. [BeMC05] transformieren das Merkmalsmodell in ein Constraint-Programming-Model um den Variabilitätsgrad (eng. variability degree) zu berechnen.
- Von der Massen und Lichter [MaLi05] haben einen Approximationsansatz erarbeitet, um den Variabilitätsgrad zu bestimmen.

In dieser Arbeit wurde ein Ansatz zur Berechnung der Kombinationsanzahl eines Variabilitätsmodells vorgestellt, um die Bestimmung der optimalen Fragenfolge zu unterstützen. Dieser Ansatz kann auch durch einen anderen, bereits existierenden, Ansatz zur Berechnung der Kombinationsanzahl eines Variabilitätsmodells ersetzt werden. Dabei wird die Vorgehensweise zur Bestimmung der optimalen Fragefolge nicht beeinflusst. Der Vorteil von dem vorgestellten Ansatz ist die einfache und leicht nachvollziehbare Vorgehensweise, welche sowohl die automatisierte als auch die manuelle Anwendung ermöglicht.

Kapitel 9

Einführungs- und Migrationsvorschlag

Dieses Kapitel beschreibt, wie die in den vorherigen Kapiteln vorgeschlagenen Ansätze (MIA-, R2F-, ZV- und ZÜ-Ansatz) in der Praxis integriert werden können. Es wird ein in der Praxis erprobter Einführungsprozess für den MBVM-Ansatz vorgestellt. Außerdem wird gezeigt, wie die Migration von existierenden Ansätzen zu dem neuen Variantenmanagementansatz erfolgen kann. Im zweiten Teil dieses Kapitels wird eine Übersicht der auf dem Markt erhältlichen Werkzeuge für das Variantenmanagement in Anforderungsdokumenten gegeben. Am Ende wird ein neues Werkzeug für die MBVM- und R2F-Ansätze, welches als eine Erweiterung der Funktionalität des Anforderungsmanagementtool DOORS entwickelt wurde, vorgestellt.

9.1 Einführungsprozess

In der Industrie fehlen oft das Wissen und die Erfahrung zur Einführung von in der Forschung vorgestellten Ansätzen. Um die Einführung der in den vorherigen Kapiteln beschriebenen Ansätze zu erleichtern, präsentiert diese Arbeit einen möglichen Einführungs- und Migrationsprozess sowie die Erfahrungen, welche bei der Einführung der davor beschriebenen Ansätze mit diesem Einführungsprozess gemacht wurden.

Die MBVM für Anforderungsdokumente wurde bei der Daimler AG für die mechanische Komponente im Rahmen des großen internen Projektes REQUEST [BoHo11] eingeführt.

Der im Rahmen der vorliegenden Dissertation erarbeitete MBVM-Einführungsprozess besteht aus folgenden vier Schritten:

- Begriffsklärung für eine Domäne;
- Definition der Methodenstufen und der Anwendungskriterien;
- Konzeption des Migrationsprozesses;
- Auswahl der Tool-Unterstützung.

In den nächsten Abschnitten werden diese vier Schritte detailliert vorgestellt.

9.1.1 Begriffsklärung

Ein großes Problem bei der Einführung neuer Methoden ist es oft, eine gemeinsame Sprache für alle Beteiligten zu finden. Deswegen ist es sinnvoll, am Anfang mit allen Beteiligten die Begrifflichkeiten zu klären. Zum Beispiel, in der Forschung wurden die variablen Eigenschaften mit den Begriffen *Merkmal*, *Feature* oder *Variationspunkt* bezeichnet.

Bei der Begriffsklärung wurde sofort festgestellt, dass die oben genannten Begriffe für die Beschreibung der Variabilität nicht verwendet werden konnten:

1. Der Begriff *Feature* hat mehrere andere Bedeutungen (u.a. eine Eigenschaft und eine Funktionalität).
2. Der Begriff *Merkmal* war wenig aussagekräftig und die meisten Beteiligten hatten kein Verständnis/Assoziation für diesen Begriff.
3. Der Begriff *Variationspunkt* wurde von den meisten Beteiligten als variable Stelle in einem Dokument verstanden und nicht als die variable Eigenschaft.

Deswegen wurden zuerst die Begriffe definiert, welche als Grundlage der Kommunikation zwischen Projektbeteiligten verwendet werden sollen. Tabelle 26 gibt einen Überblick der grundlegenden Begriffe, welche bei der Einführung des MBVM-Ansatzes bei der Daimler AG festgelegt wurden. Viele dieser Begriffe wurden bereits im Kapitel 2 dieser Arbeit vorgestellt.

Tabelle 26: Begriffe der Variabilitätsbeschreibung

Begriff	Bedeutung
Variabilität	beschreibt die Unterschiede der verschiedenen Ausprägungen bzw. Kombinationen der Ausprägungen des Bezugsgegenstandes - ist die Menge aller zulässigen Varianten eines Bezugsgegenstandes
Varianz	stellt jeden Unterschied eines Objekts zu einem anderen oder zu einem Standardwert dar
Bezugsgegenstand	ist ein (Entwicklungs-)Artefakt, der variieren kann
Variabilitätskriterium	ist eine Eigenschaft eines Bezugsgegenstandes, das variieren kann
Ausprägung des Variabilitätskriteriums	ist ein zulässiger Wert des Variabilitätskriteriums
Variante	ist ein Objekt mit einer Kombination der Ausprägungen der Variabilitätskriterien eines Bezugsgegenstandes, der sich von allen anderen Objekten in mindestens einem Merkmal unterscheidet
Gültige Variante	besteht aus einer zulässigen Kombination der Ausprägungen der Variabilitätskriterien
Produkt	ist ein Produkt mit Sachnummer und Teil einer Produktlinie
Produktfamilie	ist eine Menge von Produktlinien
Produktlinie	ist eine Gruppe von Produkten, die sowohl eine Reihe von Gemeinsamkeiten als auch erhebliche Unterschiede besitzt.
Variationspunkt	ist die Stelle in der Konfiguration eines Bezugsgegenstandes, an der die Auswahl einer oder mehrerer Varianten möglich ist

9.1.2 Ausprägungen der MBVM-Methode

Bereits während der Analyse der existierenden Ansätze (Kapitel 3.2.1) wurde klar, dass eine Methode in einer Ausprägung allein nicht zu großer Akzeptanz bei den Anwendern führen wird. Der Grund dafür sind unterschiedliche Variabilitätsmanagement-

Bedürfnisse, welche sowohl von der Anzahl der Variabilitätskriterien, der Anzahl der Beziehungen zwischen diesen Variabilitätskriterien als auch von der Anzahl der Anforderungen abhängen.

Deswegen wurde in dem Projekt entschieden, drei unterschiedliche Ausprägungen basierend auf dem R2F- Ansatz zu definieren und den Anwendern zur Verfügung zu stellen:

1. Die *einfache Ausprägung* ist für Komponenten und Systeme mit wenigen Variabilitätskriterien (zwei bis sieben) ohne Beziehungen zwischen diesen Variabilitätskriterien geeignet.

Diese Variante ist relevant für die meisten mechanischen Komponenten, deren Variabilität durch wenige, oft unabhängige, Variabilitätskriterien beschrieben werden kann. In diesem Fall ist es möglich, für die Dokumentation der Variabilität in den Anforderungsdokumenten den Objekt-Hierarchie-Ansatz anzuwenden. Trotzdem es ist empfehlenswert, die Variabilitätskriterien in einem Variabilitätsmodell zu verwalten. Diese Variante der Variabilitätsverwaltung entspricht einer Vereinfachung des R2F-Ansatzes.

2. Die *mittlere Ausprägung* ist für Komponenten und Systeme mit einer mittleren Anzahl von Variabilitätskriterien (zwei bis fünfzehn) und wenigen Beziehungen zwischen diesen Variabilitätskriterien anwendbar.

Diese Ausprägung des R2F-Ansatzes ist relevant für viele mechatronische Komponenten und viele Systeme, deren Variabilität durch eine mittlere Anzahl abhängiger Variabilitätskriterien beschrieben werden kann. In diesem Fall kann für die Dokumentation der Variabilität in den Anforderungsdokumenten der R2F-Ansatz verwendet werden. Um die Variabilität modellieren und verwalten zu können, es ist sinnvoll ein VM-Tool anzuwenden, das mit dem AM-Tool kompatibel ist.

3. Die *höchste Ausprägung* der Basismethode ist für Komponenten und Systeme mit mehr als fünfzehn Variabilitätskriterien und mehreren Beziehungen zwischen diesen Variabilitätskriterien anwendbar.

Diese Methodenausprägung ist relevant für viele elektrische und elektronische Komponenten, weil deren Variabilität durch eine hohe Anzahl von Variabilitätskriterien und vielen Beziehungen zwischen diesen Variabilitätskriterien beschrieben werden kann. In diesem Fall kann für die Dokumentation der Variabilität in den Anforderungsdokumenten der R2F-Ansatz verwendet werden, welcher bei Bedarf um die Berücksichtigung der notwendigen Merkmale erweitert werden kann. Um die Variabilität modellieren und verwalten zu können, es ist notwendig, ein VM-Tool anzuwenden, welches die komplexen Abhängigkeiten zwischen den Variabilitätskriterien abbilden kann und mit dem AM-Tool kompatibel ist.

Ein Vergleich auf dem Markt existierender Werkzeuge für das Variantenmanagement, welche mit dem Anforderungsmanagementtool DOORS kompatibel sind, wird im Abschnitt 9.2.1 präsentiert.

9.1.3 Migrationsprozess

Bei der Einführung von MBVM ist in den meisten Fällen eine Migration von einem bereits existierenden Ansatz, wie sie in Kapitel 3.2.1 beschrieben wurden, auf den neuen merkmalsbasierten Ansatz notwendig. Der Grund dafür ist, dass neue Anforderungsdokumente nur selten von Anfang, sondern durch Wiederverwendung bestehender Anforderungsdokumente erstellt werden [Houd04].

Um den Migrationsprozess zu erleichtern, kann der in Kapitel 4 vorgestellte MIA-Ansatz verwendet werden. Die mit Hilfe von MIA identifizierten Anforderungen können für die Variabilitätsmodellierung verwendet werden.

Nach der Erstellung des Variabilitätsmodells kann der ZV-Algorithmus verwendet werden, um die initiale Dokumentation der Variabilität in einem Anforderungsdokument zu unterstützen. Werden die Anforderungen zu Variabilitätskriterien im Rahmen eines Reviews erstellt, so kann abschließend der ZÜ-Algorithmus verwendet werden, um eine mögliche fehlerhafte Zuordnung zu identifizieren. Unter Berücksichtigung der großen Anzahl von Anforderungen in einem Anforderungsdokument und der möglichen Anzahl der Variabilitätskriterien, ist die Wahrscheinlichkeit für eine fehlerhafte Zuordnung sehr hoch.

9.2 Toolunterstützung

Eine Grundvoraussetzung für ein Variantenmanagement-Tool, welches für die Dokumentation und Verwaltung der Variabilität in den Anforderungsdokumenten eingesetzt wird, ist die Kompatibilität mit dem Anforderungsmanagementtool.

Die Marktanalyse der Variantenmanagement-Tools, hat gezeigt, dass auf dem Markt nur wenige Werkzeuge verfügbar sind, welche die merkmalsbasierte Variabilitätsmodellierung unterstützen und mit DOORS kompatibel sind. Im Rahmen dieser Arbeit wurden folgende drei Werkzeuge analysiert:

- *Gears* von BigLever [gears]: Dieses Tool basiert auf einem Software Product Line Framework. Für Variabilitätsmodellierung und –management wird ein zentrales Merkmalsmodell zur einheitlichen Dokumentation der Produktvielfalt für alle Entwicklungsartefakte (Anforderungen, Architektur, Tests) verwendet. Die Zuordnung von Merkmalen zu den Anforderungen erfolgt durch logische Beschränkungen (Constraints).
- *pure::variants* von pure::systems [pure]: Dieses Tool ermöglicht eine durchgängige Modellierung und Verwaltung von Variabilität in allen Phasen der Systementwicklung. Über seine Schnittstellen können Varianteninformationen aus *pure::variants* in alle Entwicklungsartefakte (z.B. Anforderungsdokumente) importiert werden. Die Anforderungen werden zu Merkmalen, welche durch die Attribute dargestellt werden, zugeordnet.
- *Feature Modeller FM* von metadoc [meta]: Dieses Tool ist eine DXL-Erweiterung der DOORS-Funktionalität. Mit dieser Erweiterung können Merkmale in DOORS dokumentiert und verwaltet werden. Die Zuordnung von Anforderungen zu Merkmalen erfolgt über den Aufbau der Anforderungsdokumentenstruktur.

Zusätzlich zur *Anforderung nach Kompatibilität mit dem Anforderungsmanagement-Tool DOORS (TA1)* wurden folgende weiteren Tool-Anforderungen (TA) definiert:

(TA2) *Die Zuordnung von Anforderungen zu Varianten muss erkennbar sein.*

Diese Anforderung bedeutet, dass es im Anforderungsdokument selbst erkennbar sein muss, für welche Produkte (Varianten) jede einzelne Anforderung gültig ist. Diese Tool-Anforderung entspricht dem Kriterium K2 für die Bewertung der Variabilitätsdokumentationsansätze aus dem Kapitel 3.

(TA3) *Die Zuordnung von Anforderungen zu Varianten muss ohne spezielle Vorkenntnisse verständlich sein.*

Die Anforderung bedeutet, dass kein Code oder komplexe logische Ausdrücke für die Dokumentation der Variabilität in dem Anforderungsdokument verwendet werden können. Diese Tool-Anforderung entspricht dem Kriterium K3 für die Bewertung der Variabilitätsdokumentationsansätze.

(TA4) *Das VM-Tool darf zu keinen Strukturänderungen in den bestehenden Anforderungsdokumenten führen.*

Der Grund dafür ist, dass die meisten Anforderungsdokumente bereits einer erprobten und bewährten Strukturierung unterliegen, die die Zusammenarbeit zwischen Hersteller und Zulieferer unterstützt. Diese Tool-Anforderung entspricht der Anforderung A9 aus der Anforderungsliste für den neuen Variabilitätsmanagementansatz.

(TA5) *Das VM-Tool darf keine Änderung von Variabilitätsmodellen aus Anforderungsdokumenten heraus ermöglichen.*

Der Grund dafür ist, dass die Verantwortung für das Variabilitätsmodell und für das Anforderungsdokument unterschiedlichen Rollen zugewiesen ist. Außerdem wird ein Anforderungsdokument in der Industrie oft von mehreren Entwicklern gleichzeitig bearbeitet. Diese Tool-Anforderung entspricht der Anforderung A10 aus der Anforderungsliste für den neuen Variabilitätsmanagementansatz.

Die Ergebnisse der Analyse, ob die drei oben genannten Werkzeuge die Tool-Anforderungen (TA1 bis TA5) erfüllen, werden in Tabelle 27 präsentiert. Es ist leicht erkennbar, dass keines der drei Werkzeuge alle aufgestellten Anforderungen erfüllt.

Ein wichtiger Wunsch vieler Benutzer war es, die Möglichkeit zu haben mit dem gleichen Werkzeug, in dem die Anforderungen dokumentiert und verwaltet werden, die Produktvarianten zu konfigurieren und zu verwalten.

Tabelle 27: Analyse der VM-Werkzeuge

	GEARS	pure::variants	Metadoc
TA1	X	X	X
TA2	X	X	X
TA3	-	X	X
TA4	X	X	-
TA5	X	-	X

Basierend auf den Ergebnissen der Werkzeuganalyse wurde bei der Daimler AG entschieden, zwei Werkzeuge für Variabilitätsdokumentation und -management einzusetzen:

- Für die einfache und mittlere Ausprägung der R2F-Methode wurde eine DOORS-Erweiterung entwickelt, die in den nächsten zwei Abschnitten vorgestellt wird.
- Für die höchste Ausprägung der R2F-Methode wurde eine Zusammenarbeit mit *pure::systems* für die Weiterentwicklung des Werkzeugs *pure::variants* angestrebt. Als Ergebnis dieser Zusammenarbeit entstand *pure::variants Connector* für DOORS.

