



TECHNISCHE
UNIVERSITÄT
WIEN
TUUB
Vienna Austria
WIEN Universitätsbibliothek

Die approbierte Originalversion dieser Diplom-
Masterarbeit ist in der Hauptbibliothek der Tech-
nischen Universität Wien zugänglich und zugänglich.
<http://www.ub.tuwien.ac.at>

The approved original version of this diploma or
master thesis is available at the main library of the
Vienna University of Technology.

<http://www.ub.tuwien.ac.at/eng>

Diplomarbeit

Dokumentationsbasiertes Komplexitäts- und Variantenmanagement

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Diplom-Ingenieurs (Dipl.-Ing.)
eingereicht an der TU Wien, Fakultät für Maschinenwesen und Betriebswissenschaften
von

Günter SCHÖBITZ

Mat.Nr.: 00925910

Gainfarnerstrasse 1, 2542 Kottingbrunn, Österreich

unter der Leitung von

Univ.-Prof. Dipl.-Ing. Dr.-Ing. Detlef Gerhard

Institut für Konstruktionswissenschaften und technische Logistik, E307

begutachtet von

Univ.-Prof. Dipl.-Ing. Dr.-Ing. Detlef Gerhard

Institut für Konstruktionswissenschaften und
technische Logistik, E307

Getreidemarkt 9/307

1060 Wien

a.O. Univ.-Prof. Dipl.-Ing. Dr. techn. Manfred
Grafinger

Institut für Konstruktionswissenschaften und
technische Logistik, E307

Getreidemarkt 9/307

1060 Wien

Ich nehme zur Kenntnis, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass die vorliegende Arbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen von mir selbstständig erstellt wurde.

Alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, sind in dieser Arbeit genannt und aufgelistet. Die aus den Quellen wörtlich entnommenen Stellen, sind als solche kenntlich gemacht.

Das Thema dieser Arbeit wurde von mir bisher weder im In- noch Ausland einer Beurteilerin/ einem Beurteiler zur Begutachtung in irgendeiner Form als Prüfungsarbeit vorgelegt. Diese Arbeit stimmt mit der von den Begutachterinnen/Begutachtern beurteilten Arbeit überein

Wien, Mai, 2018

Unterschrift



Forschungsbereich
Maschinenbau Informatik
und Virtuelle
Produktentwicklung
Univ.-Prof. Dr.-Ing. Detlef Gerhard

Dokumentationsbasiertes Komplexitäts- und Variantenmanagement

Diplomarbeit

Günter Schöbitz

Matrikel.-Nr.: 00925910

Abkürzungsverzeichnis

BBA	Baubarkeitsanalyse
BBP	Baubarkeitsprüfung
BG	Baugruppe
BM	Baumuster
bzw.	Beziehungsweise
CD	Code
CG	Codegruppe
Def.	Definition
DIN	Deutsches Institut für Normung
DMM	Domain Mapping Matrice
DMS	Dokumentenmanagementsystem
DSM	Design-Structure-Matrice
EEA	Eigenschaft-Einfluss-Analyse
EI	Exklusivitätsindex
ERP	Enterprise-Ressource-Planning
ET	Einzelteil
Etc.	Et cetera
FIN	Fahrzeug-Identifikations-Nummer
ISL	Inter-Stücklisten-Commonality
ISO	International Organization for Standardization
IVC	Inter-Vehicle-Commonality
LKW	Lastkraftwagen
LL	Linkslenker
o.ä	oder ähnliches
PDM	Produkt-Daten-Management
PLM	Product-Lifecycle-Management
PKW	Personenkraftwagen
RL	Rechtslenker
SA	Sonderausstattung
SNR	Sachnummer

VDI	Verein Deutscher Ingenieure
VDS	Vehicle Description Sector
vgl	vergleiche
VIN	Vehicle-Identification-Number
WMI	World Manufacturer Identification
z.B.	zum Beispiel

Kurzfassung

Aufgrund der aktuellen Marktsituation im Personen- und Nutzfahrzeugbereich versuchen die Hersteller, durch technologisch fortschrittliche und sehr auf den Kundenwunsch individualisierbare Produkte größere Marktanteile zu gewinnen.

Durch diesen Trend ist die externe und interne Variantenvielfalt und mit ihr die zu bewältigende Komplexität stark gestiegen.

In dieser Arbeit werden die theoretischen Grundlagen für das Komplexitäts- und Variantenmanagement auf Basis der Dokumentationsdaten, die aus Produkt-Daten-Management- und Enterprise-Ressource-Planning-Systemen gewonnen werden können, erläutert.

Anhand dieser Grundlagen wird gezeigt, wie diese Analysen tatsächlich in der Praxis angewendet, wie die Ergebnisse interpretiert werden können, bzw. worauf geachtet werden muss. Weiters werden Maßnahmen erarbeitet, die bei der Komplexitätsbeherrschung und -reduzierung unterstützend wirken.

Dabei wurden auch Indizes entwickelt, die eine Bewertung des Ist-Zustands der Produktkomplexität ermöglichen.

Inhaltsverzeichnis

Abkürzungsverzeichnis	I
Kurzfassung	III
Inhaltsverzeichnis.....	IV
1 Einleitung	6
1.1 Motivation	6
1.2 Zielsetzung	7
1.3 Aufbau der Arbeit	7
2 Grundlagen	8
2.1 Definitionen und Begriffsabgrenzungen	8
2.1.1 Produkt	8
2.1.2 Eigenschaft, Merkmal, Ausprägung.....	8
2.1.3 Derivat, Variante, Variantentreiber	9
2.1.4 Produktprogramm, Produktportfolio.....	10
2.1.5 Baumuster, Baureihe	10
2.1.6 Variantenvielfalt, Varianz, Komplexität.....	11
2.1.7 Externe Variantenvielfalt, Varianz und Komplexität vs. interne Variantenvielfalt, Varianz und Komplexität.....	12
2.1.8 Komplexitäts- und Variantenmanagement	13
2.2 Dokumentation	14
2.2.1 Funktionsstruktur, Produktstruktur, Produktarchitektur	14
2.2.2 Stücklisten	17
2.2.3 Code, Codegruppe, Codekette, Codebedingung	21
2.2.4 Fahrzeug-Identifikations-Nummer	22
2.2.5 Produktionsprogramm, Bauprogramm	22
2.2.6 Dokumentationssysteme	23
2.3 Mathematische Grundlagen.....	26
2.3.1 Zählen.....	26
2.3.2 Boolesche Algebra und Aussagenlogik.....	26
3 Verwendete Methoden	30
3.1 Lösen der booleschen Ausdrücke.....	30
3.2 Einflussanalyse der booleschen Ausdrücke.....	31
3.3 Produktauflösung	33
3.4 Domain Mapping Matrices	34
3.5 Design-Structure-Matrices	35

3.6 Clusteranalyse.....	38
3.7 ABC-Analyse.....	38
4 Eingangsrößen.....	40
4.1 Diskussion des Bedarfs.....	40
4.1.1 Produktionsprogramm.....	40
4.1.2 Codeinformationen.....	42
4.1.3 Produktarchitektur.....	43
4.2 Pre-Processing.....	47
4.2.1 Ermitteln von Informationen aus den Datensätzen.....	47
4.2.2 Berechnung häufig verwendeter Methoden.....	48
5 Tools, Analyse und Visualisierung.....	49
5.1 Exklusivitätsanalyse.....	49
5.2 Merkmalbaum.....	51
5.3 DSM Anwendungen.....	53
5.3.1 Paketbildung.....	53
5.3.2 Inter-Vehicle-Commonality.....	59
5.3.3 Inter-Stücklisten-Commonality.....	64
5.4 Exklusivitätsindex.....	65
5.5 Eigenschaft-Einfluss-Analyse.....	69
5.6 Layouts.....	74
5.7 Abschichtung.....	78
5.8 Baubarkeitsprüfung.....	78
5.9 Baubarkeitsanalyse.....	79
6 Zusammenfassung und Ausblick.....	92
6.1 Zusammenfassung.....	92
6.2 Ausblick.....	93
6.2.1 Aktiv/Passiv Unterteilung.....	93
6.2.2 Geführte Konfiguration.....	94
6.2.3 Indizes für die Bewertung.....	95
6.2.4 Künstliche Intelligenz.....	95
7 Verzeichnisse.....	96
7.1 Literaturverzeichnis.....	96
7.2 Abbildungsverzeichnis.....	98

1 Einleitung

"Any customer can have a car painted any color that he wants so long as it is black." Henry Ford

Dieses Zitat von Henry Ford stammt ohne Zweifel aus einer Zeit, in der es sich bei der Automobilindustrie um einen Verkäufermarkt handelte. Damals war die Nachfrage stärker als das Angebot und um den eigenen Umsatz zu steigern musste man nur die Produktion erhöhen. Im Laufe der Jahre entwickelte sich der Markt hin zu einem Käufermarkt, es gibt also mehr Angebot als Nachfrage, und alle Automobilhersteller, sowohl im Segment der Personenfahrzeuge als auch der Nutzfahrzeuge, bemühen sich um höhere Marktanteile, indem sie den Kunden Kaufargumente liefern. [1]

Eines dieser Argumente ist die Erfüllung der Kundenwünsche und je individueller diese sind, desto größer ist die benötigte Variantenzahl.

1.1 Motivation

Die, von den Produktprogrammen und Produktstrukturen abgebildete, externe bzw. interne Variantenvielfalt kann größer sein, als es der Markt tatsächlich fordert. So zeigen, gerade in der Automobilindustrie, die Verbrauchsdaten von variantenreichen Bauteilen eine typische Pareto-Verteilung, das heißt bei 5% bis 15% aller verbauten Teile beträgt die Rate mehr als 80%, bei den restlichen 85% bis 95% teilweise sogar deutlich unter 1%. [2]

Ein ähnliches Bild zeigt sich bei den nicht notwendigen Produktvarianten, so kommen verschiedene Untersuchungen zu dem Ergebnis, dass rund 25% aller Produktvarianten und mehr als 35% aller Komponentenvarianten nie gebaut bzw. verbaut werden. [3], [4]

Die Diskrepanz zwischen geforderter und angebotener Variantenvielfalt führt unter anderem zu folgenden negativen Effekten:

- Es werden mehr Komponenten bzw. umfangreichere Baukästen entwickelt, als notwendig.
- Kunden sind aufgrund des angebotenen Variantenspektrums überfordert.
- Kunden mit identischen Anforderungen werden mit unterschiedlichen Produktvarianten bedient.
- Änderungen und Weiterentwicklungen werden auf den gesamten, überdimensionierten Baukasten ausgerollt.
- Aufgrund der Größe der Baukästen ist es nicht möglich im Vorhinein zu erkennen, wie viele Varianten tatsächlich baubar sind.
- Unterbrechung der Lernkurve der Produktionsmitarbeiter durch häufig wechselnde Produkte bzw. Montageumfänge

1.2 Zielsetzung

Ziel der Arbeit ist es herauszufinden, mit welchen Tools und Methoden die Datensätze, die von Produktdokumentations- und Enterprise-Ressource-Planning-Systemen geliefert werden, im Hinblick auf das Varianten- und Komplexitätsmanagement, analysiert und bewertet werden können. Dabei sollen einerseits die zugrundeliegenden Methoden genau erläutert und andererseits die verschiedenen Analysen auf unterschiedliche Betrachtungsweisen bzw. -mengen untersucht werden. Es wird dabei nur auf die Bereiche Produkt- und Teilevielfalt eingegangen, nicht auf die Bereiche der Montagevielfalt (vgl. Variantenbaum von Schuh [5]), da die dafür notwendigen Daten im Zuge dieser Arbeit nicht vorgelegen sind.

Aus den Analysen sollen geeignete Bewertungsmaßnahmen abgeleitet werden.

1.3 Aufbau der Arbeit

In Kapitel 2 werden zuerst die verwendeten Begriffe klar definiert, da es unterschiedliche Verwendungen in der Praxis und der Theorie gibt. Um also diese Arbeit mit anderen Arbeiten vergleichen und in der Praxis verwenden zu können ist diese Begriffsdefinition notwendig.

Nach der Begriffsabgrenzung folgen die Grundlagen der Dokumentation, um für Leser dieser Arbeit, die nicht vom Fach sind, den Einstieg in die Methoden und Analysen zu vereinfachen.

Gleiches gilt für die mathematischen Grundlagen, die in den Methoden Anwendung finden. Die Erklärung dient einerseits dem leichteren Verständnis der Methoden, andererseits werden auch hier die Unterschiede zwischen Theorie und Praxis aufgezeigt.

Kapitel 3 liefert eine Erklärung der verwendeten Methoden, die für die Analysen bzw. Berechnungstools notwendig sind und liefert eine theoretische Grundlage, wie solche Analysen und Tools angewendet bzw. für welche Methoden etwaige Algorithmen programmiert werden müssen.

In Kapitel 4 wird diskutiert, welche Informationen aus den Dokumentationssystemen zwingend notwendig sind, um die Analysen durchführen zu können und welche Informationen optional sind und einen Mehrwert liefern um diverse Schnittmengen über die Produktdokumentation legen zu können.

Zusätzlich wird darauf eingegangen, wie aus den zwingenden Informationen über ein Pre-Processing optionale Informationen gewonnen werden können bzw. welche Methoden aus Kapitel 3 im Vorhinein den Analysen für die gesamte oder einen Großteil der Dokumentation angewendet werden, um die Rechenzeit zu reduzieren.

Kapitel 5 liefert schließlich einen detaillierten Einblick darin, wie die Dokumentation mit den beschriebenen Methoden analysiert und wie die Ergebnisse bewertet bzw. interpretiert werden können. Dabei wird sowohl auf bestehende Methoden und Analysen aus der Literatur eingegangen, wie z.B. der Merkmalbaum von Schuh (Kapitel 5.2), oder die Paketbildung nach Kesper (Kapitel 5.3.1), als auch im Zuge der Diplomarbeit neu entstandene Analysen, wie die Inter-Vehicle-Commonality (Kapitel 5.3.2), Inter-Stücklisten-Commonality (Kapitel 5.3.3), Baubarkeitsanalyse (Kapitel 5.9), Exklusivitätsindex (Kapitel 5.4) etc. eingegangen.

2 Grundlagen

In dem folgenden Kapitel werden die, für diese Arbeit relevanten, Begriffe, Dokumentations- und mathematischen Grundlagen erläutert. Dabei wird einerseits auf die Definitionen, wie sie in der Literatur zu finden sind und andererseits auf, in der Praxis gebräuchliche, Anwendungen eingegangen.

2.1 Definitionen und Begriffsabgrenzungen

2.1.1 Produkt

In der Literatur finden sich unterschiedliche Verwendungen und Definitionen des Begriffs „Produkt“. Die VDI Richtlinie 2221 definiert das Produkt als Erzeugnis, das als Ergebnis des Entwickelns und Konstruierens hergestellt oder angewendet wird, wobei es sich dabei sowohl um ein materielles (Maschinen, etc.) als auch um ein immaterielles (Programme o.ä.) handeln kann [6]

Eine weitere, von Kesper beschriebene, breitere Begriffsbestimmung weist darauf hin, dass der Begriff „Produkt“ in der Literatur und Praxis nicht nur Anwendung für Sachleistungen, sondern auch für Dienstleistungen und sogenannten Hybridleistungen (Kombination aus Sach- und Dienstleistungen), findet [1].

Zwei sehr ähnliche Definitionen werden von Sabisch und Vogel genannt. Sie bezeichnen das Produkt als eine von einem Unternehmen am Markt angebotene Sachleistung zur Befriedigung von Kundenbedürfnissen [7]

In Anlehnung an die VDI Richtlinie und die Definitionen von Sabisch und Vogel wird das Produkt als Erzeugnis bzw. Sachleistung definiert, das als Ergebnis des Entwickelns und Konstruierens hergestellt und am Markt, zur Befriedigung von Kundenbedürfnissen, angeboten wird.