Die Dokumentation der variablen Eigenschaften basiert in beiden Werkzeugen auf Grundlagen der Merkmalsmodellierung und dem R2F-Ansatz. Somit ist ein Wechsel des Werkzeugs bei Bedarf jederzeit ohne großen Aufwand möglich.

9.2.1 Erweiterung der DOORS-Funktionalität

Für die Dokumentation und Verwaltung der Variabilität bei kleinerer bis mittlerer Anzahl von Variabilitätskriterien wurde bei der Daimler AG ein Varianten-Modul entwickelt, das

eine Erweiterung der DOORS-Funktionalitäten ermöglicht. Das Varianten-Modul ist ein formales DOORS-Modul, das die Entwickler bei der Erstellung der Spezifikationen von Produktvarianten unterstützt. Der Aufbau und die Funktionalität dieses Varianten-Moduls werden in diesem Abschnitt detailliert vorgestellt.

Für die Darstellung der Variabilitätskriterien und der dazugehörigen Beziehungen wurden in dem Varianten-Modul das Attribut *Object Type* und Doors-Links verwendet. In Tabelle 28 werden die Werte des Attributs *Object Type* vorgestellt, welche den Benutzern zur Verfügung stehen.

Tabelle 28: Attribut Object Type

Werte des Attributs <i>Object Type</i>	Bedeutung
Variant Document	Dieser Wert markiert das erste Objekt im Variantenmodul. Es ist die Voraussetzung für die Herstellung der Verknüpfung zum Spezifikationsmodul.
Criteria	Dieser Wert wird den variablen Eigenschaften eines Produktes (Variabilitätskriterium) zugeordnet.
Criteria Value	Jede variable Eigenschaft hat mindestens zwei (disjunkte) Ausprägungen.
Combination	Eine Kombination ist eine Auswahl der Ausprägungen der Variabilitätskriterien unter Berücksichtigung der Regeln. Die einzelnen Ausprägungen der Variabilitätskriterien werden über Links miteinander verknüpft (AND).
Configuration	Eine Konfiguration ist eine Selektion von Ausprägungen der Variabilitätskriterien und/oder Kombinationen, die in einer bestimmten Produktvariante enthalten sind. Die einzelnen Elemente werden über Links miteinander verknüpft (AND).
Negation	Eine Negation ist ein Hilfskonstrukt, das den logischen Operator NOT darstellt. Der ausgehende Doors-Link vom Negationsobjekt zeigt auf die negierte Merkmalsausprägung.
Group	Eine Gruppierung ist ebenfalls ein Hilfskonstrukt und stellt den logischen Operator XOR dar.
Rule	Eine Regel beschreibt die Beziehungen zwischen zwei Ausprägungen der Variabilitätskriterien. Der eingehende Link auf Rule wird als Bedingung (IF) und der ausgehende Link von Rule auf einer Ausprägung als Wirkung (THEN) interpretiert.

Zusätzlich zum Attribut *Object Type* können folgende Attribute verwendet werden, um ein Variabilitätsmodell aufzubauen:

- *Simple Formula*: Dieses Attribut stellt die formale Zusammensetzung der Negationen, Groups, Rules, Combinations und Configurations dar.
- *Object Heading* und *Object Text*: Diese Attribute ermöglichen, die Variabilitätskriterien, Regeln, Kombinationen und Konfigurationen textuell zu beschreiben.
- *Short Text*: Dieses Attribut ermöglicht, die Texte aus dem vorherigen Attribut durch Abkürzungen zu ersetzen, um Formeln lesbarer zu gestalten.
- *Visible*: Dieses Attribut mit den Werten „true“ und „false“ erlaubt die Steuerung des Imports der Variabilitätskriterien in das Spezifikationsmodul.

Im nächsten Abschnitt werden die Anwendungsmöglichkeiten des Varianten-Moduls präsentiert.

9.2.2 Anwendung des Varianten-Moduls

Das Varianten-Modul wird bei der Dokumentation und Verwaltung der Variabilität in Anforderungsdokumenten eingesetzt. In diesem Kapitel wird beschrieben, wie das Varianten-Modul bei der Anwendung des R2F- Ansatzes verwendet werden kann.

Der R2F-Prozess besteht aus fünf Schritten (siehe Kapitel 7.2). In den folgenden Abschnitten wird beschrieben, wie das Varianten-Modul den R2F-Ansatz in einzelnen Prozessschritten unterstützt.

1. Variabilitätskriterien identifizieren

Um die Variabilitätskriterien zu identifizieren, kann entweder ein Spezifikationsreview durchgeführt oder das MIA-Tool verwendet werden. Es ist auch möglich beide Maßnahmen zu kombinieren (zuerst MIA anzuwenden und darauffolgend ein Review durchzuführen), um den Variabilitätskriterien-Identifizierungsprozess effektiv und effizient zu gestalten.

2. Variabilitätskriterien dokumentieren

Die identifizierten Variabilitätskriterien werden als *Kriterium* und deren Ausprägungen als *Criteria Value* in dem Varianten-Modul dokumentiert. Die Hierarchie zwischen Objekten mit den Werten *Criteria* und *Criteria Value* beschreibt die Eltern-Kind-Beziehungen zwischen gefundenen Variabilitätskriterien und deren Ausprägungen. Dabei werden nur die untersten Eltern-Kind-Beziehungen eines Variabilitätsmodells durch die DOORS-Objekt-Hierarchie abgebildet. Alle anderen (sowohl hierarchischen als auch nicht-hierarchischen) Beziehungen des Variabilitätsmodells müssen mit Hilfe von *Rules* abgebildet werden. Abbildung 30 präsentiert den Auszug des Varianten-Moduls für die Komponente *Sonnenblende*. In diesem Varianten-Modul sind drei Variabilitätskriterien (Beleuchtung, Ausziehbarkeit und Dekor) dokumentiert. Im unteren Bereich („1.2 Teil-Formeln“) können die häufigsten Kombinationen der Variabilitätskriterien (*Combinations*) oder Negationen (*Negations*) der Variabilitätskriterien angelegt werden. Diese Kombinationen und Negationen werden bei der Erstellung von Konfigurationen benutzt.

Object Type	Beschreibung	Einfache Formel	Ausmultiplizierte Formel
Variant Document	1 Varianten Modul		
	1.1 Merkmale		
Criteria	1.1.1 Beleuchtung		
Criteria Value	Mit Beleuchtung	Mit Beleuchtung	Mit Beleuchtung
Criteria Value	Ohne Beleuchtung	Ohne Beleuchtung	Ohne Beleuchtung
Criteria	1.1.2 Ausziehbarkeit		
Criteria Value	ausziehbar	ausziehbar	ausziehbar
Criteria Value	nicht ausziehbar	nicht ausziehbar	nicht ausziehbar
Criteria	1.1.3 Dekor		
Criteria Value	PVC	PVC	PVC
Criteria Value	Alcantara	Alcantara	Alcantara
Criteria Value	Stoff	Stoff	Stoff
Combination Cluster	1.2 (Teil-)Formeln		
Combination	ausziehbare PVC-Blende	(ausziehbar) AND (PVC)	(ausziehbar) AND (PVC)
Negation	nicht ausziehbare PVC-Blende	NOT (ausziehbare PVC-Blende)	NOT ((ausziehbar) AND (PVC))
Group	PVC oder Stoff-Blende	(PVC) OR (Stoff)	(PVC) OR (Stoff)
Combination	ausziehbare PVC oder Stoff-Blende	(PVC oder Stoff-Blende) AND (ausziehbar)	((PVC) OR (Stoff)) AND (ausziehbar)

Abbildung 30: Dokumentation der Variabilitätskriterien im Varianten-Modul

3. Anforderungen zu Variabilitätskriterien und Ausprägungen zuordnen

Nachdem die Variabilitätskriterien identifiziert und dokumentiert sind, muss das Varianten-Modul mit dem entsprechenden Anforderungsmodul verknüpft werden. Es ist möglich ein Varianten-Modul mit mehreren Anforderungsmodulen zu verbinden. Ein Anforderungsmodul kann sich jedoch nur auf ein Variantenmodul VM beziehen. Die Verknüpfung erfolgt über ein DOORS-proprietäres Linkmodul. Die Variabilitätskriterien werden über die Import-Funktion als DOORS-Attribute des Typs Multi-Enumeration angelegt. Die Ausprägungen der Variabilitätskriterien stehen dann als Werte des Attributes zur Verfügung. Initial gelten die Anforderungen für alle Variabilitätskriterien.

Die variablen Anforderungen werden den relevanten Variabilitätskriterien bzw. deren Ausprägungen zugeordnet. Dabei stehen mehrere Funktionen (z.B. *Apply to all Children*, *Apply to Parent*) zur Verfügung, die diesen Vorgang unterstützen. Außerdem gibt es mehrere Funktionen, die die semi-automatische Validierung der Zuordnung ermöglichen. Durch die Zuordnung von Anforderungen zu Variabilitätskriterien wird der Gültigkeitsbereich einer Anforderung reduziert.

Abbildung 31 präsentiert einen Auszug des Spezifikationsmoduls der Komponente *Sonnenblende*. Dieses Spezifikationsmodul wurde mit dem entsprechenden Varianten-Modul verknüpft, und die Variabilitätskriterien wurden importiert. Danach wurden die Anforderungen den Variabilitätskriterien zugeordnet. Zum Beispiel ist die dritte Anforderung nur gültig für die *Sonnenblenden-Varianten*, die das Dekor *Stoff* haben und *nicht ausziehbar* sind.

Anforderungen	Dekor	Beleuchtung	Ausziehbarkeit
1 Beleuchtung		mit Beleuchtung	
Die Beleuchtung des zu spiegelnden Körpers wird über eine Leuchte in ZB Dachverkleidung gewährleistet.		mit Beleuchtung	
Aktiviert wird die Leuchte durch das Betätigen des klappbaren Spiegeldeckels, wodurch eine Stromschaltung (z.B. über Mikroschalter) erfolgt.		mit Beleuchtung	
Die elektrische Kontaktierung der Sonnenblende zum Leitungssatz/Leuchte soll über Gegenlager erfolgen.	Stoff	mit Beleuchtung	nicht ausziehbar
Die elektrische Kontaktierung der Sonnenblende zum Leitungssatz/Leuchte soll über Schwenklager erfolgen.	Alcantara	mit Beleuchtung	ausziehbar

Abbildung 31: Spezifikationsmodul mit Zuordnungen

4. Konfigurationen erstellen

Um ein Lastenheft für ein Produkt aus dem Anforderungsdokument zu erstellen, müssen alle allgemeinen und die für das Produkt gültigen variablen Anforderungen aus dem Anforderungsdokument ausgewählt werden. Um die für das Produkt gültigen variablen Anforderungen auswählen zu können, muss in dem Varianten-Modul eine neue Konfiguration angelegt werden. Eine Konfiguration beinhaltet alle Variabilitätskriterien des neuen Produktes. Zur Unterstützung des Entwicklers steht die syntaktische Validierung der erstellten Konfiguration sowie des gesamten Varianten-Moduls zur Verfügung. Bei der Validierung werden folgende Aspekte überprüft:

- Stimmt die Hierarchie der Werte des Attributs *Object Type*?
- Stimmen die Typen und die Anzahl der Linkziele?
- Werden bei Kombinationen und Konfigurationen keine leeren Mengen definiert?
- Gibt es eine Schnittmenge zwischen Bedingung und Implikation?

5. Vergabelastenheft für neues Produkt erstellen

Nachdem die Konfiguration für ein neues Produkt erstellt wurde, kann auch das Lastenheft innerhalb weniger Minuten aus einem Anforderungsdokument erstellt werden. Im Varianten-Modul wird die erstellte Konfiguration als Filter in das Spezifikationsmodul

importiert. Für die Erstellung des Lastenheftes für das neue Produkt muss nur der entsprechende Filter aktiviert werden. Die gefilterten Inhalte müssen in einer neuen DOORS-Sicht gespeichert werden und stellen die Basis für den Austausch mit dem Zulieferer dar.

9.3 Zusammenfassung

Der in diesem Kapitel beschriebene Einführungsprozess des MBVM wurde bei der Daimler AG in einem internem Projekt (REQUEST) angewendet.

REQUEST ist ein großes internes Projekt, das die Einführung des tool-basierten Anforderungsmanagements in verschiedenen Bereichen der PKW-Entwicklung unterstützt. Im Rahmen dieses Projektes wurde unter anderen folgenden Aktivitäten durchgeführt [BoHo11]:

- Einführung von DOORS;
- Schulungen für Entwickler;
- Anpassung der wissenschaftlichen Methoden an die Bedürfnisse der Industrie.

In der ersten Phase des Projektes wurde die neue Generation der Anforderungsdokumente erstellt. Dafür wurden

- (1) die existierenden Anforderungsdokumente in DOORS importiert,
- (2) die Anforderungsdokumente anhand von Qualitätskriterien überarbeitet und
- (3) die Anforderungsdokumente an die Bedürfnisse neuer Fahrzeugprojekte angepasst.

In der zweiten Phase vom REQUEST, bei der Erstellung der nächsten Generation von Anforderungsdokumenten, wurde die einfache Ausprägung der Variantendokumentation (Spalten-Ansatz) eingeführt. Dieser einfache Ansatz zeigte Entwicklern, dass die Anforderungen wiederverwendet werden können. Dabei wurden schnell die Nachteile des Spalten-Ansatzes (siehe 3.2) identifiziert. Zusätzlich wurde erkannt, dass die Zuordnung von Anforderungen zu Fahrzeugmodellen zwar notwendig ist für die Erstellung der entsprechenden Lastenhefte, jedoch diese Zuordnung den Grund der Variabilität einer Anforderung nicht dokumentiert.

Nachdem die Entwickler die Möglichkeit der Wiederverwendung von Anforderungen akzeptiert hatten und der Spalten-Ansatz an seine Grenzen gestoßen war, wurde entschieden den MBVM-Ansatz einzuführen. Die einzelnen Schritte dieses Einführungsprozesses wurden in Kapitel 9.1 vorgestellt. Der MBVM-Ansatz wurde nicht nur unter Berücksichtigung der Bedürfnisse der mechanischen, sondern auch der elektrischen und elektronischen Komponenten und Systeme erarbeitet. Um dabei die unterschiedlichen methodischen Bedürfnisse abzudecken, wurden unterschiedliche Methodenausprägungen definiert (9.1.1). Der für das Projekt entwickelte Migrationsprozess ermöglichte es, innerhalb kürzester Zeit und mit einem vertretbaren Aufwand den MBVM-Ansatz in den bestehenden Anforderungsdokumenten einzuführen.

Wegen der Komplexität der Variabilitätsmodelle wurde für das Projekt entschieden, den KDVM-Ansatz anzuwenden. Die ersten Erfahrungen zeigten, dass das die richtige Entscheidung war, denn innerhalb kürzester Zeit wurde dadurch die effiziente Wiederverwendung von Anforderungen auf System- und Komponentenebene ermöglicht. Die nächsten Schritte werden die Unterstützung des Konfigurationsprozesses auf allen Abstraktionsebenen sowie die Synchronisation der Änderungen in den Variabilitätsmodellen sein.

Die ersten Erfahrungen mit dem MBVM-Ansatz waren positiv. Dieser Ansatz wurde von den Pilot-Anwendern in den unterschiedlichen Bereichen gut akzeptiert.

Als Vorteile des MBVM-Ansatzes nannten die Pilot-Anwender:

- Deutliche Reduktion der Zeit bei der Erstellung eines Lastenhefts anhand des existierenden Anforderungsdokumentes;
- Unterstützung der Anforderungswiederverwendung;
- Gewinnung des Wissens über Ursachen der Variabilität.

Der im Rahmen dieser Arbeit entwickelte MBVM-Ansatz ist unabhängig von dem verwendeten Anforderungsmanagement-Tool aber die im Rahmen des Projektes erstellte Erweiterung der DOORS-Funktionalität unterstützte die Akzeptanz des MBVM-Ansatzes bei den Anwendern.

Zu den Nachteilen des MBVM-Ansatzes gehören folgende Aspekte:

- Aufwand für die Schulungen von Anwendern und Support;
- initialer Aufwand für die Erstellung der Variabilitätsmodelle;
- Notwendigkeit der Anpassung der Prozesse auf den Anforderungsmanagementprozess.

Bei der Einführung des MBVM-Ansatzes wurde besonders auf die Akzeptanz bei den Entwicklern und dem Anforderungsmanagement-Support geachtet. Nach anfänglichen Akzeptanz-Problemen beim Support wurde innerhalb eines Jahres durch mehrere Präsentationen und eine spezielle Betreuung durch das VM-Team ein Umdenken erreicht. Bei den Entwicklern, die die Notwendigkeit einer Verwaltung mehrerer Varianten hatten, war die Akzeptanz von Anfang an hoch. Die Entwickler hatten nur sehr wenige Probleme mit den Ideen und der Anwendung des MBVM-Ansatzes.

Kapitel 10

Evaluation

Das vorliegende Kapitel präsentiert die Evaluation der Ansätze für das Variantenmanagement in den bestehenden Anforderungsdokumenten, welche im Rahmen dieser Arbeit vorgestellt wurden. Die erarbeiteten Ansätze wurden während zweier Fallstudien und ersten Pilot-Anwendungen bei der Daimler AG evaluiert. Leider war die Anzahl der zur Verfügung stehenden Probanden sehr klein und somit können die Ergebnisse angezweifelt werden. Auf der anderen Seite haben die Ergebnisse sowohl von einer empirischen Studie an der Universität Ulm als auch der Fallstudie und der Pilotanwendungen bei der Daimler AG die gleiche Tendenz gezeigt und können somit als Bestätigung der in dieser Arbeit vorgestellten Methoden und Vorgehensweisen gesehen werden. Dieses Kapitel stellt die Rahmenbedingungen der durchgeführten Fallstudien dar und präsentiert die Ergebnisse und die Erkenntnisse der Evaluation.

10.1 Evaluation mit Studenten an der Universität Ulm

Um erste Erfahrungen vor dem produktiven Einsatz in der Praxis mit dem neuen Variantenmanagementansatz für Anforderungsdokumente zu sammeln, wurde im Wintersemester 2010/2011 eine empirische Fallstudie mit Studenten an der Universität Ulm durchgeführt.

10.1.1 Design der Fallstudie

An dieser Fallstudie haben sechs Studenten der Universität Ulm teilgenommen. Alle Studenten waren im Hauptstudium und hatten bereits die Vorlesung Requirements Engineering besucht. Kein Student hatte Vorkenntnisse im Bereich der Erstellung von natürlichsprachlichen Anforderungsdokumenten oder Variabilitätsmanagement gehabt.

Die Fallstudie dauerte 16 Wochen und bestand sowohl aus vier Präsenzstunden pro Woche, als auch aus Offline-Stunden. In den ersten fünf Wochen wurde das notwendige Wissen im Anforderungsmanagement (insbesondere im Bereich der Erstellung von Anforderungsdokumenten), Produktlinien und Merkmalsmodellierung aufgebaut. In den nächsten sieben Wochen waren die Studenten mit den Aufgaben zur Erstellung von Anforderungsdokumenten und Variabilitätsmodellen für die Produktlinien beschäftigt. In den darauf folgenden zwei Wochen waren die Studenten mit Aufgaben für die Erstellung

von Anforderungsdokumenten und Variabilitätsmodellen für die einzelnen Produkte beschäftigt. Abschließend gab es eine Präsentations- und Diskussionsrunde, wo die erarbeiteten Ergebnisse begutachtet wurden.

Die Aufgabe für die Studenten war es, in drei Gruppen die Anforderungsdokumente und Variabilitätsmodelle für die vorgegebenen Fahrzeugsysteme und -komponenten zu erstellen. Jede Gruppe bekam das Portfolio eines fiktiven Fahrzeugherstellers mit Angabe der aktuellen und geplanten Fahrzeugmodelle, Zielgruppen und Einsatzmärkte. Außerdem wurde für jede Gruppe ein System mit zwei beteiligten Komponenten vorgegeben:

- Gruppe 1 hatte das System *Außenlicht* mit den Komponenten *Scheinwerfer und Außenspiegel*.
- Gruppe 2 hatte das System *Schließung* mit den Komponenten *Schlüssel und Schloss*.
- Gruppe 3 hatte das System *Wischen/Waschen* mit den Komponenten *Scheinwerferwaschanlage und Wischer*.

Das Ziel dieser Fallstudie war die Überprüfung der folgenden fünf Hypothesen:

- (H1) Die Erstellung von dezentralen, konfigurationsgekoppelten Variabilitätsmodellen (KDVM) ist schnell erlernbar.
- (H2) Die Dokumentation der Änderungen in KDVM benötigt weniger Zeit als in einem zentralen Variabilitätsmodell.
- (H3) Die Erstellung der Konfigurationen beim KDVM-Ansatz ist weniger zeitaufwändig als bei der zentralen Variabilitätsmodellierung.
- (H4) Die Verwendung des R2F-Ansatzes zur Dokumentation der Variabilität in Anforderungsdokumenten ist leicht erlernbar und reduziert den zeitlichen Aufwand für die Dokumentation im Vergleich zu anderen in der Praxis verwendeten Matrix-Ansätzen.
- (H5) Die Verwendung des MIA-Ansatzes reduziert den zeitlichen Aufwand für die Identifikation der Merkmale in bestehenden Anforderungsdokumenten.

10.1.2 Ablauf der Fallstudie

In der Vorbereitungsphase bekamen die Studenten eine Einführung in die Erstellung von Anforderungsdokumenten und lernten die Möglichkeiten der Variantenverwaltung in den Anforderungsdokumenten an Beispielen für einfache Produkte (z.B. Handpresse, Mikrowelle) kennen. Dabei wurde die theoretische Wissensvermittlung stets mit praktischen Aufgaben kombiniert.

Im ersten Teil der Fallstudie sollten die Studenten die Anforderungsdokumente für die Produktlinien der ihnen zugeordneten Fahrzeugsysteme und -komponenten erstellen. Als Eingangsinformationen für diese Aufgabe erhielten die Studenten die Skizzen von Systemlastenheften. Außerdem wurden mehrere aktuelle Fahrzeugmodelle von Mercedes-Benz betrachtet, um die Funktionalität, die Eigenschaften und die Zusammenarbeit der vorgegebenen Komponenten und Systeme, zu verdeutlichen. Anhand der Ausgangsdaten und eigenständiger Recherchen mussten die Studenten die Lastenhefte entweder in MS Word oder mit dem Anforderungsmanagement-Tool DOORS erstellen. Bei der Erstellung der Systemlastenhefte wurde Gruppenarbeit und bei der Erstellung der Komponentenlastenhefte individuelle Arbeit gefordert.

Im zweiten Teil der Fallstudie sollten die Studenten die Merkmalsmodelle für die zuvor verfassten Anforderungsdokumente erstellen. Als weiteren Input für diese Aufgabe mussten die Studenten das Portfolio des Fahrzeugherstellers mit den aktuellen und geplanten Fahrzeugmodellen, Zielgruppen und Absatzmärkten beachten und bei Bedarf weitere variable Anforderungen in die erstellten Anforderungsdokumente aufnehmen. Die erstell-

ten Merkmalsmodelle sollten die Beziehungen zwischen unterschiedlichen Systemen und Komponenten berücksichtigen.

Im dritten Teil der Fallstudie sollten die Studenten anhand der vorgegebenen Produkteigenschaften die entsprechenden Konfigurationen der Merkmalsmodelle und Anforderungsdokumente erstellen.

Alle Studenten hatten zu Beginn der Fallstudie kein Wissen im Bereich Variabilitätsmodellierung und Variabilitätsmanagement. Am Ende der Fallstudie konnten alle Studenten sowohl die Merkmalmodelle erstellen als auch den R2F-Ansatz anwenden. Bei der Lösung der Aufgaben im dritten Teil der Fallstudie haben alle Studenten die KDVM-Methode ausgewählt, weil sie diese im Vergleich zu Ansätzen der zentralen Variabilitätsmodellierung als einfacher empfunden haben. Dabei wurden von den Studenten folgende Entscheidungsgründe genannt:

- Bei den dezentralen Variabilitätsmodellen ist es einfacher, den Überblick zu behalten.
- Bei den dezentralen Variabilitätsmodellen ist es leichter, die Änderungen zu dokumentieren.

Diesen Aussagen bestätigte die erste Hypothese (H1) leider nur indirekt.

Am Anfang der Fallstudie wurde eine Übung gemacht, bei der die Studenten zuerst eine Konfiguration eines Anforderungsdokumentes mit Hilfe der Matrix-Methode und danach eine andere mit Hilfe des R2F-Ansatzes erstellen sollten. Zu diesem Zeitpunkt hatten die Studenten kein Wissen über die Merkmalsmodellierung und den R2F-Ansatz. Nach dieser Übung haben alle Studenten zunächst die Matrix-Methode als einfacher empfunden.

Nach der Vermittlung des erforderlichen Wissens über die Merkmalsmodellierung und den R2F-Ansatz haben alle Studenten im dritten Teil der Fallstudie den R2F-Ansatz für die Dokumentation der Variabilität ausgewählt. Der Grund dafür waren die Vorteile des R2F-Ansatzes (z.B. Reduzierung des Aufwandes bei der wiederholten Konfigurationserstellung). Somit wurde auch die vierte Hypothese (H4) bestätigt.

Aufgrund des großen Anteils an Heimarbeit waren keine realistischen Messungen des Zeitaufwandes möglich. Die Messungen, welche während der Präsenzstunden gemacht wurden, sind nicht repräsentativ. Somit konnten die Hypothesen H1, H2, H3 und H5 in der Fallstudie mit Studenten nicht überprüft werden. Deswegen wurde anschließend eine weitere Fallstudie mit zwei wissenschaftlichen Mitarbeitern an der Universität Ulm durchgeführt.

10.2 Evaluation mit wissenschaftlichen Mitarbeitern

Als Input-Daten für die Untersuchung durch wissenschaftliche Mitarbeiter wurden die besten Spezifikationen aus der studentischen Fallstudie für *Scheinwerferwaschanlage* (DS5) und *Wischer* (DS6) ausgewählt.

10.2.1 Vergleich des R2F- und Matrix-Ansätze

Die erste Aufgabe für die wissenschaftlichen Mitarbeiter war, zwei Lastenhefte (P1 und P2) aus einem Anforderungsdokument mit Hilfe des R2F-Ansatzes bzw. des Matrix-Ansatzes zu erstellen. Die Ergebnisse dieser Untersuchung sind in Tabelle 29 dokumentiert. Der zeitliche Aufwand für die Erstellung einer Produktspezifikation aus einem Anforderungsdokument für eine Produktlinie ist mit dem R2F-Ansatz viel effektiver als mit dem Matrix-Ansatz. Die Anwendung des R2F-Ansatzes erlaubte während kürzerer Zeit und ohne Fehler die Lastenhefte für einzelne Produkte aus einem Anforderungsdokument

für eine Produktlinie zu erstellen. Die Ergebnisse dieser Aufgabe bestätigen zwar die Hypothese H4, können wegen der Probandenanzahl aber nicht als statistisch signifikant anerkannt werden.

Tabelle 29: Erstellung der Produktspezifikationen

Datensatz	Ansatz	Messgröße	Person 1	Person 2
DS5(P1)	Matrix	Zeit	00:19	-
		Fehler	18	-
	R2F	Zeit	-	00:14
		Fehler	-	0
DS5(P2)	Matrix	Zeit	00:07	-
		Fehler	26	-
	R2F	Zeit	-	00:01
		Fehler	-	0
DS6(P1)	Matrix	Zeit	-	00:12
		Fehler	-	13
	R2F	Zeit	00:05	-
		Fehler	0	-
DS6(P2)	Matrix	Zeit	-	00:03
		Fehler	-	0
	R2F	Zeit	00:03	-
		Fehler	0	-

10.2.2 Identifikation der Merkmale

Die zweite Aufgabe für die wissenschaftlichen Mitarbeiter war, die Merkmale in einem Anforderungsdokument zu identifizieren. Dabei mussten sie die Merkmale einmal mit Hilfe der Merkmalskandidatenliste, welche mit dem MIA-Ansatz generiert wurde, und einmal ohne die Merkmalskandidatenliste identifizieren. Tabelle 30 dokumentiert den benötigten Zeitaufwand für diese Aufgabe. Die Ergebnisse dieser Überprüfung waren nicht eindeutig, deswegen wurde entschieden, eine weitere erweiterte Fallstudie bei der Daimler AG durchzuführen. Diese Fallstudie wird im Abschnitt 10.3 beschrieben.

Tabelle 30: Erstellung der Produktspezifikationen

	Methode	Person 1	Person 2
DS3	Ohne	00:26	-
	MIA	-	00:26
DS4	Ohne	-	00:40
	MIA	00:15	-

Die dritte Aufgabe für die wissenschaftlichen Mitarbeiter bestand aus zwei Teilaufgaben mit dem Ziel die Hypothesen H2 und H3 zu überprüfen:

- Im ersten Teil der Aufgabe sollte man die vorgegebenen Änderungen in ein vorliegendes Variabilitätsmodell einpflegen.
- Im zweiten Teil musste man drei Konfigurationen (K1-K3) des Variabilitätsmodells anhand der vorgegeben Merkmalslisten erstellen. Dabei wurde zum einen das zentrale Variabilitätsmodell und zum anderen eine Menge der dezentralen konfigurationsgekoppelten Variabilitätsmodelle zur Verfügung gestellt.

Die Ergebnisse der dritten Aufgabe sind in der Tabelle 31 dokumentiert. Die Ergebnisse haben die aufgestellten Hypothesen bestätigt, aber wegen der zu kleinen Anzahl der Probanden müssen diese in weiteren Fallstudien überprüft werden.

Tabelle 31: Erstellung der Produktspezifikationen

Aufgabe	Zentral	Dezentral
Teil 1	00:19	00:15
Teil 2 K1	00:17	00:05
Teil 2 K2	00:10	00:15
Teil 2 K3	00:42	00:32

10.3 Evaluation bei der Daimler AG

Die ersten Ergebnisse aus den Fallstudien an der Universität Ulm haben gezeigt, dass die Anwendung der mit Hilfe des MIA-Ansatzes generierten Merkmalskandidatenliste keine Verbesserung, weder bei dem zeitlichen Aufwand noch bei der Genauigkeit der Ergebnisse (Anzahl der korrekt identifizierten Merkmale), bringt. Ein Grund für diese Ergebnisse könnte in der Qualität der verwendeten Anforderungsdokumente (Umfang, Dokumentationsstil, Realität) liegen, weil die analysierten Anforderungsdokumente durch Studenten erstellt wurden. Deswegen wurde eine weitere Fallstudie zur Evaluation des MIA-Ansatzes mit der gleichen Fragestellung unter Verwendung von echten Anforderungsdokumenten bei der Daimler AG durchgeführt.

10.3.1 Fallstudie mit Werkstudenten

In Rahmen dieser Fallstudie mussten sechs Werkstudenten der Daimler AG in vorliegenden Anforderungsdokumenten die Merkmale (1) durch einfaches Review oder (2) durch ein Review mit Hilfe einer Merkmalskandidatenliste identifizieren. Für diese Fallstudie wurden zwei reale Anforderungsdokumente (DS1 und DS2) verwendet, deren Eigenschaften bereits im Kapitel 4 vorgestellt wurden. Die Ergebnisse dieser Fallstudie sind in Tabelle 32 zusammengefasst.