2.1.2 Eigenschaft, Merkmal, Ausprägung

Systeme weisen Eigenschaften auf und jede Eigenschaft setzt sich aus einem Merkmal und dessen Ausprägung zusammen. Die Ausprägung eines Merkmals kann wiederum von den Eigenschaften eines untergeordneten Systems abhängen. Als System kann sowohl ein Produkt, eine Baugruppe als auch nur ein einzelnes Bauteil gewählt werden. Merkmale können nicht nur von Subsystemen abhängen, sie werden auch nach Beschaffenheits-, Funktions- und Relationsmerkmalen unterschieden. Beschaffenheitsmerkmale werden direkt von der Entwicklung festgelegt und werden daher auch „direkte Merkmale“ genannt. Funktionsmerkmale beschreiben die vom System durchführbaren Handlungen und ergeben sich, genauso wie die Relationsmerkmale, direkt aus den Beschaffungsmerkmalen. Relationsmerkmale sind Eigenschaften eines Systems, die sich erst im Zusammenhang mit anderen Systemen ergeben. Aufgrund ihrer Abhängigkeit von Beschaffungsmerkmalen werden Funktions- bzw. Relationsmerkmale auch als „indirekte Merkmale“ bezeichnet [8].

Zur Erläuterung der oben genannten Definitionen sei folgendes Beispiel genannt:

Ein Motor besitzt die Eigenschaften 190 PS Leistung und 2 Liter Hubraum. Hierbei sind die Merkmale „Leistung“ und „Hubraum“ in den Ausprägungen „190 PS“ bzw. „2 Liter“ anzutreffen. Da sich diese Merkmale unter anderem durch das Zusammenspiel der Eigenschaften „Zylinderanzahl 4“, „Bohrungsdurchmesser 6 cm“ und „Kolbenhub 8,6 cm“ ergeben, handelt es sich um Relationsmerkmale. Zylinderzahl, Bohrungsdurchmesser und Kolbenhub sind Beschaffungsmerkmale, da diese direkt vom Entwickler festgelegt werden. Bei dem Kolbenhub handelt es sich um ein Funktionsmerkmal, das von den Beschaffungsmerkmalen der Kurbelwelle (Kröpfung) abhängt. [1]

2.1.3 Derivat, Variante, Variantentreiber

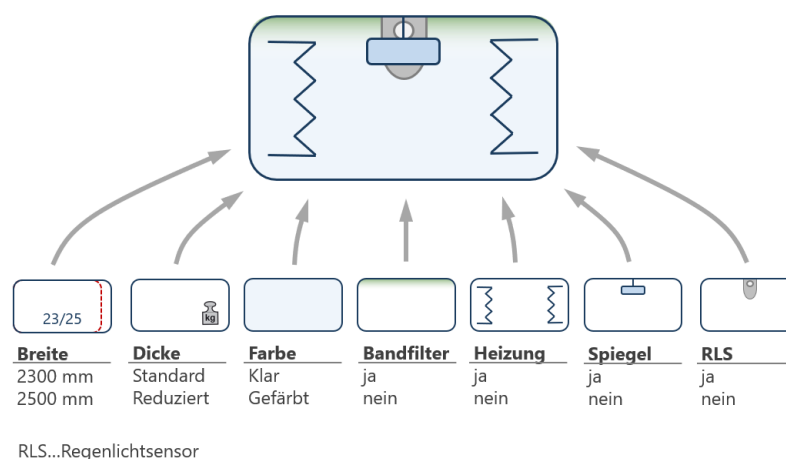
In der DIN 199-1:2002-03 wird die *Variante* definiert als Gegenstand ähnlicher Form bzw. Funktion, mit einem hohen Anteil an Gleichteilen [9].

Der Begriff *Derivat* wird in der Automobilindustrie oft im Zusammenhang mit den Varianten eines Basismodells verwendet. [10]

Im Sinne des einheitlichen Verständnisses wird in dieser Arbeit nur der Begriff *Variante* nach DIN 199-1:2002-03 angewendet.

Die variantenbildenden Merkmale werden von Feldhusen als *Variantentreiber* bezeichnet, dabei sind variantentreibende Merkmale z.B. unterschiedliche Farben, Karosserieformen, Ausstattungsmerkmale etc. In Abbildung 2.1-1 sind beispielhaft die Variantentreiber und ihre Ausprägungen einer Windschutzscheibe für einen LKW dargestellt. Die mögliche Anzahl an Varianten ergibt sich aus dem Produkt der Ausprägungen der Variantentreiber. Im Beispiel der Windschutzscheibe also zu $2^7 = 128$. Daran ist zu erkennen, dass Konfigurationslogiken benötigt werden um das exponentielle Wachstum der Variantenanzahl zu verringern. Die Variantentreiber und Konfigurationslogiken müssen bereits in der Planungsphase klar definiert werden, da diese bei komplexen Produkten (z.B. Nutzfahrzeugen) schnell unbeherrschbar werden und nur aufwendig gepflegt bzw. ermittelt werden können[11].

Abbildung 2.1-1: Variantentreiber der Windschutzscheibe



2.1.4 Produktprogramm, Produktportfolio

Während im Duden das *Produktportfolio* als Produktpalette, ferner als Auswahl an Produkten [12] und im online Wirtschaftslexikon Onpulso als, von einem Unternehmen angebotenes Produktspektrum [13], definiert wird, findet in der Literatur überwiegend der Begriff Produktprogramm Verwendung.

Homburg definiert das *Produktprogramm* als Gesamtheit der zu einem bestimmten Zeitpunkt von einem Unternehmen angebotenen Produkte [14], Jonas bezeichnet es als die Gesamtheit aller Erzeugnisse bzw. Sach- und Dienstleistungen, die ein Unternehmen auf dem Markt anbietet [15].

In Anlehnung an diese Definitionen wird das Produktprogramm als alle, zu einem bestimmten Zeitpunkt angebotenen, Produkte bzw. Produktvarianten definiert.

2.1.5 Baumuster, Baureihe

In der Literatur und in der Produktion finden sich sehr unterschiedliche Verwendungen des *Baumusters*. Es wird einerseits im Zusammenhang mit der EG-Baumusterprüfung der Europäischen Union verwendet und somit als Muster, dem alle weiteren Exemplare in der Produktion entsprechend nachgebaut werden müssen, wobei die relevanten Normen definierte Varianten eines Baumusters zulassen [16].

In der Automobilindustrie wird das *Baumuster* jedoch verwendet, um die Art des zu bauenden Fahrzeuges zu beschreiben, in dem laut Kropik oft ein Verweis auf Plattformen, Art der Lenkung (links oder rechts) oder vorgesehener Motor zu finden ist [17].

Angelehnt an Kropik wird es für diese Arbeit wie folgt definiert:

Das *Baumuster* legt eine bestimmte Anzahl von Eigenschaften bzw. die Ausprägung bestimmter Merkmale, z.B. Anzahl der Achsen, Anzahl der angetriebenen Achsen, höchstzulässiges Gesamtgewicht, Art des Aufbaus, etc. fest. Die Baumusterbezeichnung ist üblicherweise auch Teil der Seriennummer bzw. im Fall der Automobilindustrie Teil der Fahrzeug-Identifizierungsnummer.

In diesem Zusammenhang ist, sowohl in der Praxis als auch in der Literatur, zusätzlich der Begriff Baureihe zu finden. So wird im Bereich der PKW eher der Begriff Baureihe verwendet, bei den Nutzfahrzeugen hingegen Baumuster [17], [18].

Um Verwechslungen zu vermeiden wird die Baureihe im Zuge dieser Arbeit gemäß der ebenfalls in der Literatur anzutreffenden Definition als Reihe von Bauteilen, die in Funktion, Material, konstruktiver Lösung und Fertigungsprozess möglichst gleich sind und sich nur in ihrer Größe unterscheiden, verstanden [19], [20]

Die Bezeichnung des Baumusters dient auf jeden Fall für den eindeutigen Bezug, kann jedoch auch Informationen über die Eigenschaften beinhalten.

Als Beispiel nehmen wir ein Baumuster mit folgenden Eigenschaften an:

- Höchstzulässiges Gesamtgewicht: 3,5t
- Karosserieform: Stufenheck
- Antriebsart: Elektrofahrzeug, Allrad

Dieses Baumuster könnte man einfach „BM1“ nennen, oder „SH35EA“.

2.1.6 Variantenvielfalt, Varianz, Komplexität

Variantenvielfalt und *Komplexität* werden speziell im deutschen Sprachraum häufig gleichgesetzt bzw. für die gleichen Problemstellungen verwendet. Für die Bewältigung des Varianten- und Komplexitätsmanagements ist es jedoch unerlässlich zwischen beiden Begriffen zu differenzieren.

Die Menge an verschiedenen möglichen Varianten von Bauteilen, Baugruppen bzw. Produkten wird in dieser Arbeit als Variantenvielfalt verstanden. [1] Dabei wird nicht nur zwischen der Andersartigkeit in Bezug auf die Bauteileigenschaften (Material, Geometrie etc.) sondern auch der Bezugsart (Kaufteil, Eigenbau, etc.) unterschieden.

Ein Begriff der ebenfalls in der Literatur und Praxis oft mit der Variantenvielfalt synonym verwendet wird ist die Varianz [21].

Handelt es sich bei der Variantenvielfalt um die Anzahl aller möglichen Varianten, wird in dieser Arbeit unter der Varianz die empirische Varianz aus der Statistik, also der Streuung der Stichprobe und somit die tatsächliche Anzahl der benötigten bzw. verwendeten Varianten verstanden. [22]

Nach Lindemann kann die *Komplexität* eines Systems durch verschiedene Merkmale beschrieben werden[8]:

- Elemente: Anzahl, Ungleichmäßigkeit der Aufteilung sowie Art und Verschiedenartigkeit der Elemente des Systems.
- Relationen: Anzahl, Ungleichmäßigkeit der Aufteilung sowie Art und Verschiedenartigkeit der Relationen der Elemente des Systems
- Dynamik des Systems
- Art und Anzahl der Zustände des Systems

Möchte man die Komplexität des Systems „Produktprogramm“ anhand dieser Beschreibung festlegen, können die Merkmale wie folgt zugeordnet werden:

- Elemente entsprechen der Variantenvielfalt
- Relationen entsprechen den Konfigurationsbedingungen bzw. den Stücklisten. Die Konfigurationsbedingungen schränken die Konfigurierbarkeit ein, damit nicht jede Eigenschaft beliebig konfiguriert werden kann. Die Relationen in den Stücklisten beziehen sich darauf, dass bestimmte Bauteile und Baugruppen in unterschiedlichen Ebenen in unterschiedlichen Stücklisten vorkommen können.

- Dynamik des Systems kann durch die Änderungen im Konfigurationsmanagement bzw. in den Stücklisten beschrieben werden. Dies beinhaltet einerseits die Modellpflege (z.B. Facelift), Änderung der Lieferanten, externe Einflüsse durch den Gesetzgeber, etc.

2.1.7 Externe Variantenvielfalt, Varianz und Komplexität vs. interne Variantenvielfalt, Varianz und Komplexität

Die oben beschriebenen Begriffe Variantenvielfalt, Varianz und Komplexität können sowohl aus Kundensicht, also extern, als auch aus Unternehmenssicht, also intern, betrachtet werden. Diese Betrachtungsunterschiede sind in der Literatur durchgängig zu finden. [1]

Variantenvielfalt:

Die *externe Variantenvielfalt* beschreibt, die vom Kunden erlebte oder wahrgenommene Variantenvielfalt und somit alle am Markt angebotenen Produktvarianten und entspricht damit dem Produktprogramm. [1] Es sei hier jedoch darauf hingewiesen, dass die externe Variantenvielfalt in vielen Fällen größer ist, als sie von Kunden tatsächlich wahrgenommen werden kann. Es herrscht sozusagen ein Überangebot, da das Ziel ist, so viele individuelle Kundenwünsche wie möglich zu befriedigen, die vom Wettbewerber eventuell nicht erfüllt werden können.[2]

Unter der *internen Variantenvielfalt* versteht man alle Bauteile und Baugruppen, die zur Abbildung der externen Variantenvielfalt notwendig sind. Sie entspricht der Menge aller Sachnummern, die für die Produktion freigegeben sind bzw. freigegeben waren. [1]

Die interne Variantenvielfalt wird von der externen Variantenvielfalt maßgeblich beeinflusst, da eine große externe Variantenvielfalt in der Regel auch eine größere interne Variantenvielfalt benötigt. Wie stark die interne Variantenvielfalt zunimmt, hängt jedoch stark von der Produktgestaltung ab. [1]

Varianz:

Analog zur Variantenvielfalt kann die externe und interne Varianz definiert werden.

Die *externe Varianz* ist die Streuung der tatsächlich gebauten Varianten aus Kunden bzw. Funktionssicht und die Aufteilung auf die Code (Def. Kapitel 2.2.3). Sie lässt sich direkt aus dem Bauprogramm ableiten und führt zu Coden die häufig oder selten gewählt wurden. Code die nie gewählt wurden sind nicht Teil der externen Varianz.

Die *interne Varianz* ist die Streuung der benötigten Stücklisten und in weiterer Folge benötigten Sachnummern. Auch hier kann wieder zwischen häufig und selten gewählten Stücklisten unterschieden werden. Hier gilt zu beachten, dass selten gewählte Stücklisten nicht automatisch zu selten gewählten Sachnummern führen.

Komplexität:

Während bei den Begriffen Variantenvielfalt und Varianz der Unterschied zwischen extern und intern an der Betrachtungsweise liegt, werden in der Literatur unter der externen und internen Komplexität durchwegs die entstandene Komplexität durch interne und externe Einflussfaktoren verstanden. [5], [19], [23]

Unter der *externen Komplexität* werden also externe Ursachen für die Komplexität verstanden, die von dem Unternehmen nicht beeinflusst gehören. Diese sind[5], [19], [23], [24]:

- Steigerung der Marktkomplexität durch die Fokussierung auf den Kunden, welcher eine hohe Individualisierung bei gleichbleibenden oder geringeren Kosten fordert. Dies führt zu einer steigenden Variantenvielfalt und hohem Kostendruck.
- Globalisierung erfordert einerseits die Anpassung des Produktes an unterschiedliche Märkte mit anderen Kundenanforderungen und Gesetzen, andererseits führt die Globalisierung zu einer höheren Anzahl an Konkurrenzprodukten, sowie zu kürzeren Produktlebenszyklen.
- Technologischer Wandel, der einerseits erzwungen wird, da die Hersteller versuchen mit technologischen Neuheiten und Vorsprung gegenüber der Konkurrenz Marktanteile zu generieren, andererseits durch die Gesetzgeber gefordert z.B. durch strengere Anforderungen im Umwelt- oder Kundenschutz.

Die *interne Komplexität* lässt sich in die folgenden Komplexitätsfelder einteilen[2]:

- Angebotskomplexität
- Produktkomplexität
- Prozesskomplexität
- Entscheidungskomplexität

Der Fokus dieser Arbeit liegt auf der Produktkomplexität.

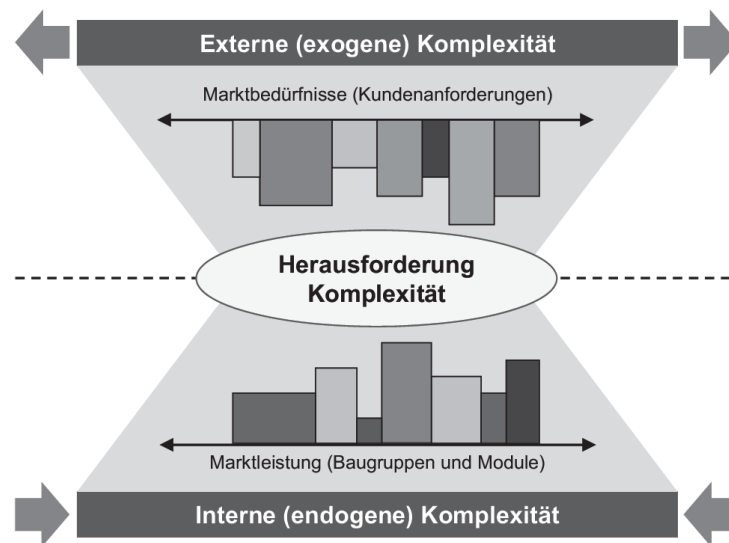
Als einer der Haupttreiber der Produktkomplexität kann die steigende Variantenvielfalt, sowohl intern als auch extern, identifiziert werden. Die steigende externe Variantenvielfalt erfordert eine hohe interne Variantenvielfalt. Um die Prozesskomplexität zu reduzieren wird in den Unternehmen die Fertigungstiefe verringert und ein Teil der Komplexität an die Zulieferer ausgelagert in dem z.B. ganze Module von einem Zulieferer montiert werden (vgl. [23]), allerdings sinkt dadurch die Wiederholrate der einzelnen Teile, was zu geringeren Skaleneffekten im Einkauf führt.[5]

2.1.8 Komplexitäts- und Variantenmanagement

Nach Schuh kann das Komplexitätsmanagement in verschiedene Kategorien entsprechend dem Produktlebenszyklus unterteilt werden. Während der Planung geht es primär um Komplexitätsvermeidung und wirtschaftliche Beherrschung der Komplexität, in der Produktionsphase geht es ebenfalls um Beherrschung und um systematische Reduktion der Komplexität.