Tabelle 32: Fallstudie MIA bei Daimler AG

Person	Ohne MIA (1)			Mit MIA (2)		
	Zeit	Anzahl Merkmale	Anzahl korrekter Merkmale	Zeit	Anzahl Merkmale	Anzahl korrekter Merkmale
Datensatz 1						
P1	00:40	20	9	-	-	-
P2	00:49	20	7	-	-	-
P3	00:49	26	7	-	-	-
P4	-	-	-	00:30	25	9
P5	-	-	-	01:10	21	9
P6	-	-	-	01:30	56	9
Datensatz 2						
P1	-	-	-	2:00	96	42
P2	-	-	-	5:00	234	50
P5	03:15	83	54	-	-	-
P6	04:00	107	45	-	-	-

Die Ergebnisse beider Fallstudien zur Evaluation des MIA-Algorithmus zeigten, dass die Ergebnisse sehr personen-abhängig sind und je nach Erfahrung und Wissensstand der Person sowohl zur deutlichen Reduzierung des Aufwandes bei der Merkmalsidentifizierung als auch zur Erhöhung des zeitlichen Aufwandes führen können. Ein weiterer Grund für die fehlende Bestätigung der Hypothese H5 kann in dem Umfang und der Qualität der generierten Merkmalskandidatenlisten liegen. Eine Weiterentwicklung des MIA-Algorithmus ist notwendig, um die Menge der Merkmalskandidaten zu reduzieren.

10.3.2 Pilotanwendungen

Parallel zu den vorgestellten Fallstudien wurden die weiteren, durch MIA erstellten Merkmalskandidatenlisten als Grundlagen für Interviews mit Ingenieuren in den vier Pilot-Projekten bei der Daimler AG verwendet.

Der Input für MIA waren vier Komponenten-Anforderungsdokumente. Die Beschreibung der Ergebnisse dieser Evaluation erfolgte bereits in Kapitel 4. Der Variantenmanager konnte anhand der generierten Merkmalskandidatenliste, ohne Fachwissen über die betroffene Komponente zu besitzen, die möglichen Beispiele der Merkmale dem Ingenieur nennen und ihm damit den Einstieg in den Merkmalsidentifikationsprozess erleichtern.

Anhand der identifizierten Merkmalslisten wurden die Varianten-Module aufgebaut und die variablen Anforderungen in den Anforderungsdokumenten mit Hilfe der R2F-Methode den Merkmalen zugeordnet. Diese Vorgehensweise fand große Akzeptanz bei den Anwendern, weil die Vorteile (z.B. geringerer zeitlicher Aufwand für die Erstellung eines neuen Lastenheftes) für alle Anwender offensichtlich sind. Die R2F-Methodik ermöglicht den Entwicklern, die neuen Lastenhefte für die neuen Fahrzeugmodelle, bei 100% Wiederverwendung der Anforderungen, in wenigen Minuten zu erstellen. Der initiale Aufwand für das Erlernen der neuen Methodik wurde durch intensive Betreuung des Variantenmanagement-Support-Teams unterstützt [BoHo11].

Kapitel 11

Zusammenfassung und Ausblick

Das vorliegende Kapitel fasst die Zielsetzungen und die erarbeitete Beiträge dieser Arbeit zusammen und bietet einen Ausblick für die Weiterentwicklung sowohl der erarbeiteten Ansätze als auch noch offener Problemfelder auf dem Gebiet des Variantenmanagements in Anforderungsdokumenten.

11.1 Zusammenfassung

Am Anfang dieser Arbeit wurde die Zielstellung folgendermaßen definiert:

Die Erarbeitung von methodischen Ansätzen, welche die Anwendung des merkmalsbasierten Variantenmanagements in Anforderungsdokumenten hochkomplexer und variantenreicher Produktentwicklung praxistauglich machen.

Durch Befragungen mehrerer Experten aus der Praxis und die Analyse der bereits durchgeführten Studien (vgl. [ChAt07], [Bout08], [RaGD10]) wurden sechs aktuelle Problemfelder (P1-P6) identifiziert, zu denen noch keine zufriedenstellenden Lösungen sowohl in der Forschung als auch in der Praxis existieren. Der Fokus dieser Arbeit wurde auf fünf Problemfelder (P1-P5) beschränkt. Jedes Problemfeld wurde detailliert beschrieben und ein möglicher Ansatz zur Lösung des jeweiligen Problems entwickelt.

Identifikation der Merkmale in den bestehenden Anforderungsdokumenten (P1)

Problemfeld P1 beschäftigt sich mit der Identifikation der Merkmale und der Erstellung von Merkmalsmodellen auf Basis von existierenden Anforderungsdokumenten. Der vorhandene Prozess der Identifikation von Merkmalsmodellelementen vor dieser Arbeit basierte auf einem manuellen Review [BoHo11], welches großen zeitlichen Aufwand benötigt. Die vorliegende Arbeit stellt einen semi-automatischen Ansatz Merkmal-Identifikations-Algorithmus, kurz MIA, für die Identifikation der Merkmale in den bestehenden Anforderungsdokumenten auf Basis lexikalischer Analysen vor.

Die Grundvoraussetzung für die Anwendung des MIA-Ansatzes ist, dass der Entwicklungs- und Spezifikationsprozess nicht auf den Konzepten der Produktlinienentwicklung aufbaut. Des Weiteren kann der MIA-Ansatz nur für Anforderungsdokumente in natürlicher Sprache verwendet werden, Grafiken und andere Visualisierungselemente können dabei nicht berücksichtigt werden. Die Variabilität muss durch Wörter dokumentiert sein.

Die ersten Erfahrungen mit dem vorgestellten Ansatz wurden sowohl im Rahmen zweier Fallstudien als auch während der ersten Pilot-Anwendungen bei der Daimler AG gesammelt [BoHo11].

Die Evaluation der Ergebnisse bei der Identifikation von Merkmalen hat gezeigt, dass die Trefferquote von MIA gemäß der Klassifikation von Hayes et al. [HaDS05] für semi-automatische Ansätze in die Gruppen “akzeptabel” bis “exzellent“ gehört. Bei der Genauigkeit kann MIA höchstens in die Gruppe “akzeptabel” eingestuft werden.

Um bessere Ergebnisse bei der Identifikation der Merkmale zu erzielen, können mehrere Verbesserungsmöglichkeiten ausprobiert werden. Die Gruppierung von Wörtern mit dem gleichen Stamm könnte beispielsweise die Anzahl der Merkmalskandidaten reduzieren.

Allerdings werden durch den MIA-Ansatz immer mehr Kandidaten als nötig vorgeschlagen, da auch die notwendigen Merkmale als Merkmalskandidaten vorgeschlagen werden. Und trotz der großen Menge der vorgeschlagenen Merkmalskandidaten, wird MIA nie alle Merkmale finden können. Der Grund dafür sind die impliziten Merkmale, aber auch Merkmale, die durch Zahlen, Abkürzungen, Adjektive oder Kombinationen aus Adjektiven und Substantiven dokumentiert sind.

Die Ergebnisse der Evaluation der Anwendung des MIA-Ansatzes mit dem Ziel, den zeitlichen Aufwand bei der Identifikation der Merkmale zu reduzieren, hat gezeigt, dass die gewonnenen Ergebnisse sehr personenabhängig sind. Somit kann die Anwendung des MIA-Ansatzes auf der einen Seite zur deutlichen Reduzierung des zeitlichen Aufwandes bei der Merkmalsidentifizierung und auf der anderen Seite zur Erhöhung des zeitlichen Aufwandes, führen.

Eine alternative Lösung für dieses Problemfeld wäre eine Prozessumstellung: im ersten Schritt die Merkmale identifizieren, und erst danach ein Anforderungsdokument erstellen. Die Herausforderung in der Praxis besteht jedoch im Umgang mit bestehenden Anforderungsdokumenten. Somit ist diese alternative Lösung in der aktuellen Praxis nicht akzeptabel. Allerdings könnte diese Lösung in der Zukunft anwendbar sein.

Handhabung der Anzahl der Variabilitätsmodelle (P2)

Zur Lösung des Problemfeldes P2, das die Handhabung der Anzahl der Variabilitätsmodelle beschreibt, wurde in Rahmen der vorliegenden Arbeit ein konfigurationsgekoppelter Variabilitätsmodellierungsansatz (KDVM) vorgestellt. Der KDVM-Ansatz bietet eine mögliche Lösung des Komplexitätsproblems und basiert auf den Ideen der dezentralen Variabilitätsmodellierung, kombiniert mit dem Vorschlag zur Kopplung der einzelnen Variabilitätsmodelle über Konfigurationen.

Die ersten Erkenntnisse aus den Fallstudien und Pilot-Anwendungen haben gezeigt, dass der vorgestellte Ansatz leicht erlernbar ist und eine stufenweise Einführung in einem gewachsenen industriellen Umfeld ermöglicht. Insbesondere ist das Thema Änderungs-pflege in einem Variabilitätsmodell bei der Anwendung dieses Ansatzes leichter umsetzbar als bei der zentralen Variabilitätsmodellierung.

Eine weitere mögliche Alternative zur Lösung des Komplexitätsproblems wäre die Erweiterung des Multi-Level-Feature-Tree-Ansatzes, kurz MLFT, von Reiser [Reis08] um mehrere Referenzmodelle zu einem referenzierten Modell und die Erarbeitung eines entsprechenden Prozesses.

Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten (P3)

Eine mögliche Lösung des Problemfeldes P3, das sich mit der Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten beschäftigt, wurde im Kapitel 7 vorgestellt. In dieser Arbeit wurde ein neuer Vorschlag für eine Dokumentationsform der Variabilitätsinformationen in den Anforderungsdokumenten unterbreitet. Der vorgestellte Requirement-to-Feature (R2F)-Ansatz für das Variantenmanagement in Anforderungsdokumenten

kumenten basiert auf den Ideen der Merkmalsmodellierung [Kang90] und ermöglicht die effiziente Dokumentation von Variabilitätsinformationen in Anforderungsdokumenten. Zusätzlich wurden zwei Algorithmen entwickelt, welche den Autor des Anforderungsdokuments bei der Dokumentation der Variabilität und bei der Überprüfung der Korrektheit der erstellten Zuordnung in den Anforderungsdokumenten unterstützen.

Die Anwendung des R2F-Ansatzes wurde in mehreren Pilot-Anwendungen bei der Daimler AG evaluiert. Die vorgestellte Vorgehensweise zur Dokumentation und Verwaltung der Variabilität ist leicht erlernbar und wird schnell von den Anwendern akzeptiert. Voraussetzung zur erfolgreichen Umsetzung des R2F-Ansatzes ist der initiale Aufwand für das Erlernen der neuen Methode [BoHo11].

Die vorgestellte Methode ermöglicht es den Entwicklern, neue Lastenhefte auf Basis der vorhandenen Anforderungsdokumente bei der 100%-tiger Wiederverwendung der Anforderungen in wenigen Minuten zu erstellen. Aber auch bei einem niedrigeren Grad der Wiederverwendung von Anforderungen ist die Reduzierung des zeitlichen Aufwandes für die Erstellung eines neuen Lastenheftes erkennbar.

Eine mögliche alternative Lösung dieses Problems wäre die Umstrukturierung der Anforderungsdokumente anhand der Variabilitätsinformationen. In Anforderungsdokumenten mit einer geringen Anzahl von Variabilitätskriterien wäre das möglich, aber in Anforderungsdokumenten für hochkomplexe Produkte ist es jedoch nicht mehr durchführbar, da eine Anforderung zu mehreren Variabilitätskriterien zugeordnet wird.

Die in der Arbeit vorgestellte Dokumentationsmöglichkeit der Variabilitätsinformationen in Anforderungsdokumenten wurde mit mehreren Anforderungsmanagement-Tool-Herstellern diskutiert. Als Ergebnis des Wissensaustauschs, erweiterte die Firma pure::systems ihr Tool pure::variantes [pure] um die Funktion, die Variabilitätsinformationen in Anforderungsdokumenten in DOORS zu dokumentieren.

Anwenderunterstützung während der Konfigurationserstellung (P4)

Für die Lösung des Problemfeldes bezüglich der Anwenderunterstützung während der Konfigurationserstellung (P4), wurde eine Vorgehensweise zur Unterstützung der Anwender beim Konfigurationserstellungsprozess vorgestellt. Die vorgestellte Vorgehensweise basiert auf der optimalen Fragestellung zur Auswahl der Merkmale bei der Konfigurationserstellung. Für die Generierung der optimalen Fragenfolge wurde ein Ansatz entwickelt, welcher auf der Analyse der Beziehungen zwischen einzelnen Merkmalen in einem oder mehreren Merkmalsmodellen und der Berücksichtigung der Auswahlwahrscheinlichkeit eines Merkmals basiert.

Die Anwendung des in dieser Arbeit vorgestellten Ansatzes zur Bestimmung der optimalen Fragenfolge kann ohne zeitliche Nachteile verwendet werden, weil sich die Dauer der Konfigurationserstellung nie verlängert, und im schlimmsten Fall der Dauer des manuellen Konfigurationsprozesses entspricht.

In der Forschung existieren mehrere Ansätze zur Lösung unterschiedlicher Optimierungsprobleme, das Thema der Optimierung des Konfigurationsprozesses von Variabilitätsmodellen ist jedoch noch nicht erforscht. Zum Zeitpunkt der Erstellung dieser Arbeit wurden lediglich zwei Ansätze ([NöEg11] und [ChEr11]) gefunden, welche sich mit diesem Thema beschäftigen, wobei beide alternative Ansätze noch nicht in der Praxis erprobt wurden.

Fehlende Beschreibungen der möglichen Einführungsstrategie (P5)

Zum Abschluss der vorliegenden Arbeit wurde ein Leitfaden zur Einführung des merkmalsbasierten Variantenmanagements in Anforderungsdokumenten inklusive möglicher Migrationsszenarien von bereits existierenden Ansätzen vorgestellt. Dabei wurden so-

wohl die möglichen unterschiedlichen Stufen der Einführung präsentiert als auch die mögliche Werkzeug-Unterstützung bei der Dokumentation und Verwaltung der Variabilität in den Anforderungsdokumenten diskutiert. Anschließend wurden die Erfahrungen aus dem internen Projekt bei der Daimler AG präsentiert.

Während der Erstellung dieser Arbeit wurde kein alternativer Einführungs- bzw. Migrationsleitfaden in der Literatur gefunden.

Die wesentlichen Beiträge der vorliegenden Arbeit (B1-B5) lassen sich also in der Kurzform wie folgt darstellen:

- (B1) Ein konfigurationsgekoppelter Variabilitätsmodellierungsansatz zur Lösung des Komplexitätsproblems.
- (B2) Ein semi-automatischer Ansatz für die Identifikation von Merkmalen (MIA) in bestehenden Anforderungsdokumenten.
- (B3) Eine geeignete Dokumentationsform für Variabilitätsinformationen in Anforderungsdokumenten.
- (B4) Ein Ansatz zur Unterstützung des Konfigurationsprozesses.
- (B5) Empfehlungen für den Aufbau und die Einführung eines merkmalsbasierten Variantenmanagements in einem gewachsenen industriellen Umfeld.

Die erarbeiteten Beiträge bieten die methodischen Ansätze, welche die Anwendung des merkmalsbasierten Variantenmanagements in Anforderungsdokumenten einer hochkomplexen und variantenreichen Produktentwicklung in der Praxis ermöglichen – was dem Ziel dieser Arbeit entspricht.

Alle erarbeiteten Ansätze sind flexibel, effizient sowie tool- und domänenunabhängig. Die Ergebnisse der ersten Pilot-Anwendungen zeigen, dass alle Ansätze sofort in der Praxis umgesetzt werden können.

11.2 Ausblick

Das Thema des Variantenmanagements ist ein Schwerpunktthema in der Produktentstehung – auch in der Automobilindustrie. Alle Automobilhersteller erweitern von Jahr zu Jahr ihre Produktpalette, um bestehende Marktanteile zu behalten und um neue Kunden zu gewinnen. Der Erfolg eines breiten Produktportfolios wiederum hängt in erster Linie von der Effizienz in der Produktentwicklung und Produktherstellung ab. Für die Steigerung der Effizienz der Produktentwicklung und Produktherstellung sind besonders zwei Themen von großer Bedeutung: Reduzierung der technischen Variantenvielfalt bei steigender Variantenzahl von Endprodukten und Beherrschung der Variabilität in diesem komplexen Umfeld.

Variantenreduzierung

Um die Variantenzahl zu reduzieren entwickeln fast alle Fahrzeughersteller heutzutage neue Produkte auf der Basis sogenannter Plattformstrategien. Die Idee hinter der Plattformstrategie ist es nicht für jedes Modell eine eigene Bodengruppe mit passenden Motoren, Getrieben und Komponenten zu entwickeln, sondern ähnliche Fahrzeuge aus gleichen Komponenten zu bauen. Zum Beispiel baut die Porsche AG die Modelle Boxster und Cayman auf der gleichen Plattform, die Fahrzeuge haben jedoch ein unterschiedliches Aussehen. Die gleiche Plattformstrategie wird auch von unterschiedlichen Fahrzeugherstellern genutzt, die in einer strategischen Allianz sind, um Wettbewerbsvorteile zu bekommen. Zum Beispiel baute Ford (Fahrzeugmodelle C-MAX und Kuga), Volvo (Fahrzeugmodelle S40/V50, C30, C70 II) und Mazda (Fahrzeugmodelle Mazda 3 und 5) ihre unterschiedlichen Produkte auf der gleichen Plattform namens C1[Rees12].

Modellhaft beschrieben, kann man sagen, dass die Volkswagen AG die Plattformstrategie auf der Idee eines Lego-Baukastens entwickelt, wobei sich in dem Baukasten festgelegte Fahrzeugteile (z.B. Boden, Motoren, Getriebe, Sitze, Assistenzsysteme) befinden. Die Teile aus dem Baukasten werden benutzt um unterschiedliche Fahrzeugmodelle von verschiedenen Konzernmarken zu bauen. Auf diese Weise wurden von der Volkswagen AG verschiedene Modelle, wie VW New Beetle, Bora, Caddy Life, Seat-Leon und Toledo, Skoda Octavia, Audi A3 und Audi TT entwickelt [Rees12].

Bei der Benutzung einer Plattformstrategie kommt es vor allem darauf an, dass sich die Endprodukte aus Kundensicht unterscheiden. Der Kunde muss auf den ersten Blick die Unterschiede zwischen Audi, VW, Seat oder Skoda erkennen. Ansonsten kann es zum Verlust von Kunden in bestimmten Marktsegmenten kommen. 2001 wurde beispielsweise bei der Herstellung des Modells Jaguar X-Type Teile des Modells Ford Mondeo benutzt, was für die Kunden offensichtlich war. Das wurde als einer der Gründe für den Misserfolg des Jaguar X-Type auf dem Markt identifiziert und das Modell wurde 2009 ohne Nachfolger eingestellt [Rees12].

Auf Basis der neuen Plattform plant die Volkswagen AG im Jahr 2018 über 40 Modelle der Kompakt- und Mittelklasse zu produzieren und rechnet dabei mit Stückzahlen von bis zu sechs Millionen Fahrzeugen jährlich. Mit der Strategie des Modularen Querbaukastens will die Volkswagen AG den einmaligen Aufwand und die Kosten pro Einheit um 20% senken und den Zeitaufwand für die Fertigung jedes Fahrzeugs um 30% reduzieren [Rees12].

Variantenmanagement

Die in der vorliegenden Arbeit entwickelten Ansätze liefern einen Beitrag zur Beherrschung des Variantenmanagements in Anforderungsdokumenten. Alle erarbeiteten Ansätze können sowohl einzeln als auch kombiniert verwendet werden. Eine mögliche Weiterentwicklung der vorliegenden Arbeit ist die Erarbeitung eines Frameworks, in welchem die vorgestellten Ansätze integriert werden, mit dem Ziel eine durchgängige Methode für das Variantenmanagement in allen Anforderungsdokumenten umzusetzen.

Der in dieser Arbeit entwickelte MIA-Algorithmus hat hohes Weiterentwicklungspotential. Ein Ziel der Weiterentwicklung kann sein, die Qualität der Kandidatenliste zu erhöhen, wobei die Anzahl der möglichen Merkmalskandidaten so klein wie möglich zu halten ist. Eine mögliche Richtung der Weiterentwicklung zur Erreichung dieses Ziels, ist die Gruppierung von Synonymen und Begriffen mit gleichem Wortstamm. Eine Überprüfung, ob Data-Mining-Ansätze hier angewendet werden könnten, wäre empfehlenswert. Ein weiteres Ziel der Weiterentwicklung des MIA-Ansatzes kann sein, eine Methode zur Identifikation der Beziehungen zwischen Merkmalskandidaten zu entwickeln. In diesem Fall wäre es möglich, nicht nur die Merkmale aufzulisten, sondern die kompletten Merkmalsmodelle zu erstellen.

In dieser Arbeit wurden nur zwei grundlegende Beziehungstypen betrachtet (benötigt und schließt aus). In der Literatur existieren mehrere Arbeiten (z.B. [KCNP90, StRP03, LeKa04]) die eine feinere Typisierung der möglichen Beziehungen zwischen Merkmalen vorschlagen. In der vorgestellten Lösung zur Berechnung der optimalen Fragenfolge wurden dementsprechend nur die grundlegenden Beziehungstypen berücksichtigt. Der Algorithmus zur Bestimmung der optimalen Fragenfolge basiert auf einer festgelegten Priorisierung der betrachteten Beziehungstypen. Bei der Berechnung der Anzahl von Konfigurationen wurden wiederum die Beziehungen zwischen einzelnen Merkmalen nicht berücksichtigt. Eine mögliche Weiterentwicklung des vorgestellten Ansatzes zur Bestimmung der optimalen Fragenfolge ist die genaue Berechnung der Anzahl von Konfigurationen unter Berücksichtigung der Beziehungen zwischen einzelnen Merkmalen. Eine weitere mögliche Weiterentwicklung des Ansatzes zur Bestimmung der optimalen

Fragenfolge, wäre die Betrachtung der Verfeinerung von Beziehungstypen zwischen den Merkmalen.

Die vorliegende Arbeit erweitert das Anwendungsgebiet der Merkmalsmodellierung um die Welt der Anforderungen. Die Verwendung der Merkmalsmodelle für die Dokumentation der Variabilität in den Anforderungsdokumenten bietet die Möglichkeit, einen Ansatz für ein durchgängiges Variantenmanagement in dem kompletten Entwicklungsprozess zu erarbeiten und zu etablieren.

Die in der vorliegenden Arbeit vorgestellten KDVM- und MIA-Ansätze sowie die Ergebnisse der empirischen Untersuchungen von MIA- und R2F-Ansätzen, liefern wertvolle Informationen für die weitere Forschung auf diesem Gebiet.

Für die Praxis sind insbesondere der R2F-Ansatz und der Einführungs- und Migrationsleitfaden interessant. Die Anwendung der vorgestellten Ansätze kann sowohl die Effizienz des Variabilitätsmanagements in Anforderungsdokumenten steigern, als auch die Qualität der Anforderungsdokumente durch die Reduzierung der redundanten Anforderungen erhöhen. Ein großes Interesse für die Praxis stellt auch die Frage der Kopplung von Anforderungen und Stücklisten dar. Die Verbindung von einzelnen Anforderungen mit Stücklistennummern hätte die Nachvollziehbarkeit der Anforderungsumsetzung für jede Komponenten- bzw. Systemvariante erweitert. Es wäre möglich, im Fall eines Fehlers die ursprünglichen Anforderungen eindeutig zu identifizieren.

Die in dieser Arbeit nicht adressierte Frage der Evolution von Merkmalsmodellen stellt die nächste Herausforderung für Forschung und Praxis dar. Die erarbeiteten Ansätze stellen keine zusätzlichen Herausforderungen an die Evolution von Merkmalsmodellen. Der dezentrale Variabilitätsmodellierungsansatz kann sogar die Lösung des Evolutionsproblems unterstützen, da bei einem zentralen Variabilitätsmodell die Änderungen im ganzen Modell gepflegt werden müssen, während bei dezentralen Variabilitätsmodellen die Änderungspflege nur in einzelnen betroffenen Modellen erfolgt.

Anhang

In diesem Anhang wird die Tabelle mit Endungen für die Substantive vorgestellt [Leiz96], welche bei dem Normalisierungsschritt vom MIA-Tool verwendet wird.

Nr	Singular				Plural				Gen	Uml
	Nom	Gen	Dat	Akk	Nom	Gen	Dat	Akk		
1	-	s	-	-	-	-	n	-	m	-
2	-	s	-	-	-	-	n	-	m	u
3	-	s/es	e/-	-	e	e	en	e	m	-
4	-	s/es	e/-	-	e	e	en	e	m	u
5	-	s	-	-	-	-	-	-	m	-
6	-	s	-	-	-	-	-	-	m	u
7	-	s/es	e/-	-	er	er	ern	er	m	-
8	-	s/es	e/-	-	er	er	ern	er	m	u
9	-	s/es	e/-	en	en	en	en	en	m	-
10	-	s	-	-	n	n	n	n	m	-
11	-	en	en	en	en	en	en	en	m	-
12	-	n	n	n	n	n	n	n	m	-
13	e	en	en	en	en	en	en	en	m	-
14	e	ens	en	en	en	en	en	en	m	-
15	en	ens	en	en	en	en	en	en	m	-
16	en	ens	en	en	en	en	en	en	m	u
17	-	s	-	-	s	s	s	s	m	-
18	-	ses	-	-	se	se	sen	se	m	-
19	-	s	-	-	-	-	n	-	n	-
20	-	s	-	-	-	-	n	-	n	u
21	-	s/es	e/-	-	e	e	en	e	n	-
22	-	s/es	e/-	-	e	e	en	e	n	u
23	-	s	-	-	-	-	-	-	n	-
24	-	s/es	e/-	-	er	er	ern	er	n	-
25	-	s/es	e/-	-	er	er	ern	er	n	u
26	-	s/es	e/-	-	en	en	en	en	n	-
27	en	ens	en	en	en	en	en	en	n	-
28	-	s	-	-	s	s	s	s	n	-
29	e	es	e	e	e	e	en	e	n	-
30	e	es	e	e	en	en	en	en	n	-
31	n	ns	n	n	n	n	n	n	n	-
32	-	ens	en	-	en	en	en	en	n	-
33	-	ses	-/se	-	se	se	sen	se	n	-
34	-	-	-	-	e	e	en	e	f	-
35	-	-	-	-	e	e	en	e	f	u
36	-	-	-	-	-	-	n	e	f	u
37	-	-	-	-	en	en	en	en	f	-
38	-	-	-	-	n	n	n	n	f	-
39	e	e	e	e	en	en	en	en	f	-
40	-	-	-	-	s	s	s	s	f	-
41	-	-	-	-	se	se	sen	se	f	-
42	-	-	-	-	nen	nen	nen	nen	f	-
43	-	-	-	-	-	-	-	-	f	-
44	-	-	-	-	-	-	-	-	m	-
45	-	s	-	-	ein	ein	ien	ein	n	-
46	a	a	a	a	en	en	en	en	f	-
47	a	as	a	a	en	en	en	en	n	-

Nr	Singular				Plural				Gen	Uml
	Nom	Gen	Dat	Akk	Nom	Gen	Dat	Akk		
48	um	ums	um	um	en	en	en	en	n	
49	um	ums	um	um	a	a	a	a	n	-
50	is	is	is	is	en	en	en	en	n	-
51	on	ons	on	on	a	a	a	a	n	-
52	us	us	us	us	ora	ora	ora	ora	m	-
53	us	us	us	us	een	een	een	ren	m	-
54	o	o	o	o	en	en	en	rn	n	-
55	us	us	us	us	en	en	en	rn	m	-
56	os	os	os	os	en	en	en	rn	n	-
57	x	x	x	x	zen	zen	zen	zen	f	-
58	s	s	s	s	nten	nten	nten	nten	m	-
59	-	s	-	-	nen	nen	nen	nen	n	-
60	s	s	s	s	en	en	en	en	n	-
61	-	-	-	-	-	-	-	-	n	-
62	-	s	-	-	ten	ten	ten	ten	m	-

In diesem Anhang sind die Ergebnisse der Evaluierung des MIA-Algorithmus in zwei Durchläufen dokumentiert.

Phase 1

Erster Durchlauf

Werte	DS1				DS2			
	Manuell	MIA	R	P	Manuell	MIA	R	P
Anzahl Anforderungen	1124	1124			543	543		
Anzahl von Merkmalskandidaten		241				380		
Anzahl von Merkmalen	98	44	0,45	0,18	9	8	0,89	0,02
Anzahl von Merkmalen mit Kinder-Merkmalen	15	8	0,53	0,03	4	3	0,75	0,01
Anzahl von Merkmalen ohne Kinder	83	36	0,43	0,15	5	5	1,00	0,01
Dauer in Minuten		25				36		

Werte	DS3				DS4			
	Manuell	MIA	R	P	Manuell	MIA	R	P
Anzahl Anforderungen	712	712			274	274		
Anzahl von Merkmalskandidaten		864				347		
Anzahl von Merkmalen	46	17	0,37	0,02	42	8	0,19	0,02
Anzahl von Merkmalen mit Kinder-Merkmalen	16	5	0,31	0,01	14	3	0,21	0,01
Anzahl von Merkmalen ohne Kinder	30	12	0,4	0,01	28	5	0,18	0,01
Dauer in Minuten		90				24		

Zweiter Durchlauf

Werte	DS1				DS2			
	Manuell	MIA	R	P	Manuell	MIA	R	P
Anzahl Anforderungen	1012	1012			56	56		
Anzahl von Merkmalskandidaten		178				48		
Anzahl von Merkmalen	98	40	0,41	0,22	9	8	0,89	0,17
Anzahl von Merkmalen mit Kinder-Merkmalen	15	6	0,40	0,15	4	3	0,75	0,06
Anzahl von Merkmalen ohne Kinder	83	34	0,41	0,19	5	5	1,00	0,10
Dauer in Minuten		16				1		

Wert	DS4			
	Manuell	MIA	R	P
Anzahl Anforderungen	256	256		
Anzahl von Merkmalskandidaten		324		
Anzahl von Merkmalen	42	8	0,19	0,02
Anzahl von Merkmalen mit Kinder-Merkmalen	14	3	0,21	0,01
Anzahl von Merkmalen ohne Kinder	28	5	0,18	0,02
Dauer in Minuten		21		

Phase 2

Erster Durchlauf

Werte	DS1				DS2			
	Manuell	MIA	R	P	Manuell	MIA	R	P
Anzahl Anforderungen	1124	1124			543	543		
Anzahl von Merkmalskandidaten		211				205		
Anzahl von Merkmalen	98	44	0,45	0,21	9	8	0,89	0,04
Anzahl von Merkmalen mit Kinder-Merkmalen	15	8	0,53	0,04	4	3	0,75	0,01
Anzahl von Merkmalen ohne Kinder	83	36	0,43	0,17	5	5	1,00	0,02
Dauer in Minuten		20				35		