Er definiert das Komplexitätsmanagement als Gestaltung, Steuerung und Entwicklung der Vielfalt des Leistungsspektrums im Unternehmen. Es wird angestrebt, dass die Vielfalt in allen Wertschöpfungsstufen beherrscht wird, um einen maximalen Beitrag zum Kundennutzen bei zugleich hoher Wirtschaftlichkeit zu erzielen. Dabei müssen die externe und die interne Komplexität aufeinander abgestimmt werden, was zusätzliche Komplexität hervorruft (vgl. Abbildung 2.1-2). Besonders hervorzuheben ist, dass die Produktkomplexität nicht nur auf das Produkt beschränkt werden darf, also nicht nur auf die reine Variantenvielfalt, sondern die gesamte Wertschöpfungskette Einfluss darauf hat und betrachtet werden muss. [5]

Abbildung 2.1-2: Interne vs. externe Komplexität[5]



Da die Variantenvielfalt einen erheblichen Einfluss auf die Produktkomplexität hat, kommt ihr und dem Variantenmanagement besondere Bedeutung zu.

Ehrlenspiel definiert das Variantenmanagement als alle Maßnahmen, mit denen die Variantenvielfalt innerhalb eines Unternehmens bewusst beeinflusst wird, mit dem Ziel der Reduzierung und Beherrschung der Produkt- bzw. Prozesskomplexität. Dies bedeutet genauer minimale Komplexität bei ausreichend großer externer Variantenvielfalt für den Kunden. Das Variantenmanagement kann dabei sowohl Einfluss auf die interne als auch auf die externe Variantenvielfalt nehmen. [19]

Auch Schuh versteht unter dem Variantenmanagement die Entwicklung, Gestaltung und Strukturierung von Produkten mit dem Ziel, die vom Produkt ausgehende, wie auch die aufs Produkt wirkende, Komplexität mittels geeigneter Methoden und Werkzeuge zu managen.[5]

Ehrlenspiel beschreibt die Ziele des Variantenmanagements genauer als [19]:

- Bedienen der am Markt erforderlichen Varianz (externe Varianz)
- Erkennen der unnötigen Varianten und Reduzieren dieser
- Verringern der Durchlaufzeit und der indirekten Kosten der benötigten Varianten

2.2 Dokumentation

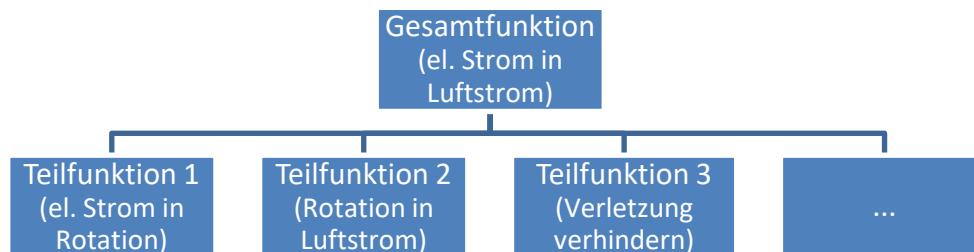
2.2.1 Funktionsstruktur, Produktstruktur, Produktarchitektur

Ein Produkt kann aus zwei Perspektiven, einerseits die funktionale, andererseits die physische, betrachtet und beschrieben werden. Die funktionale Beschreibung, auch Funktionsstruktur genannt, betrachtet das Produkt aus Marktsicht und gibt die Ausprägungen der Funktions- und Relationsmerkmale wieder. Die Produktstruktur betrachtet es aus der Produktionssicht und stellt die physische Zusammensetzung des Produkts aus Baugruppen

und Komponenten dar. Wie in Abbildung 2.2-1 dargestellt, werden in der Funktionsstruktur aus der Gesamtfunktion Unterfunktionen bzw. Teilfunktionen abgeleitet.

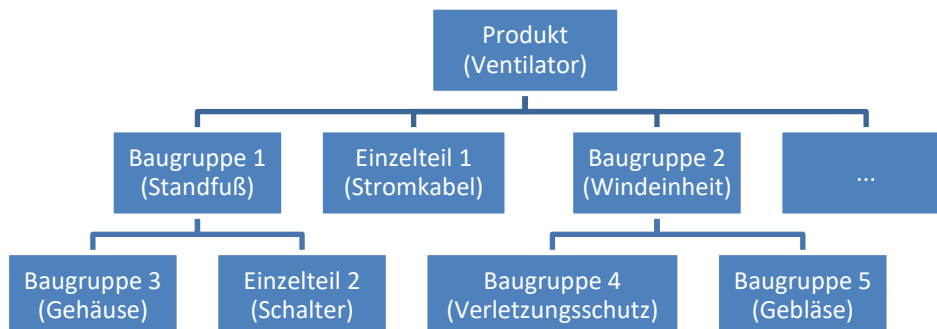
Als Beispiel sei an dieser Stelle ein sicherer Ventilator genannt, dessen Gesamtfunktion es ist, elektrischen Strom in einen Luftstrom umzuwandeln mit dem Zusatz, dass man sich nicht daran verletzen kann. Teilt man diese Gesamtfunktion in Teilfunktionen auf, findet man den Elektromotor, dessen Funktion es ist, elektrischen Strom in eine Rotation umzuwandeln, die Ventilatorblätter, deren Funktion es ist die Rotation in einen Luftstrom umzuwandeln und den Kundenschutz z.B. in Form eines Gitters. Neben den genannten Funktionen gibt es noch weitere Funktionen (z.B. Standfestigkeit, etc.).

Abbildung 2.2-1: Funktionsstruktur



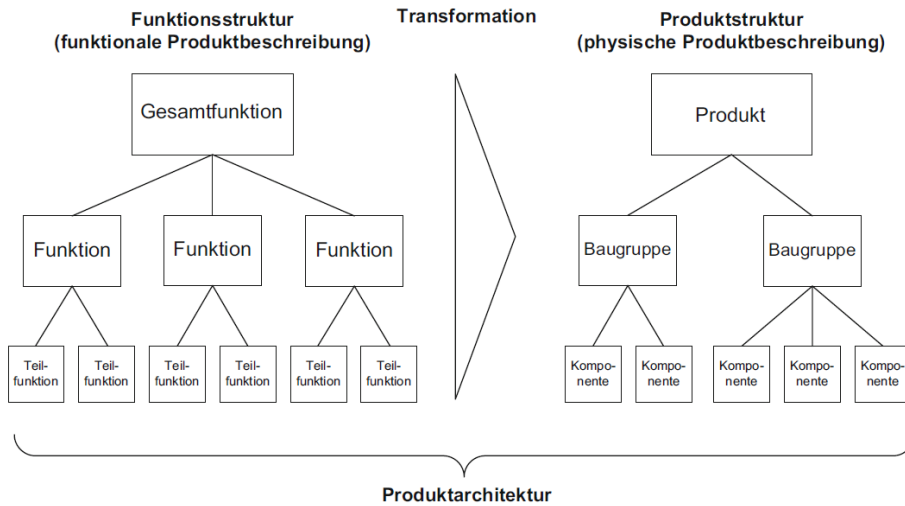
Wie in Abbildung 2.2-2 dargestellt, wird in der Produktstruktur das Produkt in verschiedenen Ebenen bis auf Bauteilebene aufgespalten und zeigt *eine* mögliche physische Lösung zur Erfüllung der geforderten Funktionen.

Abbildung 2.2-2: Produktstruktur



Die Produktarchitektur verknüpft diese beiden Strukturen bzw. transformiert die Funktionsstruktur in die Produktstruktur wie in Abbildung 2.2-3 dargestellt.

Abbildung 2.2-3: Produktarchitektur [11]

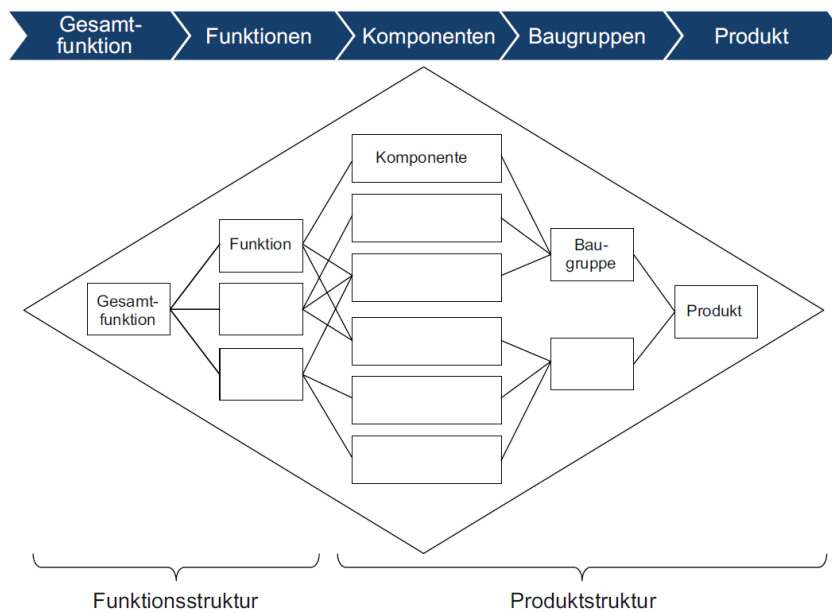


Eine weitere Darstellungsform der Produktarchitektur ist die sogenannte *METUS-Raute* (Abbildung 2.2-4). In ihr wird die Verknüpfung zwischen Funktionsstruktur und Produktstruktur dargestellt, daraus sind folgende Kombinationen ersichtlich:

- Eine Funktion kann mehrere Komponenten auslösen.
- Eine Komponente kann mehrere Funktionen erfüllen.
- Eine Funktion löst genau eine Komponente aus.

Aus diesen Kombinationen ergeben sich *funktionale Beziehungen* zwischen den Komponenten.

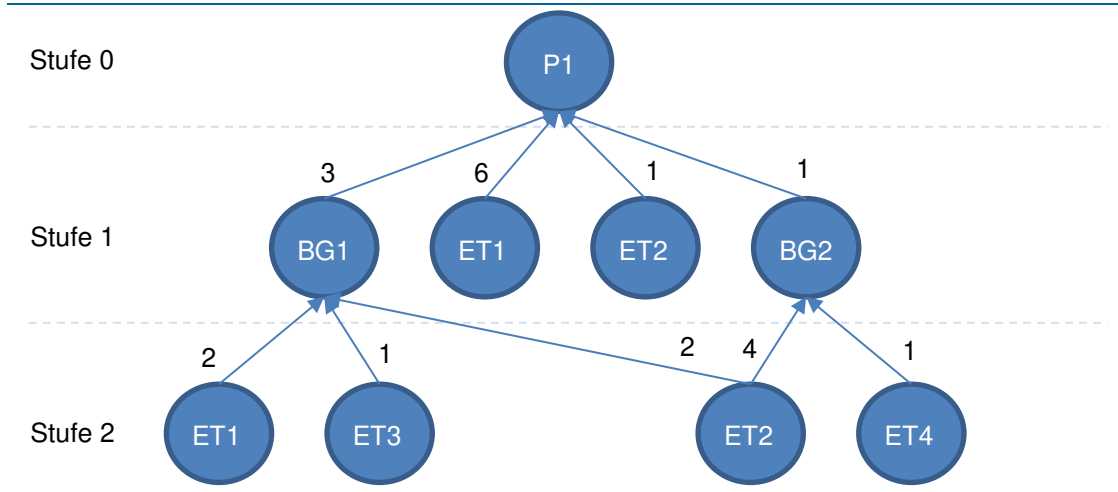
Abbildung 2.2-4: METUS-Raute[11]



Zu beachten ist, dass in dieser Darstellung die Produktstruktur montageorientiert dargestellt ist, woraus folgt, dass eine Komponente nur in einer Baugruppe vorhanden sein kann. Ist eine Komponente in mehreren Baugruppen vorhanden (z.B. eine Befestigungsschraube), so ist sie in jeder Baugruppe separat dargestellt. Diese Darstellung der Produktstruktur entspricht der Dokumentation in Form von Strukturstücklisten (vgl. Kapitel 2.2.2).[11]

Im Gegensatz zu obiger Darstellung ist der Gozintograph nicht Montageorientiert (Abbildung 2.2-5). Er lässt sich aus der Strukturstückliste ableiten und in ihm ist ersichtlich, aus welchen Einzelteilen (ET) und Baugruppen (BG) ein Produkt (P) besteht. Der Gozintograph hat nicht zwingend eine Baumstruktur, da man auch die Gozintographen mehrerer Produktvarianten bzw. einzelner Baugruppen übereinander legen kann um Gleichteile bzw. das Potential dafür erkennen zu können [25].

Abbildung 2.2-5: Beispiel eines Gozinto-Graphen



In der Dokumentation der Produktarchitektur werden die Codebedingungen festgelegt bzw. dargestellt, die es ermöglichen von einer Produktkonfiguration durch eine Codekette mittels Produktauflösung die benötigten Stücklisten zu erhalten (vgl. Kapitel 3.3).

Es sei hier darauf hingewiesen, dass es sich bei den Codebedingungen nicht um die Transformationsbeziehungen handelt, um von der Funktionsstruktur auf die Produktstruktur zu gelangen.

2.2.2 Stücklisten

Aus der Produktstruktur bzw. dem Gozinto-Graphen lassen sich Stücklisten erstellen. Nach Eigner und Stelzer zeigt die Stückliste für eine vorgegebene Sache alle untergeordneten Sachen und stellt eine elektronisch verwend- und verwertbare Abbildung der Produktstruktur dar. Diese können grundsätzlich in folgende Arten von Stücklisten unterteilt werden[26]:

- **Mengenstückliste:**
Darstellung der Gesamtmengen aller untergeordneten Elemente, ohne jegliche Struktur
- **Baukastenstückliste:**
Darstellung nur der direkt untergeordneten Elemente und deren Mengen. Führt zu mehreren

Stücklisten, wenn ein Endprodukt aus mehreren Baugruppen mit Stücklisten besteht und nicht nur aus Einzelteilen.

■ **Strukturstückliste:**

Kombination aus Mengenstückliste und Baukastenstückliste, zeigt sowohl die Mengen der Einzelteile und Baugruppen, als auch die Struktur der Erzeugnisse und Baugruppen. Es gilt zu beachten, dass nur die Mengen in der jeweiligen Ebene und nicht die Gesamtmengen abgebildet sind.

Aus der Baukastenstückliste lässt sich einfach die Strukturstückliste ableiten und umgekehrt, aus beiden Stücklisten lässt sich die Mengenstückliste ableiten, aus der Mengenstückliste lässt sich umgekehrt weder die Baukasten- noch die Strukturstückliste ableiten, da in ihr jegliche Struktur verloren geht.

In Abbildung 2.2-6 sind die oben genannten Arten der Stücklisten in Anlehnung an Eigner und Stelzer für den Gozinto-Graphen aus Abbildung 2.2-5 zur Veranschaulichung dargestellt.