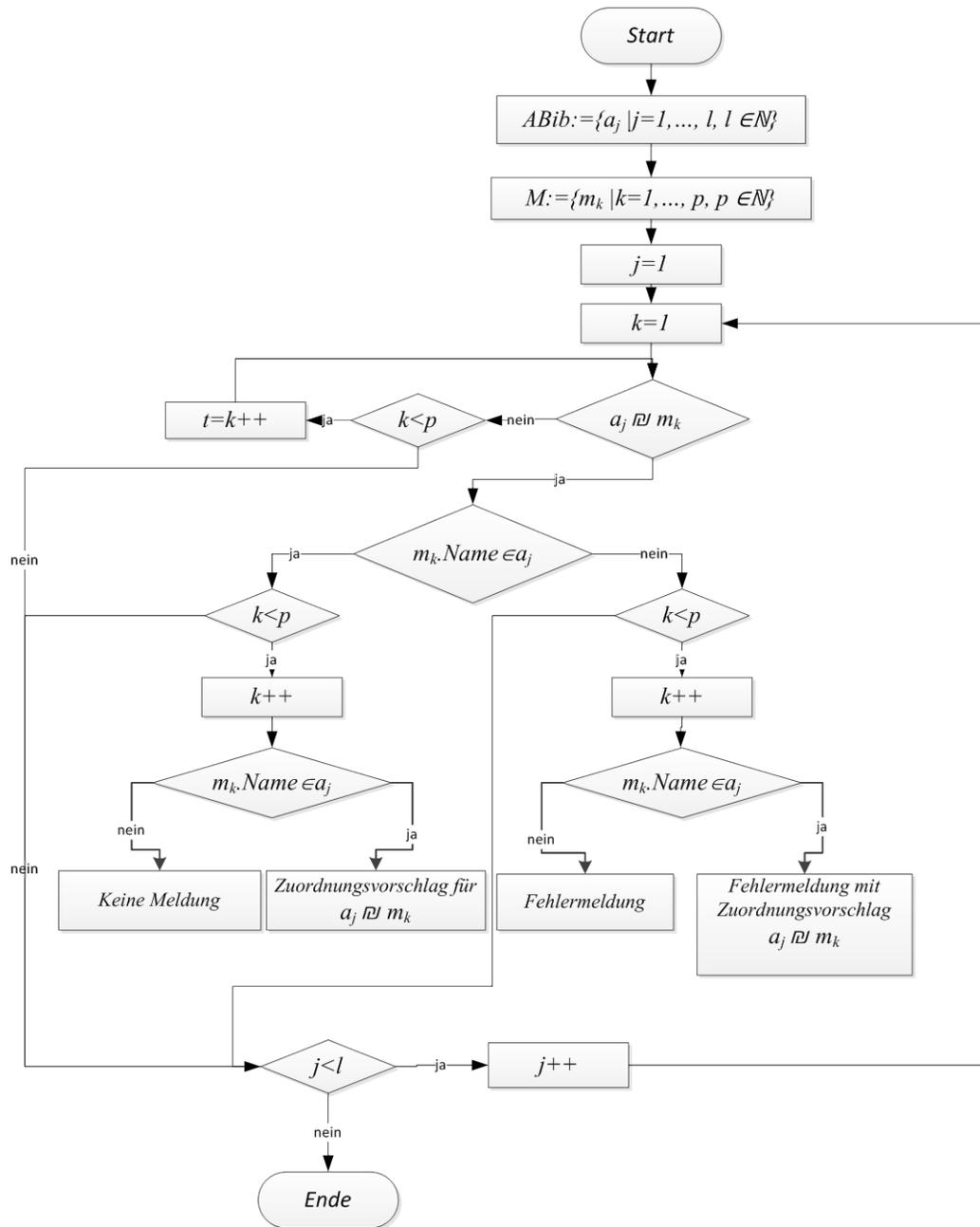
Werte	DS3				DS4			
	Manuell	MIA	R	P	Manuell	MIA	R	P
Anzahl Anforderungen	712	712			274	274		
Anzahl von Merkmalskandidaten		629				217		
Anzahl von Merkmalen	46	19	0,41	0,03	42	13	0,31	0,06
Anzahl von Merkmalen mit Kinder-Merkmalen	16	6	0,38	0,01	14	3	0,21	0,01
Anzahl von Merkmalen ohne Kinder	30	13	0,43	0,02	28	10	0,36	0,05
Dauer in Minuten		91				23		

Zweiter Durchlauf

Werte	DS1				DS2			
	Manuell	MIA	R	P	Manuell	MIA	R	P
Anzahl Anforderungen	1012	1012			56	56		
Anzahl von Merkmalskandidaten		164				40		
Anzahl von Merkmalen	98	50	0,51	0,30	9	8	0,89	0,20
Anzahl von Merkmalen mit Kinder-Merkmalen	15	11	0,73	0,22	4	3	0,75	0,08
Anzahl von Merkmalen ohne Kinder	83	39	0,47	0,24	5	5	1,00	0,13
Dauer in Minuten		17				1		

Werte	DS4*			
	Manuell	MIA	R	P
Anzahl Anforderungen	256	256		
Anzahl von Merkmalskandidaten		203	164	
Anzahl von Merkmalen	42	13	0,31	0,06
Anzahl von Merkmalen mit Kinder-Merkmalen	14	3	0,21	0,01
Anzahl von Merkmalen ohne Kinder	28	10	0,36	0,05
Dauer in Minuten		22		

In diesem Anhang ist der Zuordnungsvorschlagalgorithmus (ZV-Algorithmus) vorgestellt.



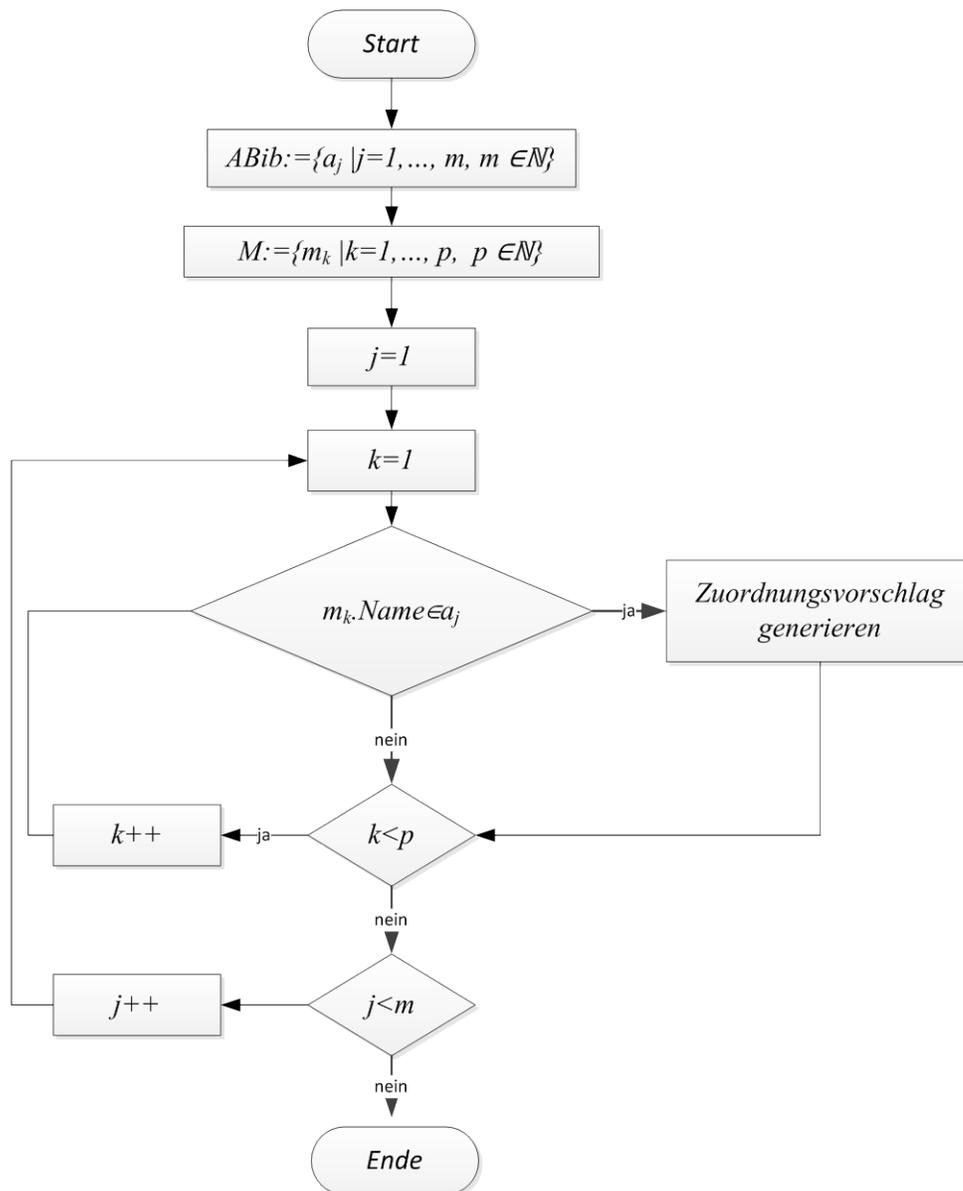
In diesem Anhang sind die Ergebnisse der Evaluierung des ZV-Algorithmus dokumentiert.

DS1	Manuell	R2F_DM			R2F_L		
		#	R	P	#	R	P
Anzahl Merkmale	98						
Dauer	4:00	00:30			00:25		
Anzahl Zuordnungen	998	695			260		
Anzahl korrekte Zuordnungen		330	0,47	0,33	175	0,67	0,18
- 100%korrekte Zuordnungen		126	0,18	0,12	133	0,51	0,13
- teilweise korrekte Zuordnungen		204	0,29	0,20	42	0,16	0,04
Anzahl falsche Zuordnungen		365	0,53		55	0,21	
Anzahl fehlende Zuordnungen		668		0,67	865		0,87

DS5	Manuell	R2F_DM			R2F_L		
		#	R	P	#	R	P
Anzahl Merkmale	16	6	0,38		6	0,38	
Dauer	00:15	00:01			00:01		
Anzahl Zuordnungen	88	31			14		
Anzahl korrekte Zuordnungen		26	0,84	0,30	13	0,93	0,15
- 100%korrekte Zuordnungen		21	0,68	0,24	6	0,43	0,07
- teilweise korrekte Zuordnungen		5	0,19	0,06	7	0,54	0,08
Anzahl falsche Zuordnungen		0	0		1	0,07	
Anzahl fehlende Zuordnungen		62		0,7	74		0,84

DS6	Manuell	R2F_DM			R2F_L		
		#	R	P	#	R	P
Anzahl Merkmale	19	8	0,42		4	0,21	
Dauer	00:15	00:01			00:01		
Anzahl Zuordnungen	94	42			15		
Anzahl korrekte Zuordnungen		25	0,60	0,27	12	0,80	0,13
- 100%korrekte Zuordnungen		19	0,45	0,20	9	0,60	0,10
- teilweise korrekte Zuordnungen		6	0,14	0,06	3	0,20	0,03
Anzahl falsche Zuordnungen		17	0,40		2	0,13	
Anzahl fehlende Zuordnungen		52		0,55	82		0,85

In diesem Anhang ist der Zuordnungsüberprüfungsalgorithmus (ZÜ-Algorithmus) vorgestellt.



Anhang IV Algorithmus zur Bestimmung der optimalen Fragenreihenfolge

Für das Vorgehen zur Bestimmung der optimalen Fragenfolge aus dem Kapitel 8 wird eine Entscheidungstabelle erstellt. Diese Entscheidungstabelle beinhaltet fünf Vorbedingungen und drei Aktionen. Da zwei Vorbedingungen nicht unabhängig sind, ist die Anzahl der möglichen Kombinationen gleich $2 \cdot 3 \cdot 2 \cdot 2 = 24$. Nach der Konsolidierung sind acht Kombinationen der Vorbedingungen relevant, diese sind in der Tabelle dargestellt.

		1	2	3	4	5	6	7	8
Vorbedingungen									
A	Es gibt genau ein Merkmal, welches an der maximalen Anzahl der „schließt_aus“-Beziehungen beteiligt ist.	j	j	n	n	n	n	n	n
B	Es gibt genau ein Merkmal mit der maximalen Anzahl der eingehenden „benötigt“-Beziehungen.	-	-	j	j	n	n	n	n
C	Es gibt mehrere Merkmale mit der maximalen Anzahl der eingehenden „benötigt“-Beziehungen	-	-	-	-	j	j	j	n
D	Es gibt ein Merkmal mit der größten Auswahlwahrscheinlichkeit	-	-	-	-	j	j	n	-
E	Das Merkmal m hat mindestens ein Voraussetzungsmerkmal.	j	n	j	n	j	n	-	-
Aktion									
1	Es wird nach dem Merkmal gefragt, welches als Einstiegspunktmerkmal für das Voraussetzungsmerkmal bestimmt wird.	x		x		x			
2	Es wird nach dem Merkmal gefragt, welches als Einstiegspunktmerkmal für das Merkmal bestimmt wird.		x		x		x		
3	Es wird nach einem beliebigen Merkmal gefragt, von welchem Vorfahren bereits ausgewählt sind.							x	x

In diesem Anhang sind die Eigenschaften von Anforderungsdokumenten und Merkmalsmodellen, welche von Studenten während der Fallstudie erstellt wurden, dokumentiert.

Studentengruppe 1:

Anzahl	Scheinwerfer				Spiegel				Außenlicht			
	PL	P1	P2	P3	PL	P1	P2	P3	PL	P1	P2	P3
Objekte	200	117	100	171	98				122			
Anforderungen	121	73	58	107	57				71			
variable Anforderungen	93	51	57	78	46				67			
Merkmale	32	-	-	-	32	-	-	-	26	-	-	-
relevante Merkmale	19	-	-	-	17	-	-	-	17	-	-	-
davon Eltern-Merkmale	9	-	-	-	9	-	-	-	7	-	-	-
davon Kinder Merkmale	23	-	-	-	23	-	-	-	19	-	-	-
Beziehungen im VM	8	-	-	-	3	-	-	-	6	-	-	-
Beziehungen zwischen VM	1	-	-	-	3	-	-	-	1	-	-	-

Studentengruppe 2:

Anzahl von	Schlüssel				Schloss				Schließung			
	PL	P1	P2	P3	PL	P1	P2	P3	PL	P1	P2	P3
Objekte	115	92	102	105	213	184	184	203	284	176	212	230
Anforderungen	74	62	66	71	166	144	145	158	189	106	132	188
variable Anforderungen	24	2	5	11	43	18	18	34	100	16	41	58
Merkmale	22	-	-	-	14	-	-	-	17	-	-	-
relevante Merkmale	18	-	-	-	12	-	-	-	15	-	-	-
davon Eltern-Merkmale	6	-	-	-	3	-	-	-	5	-	-	-
davon Kinder Merkmale	16	-	-	-	11	-	-	-	12	-	-	-
Beziehungen im VM	7	-	-	-	0	-	-	-	3	-	-	-
Beziehungen zwischen VM	1	-	-	-	1	-	-	-	1	-	-	-

Studentengruppe 3:

Anzahl von	Scheinwerferreinigung				Scheibenwischer				Wischen-Waschen			
	PL	P1	P2	P3	PL	P1	P2	P3	PL	P1	P2	P3
Objekte	195	155	161	161	238	196	194	187	189	170	171	128
Anforderungen	109	85	92	89	175	135	131	124	133	119	120	87
variable Anforderungen	49	21	27	25	84	31	33	28	62	46	47	14
Merkmale	15	-	-	-	30	-	-	-	24	-	-	-
relevante Merkmale	11	-	-	-	30	-	-	-	24	-	-	-
davon Eltern-Merkmale	6	-	-	-	10	-	-	-	9	-	-	-
davon Kinder Merkmale	5	-	-	-	20	-	-	-	15	-	-	-
Beziehungen im VM	1	-	-	-	14	-	-	-	0	-	-	-
Beziehungen zwischen VM	6	-	-	-	0	-	-	-	3	-	-	-

In diesem Anhang sind die Zeitaufwände für die einzelnen Aufgaben, welche von Studenten während der Heimarbeit erfasst wurden, dokumentiert.

Zeitaufwände für Komponenten

Aufwand für	P1	P2	P3	P4	P5	P6
Vorbereitung	4,25	2,5	3	5	0,4	10,5
Dokumentation der Komponentenanforderungen	9	9,8	2,5	3	1,88	17,5
Identifikation der Komponentenmerkmale	0	1	2	2	0,1	0
Identifikation der Beziehungen zwischen Komponentenmerkmalen	0,5	0	0,5	0,5	0,05	0
Erstellung Komponenten-Variantenmodell	3,1	3,2	1	5	1,55	4
Zuordnung der Komponentenanforderungen zu Merkmalen	0,5	0	0,75	1	0	0
Einarbeitung der Änderungen in Komponentenlastenheft	2	1	3,5	9	0	0
Erstellung eine Komponentenlastenheft	0,5	2	15,5	1	0,4	7,5

Zeitaufwände für Systeme

Aufwand für	P1+P2	P3 + P4	P5 + P6
Dokumentation der Systemanforderungen	17	3	7
Identifikation der Systemmerkmale		2	4,5
Identifikation der Beziehungen zwischen Systemmerkmalen		0,5	2
Erstellung des System-Variantenmodell	8	3	6
Zuordnung der Systemanforderungen zu Merkmalen		3,25	3
Einarbeitung der Änderungen in Systemlastenheft	1,5	3	27
Erstellung eines System-Lastenheftes	3	15,16	3,5

Literaturverzeichnis

- [ABBK02] C. Atkinson, J. Bayer, C. Bunse, E. Kamsties, O. Laitenberger, R. Laqua, D. Muthig, B. Paech, J. Wüst, J. Zettel: „Component-based product-line engineering with UML”. Addison-Wesley, Amsterdam, (2002).
- [Abbo83] R.J. Abbott: „Program design by informal English descriptions”. In: Communications of the ACM, 26(11), S. 882-894, (1983).
- [AgBe90] C.S. Aguilera, D.M. Berry: „The use of a repeated phrase finder in requirements extractions”. In: Systems and Software, 13(9), S. 209-230, (1990).
- [Agui87] C.S. Aguilera: „Finding abstractions in problem descriptions using findphrases. Master Thesis, Computer Science Department, UCLA, Los Angeles, (1987).
- [Allm07] C. Allmann: „Anforderungen auf Kundenfunktionsebene in der Automobilindustrie“. In: SE 2007, Workshop "Wie viel Requirements Engineering steckt im Software Engineering?", (2007).
- [AIOp2008] C. Allmann, N. Oppermann: „Leichtgewichtiges Anforderungsmanagement in der automotiven Vorentwicklung“. In: H. Korbinian, B. Bruegge (Hrsg.): GI-Fachtreffen Softwaretechnik, SE2008, München, 2008, S. 61-66, (2008).
- [AsSM04] T. Asikainen, T. Soininen, T. Männistö: „A Koala-Based Approach for Modelling and Deploying Software Product Families”. In: 5th Workshop on Product Family Engineering (PFE-5), Springer Verlag, LNCS 3014, May 2004, S. 225–249, (2004).
- [AtBM00] C. Atkinson, J. Bayer, D. Muthig: „Component-Based Product-Line Development: The KobrA Approach”. In: P. Donohoe (Hrsg.): Software Product Lines: Experiences and Research Directions, Kluwer Academic Publishers, S. 289-309, (2000).
- [Bato05] D. Batory: „Feature models: Grammars and propositional formulas”. In: H. Obbing, K. Pohl (Hrsg): 9th International Software Product Lines Conference, (SPLC 2005), LNCS3714, Springer Verlag Berlin Heidenheim, S. 7-20, (2005).
- [BeYY87] D.M. Berry, N.M. Yavne, M. Yavne: „Application of program design language tools to Abbott’s method of program design by informal nature language descriptions. In: Software and Systems, 7, S. 221-247, (1987).
- [BGLN03] F. Bachmann, M. Goedicke, J. Leite, R. Nord, K. Pohl, R. Ramesh, A. Vilbig: „A Meta-Model for Representing Variability in Product Family Development”. In: 5th Intl. Workshop on Product Family Engineering (PFE-5), Italy, (2003).
- [BHLK11] E. Boutkova, S.Hami-Nobari, L.Löffler, B.Kolb: „Merkmalsmodellierung für Anforderungsartefakte mit Doors”. In: Softwaretechnik-Trends 31(1), (2011).
- [BHPW04] S. Bühne, G. Halmans1, K. Pohl1, M. Weber, H. Kleinwechter, T. Wierczoch: „Defining Requirements at Different Levels of Abstraction”. In: Proceedings of the 12th IEEE International Requirements Engineering Conference (RE’04), (2004).
- [BigL10] bigliver <http://www.biglever.com/>
- [Birk80] H. Birkhofer: „Analyse und Synthese der Funktionen technischer Produkte“. Dissertation, TU Braunschweig (1980).
- [BKPS04] G. Böckle, P. Knauber, K. Pohl, K. Schmidt: „Software-Produktlinien: Methoden, Einführung und Praxis“. Dpunkt Verlag, (2004).

- [BLPW04] S. Bühne, K. Lauenroth, K. Pohl, M. Weber: „Modelling features for multi-criteria product-lines in automotive industry”. In: Proceedings of the workshop on software engineering for automotive systems (SEAS), ICSE 2004, (2004).
- [BeMC05] D. Benavides, P.T. Martin-Arroyo, A. Ruiz-Cortes: „Automated reasoning on feature models“. In: Proceeding of the 17th International Conference of Advanced Information Systems Engineering, (CAiSe2005), S. 491-503, (2005).
- [Bout09] E. Boutkova: „Variantenmanagement in Anforderungsdokumenten: State of the Practice”. In: Softwaretechnik-Trends, 29 (1), (2009).
- [Bout10a] E. Boutkova: „Herausforderungen für Variantenmanagement in Anforderungsdokumenten“. In: Proceeding of Software Engineering Conference (PIK-Workshop), Paderborn, (2010).
- [Bout10b] E. Boutkova: „Variantenmanagement in Anforderungsdokumenten“. In: Proceeding of SE 2010 (Doktorandensymposium), Paderborn, (2010).
- [Bout11a] E. Boutkova: „Pragmatic Variability Management Approach in Requirement Specifications with DOORS“. In: Post-Proceeding REFSQ 2011 (REEW 2011), 2011.
- [Bout11b] E. Boutkova: „Experience with Variability Management in Requirement Specifications”. In: Proceeding of the SPLC 2011, München, (2011).
- [BoHo11] E. Boutkova, F. Houdek: „Semi-Automatic Identification of Features in Requirement Specifications”. In: Proceeding of RE2011, Trento, Italien, (2011)
- [Bran00] T. Brants: „TnT – a statistical part of-speech tagger”. In: Proceeding of 6th Conference. on Applied Natural Language Processing, S. 224–231, (2000).
- [BrRa00] P. Braun, M. Rappl: „Abstraction levels of embedded systems”. <http://www2.cs.unibw.de/Proj/GROOM/OMER-2/papers/OMER2-BraunRappl.pdf> abgerufen am 10.05.2011
- [BüLP04] S. Bühne, K. Lauenroth, K. Pohl: „Anforderungsmanagement in der Automobilindustrie: Variabilität in Zielen, Szenarien und Anforderungen“. GI Jahrestagung 2004, S. 23-27, (2004).
- [ChAt07] B.H.C. Cheng, J.M. Atlee: „Research Directions in Requirements Engineering.” In: Future of Software Engineering (FOSE’2007), IEEE, (2007).
- [ChBa10] L. Chen, M.A. Babar: „Variability management in Software Product Lines: An Investigation of Contemporary Industrial Challenges”. In Bosch, J., Lee, J. (Hrsg.) SPLC 2010, LNCS 6287, S.166-180, (2010).
- [Chen83] P.Chen: English sentence structure and entity relationship diagram. In: Information Sciences, 1(1), S. 127-149, (1983)
- [ChEr11] S. Chen, M. Erwig : „Optimizing the Product Derivation Prozess“. In: Proceeding of Software Product Line Conferense, SPLC 2011, München, S. 35-44, (2011).
- [ChZM05] K. Chen, W. Zhang, H. Zhao, H. Mei: „An Approach to Constructing Feature Models Based on Requirements Clustering”. In: Proceeding of the 13th IEEE International Conference on Requirements Engineering, (2005).
- [Clau01] M. Clauß: „A Proposal for Uniform Abstract Modelling of Feature Interactions in UML”. In: E. Pulvermüller, A. Speck, J. Coplien, M. D’Hondt, W. De Meuter (Hrsg.): Proceedings of the ECOOP 2001 Workshop on Feature Interaction in Composed Systems (FICS 2001), Budapest, Hungary, Tech-

- nical Report 2001-14 University of Karlsruhe, Institut für Programmstrukturen und Datenorganisation, S. 21-25, (2001).
- [CINo01] P. Clements, L. Northrop: „Software Product Lines: Practices and Patterns”. AddisonWesley, (2001).
- [CoJB00] M. Coriat, J. Jourdan, F. Boisbordin: „The SPLIT Method: Building Product Lines for Software-Intensive Systems”. In: P. Donohoe (Hrsg.): Software Product Lines: Experiences and Research Directions, S. 289-309, Norwell, Kluwer Academic Publishers, (2000).
- [Cole94] D. Coleman et al, Object Oriented Development — The FUSION Method, Prentice-Hall, 1994
- [CzEi00] K. Czarnecki, U. Eisenecker: „Generative Programming“. Addison-Wesley, (2000).
- [CzHE05] K.C. Czarnecki, S. Helsen, E. Eisenecker: „Staged Configuration through Specialization and Multi-Level Configuration of Feature Models”. In: Software Process: Improvement Practice, 10(2), S. 143-169, (2005).
- [CzHE05] K. Czarnecki, S. Helsen, U. Eisenecker: „Formalizing cardinality-based feature models and their specialization”. In: Software Process Improvement and Practice 10 (1), S.7–29, (2005).
- [DhGR11] D.Dhungana, P. Grünbacher, R. Rabiser „The Doppler meta-tool for decision-orineted variability modeling: a multiple case study. In: Automated Software Engineering, 18, S. 77-114, (2011).“
- [DLPW08] C. Dziobek, J. Loew, W. Przystas, J. Weiland: „Von Vielfalt und Variabilität.“ In: Elektronik Automotive (2), www.elektroniknet.de, (2008).
- [DIN69905] Deutsche Institute für Normung e.V, „ DIN 66 905 – Projektwirtschaft - Projektabwicklung – Begriffe. Normungsausschuss Informationsverarbeitungssysteme (NI)“. Beuth-Verlag, Berlin, (1997).
- [ErBB05] M. Eriksson, J. Börstler, K. Borg: „The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations. In: H. Obbing, K. Pohl (Hrsg.): 9th International Software Product Lines Conference, SPLC 2005, LNCS3714, Springer Verlag Berlin Heidenheim, S. 33-44, (2005).
- [Fran02] H.J. Franke: „Variantenmanagement in der Einzel- und Kleinserienfertigung“. Hanser Verlag, (2002).
- [FeFB02] D. Fey, R Fajta, A. Boros: „Feature modelling: A meta-model to enhance usability and usefulness”. In SPLC2, LNCS 2379, S. 198–216, (2002).
- [FHPR05] A. Fleischmann, J. Hartmann, C. Pfaller, M. Rappl, S. Rittmann, D. Wild: „Concretization and Formalization of Requirements for Automotive Embedded Software Systems Development. In: K. Cox, J.L. Cybulski et.al (Hrsg.): The Tenth Australian Workshop on Requirements Engineering (AWRE). Deakin University, Melbourne, Australia, S. 60-65, (2005).
- [FrBY92] W.B. Frakes, R. Baeza-Yates: „Information Retrieval Data Structures & Algorithms”. Prentice Hall, (1992).
- [GaSC10] R. Gacitua, P. Sayer, V. Gervasi: „On the Effectiveness of abstraction identification in requirements engineering. In: Proceeding of the 18th IEEE International Requirements Engineering Conference, (2010).
- [GiMa04] J. Giménez, L. Màrquez: „SVMTool: „A general POS tagger generator based on support vector machines”. In: Proceedings of the 4th International Conference on Language Resources and Evaluation, (2004).

- [Glin06] M. Glinz: „Vorlesungsskript Spezifikation und Entwurf von Software“. Universität Zürich, (2006).
- [GrEv09] E. Giesbrecht, S. Evert: „Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS-Taggers for the German Web as Corpus“. Web as Corpus Workshop (WAC5), (2009).
- [GrFA98] M. Griss, J. Favaro, M. d’Alessandro: „Integrating Feature Modelling with the RSEB“. In: Proceedings of the Fifth International Conference on Software Reuse, S.76–85, (1998).
- [Grim03] K. Grimm: „Software Technology in an Automotive Company – Major Challenges“. In: Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society, (2003).
- [GoBe94] L. Golding, D.M. Berry: „AbstFinder: A prototype abstraction finder for natural language text for use in requirements elicitation: design, methodology and evaluation“. In: Proceeding of the 1st international conference of requirements engineering, Colorado Springs, IEEE Computer Society, S. 84-93, (1994).
- [GoWe04] H. Gomaa, D. Webber: „Modelling Adaptive and evolvable software product lines using the variation point model“. In: Proceedings of the Hawaii International Conference on System Sciences, Hawaii, (2004).
- [GoWo06] T. Gorschek, C. Wohlin: „Requirements Abstraction Model“. In: Requirements Engineering 11 (2006), S. 79–101, (2006).
- [HaDS05] J.H. Hayes, A. Dekhtyar, S.K. Sundaram: „Improving After-the-Fact Tracing and Mapping: Supporting Software Quality Predictions“, IEEE Software, S. 30-37, (2005).
- [HaPo03] G. Halmans, K. Pohl: „Communicating the variability of a software product family to customers“. In: Software and Systems Modelling, 2 (1), S. 15–36, (2003).
- [HaRH01] J. Hammond, R. Rawlings, A. Hall: „Will It Work?“ In: Proceeding of the 5th IEEE International Symposium on Requirements Engineering (RE 2001), Toronto, Canada, (2001).
- [HeHo03] N. Heumesser, F. Houdek: „Towards Systematic Recycling of Systems Requirements“. In: Proceeding of the ICSE 2003, S. 512-519, (2003).
- [HeHo04] N. Heumesser, F.Houdek: „Experiences in Managing an Automotive Requirements Engineering Process“. In Proceeding of the RE 2004, S. 322-327, (2004).
- [HKWN06] L. Hotz, T. Krebs, K. Wolter, J. Nijhuis, S. Deelstra, M. Sinnema, J. MacGregor: „Configuration in Industrial Product Families –The ConIPF Methodology“, IOS Press, (2006).
- [HMRV07] C. Hood, S. Mühlbauer, C. Rupp, G. Versteegen: „iX-Studie Anforderungsmanagement“. 2. Auflage, Hannover, (2007).
- [HSVM00] A. Hein, M. Schlick, R. Vinga-Martins: „Applying feature models in industrial settings. In P. Donohoe (Hrsg.), Software Product Lines: Experience and Research Directions, Kluwer Academic Publishers, S. 47–70, (2000).
- [Houd03] F. Houdek: „Requirements Engineering Erfahrungen in Projekten der Automobilindustrie“. In: Softwaretechnik-Trends 23(1), 2003.
- [Houd10] F. Houdek: „Challenges in Automotive Requirements Engineering“. Industrial Presentations by REFSQ 2010, Essen, (2010).