Abbildung 2.2-6: Stücklistendarstellungen

Mengenstückliste	
P1	
Teil	Menge
B1	3
B2	1
E1	12
E2	11
E3	3
E4	1

Baukastenstückliste	
P1	
Teil	Menge
B1	3
B2	1
E1	6
E2	1

Strukturstückliste	
P1	
Teil	Menge
B1	3
E1	2
E2	2
E3	1
E1	6
E2	1
B2	1
E2	4
E4	1

Baukastenstückliste	
B1	
Teil	Menge
E1	2
E2	2
E3	1

Baukastenstückliste	
B2	
Teil	Menge
E2	4
E4	1

Für die Darstellung unterschiedlicher Varianten eines Produktes lassen sich aus der Baukasten- bzw. Strukturstückliste die folgenden Möglichkeiten ableiten [26]:

■ **Vollständige Auflösung aller Varianten:**

Es wird für jede Produktvariante eine eigene Strukturstückliste erstellt. Aufgrund des hohen Aufwands in der Ablage der Stücklisten und der auftretenden Redundanzen, weil einzelne Baukastenstücklisten mehrfach abgelegt werden, findet diese Darstellungsform in der Praxis kaum mehr Verwendung

■ **Gleichteilstückliste:**

Für alle Baugruppen und Einzelteile, die in allen Varianten des Produktes vorkommen, wird eine Gleichteilstückliste erstellt. Die Stückliste einer Variante besteht dann aus der Gleichteilstückliste und zusätzlichen Baugruppen/Einzelteilen.

■ **Plus-Minus-Stückliste:**

Die Plus-Minus-Stückliste besteht aus allen Komponenten und jede Komponente, die in einer Variante vorkommt wird in der Stückliste mit einer positiven oder negativen Menge, in Bezug auf den Gleichteileanteil, gekennzeichnet.

■ **Variantenstückliste:**

Alle Bauteile werden mit einem Kenner versehen, dabei gibt es Bauteile die in allen Varianten gleich sind und daher fixer Bestandteil aller Varianten sind (F), optionale Bauteile die abhängig von der Ausprägung gewählt werden (O) und Muss-Varianten Bauteile, alternative Bauteile von denen immer eine Variante gewählt werden muss (V). Die Auswahlmöglichkeiten sind in der Variantenleiste dargestellt. Für jeden Muss-Varianten Bauteil existiert eine eigene Variantenleiste. Die Stückliste für eine Variante wird Variantenausprägung genannt.

■ **Maximalstückliste[27]:**

In der Maximalstückliste sind alle Bauteile mit ihren maximal vorkommenden Mengen dargestellt, dabei kann es sich auch um nicht baubare Varianten handeln, da z.B. zu viele Türen vorhanden sind. Entsprechend der benötigten Mengen in den Varianten werden die überschüssigen Mengen abgezogen woraus schließlich eine baubare Stückliste entsteht.

Zur Veranschaulichung der Stücklistenarten werden Variationen des Produktes aus Abbildung 2.2-5 in den folgenden Abbildungen dargestellt. Abbildung 2.2-7 zeigt die Variationen bereits in vollständiger Auflösung.

Abbildung 2.2-7: Vollständige Auflösung der Varianten P1, P2 und P3

Strukturstückliste		Strukturstückliste		Strukturstückliste	
P1		P2		P3	
Teil	Menge	Teil	Menge	Teil	Menge
B1	3	B1	3	B1	3
E1	2	E1	2	E1	2
E2	2	E2	2	E2	2
E3	1	E3	1	E3	1
E1	6	E1	6	E1	4
E2	1	B3	2	B2	2
B2	1	E3	1	E2	4
E2	4	E4	3	E4	1
E4	1	E5	2	E5	2

Anhand der Gleichteilstückliste in Abbildung 2.2-8 ist zu erkennen, dass diese Darstellungsform bereits wesentlich kompakter ist als die vollständige Auflösung der Varianten und bereits negative Mengen auftreten können.

Abbildung 2.2-8: Gleichteilstückliste der Varianten P1, P2 und P3

Gleichteilstückliste	
GT	
Teil	Menge
B1	3
E1	6

Baukastenstückliste	
P2	
Teil	Menge
GT	1
B3	2

Baukastenstückliste	
P1	
Teil	Menge
GT	1
E2	1
B2	1

Baukastenstückliste	
P3	
Teil	Menge
GT	1
E1	-2
B2	2
E5	2

Abbildung 2.2-9 zeigt die Plus-Minus-Stückliste und die Variantenstückliste, darin ist die Kompaktheit der Plus-Minus-Stückliste deutlich zu erkennen.

Abbildung 2.2-9: Plus-Minus-Stückliste und Variantenstückliste der Varianten P1, P2 und P3

Plus-Minus-Stückliste				
Teil	GT	P1	P2	P3
B1	3			
B2		1		2
B3			2	
E1	6			-2
E2		1		
E5				2

Variantenstückliste P		
Teil	Kenner	Menge
B1	F	3
B	V	
E	V	
E2	O	1
E5	O	2

Zugehörig zu der Variantenstückliste aus Abbildung 2.2-9 sind in Abbildung 2.2-10 die Variantenleisten und zwei Variantenausprägungen dargestellt.

Abbildung 2.2-10: Variantenleisten und Variantenausprägungen P1 und P3

Variantenleiste B	
Teil	Menge
B2	1
B2	2
B3	2

Variantenleiste E	
Teil	Menge
E1	4
E1	6

Variantenausprägung P(1)	
Teil	Menge
B1	3
E1	6
B2	1
E2	1

Variantenausprägung P(3)	
Teil	Menge
B1	3
E1	4
B2	2
E5	2

In Abbildung 2.2-11 ist die Maximalstückliste und die Stückliste für eine Variante nach der Konfiguration dargestellt.

Abbildung 2.2-11: Maximalstückliste mit Konfiguration von Variante P3

Maximalstückliste	
Teil	Menge
B1	3
B2	2
B3	2
E1	6
E2	1
E5	2

Konfiguration

Stückliste Variante P3	
Teil	Menge
B1	3
B2	2
B3	2
E1	4
E2	1
E5	2

2.2.3 Code, Codegruppe, Codekette, Codebedingung

Code (CD), auch Sonderausstattungscode (SA-Code) oder PR-Nummern, sind 2- bis 5-stellige alphanumerische Kombinationen, die von den meisten Fahrzeugherstellern genutzt werden, um Ausstattungsmerkmale übersichtlicher abzubilden. So hat jedes wählbare Merkmal (z.B. Standardsitz, Sportsitz, oder unterschiedliche Felgen) einen eigenen Code.

Die im Verkauf mit den Kunden konfigurierten Fahrzeuge werden über eine sogenannte Codekette oder Codeliste abgebildet, durch sie werden alle Ausstattungsdetails eines Fahrzeugs vollständig beschrieben. Code werden häufig nach ihrer funktionalen Übergruppe zu Codegruppen (CG), auch Codefamilien oder Codetypen genannt, ebenfalls mit einer eigenen ID, zusammengefasst. Diese Codegruppen können, müssen allerdings nicht, mit den Variantentreibern korrelieren.

Zusätzlich zu den Ausstattungscodes gibt es Werkssteuer-codes, z.B. werden damit extra Tests, die im Werk durchgeführt werden, zugesteuert [17].

Bei Codebedingungen handelt es sich um boolesche Funktionen (vgl. Kapitel 2.3.2) in denen die vorkommenden Variablen durch Code dargestellt werden. Diese Codebedingungen finden sowohl im Konfigurationsmanagement, als auch in der Produktauflösung von der Codekette in die verwendeten Stücklisten Anwendung.

2.2.4 Fahrzeug-Identifikations-Nummer

Bei der Fahrzeug-Identifikations-Nummer (FIN) oder auch Vehicle-Identification-Number (VIN) handelt es sich um eine genormte Seriennummer für Kraftfahrzeuge. Sie muss laut EU-Verordnung 19/2011 an allen vollständigen und unvollständigen Fahrzeugen der Klassen M, N und O an mehreren Stellen im Fahrzeug angebracht werden.[28]

Der Aufbau der FIN ist in der ISO-Norm 3779:2009 festgelegt. Laut ihr besteht die FIN aus einem 17-stelligen alphanumerischen Code der sich wie folgt zusammensetzt[29]:

Die Position...

- 1 bis 3 enthält den Weltherstellercode (world manufacturer identification – WMI) und ordnet das Fahrzeug einem Hersteller zu. Die WMI wird von der ISO-Norm 3780 festgelegt.
- 4 bis 9 bildet den fahrzeugbeschreibenden Sektor (Vehicle description sector – VDS), dieser Bereich wird vom Fahrzeughersteller festgelegt und muss Informationen bezüglich des Aufbaus der Fahrzeuge beinhalten, daher wird hier oft auf die Bezeichnung der Baumuster bzw. Baureihen zurückgegriffen
- 10 ist der Modelljahrescode. Dieser wird von der ISO-Norm 3779 festgelegt und beschreibt in welchem Modelljahr das Fahrzeug produziert wurde. Das Modelljahr muss sich nicht immer mit dem Kalenderjahr decken.
- 11 stellt den Bezug zum Produktionswerk her. Der Hersteller legt fest, welcher Code für welches Werk steht, alle Fahrzeuge aus diesem Werk haben an dieser Stelle die gleiche Ziffer oder Nummer
- 12 bis 17 ist eine fortlaufende Nummerierung, die mit den restlichen Positionen eine eindeutige Identifizierung des Fahrzeugs ermöglicht.

Anhand des Aufbaus der FIN sind aus dieser bereits die Informationen Baumuster/Baureihe, Produktionswerk und Produktionsjahr feststellbar.

2.2.5 Produktionsprogramm, Bauprogramm

Nach allgemeiner Auffassung in der Literatur, wird in der Produktionsprogrammplanung festgelegt, welche Erzeugnisse zu einem bestimmten Termin in einer bestimmten Menge hergestellt werden. Während im Produktprogramm festgelegt wird, welche Erzeugnisse in einem bestimmten Zeitraum möglich sind (vgl. Kapitel 2.1.4), beschreibt das Produktionsprogramm alle tatsächlich hergestellten Erzeugnisse. [17], [30], [31].

Das Bauprogramm findet Verwendung im Zusammenhang mit Baukastensystemen, so sind geschlossene Baukastensysteme mit endlichen, festgelegten Konfigurationsmöglichkeiten in einem Bauprogramm dargestellt.[11], [32]

Das Produktionsprogramm kann bei speziellen Kundenanforderungen angepasst werden, das Ist-Produktionsprogramm für einen bestimmten Betrachtungszeitraum besteht jedoch aus einer endlichen, bestimmten Anzahl an fertigen Erzeugnissen in festgelegten Konfigurationen und wird daher nach obiger Definition, in der Praxis, ebenfalls als Bauprogramm bezeichnet.

2.2.6 Dokumentationssysteme

Im Bereich des Komplexitäts- und Variantenmanagements werden Produkt-Daten-Management (PDM)-, Produkt-Lifecycle-Management (PLM) - und Enterprise-Ressource-Planning (ERP) -Systeme eingesetzt.

PDM-Systeme übernehmen nach Eigner und Stelzer das Management von produktdefinierenden Daten in Verbindung mit der Abbildung und dem Management von Geschäftsprozessen. Das dadurch ermöglichte kollaborierende Prozess- und Produktmanagement erlauben eine Rekonfiguration von Konfigurations- und Fertigungsständen entlang des gesamten Produktlebenszyklus. Dabei werden die unterschiedlichen Dokumente mit divergierenden Dateiformaten, die während dem Produktlebenszyklus entstehen, auf Datenbanksystemen (Dokumenten-Management-System (DMS)) abgelegt und die Beziehungen zueinander verwaltet. Die unterschiedlichen bedienten Bereich sind: [26]

- Produkt und Dokumentenmanagement:
 - Zuordnung von produktbezogenen Dokumenten aller Arten
- Prozessmanagement:
 - Abbildung, Verwaltung und Dokumentation der zeitlichen Veränderung des Produktmodells, Steuerung des Informationsflusses zwischen den beteiligten Akteuren
- Projektmanagement:
 - Gliedern und delegieren von Aufgaben
- Arbeitsmanagement:
 - Erfassung und Abbildung aller Änderungen durch Versionen
- Konfigurationsmanagement:
 - Systematische Verwaltung des Konfigurationsmodells durch Überwachen des Lebenszyklus der Produkte

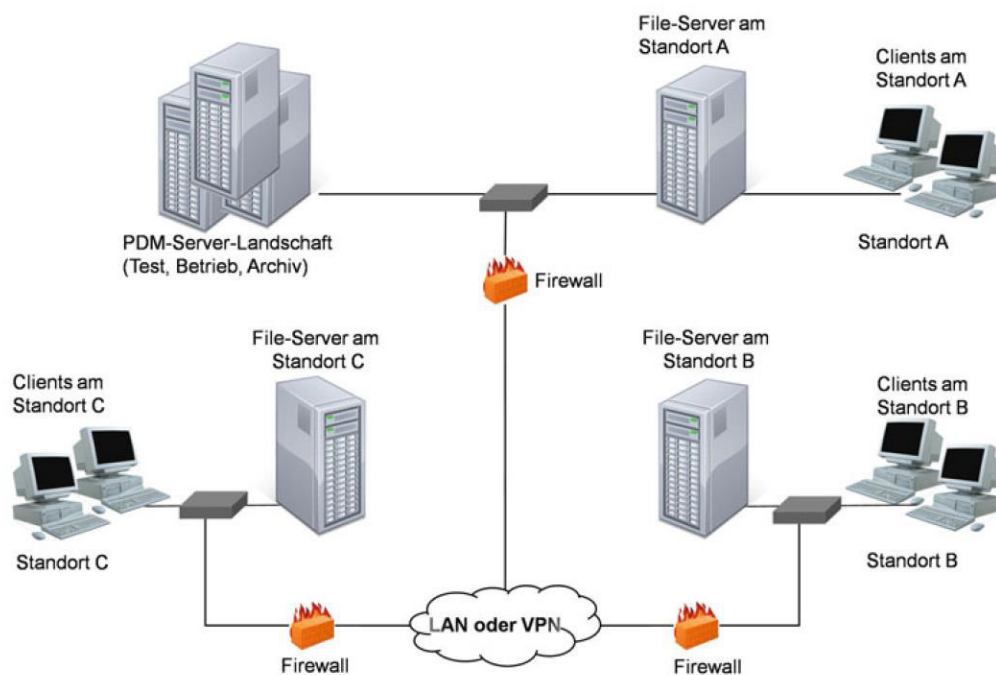
Nach Eigner und Stelzer bezieht sich das PDM nur auf die Entwicklungs- und Planungsphase des Produktlebenszyklus, eine Erweiterung des PDM auf den gesamten Produktlebenszyklus nennt er Produkt-Lifecycle-Management. [26]

Nach Feldhusen et al. steuern PDM-Systeme den Zugriff auf die Produktdaten durch unterschiedliche Personen. Sie basieren auf Datenbanken, in denen die unterschiedlichen Dokumente abgelegt werden. Zusätzlich zu den Dokumenten werden auch beschreibende Informationen, genannt Metadaten, wie Teilenummer, Version, Freigabedatum, Gültigkeitsstatus usw. abgespeichert. Diese Metadaten werden entweder durch die Personen oder das System hinzugefügt.