- [HuLi04] M. Hu, B. Liu: „Mining and Summarizing Customer Reviews”. In: Proceeding of KDD, Seattle, USA, (2004).
- [IEEE610] Institute of Electric and Electronic Engineers: IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). IEEE Computer Society, New York, (1990).
- [IEEE830] Institute of Electric and Electronic Engineers: IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998). IEEE Computer Society, New York, (1998).
- [IEEE 1233] Institute of Electric and Electronic Engineers: IEEE Guide for Development System Requirements Specifications (IEEE Std 1233-1998). IEEE Computer Society, New York, (1998).
- [IEEE1362] Institute of Electric and Electronic Engineers: Guide for Information Technology System Definition (IEEE Std 1369-1998). IEEE Computer Society, New York, (1998).
- [IIEO01] I. Iliopoulos, A. J. Enright, C. A. Ouzounis: „Textquest: Document clustering of Medline abstracts for concept discovery in molecular biology”. Pac Symp Biocomput, (2001).
- [Inou04] R.B. Inouye: “Minimizing the length of non-mixed initiative dialogs”. In: ACL workshop on student research, Barcelona, Spain, S.39-43, (2004).
- [JoMu02] I. John, D. Muthig: „Tailoring Use Cases for Product Line Modelling.” In: Proceedings of the International Workshop on Requirements Engineering for Product Lines 2002 (REPL'02), (2002).
- [JaGJ97] I. Jacobson, M. Griss, P. Jonsson: „Software Reuse: Architecture, Process, and Organization for Business Success”. Addison-Wesley, ACM Press Books, Harlow, (1997).
- [KKLK98] K.C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, M. Huh: „Form: A feature-oriented reuse method with domain-specific reference architectures”. In: Annals of Software Engineering, 5, S. 143–168, (1998).
- [KLMM97] G.Kiczales, J. Lamping, A.Mendhekar, C. Maeda, C. Videira, J. Irwin, J.-M. Longetier: „Aspect-oriented Programming”, (1997).
- [KMPS08] M. Kirchmayr, M. Müller, B. Penzenstadler, E. Sikora, T. Weyer: „Essenzieller REMsES-Leitfaden“. Projekt REMsES, Förderkennzeichen: 01 IS F06 D, REMsES-Konsortium, (2008).
- [KPKH02] A. von Knethen, B. Paech, F. Kiedaisch, F. Houdek: „Systematic Requirements Recycling through Abstraction and Traceability“. In: Proceeding of RE 2002, S. 273-281, (2002).
- [KCHN90] K.C. Kang, G. Cohen, S.J.A Hess, W.E. Novak, A. Spencer: „Feature-Oriented Domain Analysis (FODA) Feasibility Study”. Peterson, (1990).
- [Leiz96] W. Lezius: „Morphologie System Morphy“. In: R. Hausser (Hrsg.) Linguistische Verifikation, S. 25-35. (1996)
- [Leco00] R. Lecoecue: „Finding comparatively important concepts between texts”. In: Proceeding of the 15th IEEE International conference on Automated Software engineering, Grenoble, France, S. 55-60, (2000).
- [LeKa04] K. Lee, K.C. Kang: „Feature Dependency Analysis for Product Line Component Design”. In: J. Bosch, C. Krueger (Hrsg.): ICSR 2004, LNCS 3107, S. 69-85, (2004).
- [LeKL02] K. Lee, C. Kang, J. Lee: „Concepts and guidelines of feature modelling for product line software engineering”. In LNCS 2319, S. 62–77, (2002).

- [LiYT10] Y.Lin, H. Ye, J. Tang: „An approach to efficient product configuration on software product line“. In: Proceeding of the 14th Software Product Line Conference (SPLC22010), Jesu Island, South Korea, S. 435-438, (2010).
- [LKKC00] K. Lee, K.C. Kang, E. Koh, W. Chae, B. Kim, B. Wook Choi: „Domain-oriented engineering of elevator control software“. In P. Donohoe (Hrsg.), Software Product Lines. Experience and Research Directions, Kluwer Academic Publishers, S. 3–22, (2000).
- [LMNW03] H. Lichter, T. von der Maßen, A. Nyßen, T. Weiler: „Vergleich von Ansätzen zu Feature Modellierung bei Software-Produktlinienentwicklung“, (2003).
- [MaBe88] Y. Maarek, D.M. Berry: „The use of lexical affinities in requirements extraction“. Technical report. Faculty of computer Science, Technion, Haifa, Israel, (1988).
- [MaLi02] T. von der Maßen, H. Lichter: „Modeling Variability by UML Use Case Diagrams, Modeling Variability by UML Use Case Diagrams“. In: Proceedings of the International Workshop on Requirements Engineering for Product Lines 2002, Technical Report ALR-2002-033, (2002).
- [MaLi04] T. von der Maßen, H. Lichter: „RequiLine: A Requirements Engineering Tool for Software Product Lines. In: Proceeding of the 5th Workshop on Product Family Engineering (PFE-5), Springer Verlag Lecture Notes on Computer Science, vol. 3014 (LNCS 3014), May 2004, S.168–180, (2004).
- [MaLi04a] T. von der Maßen, H. Lichter: „Deficiencies in Feature Models“. In Proceedings of SPLC’04 Workshop on Software Variability Management for Product Derivation - Towards Tool Support, (2004).
- [MaLi05] T. von der Maßen, H. Lichter: „Determining the variation degree of feature models“. In: LNCS Software Product Lines, S. 82-88, (2005).
- [Mann02] M.Mannion Using first-order logic for product line model validation“. In: Proceeding of the 2nd Software Product Line Conference (SPLC2002), S. 176-187, (2002).
- [MaSM93] M. Marcus, B. Santorini, M. Marcinkiewicz: „Building a large annotated corpus of English: The Penn Treebank“. In: Computational Linguistics, 19(2), S. 313-330, (1993).
- [MeHP07] A. Metzger, P. Heymans, K. Pohl, P.Y. Schobbens, G. Saval: „Disambiguating the Documentation of Variability in Software Product Lines: A Separation of Concerns, Formalization and Automated Analysis“. In: Proceeding of the 15th IEEE International Requirements Engineering Conference (RE 2007), New Delhi, India, S. 243-253, (2007).
- [MiFN04] L. Mich, M. Franch, I. Novi: „Market research for requirements analysis using linguistic tools“. In: Requirements Engineering, 9 (2), S. 40–56, (2004).
- [Monz08] A. Monzon: „A practical approach to requirements reuse in product families of on-board systems“. In: 16th International Requirements Engineering Conference (RE08), Barcelona, (2008).
- [MZMB08] C. Müller, T. Zesch, M.C. Müller, D. Bernhard, K. Ignatova, I. Gurevych, M. Mühlhuser: „Flexible UIMA components for information retrieval research“. In Proceedings of the LREC 2008 Workshop Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP, S. 24–27, (2008).

Literaturverzeichnis

- [NeSA03] G. Nenadic, I. Spasic, S. Ananiadou: „Terminology-driven mining of bio-medical literature”. *Bioinformatics*, 19 (8), S. 938-943, (2003).
- [NeLa03] C. J. Neill, P. A. Laplante: „Requirements engineering: The state of the practice”. In: *IEEE Software*, 20 (2), S.40–45, (2003).
- [NöEg11] A.Nöhner, A. Egyed: „Optimizing User Guidance during Decision-Making. In: *Proceeding of Software Product Line Conference, SPLC 2011, München*, S. 25-34, (2011).
- [NRWF08] E. Nyberg, E. Riebling, R.C.Wang, R. Frederking: „Integrating a natural language message pre-processor with uima”. In *Proceedings of the LREC 2008 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP'*, S. 28-31, (2008).
- [PaLi92] J.D. Palmer, Y. Liang: „Indexing and Clustering of Software Requirements Specifications”. In: *Information and Decision Technologies*, 18 (4), S. 283-299, (1992).
- [Part10] H. Partsch: „Requirements-Engineering systematisch - Modellbildung für softwaregestützte Systeme“. 2 Auflage, Springer, (2010).
- [Rees12] J. Rees: Volkswagen perfektioniert das Lego-System. In: *Wirtschaftswoche*, 28.02.2012 <http://www.wiwo.de/technologie/auto/autobau-das-erste-baukasten-fahrzeug/6260246-2.html>
- [PoBL05] K. Pohl, G. Böckle, F. van der Linden: „Software Product Line Engineering: Foundations, Principles, and Techniques”. Springer Verlag, (2005).
- [PoEt05] A.M. Popescu, O. Etzioni: „Extracting Product Features and Opinions from Review”. In: *Proceeding of Human Language Technology and Empirical Methods in Natural Language, Vancouver*, S. 339-346, (2005).
- [PoMe08] K. Pohl, A. Metzger: „Variabilitätsmanagement in Software-Produktlinien“. In: K. Herrmann, B. Brüggel (Hrsg.) *Software Engineering 2008, Fachtagung des GI-Fachbereichs Softwaretechnik, München, LNI 121*, S. 28-41,(2008).
- [PoSi07] K. Pohl, E. Sikora: „COSMOD-RE: Supporting the Co-Design of Requirements and Architectural Artefacts”. In: *Proceeding of the 15th IEEE International Requirements Engineering Conference (RE 2007), New Delhi, India*, S. 258-261, (2007).
- [Pohl07] K. Pohl: „Requirements Engineering”, dpunktverlag, Heidelberg (2007).
- [Puls02] C. Puls: „Die Konfigurations- & Verträglichkeitsmatrix als Beitrag zum Management von Konfigurationswissen in KMU. Dissertation (2002).
- [RaGD10] R. Rabiser, P. Grünbacher, D. Dhungana: „Requirements for product derivation support: Results from a systematic literature review and an expert survey”. In: *Information and Software Technology* 52, S. 324-346, (2010).
- [Reis08] M.O. Reiser: „Managing Complex Variability in Automotive Software Product Lines with Subscoping and Configuration Links”. PhD, TU Berlin, (2008).
- [ReSW08] B. Regnell, R.B. Svensson, K. Wnuk: „Can we Beat the Complexity of very Large-Scale Requirements Engineering? “. In: B. Paech, C. Rolland (Hrsg.) *Requirements Engineering: Foundation for Software Quality. LNCS 5025*, S. 123 -128, (2008).
- [ReWe05] M.O. Reiser, M. Weber: „Using Product Sets to Define Complex Product Decisions”. In: H. Obbink and K. Pohl (Hrsg.): *SPLC 2005, LNCS 3714, Springer-Verlag*, S. 21–32, (2005).

- [RFHP05] S. Rittmann, A. Fleischmann, J. Hartmann, C. Pfaller, M. Rappl, D. Wild, „Integrating Service Specifications on Different Levels of Abstraction”. In: Proceedings of the IEEE International Workshop on Service-Oriented System Engineering (SOSE), S. 63-70, (2005)
- [RiSP04] M. Riebisch, D. Streitferdt, I. Pashov: „Modeling Variability for Object-Oriented Product Lines”. In: Workshop Reader of Object-Oriented Technology at ECOOP 2003, Springer, LNCS 3013, S.165–178, (2004).
- [RuRL2003] C.Rupp, F. Rachinger, A. Lechner: „Benutzerhandbuch als Anforderungsspezifikation- der Versuch einer konstruktiven Provokation“. GI Fachgruppentreffen Softwaretechnik, (2003).
- [Schm10] K. Schmid: „Variability Modelling for Distributed Development : A Comparison with Established Practice”. In: J. Bosch and J. Lee (Hrsg.): SPLC 2010, LNCS 6287, S. 151–165, (2010).
- [SDNB04] M. Sinnema, S. Deelstra, J. Nijhuis, J. Bosch: „COVAMOF: A Framework for Modeling Variability in Software Product Families”. In: Proceedings of the Third Software Product Line Conference (SPLC2004), Springer Verlag, LNCS 3154, S.197–213, (2004).
- [SiDe06] M. Sinnema, S. Deelstra: „Classifying variability modelling techniques”. In: Information and Software Technology 49, S.717–739, (2006).
- [ScZu03] J. Schäuuffele, T. Zurawka: „Automotive Software Engineering“. Vieweg Verlag, (2003).
- [SHTB07] P.Y Schobbens, P. Heymans, J.C. Trigaux, Y. Bontemps: „Generic semantics of feature diagrams”. In: Computer Networks, 51, S. 456-479, (2007).
- [Schm95] H. Schmid: „Improvements in part-of speech tagging with an application to German”. In: Proceeding of the ACL SIGDAT Workshop, (1995).
- [Smea95] A.F. Smeaton: „Low level language processing for large scale informational retrieval: what techniques actually work. In: Proceeding of a Workshop: Terminology, Informational Retrieval und Linguistic, (1995).
- [Souza99] D. D’Souza, A. Wills: „ Objects, Components and Frameworks with UML — The Catalysis Approach”. Addison-Wesley, 1999
- [Stan09] Standish Group International, Inc.: Chaos Report, (2009).
- [Stre03] D. Streitferdt: „Family-Oriented Requirements Engineering”. Dissertation, TU Ilmenau, (2003).
- [StRP03] D. Streiferdt, M. Riebisch, I. Philippow: „Formal Details of Relations in Feature Models”.; In: Proceedings 10th IEEE Symposium and Workshops on Engineering of Computer-Based Systems, Huntsville Alabama, USA, (2003).
- [SvBo00] M. Svahnberg, J. Bosch: „Issues Concerning Variability in Software Product Lines”. In: F. van der Linden (Hrsg.): Software Architectures for Product Families, International Workshop IW-SAPF-3, Las Palmas de Gran Canaria, Spain, LNCS 1951, Springer 2000, S. 146-157, (2000).
- [ScZu02] Schulte-Zurhausen: *Organisation*. 3. Auflage. 2002, S. 562
- [Tava06] R. Tavakoli Kalagari: „Requirements Engineering für Software-Produktlinien eingebetteter, technischer Systeme“. Dissertation Universität Ulm, (2006).
- [TaRe07] R. Tavakoli Kolagari, M.O. Reiser: „Reusing Requirements: The Need for Extended Variability Models”. In: F. Arbab, M. Sirjani (Hrsg.) FSEN 2007, LNCS 4767, Springer-Verlag Berlin Heidelberg, S. 129–143, (2007).

Literaturverzeichnis

- [ThHe02] S. Thiel, A. Hein: „Modelling and Using Product Line Variability in Automotive Systems”. IEEE Software, 19, S. 66-72, (2002).
- [ToKM03] K. Toutanova, D. Klein, C.D. Manning: „Feature-rich part-of-speech tagging with a cyclic dependency network”. In: Proceeding of the 2003 Human Language Technology, (2003).
- [vGBS01] J. van Gorp, J. Bosch, M. Svahnberg: „On the notion of variability in software product lines”. In: Proceedings of the Working IEEE/IFIP Conference on Software Architectur (WICSA '01), (2001).
- [VDI/VDE3694] Verein Deutsch Ingenieure e.V. , Verband der Elektrotechnik Elektronik Informatiostechneik e.V.: Lastenheft/Pflichtenheft für den Einsatz von Automatisierungssystemen (VDI/VDE-3694), (1991).
- [WeRe09] M. Weber, M. O. Reiser: „Product Lines in Automotive Electronics”. In: N. Navet: Autmotive embedded systems handbook, CRC Press, S. (7)1-30, (2009).
- [WeWe03] M. Weber, J. Weisbrod: „Requirements Engineering in Automotive Development: Experience and Challenges”. In: IEEE Software, S. 16-24, (2003).
- [Wigg97] T.A. Wiggerts: „Using clustering algorithms in legacy systems remodularization”. In Proceedings of the 4th Working Conference on RE, S.33-43, (1997).
- [WKHH95] J. Wilkenfeld, S. Kraus, K.M. Holley, M.A. Harries: “GENIE: A decision support system for crisis negotiations“. In: Decision Support System, 14, S. 369-391, (1995).
- [WoBr09] T. Woeniger, R. Breitbarth: „SW-Lastenhefte für Motorsteuerungen. Methode und Erfahrungen“. Vortrag bei RECONF 2009, München, (2009).
- [ZeLe06] S. Zelewski: „Strategisches Produktionsmanagement“. Vorlesungsskript Universität Duisburg-Essen, (2006).
- [Zwic89] F. Zwicky: „Entdecken, Erfinden, Forschen im morphologischen Weltbild. Glarus: Fritz- Zwicky Stiftung (1989).
-
- [gears] BigLever Gears <http://www.biglever.com/>
- [door] DOORS <http://www-142.ibm.com/software/products/de/de/ratidoor/>
- [meta] Me-
tadoc <http://www.metadoc.de/anforderungsmanagement/werkzeuge/fm/73-feature-modeller>
- [pure] Pure::Variants http://www.pure-systems.com/pure_variants.49.0.html