Die PDM-Systeme basieren in der Regel auf Client/Server-Rechnerarchitekturen, mit einer Datenspeicherung auf zentralen Servern und Client-Rechnern, die über Netzwerke mit den Servern verbunden sind, wie in Abbildung 2.2-12 dargestellt. PDM-Systeme weisen in der Regel folgende Funktionen auf: [11]

- Konfliktfreie Bearbeitung von PDM-Objekten
- Ermöglichen der Betrachtung von Dokumenten und eintragen von Anmerkungen
- Workflow und Prozessmanagement
- Dokumentenmanagement
- Produktstruktur- und Konfigurationsmanagement
- Berechtigungsmanagement

Abbildung 2.2-12: PDM-System, IT-Infrastruktur [11]



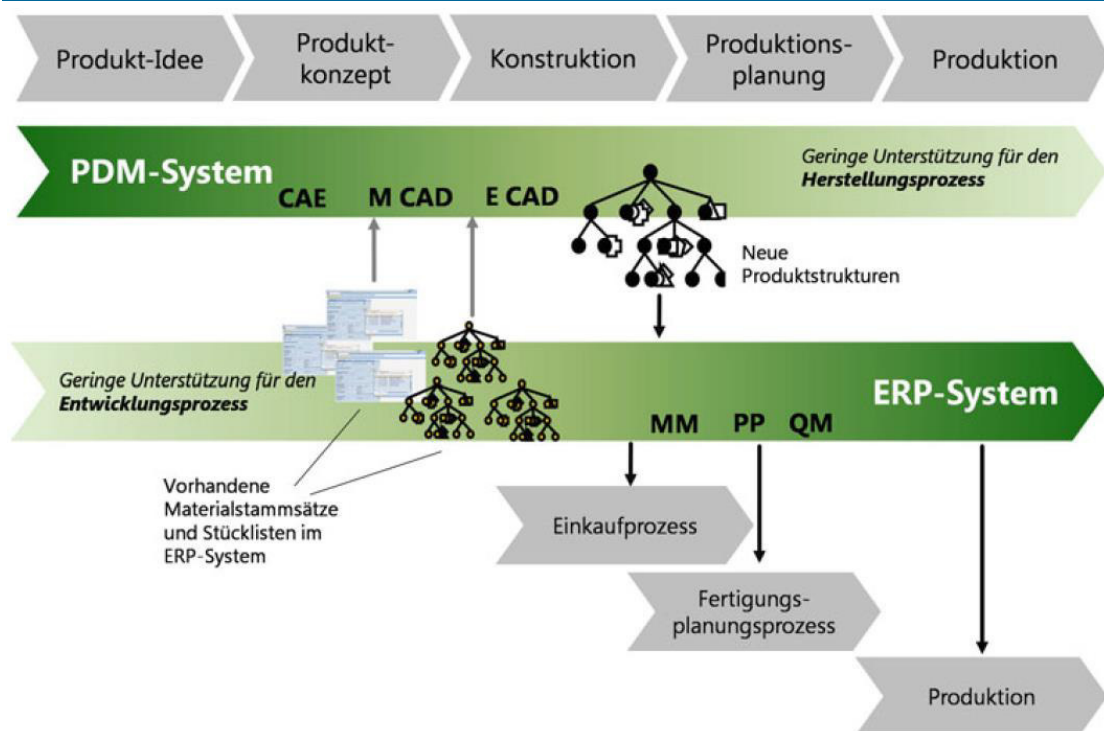
Entgegen der Definition von Eigner und Stelzer unterscheiden Feldhusen et al nicht zwischen PDM- und PLC-Systemen. [11]

In PDM-Systemen werden also alle relevanten Produktdaten abgelegt, die notwendig sind um eindeutige reproduzierbare Produktkonfigurationen zu erzeugen, sowohl aus Funktionssicht, als auch aus Teilesicht.

ERP-Systeme stellen eine Weiterentwicklung der Produktions-Planung und Steuerung (PPS)-Systemen dar und werden zur Unterstützung von Produktionsprozessen, Einkauf, Controlling, Personalwirtschaft, Vertrieb und Service bzw. Instandhaltung verwendet. Die Hauptaufgabe von ERP-Systemen ist sicherzustellen, dass bestellte Produkte zur richtigen Zeit in der richtigen Menge und Qualität zu festgelegten Kosten hergestellt werden. In ihnen werden also gewünschte Produktkonfigurationen mit den bestehenden Transformationsbeziehungen und

Stücklisten, die sie von den PDM-Systemen erhalten, verglichen woraus sich das benötigte Material ableiten lässt. Sollten während der Produktion Änderungen an einem Objekt (z.B. Stückliste) notwendig sein, werden diese ebenfalls im ERP-System dokumentiert. Während der Weiterentwicklung der Produkte benötigen die Entwickler oftmals Zugriff auf die aktuellen und eventuell geänderten Stücklisten und daher Zugriff auf das ERP-System. Das bedeutet ERP und PDM-Systeme sind oftmals sehr eng verknüpft, wie in Abbildung 2.2-13 dargestellt. Diese enge Verknüpfung hat dazu geführt, dass Anbieter von PDM- bzw. ERP-Systemen Module des jeweils anderen in ihre Systeme integriert haben.[11]

Abbildung 2.2-13: Verknüpfung von PDM- und ERP-Systemen [11]



Schuh versteht unter PLM eine unternehmensweiter Verwaltung und Steuerung aller Produkt- und Prozessbezogenen Daten des gesamten Lebenszyklus und somit einer Verschmelzung von PDM- und ERP-Systemen.[5]

2.3 Mathematische Grundlagen

In diesem Kapitel werden die mathematischen Grundlagen und Regeln erläutert, die für das Verständnis der angewendeten Methoden und Werkzeuge notwendig sind.

2.3.1 Zählen

Mittels Zählen wird die Anzahl an Elementen einer endlichen Menge M ermittelt. Für die Problemstellungen dieser Arbeit sind die Sätze 8.1 und 8.2 nach Meinel und Mundhek ausreichend [33]

Sei A eine Menge mit einer endlichen Anzahl $n = \#A$ an Elementen $A = \{a_1, a_2, a_3, a_4, \dots, a_n\}$, B eine Menge mit einer endlichen Anzahl $m = \#B$ an Elementen $B = \{b_1, b_2, b_3, b_4, \dots, b_m\}$ und C die Menge aller Kombinationsmöglichkeiten der Elemente aus A und B so lassen sich die einzelnen Elemente von C als folgendes Kreuzprodukt darstellen:

$$A \times B = \{a_1, a_2, a_3, a_4, \dots, a_n\} \times \{b_1, b_2, b_3, b_4, \dots, b_m\} = \{(a_1, b_1), (a_1, b_2), \dots, (a_n, b_m)\}$$

Für die Anzahl an Elementen von C gilt in weiterer Folge Satz 8.1:

$$\#C = \#(A \times B) = \#A * \#B$$

Sei nun k eine positive natürliche Zahl und $A_1, A_2, A_3, \dots, A_k$ endliche Mengen, dann gilt mit Satz 8.1 folgende Produktregel:

$$\#(A_1 \times A_2 \times A_3 \times \dots \times A_k) = \#A_1 * \#A_2 * \#A_3 * \dots * \#A_k = \prod_{t=1}^k \#A_t$$

2.3.2 Boolesche Algebra und Aussagenlogik

Die in diesem Kapitel beschriebenen Grundsätze orientieren sich an Meinel und Mundheks Mathematischen Grundlagen der Informatik. [33]

Die mathematische Logik oder auch Aussagenlogik bezeichnet das ermitteln der Wahrheitswerte von zusammengesetzten Verbindung aus mehreren Aussagen.

Eine Aussage kann entweder „wahr“ oder „falsch“ sein, aber nie beides zugleich.

Einfache Aussagen können durch logische Operatoren verknüpft werden, wodurch komplexe Aussagen entstehen die im gesamten entweder „wahr“ oder „falsch“ sein können.

Diese Operatoren mit ihren Symbolen sind:

- „und“ bzw. „ \wedge “

Werden zwei Aussagen durch ein „und“ verknüpft, ist die gesamte Aussage nur „wahr“, wenn beide Teilaussagen „wahr“ sind.

- „oder“ bzw. „ \vee “

Werden zwei Aussagen durch ein „oder“ Verknüpft, ist die gesamte Aussage „wahr“, wenn eine der beiden Teilaussagen oder beide Teilaussagen „wahr“ sind.

- „nicht“ bzw. „ \neg “

„nicht“ ist keine Verknüpfung, sondern negiert die nachfolgende Aussage, ist also eine Aussage „wahr“ und mit einem „nicht“ versehen, liefert sie „falsch“ und umgekehrt.

- „Implikation“ bzw. „ \rightarrow “

Eine Implikation besagt, wie aus einer Aussage eine andere Aussage geschlussfolgert werden kann, also aus „wahr“ kann „wahr“ geschlussfolgert werden. Eine Implikation ist nur dann falsch, wenn aus „wahr“ „falsch“ geschlussfolgert wird.

- „Äquivalenz“ bzw. „ \leftrightarrow “

Die Äquivalenz beschreibt die Gleichheit zweier Aussagen. Die Äquivalenz liefert „wahr“, wenn beide Teilaussagen gleich sind, also entweder beide „wahr“ oder beide „falsch“

Eine Darstellungsform der Aussagenlogik ist die sogenannte Wahrheitstabelle, in ihr werden den einfachen Aussagen die Werte „wahr“ und „falsch“ in allen Kombinationsmöglichkeiten als Eingangsgröße zugeordnet, als Ausgangsgröße erhält man die Aussage der verknüpften Aussage. Tabelle 2.3-1 zeigt die Wahrheitstabelle für die Aussagen „x“ und „y“ mit einfacher Verknüpfung über die oben erwähnten Operatoren.

Tabelle 2.3-1: Wahrheitstabelle einfacher, verknüpfter Aussagen

x	y	$\neg x$	$\neg y$	$x \rightarrow y$	$x \leftrightarrow y$	$x \wedge y$	$x \vee y$	$\neg(x \wedge y)$	$\neg(x \vee y)$
w	w	f	f	w	w	w	w	f	f
w	f	f	w	f	f	f	w	w	f
f	w	w	f	w	f	f	w	w	f
f	f	w	w	w	w	f	f	w	w

Das Ergebnis von komplexen Aussage hängt also von den Wahrheitswerten der verknüpften Aussagen und den Verknüpfungen ab.

Verknüpfte Aussagen können ebenfalls wieder mittels Operatoren verknüpft werden wodurch die Komplexität der Aussagen weiter erhöht wird. In Tabelle 2.3-2 ist die Wahrheitstabelle der einfachen Aussagen x, y und z mit den Teilaussagen der komplexen Aussage $(x \wedge y) \wedge (x \vee z) \vee (y \wedge z)$ dargestellt.

Im Zusammenhang mit der Aussagenlogik werden auch die Begriffe Tautologie und Kontradiktion verwendet.

Eine Tautologie ist eine Aussage die immer „wahr“ ist, unabhängig von den Werten ihrer Teilaussagen. Eine Einfache Tautologie ist z.B. die Aussage $(x \vee \neg x)$ da diese Aussage immer „wahr“ ist, egal ob x „wahr“ oder „falsch“ ist.

Eine Kontradiktion ist eine Aussage, die immer „falsch“ ist, unabhängig von den Werten ihrer Teilaussagen. Ähnlich zur einfachen Tautologie lässt sich auch eine einfache Kontradiktion

finden. $(x \wedge \neg x)$ ist immer „falsch“, da eine der beiden Aussagen immer „falsch“ ist und somit die gesamte Aussage nicht „wahr“ werden kann.

Tabelle 2.3-2: Wahrheitstabelle verknüpfter komplexer Aussagen

x	y	z	$(x \wedge y)$	$(x \vee z)$	$(y \wedge z)$	$(x \wedge y) \wedge (x \vee z) \vee (y \wedge z)$
w	w	w	w	w	w	w
w	w	f	w	w	f	w
w	f	w	f	w	f	f
w	f	f	f	w	f	f
f	w	w	f	w	w	w
f	w	f	f	f	f	f
f	f	w	f	w	f	f
f	f	f	f	f	f	f

Komplexe Aussagen können über folgende Äquivalenzen vereinfacht werden:

■ Kommutativität:

$$(x \wedge y) \leftrightarrow (y \wedge x)$$

$$(x \vee y) \leftrightarrow (y \vee x)$$

■ Distributivität:

$$(x \wedge (y \vee z)) \leftrightarrow ((x \wedge y) \vee (x \wedge z))$$

$$(x \vee (y \wedge z)) \leftrightarrow ((x \vee y) \wedge (x \vee z))$$

■ Doppelnegation:

$$\neg(\neg x) \leftrightarrow x$$

■ Tautologieregeln:

$$(x \wedge y) \leftrightarrow x$$

$$(x \vee y) \leftrightarrow y$$

Unter der Voraussetzung, dass y eine Tautologie ist

■ Assoziativität:

$$(x \wedge (y \wedge z)) \leftrightarrow ((x \wedge y) \wedge z)$$

$$(x \vee (y \vee z)) \leftrightarrow ((x \vee y) \vee z)$$

■ Idempotenz:

$$(x \wedge x) \leftrightarrow x$$

$$(x \vee x) \leftrightarrow x$$

■ deMorgans Regeln:

$$\neg(x \wedge y) \leftrightarrow (\neg x) \vee (\neg y)$$

$$\neg(x \vee y) \leftrightarrow (\neg x) \wedge (\neg y)$$

■ Kontradiktionsregeln:

$$(x \wedge y) \leftrightarrow y$$

$$(x \vee y) \leftrightarrow x$$

Unter der Voraussetzung, dass y eine Kontradiktion ist

Zusätzlich zu den oben beschriebenen Äquivalenzen können komplexe Aussagen vereinfacht werden aufgrund der Regel, dass \neg stärker bindet als \wedge und \wedge stärker bindet als \vee , wodurch Klammern weggelassen werden können, was die Aussagen übersichtlicher macht.

Komplexe Aussagen können auch als Formeln bezeichnet werden. Ersetzt man die Werte „wahr“ und „falsch“ durch die booleschen Konstanten „0“ und „1“ befinden wir uns im Bereich der booleschen Algebra. Boolesche Variablen können also nur die Werte „0“ und „1“ annehmen.

Äquivalent zu den Formeln werden boolesche Variablen, die durch Operatoren miteinander verknüpft sind boolesche Ausdrücke genannt. Abhängig von der Anzahl k an booleschen Variablen, die entweder „0“ oder „1“ einnehmen können, werden die Ausdrücke k -stellige boolesche Ausdrücke genannt.

Für boolesche Ausdrücke gelten dieselben Rechenregeln bzw. Äquivalenzen wie für die Formeln der Aussagenlogik, die Operatoren „und“, „oder“ und „nicht“ bekommen jedoch unterschiedliche Symbole.

- „und“ bzw. „ \wedge “ $\rightarrow \cdot$
- „oder“ bzw. „ \vee “ $\rightarrow +$
- „nicht“ bzw. „ \neg “ $\rightarrow \bar{}$

Die aussagenlogische Formel $(x \wedge y) \wedge (x \vee z) \vee \neg(y \wedge z)$ kann also transformiert werden zu dem 3-stelligen booleschen Ausdruck $(x \cdot y) \cdot (x + z) + \overline{(y \cdot z)}$.

In der booleschen Algebra gilt $1 + 1 = 1 \neq 2$, das bedeutet die boolesche Summe darf nicht mit der arithmetischen Summe verwechselt werden.

Aufgrund der Praxis in der Automobilindustrie werden in dieser Arbeit für die booleschen Operatoren die folgenden, von der Norm abweichenden, Symbole verwendet:

- „+“ für „AND“ anstelle von „ \wedge “ bzw. „ \cdot “
- „/“ für „OR“ anstelle von „ \vee “ bzw. „+“
- „-“ für „NOT“ anstelle von „ \neg “ bzw. „ $\bar{}$ “

Der boolesche Ausdruck $(x \cdot y) \cdot (x + z) + \overline{(y \cdot z)}$ wird somit zu $(x + y) + (x/z) / -(y + z)$.

Dieser Problematik der abweichenden Operatoren kann man begegnen indem entweder geeignete Algorithmen verwendet werden, oder die Datensätze entsprechend umgeformt werden.

3 Verwendete Methoden

Im folgenden Kapitel werden die Methoden und Herangehensweisen erläutert, die sich für die im Zuge dieser Diplomarbeit durchgeführten Analysen, als nützlich erwiesen haben bzw. notwendig waren.

3.1 Lösen der booleschen Ausdrücke

Wie in Kapitel 2.2.3 beschrieben, stellen die Codebedingungen für das Konfigurationsmanagement bzw. das Auflösen der Codeketten in Stücklisten, boolesche Funktionen dar, daraus folgt, dass das Lösen von booleschen Funktionen eine unerlässliche Methode für die Dokumentationsbasierte Komplexitätsanalyse darstellt.

An dieser Stelle sei, für das Verständnis dieser Arbeit, nochmal darauf hingewiesen, dass für die booleschen Operationen folgende Notation verwendet wird (vgl. Kapitel 2.3.2):

Zum Vergleich ist in Tabelle 3.1-1 eine Codebedingung in allen vier Notationen dargestellt

Tabelle 3.1-1:Gegenüberstellung Notationen der booleschen Operatoren

Notation	Codebedingung
Diese Arbeit	$CD1 + CD2/CD3 + -(CD4/-CD5)$
Text	$CD1 \text{ AND } CD2 \text{ OR } CD3 \text{ AND NOT } (CD4 \text{ OR NOT } CD5)$
Logische Operatoren	$CD1 \wedge CD2 \vee CD3 \wedge \neg (CD4 \vee \neg CD5)$
Mathematische Operatoren	$CD1 \cdot CD2 + CD3 \cdot \overline{(CD4 + \overline{CD5})}$

Das Ergebnis aufgelöster boolescher Funktionen ist immer „0“ oder „1“ bzw. „falsch“ oder „wahr“

In einem ersten Schritt muss die Codebedingung in eine lösbare boolesche Funktion umgewandelt werden, dazu wird die Codebedingung mit der zu überprüfenden Codekette verglichen, dabei muss unterschieden werden zwischen einer vollständigen Codekette (beschreibt ein vollständiges, gebautes Produkt) und einer Teilkette.

Vergleicht man eine Codebedingung mit einer vollständigen Codekette, wird überprüft ob die Codebedingung mit dieser Kette vollständig erfüllt wird. Für diese Überprüfung wird jeder Code in der Codebedingung, welcher in der zu vergleichenden Codekette vorkommt, durch „1“, jeder nicht vorkommende Code durch „0“ ersetzt.

Die Codebedingung aus Tabelle 3.1-1 verglichen mit der Codekette „CD1;CD3;CD4“, unter der Annahme, dass es sich um eine vollständige Codekette handelt, ergibt folgendes Ergebnis:

$$„CD1 + CD2/CD3 + -(CD4/-CD5)“ \rightarrow „1 + 0/1 + -(1/-0)“ \rightarrow „0“$$

Wird die Codebedingung mit einer unvollständigen Codekette verglichen, werden, analog zur vollständigen Codekette, alle vorkommenden Code durch „1“, alle nicht vorkommenden jedoch durch einen Platzhalter „X“ ersetzt, der immer den optimalen Wert einnimmt um die Funktion lösbar zu machen. Dieser Platzhalter kann sowohl den Wert „0“ als auch „1“ annehmen.

Überprüft man nun das obige Beispiel, unter der Annahme, dass es sich bei „CD1;CD3;CD4“ um eine Teilkette handelt kommt man zu folgendem Ergebnis:

$$„CD1 + CD2/CD3 + -(CD4/-CD5)“ \rightarrow „1 + X/1 + -(1/-X)“$$

Der Term „ $1 + -(1/-X)$ “ wird aufgrund der „1“ im negierten Klammersausdruck immer „0“ ergeben, egal welchen Wert „X“ annimmt, der Term „ $1 + X$ “ nimmt jedoch „1“ an, wenn „X“ „1“ annimmt, somit gilt:

„ $1 + X/1 + -(1/-X)$ “ \rightarrow „1“ das bedeutet die Codebedingung ist mit der Teilkette theoretisch erfüllbar.

Nachdem beim Vergleich einer booleschen Funktion mit einer Teilkette immer das Optimum gewählt wird, folgt unmittelbar, dass die Funktion nur dann „falsch“ werden kann, wenn in ihr Code mit negativer Beeinflussung (vgl. Kapitel 3.2) vorhanden und diese auch in der Codekette sind.

3.2 Einflussanalyse der booleschen Ausdrücke

Das Ziel der Einflussanalyse der booleschen Funktionen ist es, herauszufinden wie ein bestimmter Term bzw. Code das Ergebnis der booleschen Funktion beeinflusst.

Betrachtet man die einfache boolesche Funktion $CD1 + CD2$ so ist diese nur „wahr“, wenn CD1 und CD2 „wahr“ sind. Das bedeutet umgekehrt, wenn CD1 oder CD2 „falsch“ sind ist auch die boolesche Funktion „falsch“, CD1 und CD2 werden somit als positiv bestimmend definiert, da die Code „wahr“ sein müssen um als Ergebnis „wahr“ zu bekommen

Betrachtet man nun die boolesche Funktion $CD3/CD4$ ist diese „wahr“, wenn CD3 oder CD4 „wahr“ sind. Sind CD3 oder CD4 „falsch“, ist das Ergebnis der booleschen Funktion abhängig von dem Wert des anderen Codes, das bedeutet das Ergebnis ist nicht zwingend „falsch“. CD3 und CD4 werden somit als positiv beeinflussend definiert, da diese ein „wahr“ als Ergebnis liefern, wenn sie „wahr“ sind, aber nicht zwingend ein „falsch“ liefern, wenn sie „falsch“ sind.

Kombiniert man nun die beiden booleschen Funktionen zu $CD1 + CD2 + (CD3/CD4)$ sind also CD1 und CD2 in dieser Funktion positiv bestimmend, CD3 und CD4 positiv beeinflussend

Die boolesche Funktion $-CD5 + -CD6$ liefert „wahr“ als Ergebnis, wenn sowohl CD5 als auch CD6 „falsch“ sind. Ist einer der beiden Code „wahr“, ist das Ergebnis immer „falsch“. Die Code CD5 und CD6 werden also als negativ bestimmend definiert, da diese immer zu „falsch“ führen, wenn sie „wahr“ sind.

$-CD7/-CD8$ ist „wahr“, wenn CD7 oder CD8 „falsch“ sind. Ist einer der beiden Code „wahr“ ist das Ergebnis abhängig vom Wert des anderen Codes. CD7 und CD8 sind also negativ beeinflussend, da sie zu dem Ergebnis „falsch“ führen können, aber nicht zwingend.

In Tabelle 3.2-1 ist die Wahrheitstabelle der booleschen Funktion $CD1 + -CD2 + (CD3/-CD4)$ mit „0“ als „falsch“ und „1“ als „wahr“ dargestellt.

Tabelle 3.2-1: Wahrheitstabelle von $CD1 + -CD2 + (CD3/-CD4)$

CD1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
CD2	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
CD3	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
CD4	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
$CD1 + -CD2 + (CD3/-CD4)$	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0

In Tabelle 3.2-1 ist zu erkennen, dass das Ergebnis nur dann „1“ ist, wenn $CD1 = 1$ und $CD2 = 0$ da diese positiv bzw. negativ bestimmend sind. $CD3$ und $CD4$ sind positiv bzw. negativ beeinflussend, daher ist das Ergebnis des OR Terms nur dann „0“, wenn beide Code ungünstige Werte ($CD3 = 0$ und $CD4 = 1$) einnehmen.

Für die weitergehenden Analysen werden den bestimmenden und beeinflussenden Coden die werte 2 und -2 (bestimmend) bzw. 1 und -1 (beeinflussend) zugeordnet. Tabelle 3.2-2 zeigt die Beschreibung und zugeordneten Werte der Einflussarten.

Tabelle 3.2-2: Beschreibung der Einflussarten von Coden auf boolesche Funktionen

Einflussart	Beschreibung	Wert
Positiv bestimmend	Muss „wahr“ sein für Funktionsergebnis „wahr“	2
Positiv beeinflussend	Muss nicht „wahr“ sein für Funktionsergebnis „wahr“, kann aber zu „wahr“ führen	1
Negativ beeinflussend	Muss nicht „falsch“ sein für Funktionsergebnis „falsch“, kann aber zu „falsch“ führen	-1
Negativ bestimmend	Führt bei „wahr“ zwingend zu einem Funktionsergebnis „falsch“	-2

Die Code jeder Codebedingung können nach den oben definierten Einflussarten analysiert werden und entsprechend Tabelle 3.2-3 aufgeschlüsselt werden.

Beeinflussende Code können nicht alleine ausschlaggebend für das Ergebnis einer booleschen Funktion sein, sie benötigen immer mindestens einen zweiten Code.

Tabelle 3.2-3: Codebedingung mit beeinflussenden Coden und Einflussart

Codebedingung	Code	Einflussart
$CD1 + -CD2 + (CD3/-CD4)$	CD1	2
$CD1 + -CD2 + (CD3/-CD4)$	CD2	-2
$CD1 + -CD2 + (CD3/-CD4)$	CD3	1
$CD1 + -CD2 + (CD3/-CD4)$	CD4	-1

Tabelle 3.2-4 zeigt eine ausgeführte Analyse für Codebedingungen aus Tabelle 4.1-3 und Tabelle 4.1-7

Tabelle 3.2-4: Ausgeführte Einflussanalyse der booleschen Ausdrücke

Verweis	Boolesche Funktion	Code	Einflussart
Tabelle 4.1-3 Zeile 1	$-(CD2/CD3)$	CD2	-2
Tabelle 4.1-3 Zeile 1	$-(CD2/CD3)$	CD3	-2
Tabelle 4.1-7 Zeile 2	CD2	CD2	2
Tabelle 4.1-7 Zeile 9	$CD8 + -CD4 + (CD6/CD7)$	CD8	2
Tabelle 4.1-7 Zeile 9	$CD8 + -CD4 + (CD6/CD7)$	CD4	-2
Tabelle 4.1-7 Zeile 9	$CD8 + -CD4 + (CD6/CD7)$	CD6	1
Tabelle 4.1-7 Zeile 9	$CD8 + -CD4 + (CD6/CD7)$	CD7	1

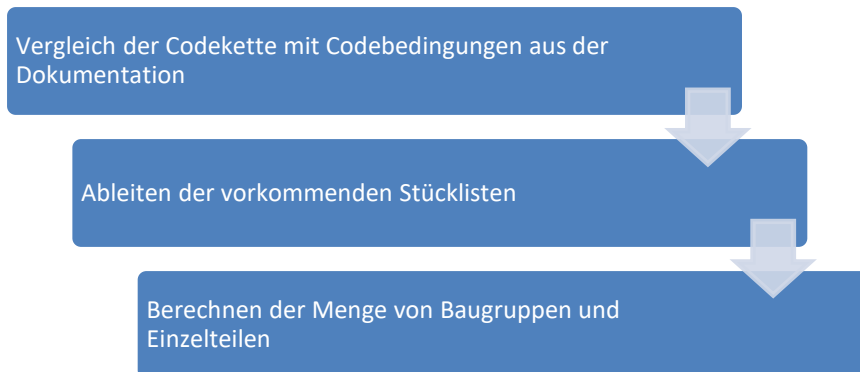
3.3 Produktauflösung

Ziel der Produktauflösung ist es, aus einer Produktkonfiguration, über die Codebedingungen, die dadurch ausgelösten Stücklisten und in weiterer Folge Einzelteile zu erhalten, aus denen ein Produkt besteht.

Es wird also eine Produktkonfiguration bzw. Codekette mit den Bedingungen aus der Dokumentation (vgl. Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**) verglichen. Alle Bedingungen, die nach dem lösen der booleschen Funktionen als Ergebnis „1“ liefern, führen folglich zu den Stücklisten, aus denen sich das Produkt zusammensetzt. Aus diesen Stücklisten lassen sich, mit den Informationen aus der Produktstruktur, die Mengen der Baugruppen bzw. Einzelteile ableiten.

Der Ablauf der Produktauflösung ist in Abbildung 3.3-1 dargestellt.

Abbildung 3.3-1: Methodenablauf Produktauflösung



Es gilt zu beachten, dass bei der Berechnung der Baugruppen- bzw. Einzelteilmengen darauf geachtet werden muss, in welcher Form die Produktstruktur vorliegt und diese entweder durch ein Pre-Processing umgebaut werden muss, oder entsprechende Algorithmen verwendet werden müssen, um auch tatsächlich auf die richtige Menge zu kommen.

Hat man Informationen bezüglich Kosten bzw. Gewicht der Einzelteile, lassen sich, unter der Voraussetzung einer lückenlosen Dokumentation, über die Produktauflösung die Gesamtkosten bzw. das Gesamtgewicht für das Produkt berechnen.

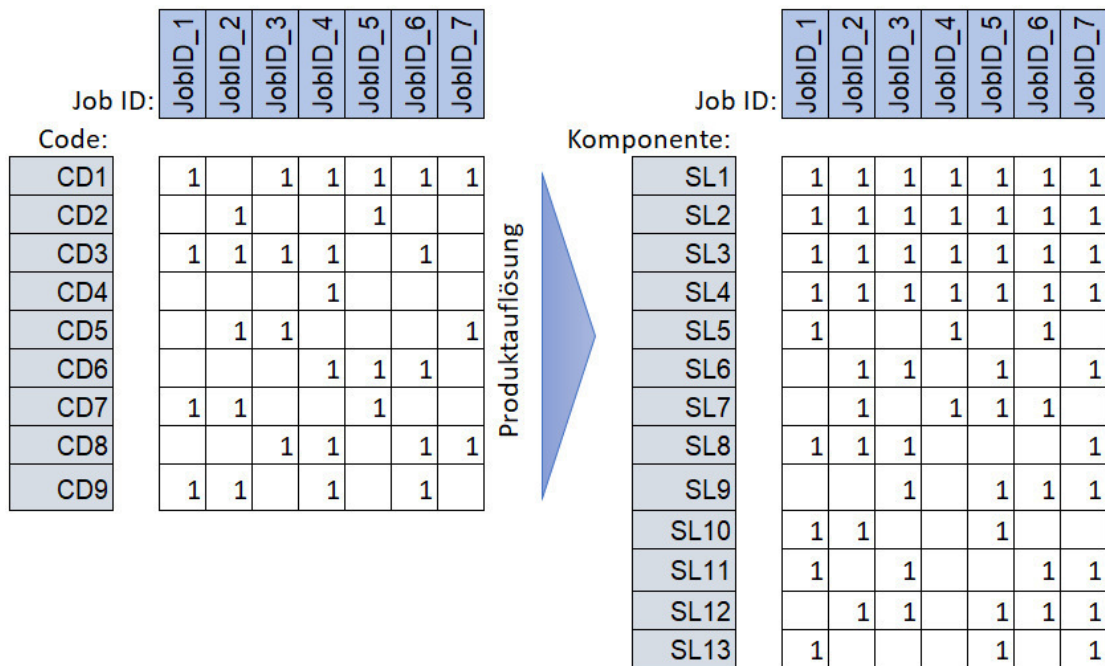
3.4 Domain Mapping Matrices

Domain Mapping Matrices (DMM) sind Matrizen in denen die unterschiedliche Elemente zweier Domänen (z.B. Code, Eigenschaften, Komponenten, Produktvarianten, etc.) und die Relationen in denen sie stehen, dargestellt werden können.[34]

Die in dieser Arbeit verwendeten Vertreter der DMM sind die Produkt-Code-DMM bzw. die Produkt-Komponenten-DMM. Die Produkt-Code-DMM stellt also dar, welche Code in einem Produkt vorkommen, die Produkt-Komponenten-DMM lässt sich durch Auflösen der Produkte analog zu der Produkt-Code-DMM erstellen und ist eine Abbildung dessen, welche Komponenten oder Stücklisten in einem Produkt vorkommen. Abbildung 3.4-1 zeigt eine Produkt-Code-DMM (links) und eine Produkt-Komponenten-DMM (rechts).

Die Produkt-Komponenten-DMM kann sowohl auf Stücklistenbasis als auch auf Einzelteilbasis dargestellt werden.

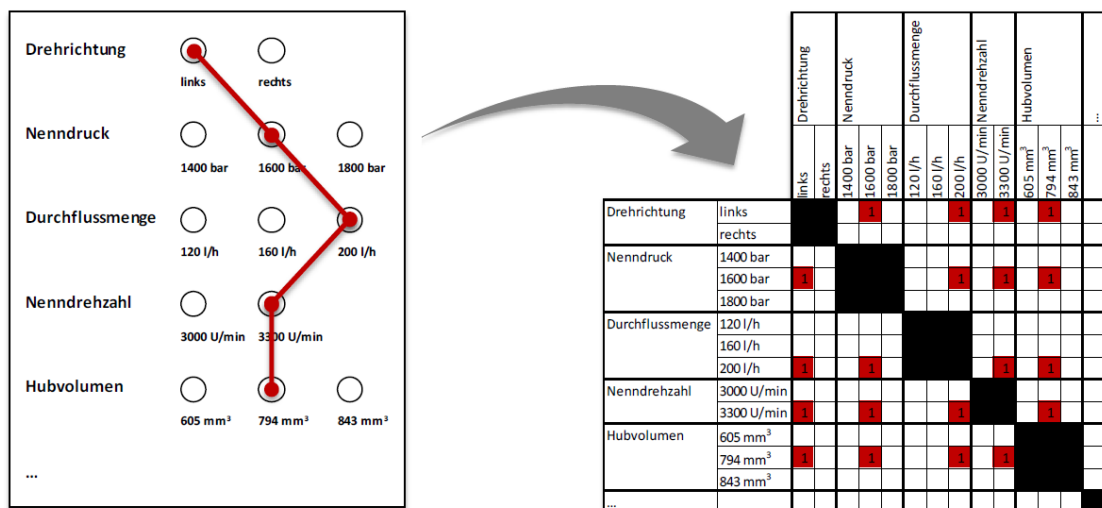
Abbildung 3.4-1: Produkt-Code-DMM und Produkt-Komponenten-DMM



3.5 Design-Structure-Matrices

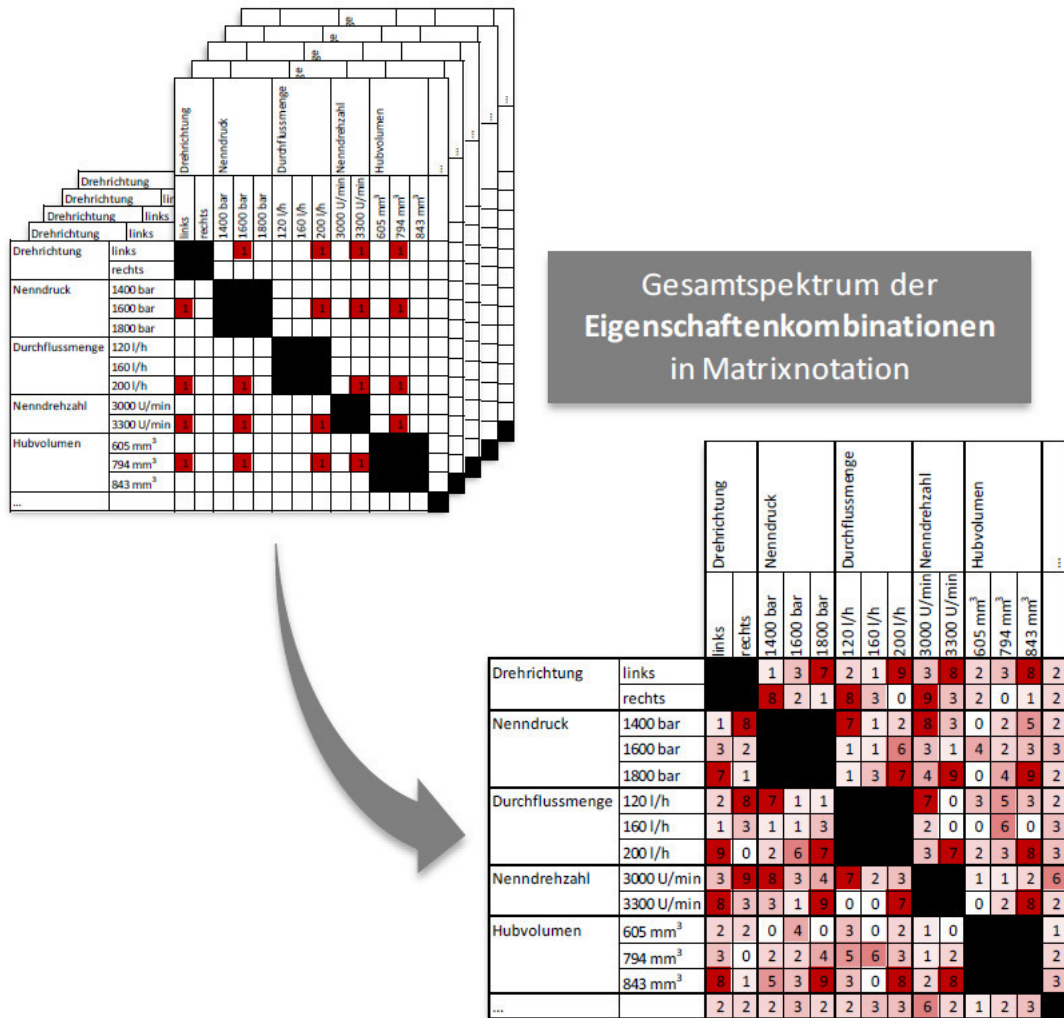
Design-Structure-Matrices (DSM) sind Matrizen, die zur Abbildung der Elemente einer Domäne und die Beziehungen untereinander dargestellt werden können. Kesper verwendet sie z.B. um einzelne Eigenschaftenkombinationen in Matrixschreibweise darzustellen, wie in Abbildung 3.5-1 dargestellt. [1]

Abbildung 3.5-1: DSM zur Darstellung von Eigenschaftenkombinationen [1]



Für weitere Analysen verwendet er aggregierte DSMs indem er die DSMs der einzelnen Varianten aufsummiert, wie in Abbildung 3.5-2 dargestellt.

Abbildung 3.5-2: Aggregierte DSM nach Kesper [1]



Da es sich bei der Eigenschaften-DSM um eine symmetrische Matrix handelt, die sich aus einem Produkt-Eigenschaft Vektor ableiten lässt, ist es hinreichend die Produkteigenschaften-DMM mit der transponierten Produkt-Eigenschaften-DMM zu multiplizieren, um die aggregierte Eigenschaften-DSM zu erhalten.

Der Beweis dafür sieht wie folgt aus:

Für eine Produktvariante a_m mit n möglichen Eigenschaften, dargestellt in Vektorform gilt:

$$a_m = \begin{pmatrix} a_{1,m} \\ a_{2,m} \\ \vdots \\ a_{n,m} \end{pmatrix} \text{ dabei erhält jede vorhandene Eigenschaft den Wert 1, jede nicht vorhandene den Wert 0. Mit } n \text{ möglichen Eigenschaften ergibt sich daraus die } n \times n \text{ Eigenschaften DSM:}$$

$$DSM_m = \begin{pmatrix} a_{1.1,m} & a_{1.2,m} & \cdots & a_{1.n,m} \\ a_{2.1,m} & a_{2.2,m} & \cdots & a_{2.n,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n.1,m} & a_{n.2,m} & \cdots & a_{n.n,m} \end{pmatrix} \text{ für die Werte gilt } a_{x,y,m} = a_{x,m} * a_{y,m} \text{ und } a_{x,y,m} = a_{y,x,m}$$

Die Summe berechnet sich also zu

$$\sum_{t=1}^m DSM_t = \begin{pmatrix} \sum_{t=1}^m a_{1.1,t} & \sum_{t=1}^m a_{1.2,t} & \cdots & \sum_{t=1}^m a_{1.n,t} \\ \vdots & \sum_{t=1}^m a_{2.2,t} & \cdots & \sum_{t=1}^m a_{2.n,t} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{t=1}^m a_{1.n,t} & \sum_{t=1}^m a_{2.n,t} & \cdots & \sum_{t=1}^m a_{n.n,t} \end{pmatrix}$$

Für ein Element ergibt sich die Summe zu $\sum_{t=1}^m a_{x,y,t} = a_{x,y,1} + a_{x,y,2} + a_{x,y,3} + \cdots + a_{x,y,m}$ und in weiterer Folge zu

$$\sum_{t=1}^m a_{x,y,t} = a_{x,1} * a_{y,1} + a_{x,2} * a_{y,2} + a_{x,3} * a_{y,3} + \cdots + a_{x,m} * a_{y,m}$$

Betrachtet man nun die Produkt-Eigenschaften-DMM in der die Spalten 1 bis m den Produktvektoren entsprechen und multipliziert diese mit der transponierten DMM

$$\begin{aligned} DMM * DMM^T &= \begin{pmatrix} a_{1.1} & a_{1.2} & \cdots & a_{1.m} \\ a_{2.1} & a_{2.2} & \cdots & a_{2.m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n.1} & a_{n.2} & \cdots & a_{n.m} \end{pmatrix} * \begin{pmatrix} a_{1.1} & a_{2.1} & \cdots & a_{n.1} \\ a_{1.2} & a_{2.2} & \cdots & a_{n.2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1.m} & a_{2.m} & a_{3.m} & a_{n.m} \end{pmatrix} = \\ &= \begin{pmatrix} \sum_{t=1}^m a_{1.t} * a_{1.t} & \sum_{t=1}^m a_{1.t} * a_{2.t} & \cdots & \sum_{t=1}^m a_{1.t} * a_{n.t} \\ \sum_{t=1}^m a_{2.t} * a_{1.t} & \sum_{t=1}^m a_{2.t} * a_{2.t} & \cdots & \sum_{t=1}^m a_{2.t} * a_{n.t} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{t=1}^m a_{n.t} * a_{1.t} & \sum_{t=1}^m a_{n.t} * a_{2.t} & \cdots & \sum_{t=1}^m a_{n.t} * a_{n.t} \end{pmatrix} \end{aligned}$$

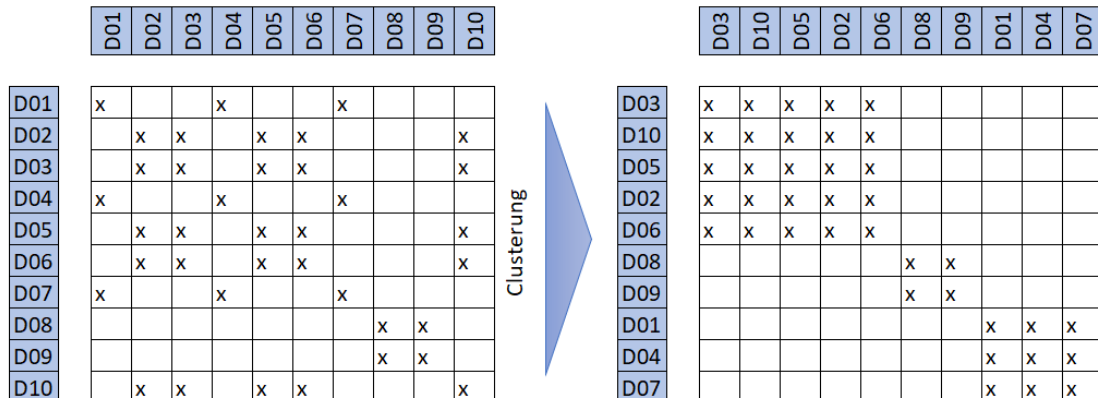
Ein Element der berechneten Matrix ergibt sich zu:

$$\sum_{t=1}^m a_{x,t} * a_{y,t} = a_{x,1} * a_{y,1} + a_{x,2} * a_{y,2} + a_{x,3} * a_{y,3} + \cdots + a_{x,m} * a_{y,m} = \sum_{t=1}^m a_{x,y,t}$$

3.6 Clusteranalyse

Die Clusteranalyse wird dazu verwendet, in einer besetzten Matrix, durch verschieben der Spalten und Zeilen, möglichst große Cluster um die Hauptdiagonale zu erhalten um z.B. Gemeinsamkeiten der Domänen der Matrix zu erkennen wie in Abbildung 3.6-1 dargestellt. (vgl. [35])

Abbildung 3.6-1: Clusterbildung



Möchte man eine Clusteranalyse einer DSM durchführen, müssen zuerst Grenzwerte definiert werden, wird dieser Grenzwert in einer Zelle der Matrix überschritten, erhält die Zelle z.B. den Wert „1“ alle anderen Zellen erhalten den Wert „0“. Anschließend kann die DSM geclustert werden

3.7 ABC-Analyse

In der ABC-Analyse, bzw. der Pareto-Analyse werden Teilmengen einer Gesamtmenge, bezogen auf ihre Eigenschaft (z.B. Kosten, Umsatzmenge, Gewicht, usw.), so geordnet, dass sich daraus drei Teilmengen ergeben:[19]

- Teilmenge A hat den größten Anteil an der Eigenschaft
- Teilmenge B einen geringeren Anteil an der Eigenschaft
- Teilmenge C den geringsten Anteil an der Eigenschaft

Die ABC-Analyse kann sowohl in Säulenform (Abbildung 3.7-1), als auch in aggregierter Form (Abbildung 3.7-2) und mit absoluten oder relativen Werten dargestellt werden.

Abbildung 3.7-1: ABC-Analyse in Säulenform

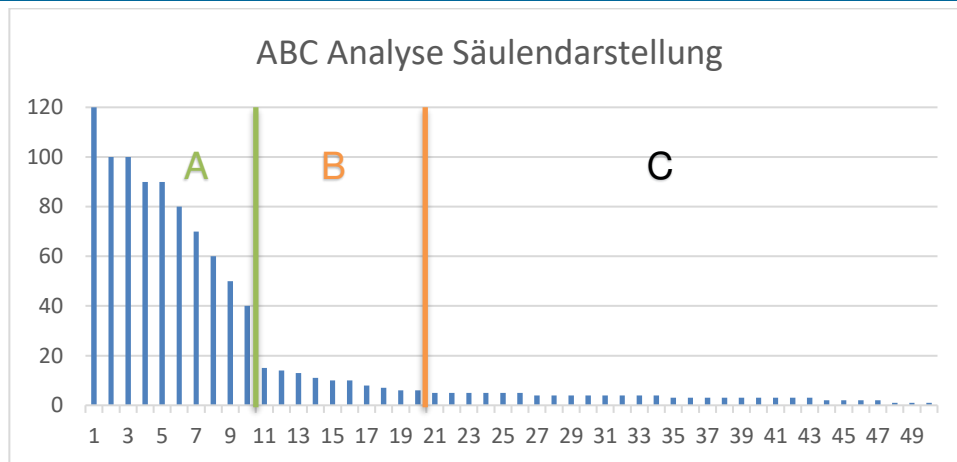
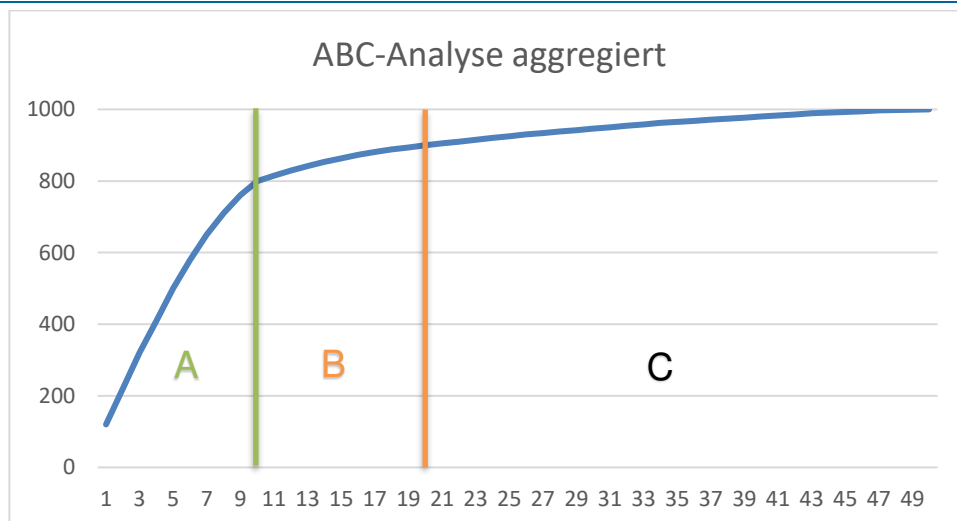


Abbildung 3.7-2: ABC-Analyse in aggregierter Form



Nach welchen Kriterien die Teilmengen eingeteilt werden muss individuell entschieden werden, eine häufige Regel, die auch für diese Beispiele gewählt wurde, ist die 80/20-Regel, also z.B. 80% der Teile machen 20% der Kosten aus, oder 20% der Produkte verursachen 80% des Umsatzes.

Würde man für ein hochvariantes Produktionsprogramm, in dem kein Produkt gleich ist, eine ABC-Analyse durchführen, ergäbe sich daraus eine Linie mit Steigung = 1 und eine Unterteilung nach den Kategorien A, B und C würde nicht funktionieren.

4 Eingangsgrößen

Im folgenden Kapitel werden die benötigten Eingangsgrößen, für die in Kapitel 4 beschriebenen Methoden und Tools, diskutiert. Zu Beginn soll vermittelt werden, welche Informationen zwingend notwendig sind um die Analysen durchzuführen.

Anschließend wird erläutert welche Informationen mittels Pre-Processing aus den Daten gewonnen werden können und welche Methoden vor den Analysen für größere Datensätze durchgeführt werden sollten, da diese häufiger verwendet werden.

4.1 Diskussion des Bedarfs

Wie an der Möglichkeit des Ableitens einer Mengenübersichtsstückliste aus der Strukturstückliste in Kapitel 2.2.2 gezeigt, kann man allgemein feststellen, je strukturierter die Daten sind, desto mehr Analysen sind möglich. Diese Ableitungen aus strukturierten Daten sind jedoch oft zeitaufwendig und keinesfalls reversibel. Wenn also die Möglichkeit besteht aus den Systemen sowohl strukturierte als auch kumulierte Daten abzurufen verringert es nicht nur den Zeitaufwand, sondern hilft auch bei dem Verständnis der Daten.

Eine weitere Information, die speziell für die Wahl von Betrachtungszeiträumen relevant ist, ist die Gültigkeit der Daten, genauer der Einsatztermin und Endtermin der Gültigkeit, da es verfälschte Ergebnisse liefern kann, wenn das Bauprogramm aus einem bestimmten Zeitraum anhand einer noch nicht, oder nicht mehr gültigen Dokumentation analysiert wird. Diese Information ist also für das Konfigurationsmanagement, die Stücklisten und Codeinformationen optional für zusätzliche Analysen. Ein Löschen veralteter Daten ist aufgrund der Dokumentationspflicht nicht zulässig.

4.1.1 Produktionsprogramm

Das Produktionsprogramm muss folgende Informationen enthalten:

- Eindeutige Identifikationsmöglichkeit des Fahrzeuges für den Primärschlüssel:
 - Es bieten sich eine Auftragsnummer, Job-ID oder die Fahrgestellnummer an. Aus der Auftragsnummer bzw. der Job-ID lässt sich oft der Zeitraum der Fertigstellung ablesen, aus der Fahrgestellnummer das Baumuster.
- Beschreibung des Fahrzeuges:
 - In der Praxis geschieht dies meist durch eine Codekette (vgl. Kapitel 2.2.2) oder einer Codekette in Kombination mit dem Baumuster. Das Baumuster ist dabei in einem extra Feld, oder der FIN zu finden.

Neben den genannten Pflichtfeldern des Bauprogramms eröffnen die folgenden Zusatzinformationen weitere Analysemöglichkeiten. Können die Informationen nicht direkt aus den Dokumentationssystemen bezogen werden, sind einige von Ihnen aus anderen ableitbar, dies ist jedoch mit einem erheblichen Mehraufwand verbunden und setzt bereits ein Verständnis für die Daten voraus, wie in Kapitel 4.2 beschrieben

- Werkskennung:

Eine Werkskennung liefert zusätzliche Information darüber, in welchem Werk das Fahrzeug produziert wurde und ermöglicht es werkspezifische Analysen durchzuführen. Diese Informationen können auch in der FIN enthalten sein (vgl. Kapitel 2.2.4).

- Länderkennung:

Kennzeichnet für welches Land bzw. welche Region das Fahrzeug gebaut wurde und ermöglicht nach Regionen gegliederte Analysen. Dies wird besonders bei der Paketbildung notwendig.

In der Codekette sind oft auch Informationen über das Zielland zu finden, da unterschiedliche Regionen verschiedene behördliche Auflagen, z.B. Warnhinweise für Airbags oder Reflektoren in der Frontschürze, haben. Diese unterschiedlichen Stücklisten und Montageanweisungen werden über Code bzw. Codebedingungen gesteuert

- Lenkungsausführung:

Informationen darüber, ob es sich um einen Linkslenker (LL) oder Rechtslenker (RL) handelt, ermöglichen es einerseits nach den technischen Ausführungen, andererseits nach den Regionen, ähnlich der Länderkennung, zu analysieren.

Diese Informationen können auf jeden Fall aus der Codekette und unter Umständen aus dem Baumuster bzw. der Fahrgestellnummer abgeleitet werden

- Produktionsdatum:

Für die Betrachtung unterschiedlicher Zeithorizonte ist ein Produktionsdatum bzw. Produktionszeitraum notwendig. Sollten diese Informationen nicht vorhanden sein, kann man aus der FIN das Produktions- bzw. Modelljahr ableiten.

In Tabelle 4.1-1 ist ein möglicher Aufbau der Dokumentation eines Bauprogramms dargestellt. Dieser Aufbau ist bereits in idealisierter Form und enthält alle Informationen um die in Kapitel 5 beschriebenen Analysen durchzuführen.

Da diese idealisierte Form nicht immer direkt aus den Dokumentationssystemen ausgeleitet werden kann, wird in Kapitel 4.2 darauf eingegangen, wie man aus den Minimalangaben möglichst viel ableiten kann.

Tabelle 4.1-1: Mögliche Dokumentation eines Produktionsprogramms

Job ID	FIN	BM	Codekette	Werk	Zielland	LKG	Prod.-Dat
1607345	WWW004551GA012933	4551	CD1;CD4;CD7;...	11	DE	LL	11.07.2016
1607346	WWW004561GA012934	4561	CD2;CD5;CD7;...	11	US	LL	11.07.2016
1608256	WWW005562GB012935	5562	CD1;CD5;CD8;...	12	UK	RL	12.08.2016

4.1.2 Codeinformationen

Die Codeinformationen sind für das Verständnis der Analysen unerlässlich und liefern zusätzliche Informationen für unterschiedliche Betrachtungsmengen.

- **Code:**

Der Code dient im Datensatz als Primärschlüssel

- **Codegruppe:**

Die Information, zu welcher Codegruppe ein Code gehört, ermöglicht es nach bestimmten Codegruppen, z.B. nach Motoren oder Radständen zu analysieren.

- **Benennung der Code bzw. der Codegruppen:**

Während die Algorithmen mit den Codes rechnen, benötigt man die Benennungen, um die Ergebnisse bzw. die Bedienung für den Benutzer zu übersetzen, da dieser die Analysen nicht nach Codes auswählt, sondern nach den Benennungen. Es wird also z.B. nicht nach „CD1“ und „CD8“ analysiert, sondern nach „Motortyp 1“ und „Hinterradantrieb“.

Tabelle 4.1-2 zeigt einen Ausschnitt einer möglichen Codeliste.

Tabelle 4.1-2: Beispiel für eine Codeauflistung

Code	Codegruppe	Code Ben	Codegruppe Ben	Gültig ab	Gültig bis
CD1	000102	Motortyp 1	Motor	01.01.2016	31.12.2100
CD2	000102	Motortyp 2	Motor	01.01.2016	31.12.2100
CD3	000102	Motortyp 3	Motor	01.01.2016	31.12.2100
CD4	002104	Manuell	Getriebe	01.06.2008	31.05.2016
CD5	002104	Wandlerautomatik	Getriebe	01.06.2010	31.05.2016
CD6	002104	Doppelkupplung	Getriebe	01.06.2016	31.12.2100
CD7	002105	Vorderradantrieb	Antrieb	01.06.2014	31.12.2100
CD8	002105	Hinterradantrieb	Antrieb	01.06.2008	31.05.2016
CD9	002105	Allradantrieb	Antrieb	01.06.2016	31.12.2100

4.1.3 Produktarchitektur

Die Dokumentation der Produktarchitektur ermöglicht es, aus dem Bauprogramm einerseits den Bedarf an Stücklisten und in weiterer Folge an Komponenten abzuleiten, andererseits kann man Baubarkeitsprüfungen durchführen.

4.1.3.1 Konfigurationsmanagement

Verknüpfung der Code durch boolesche Operatoren AND(+), OR(/), NOT(-) und ANDNOT (+-)

Es gibt unterschiedliche Arten von Bedingungsinformationen, welche auch in unterschiedlichen Datensätzen abgelegt werden können:

- Baubare Kombinationen, dies beinhaltet triviale Bedingungen wie z.B. es darf nur ein Motor verbaut werden, jedoch auch komplexe Bedingungen, die von der Entwicklung aufgrund technischer Einschränkungen, oder vom Vertrieb um die Variantenvielfalt zu reduzieren, vorgegeben werden. Bei diesen Kombinationen kann es sich sowohl um Codeketten, als auch um Baumuster-Codeketten handeln. Diese Informationen sind notwendig um die Baubarkeit eines Produktes zu überprüfen, bzw. sicherzustellen
- Bestimmte Code können durch Codeketten ausgelöst werden und sind nicht direkt vom Kunden gewählt, z.B. kann ein Kunde ein Ausstattungspaket wählen, welches einen eigenen Code hat, dieser Code löst mehrere Codes für die Sonderausstattungen aus dem Paket aus. Neben Codeketten können auch Baumuster bestimmte Code auslösen, da das Baumuster ja bereits Eigenschaften festlegt (vgl. Kapitel 2.1.5). Diese Informationen sind notwendig, um aus der Vertriebskonfiguration die tatsächliche Produktkonfiguration ableiten zu können.

In Tabelle 4.1-3 sind Codebedingungen für die Auswahl bestimmter Code dargestellt, dabei handelt es sich bei den Zeilen 1 bis 7 um triviale Bedingungen (z.B. verschiedene Motoren, Tonnagen und Radformeln), in denen Redundanzen erkennbar sind. Es kann bereits durch Zeile 4 verhindert werden, dass sowohl CD5 als auch CD6 gewählt werden und Zeile 5 ist überflüssig. In der Praxis werden diese Redundanzen jedoch beibehalten, um den Pflegeaufwand der Dokumentation geringer zu halten.

Durch diese Redundanzen können neue Code hinzugefügt werden ohne die Codebedingungen näher analysieren zu müssen. Möchte man z.B. einen weiteren Motor mit CD4 hinzufügen, kann man die Zeilen einfach um ein CD4 in der Klammer ergänzen und für CD4 die übrigen Motoren mittels $-(CD1/CD2/CD3)$ ausschließen.

Zeile 8 In Tabelle 4.1-3 könnte bedeuten, dass der Motor CD1 nur mit der Tonnage CD7 und der Radformel CD6, oder Tonnage CD8 und Radformel CD5 gewählt werden darf. Zeile 9 hingegen kann bedeuten, dass der Motor CD2 nicht mit der Tonnage CD7 oder CD8 gewählt werden darf aber auf jeden Fall das Getriebe CD9 oder CD10 benötigt.

Tabelle 4.1-3: Codebedingungen für gewählte Code

Zeilennummer	Code	Codebedingung
1	CD1	-(CD2/CD3)
2	CD2	-(CD1/CD3)
3	CD3	-(CD1/CD2)
4	CD5	-CD6
5	CD6	-CD5
6	CD7	-CD8
7	CD8	-CD7
8	CD1	CD7+CD6/CD8+CD5
9	CD2	-(CD7/CD8)+(CD9/CD10)
10	CD3	CD7+CD6/CD8+CD5

In Tabelle 4.1-4 sind sowohl Baumuster als auch Code mit auslösenden Coden dargestellt. Ein Produkt, das in der Vertriebskonfiguration als BM1;PCD2 dokumentiert ist, wird nach dieser Tabelle für die Konfiguration in der Produktion durch die Codekette CD3;CD4;CD5;CD9;CD11 beschrieben.

Tabelle 4.1-4: Baumuster und Paketcode mit auslösenden Coden

Zeilennummer	Baumuster/Paketcode	Ausgelöste Code
1	BM1	CD3;CD9;CD11
2	BM2	CD1;CD9;CD11
3	BM3	CD2;CD9;CD11
4	PCD1	CD4;CD5
5	PCD2	CD2;CD6

Die Beschreibung der baubaren Kombinationen mittels Baumuster-Code-Kombination ist in Tabelle 4.1-5 dargestellt. In diesem Fall entfallen die trivialen Codebedingungen da diese Code bereits durch die Baumuster festgelegt sind. Die trivialen Codebedingungen werden in der Praxis dennoch in einem separaten Datensatz abgelegt um zu verhindern, dass trotz Baumuster-Regeln Fehlkonfigurationen entstehen können.

Das Konfigurationsmanagement mittels Baumuster-Code-Kombination verursacht zwar einen höheren Dokumentationsaufwand, allerdings ist es dadurch möglich nach Baumustern zu filtern und Baumusterspezifisch zu analysieren, was unter anderem in der Baubarkeitsanalyse zu einer erheblichen Verkürzung der Rechendauer führt. (vgl. Kapitel 5.9)

Tabelle 4.1-5: Codebedingungen für gewählte Baumuster-Code-Kombinationen

Zeilennummer	Baumuster	Code	Codebedingung
1	BM1	CD3	;
2	BM1	CD7	CD6
2	BM1	CD8	CD5
3	BM2	CD1	;
5	BM2	CD7	CD6
6	BM2	CD8	CD5
7	BM3	CD2	;

4.1.3.2 Produktstruktur

Für das Dokumentationsbasierte Variantenmanagement unerlässlich sind die Datensätze über die Produktstruktur, genauer der Aufbau der Stücklisten und durch welche Codebedingungen sie ausgelöst werden.

Für die Analysen ist es notwendig möglichst strukturierte Daten zu verwenden, es sind alle Stücklistenarten legitim, ausgenommen der Mengenstückliste, da diese keinerlei Informationen über die Struktur liefert (vgl. Kapitel 2.2.2). Die Mengenstückliste kann verwendet werden um schnell zu berechnen aus welchen Einzelteilen ein Produkt besteht.

Die Codebedingungen können in anderen Datensätzen abgelegt werden als die Stücklisten selbst, Mindestanforderungen an die Codebedingungen für die Stücklisten sind:

- Bedingung, unter der eine Stückliste ausgelöst wird. Dabei kann es sich entweder um eine Codebedingung (Tabelle 4.1-6) oder eine Baumuster-Codebedingung (Tabelle 4.1-7) handeln. Ähnlich dem Konfigurationsmanagement sind die Code durch die booleschen Operatoren AND(+), OR(/), NOT(-) und ANDNOT (+-) verknüpft.
- Menge, wie oft die Stückliste durch die Kombination vorkommt
- Stücklisten-Identifikationsnummer(ID)

Wie in Tabelle 4.1-6 ersichtlich, unterscheiden sich einige Codebedingungen nur durch einen Code. So lösen z.B. Zeile 5 und 6 die gleiche Stückliste aus, allerdings wird sie durch Zeile 6 zwei Mal ausgelöst. Dies könnte z.B. der Fall sein, wenn CD3 für „Ein Tank“ steht und CD8 für „Zwei Tanks“

Tabelle 4.1-6: Transformationsbeziehung mit Codebedingungen

Zeilennummer	Codebedingung	Stücklistenmenge	Stücklisten ID
1	CD1+CD2/CD3+-CD8	1	SL0000201
2	CD4+(CD5/-CD3)+CD8	1	SL0000301
3	CD6+(CD5/-CD3)+CD9	2	SL0000302
4	CD2+-CD4+(CD6/CD7)	1	SL0000503
5	CD3+-CD4+(CD6/CD7)	1	SL0000504
6	CD8+-CD4+(CD6/CD7)	2	SL0000504

In Tabelle 4.1-7 sind die Äquivalenten Stücklisten aus Tabelle 4.1-6, allerdings mit Baumuster-Codebedingungen, dargestellt. Dabei werden in diesem Beispiel die Baumustercode aus Tabelle 4.1-4 verwendet.

Tabelle 4.1-7: Transformationsbeziehung mit Baumuster-Codebedingungen

Zeilennummer	Baumuster	Codebedingung	Stücklistenmenge	Stücklisten ID
1	BM1	;	1	SL0000201
2	BM2	CD2	1	SL0000201
3	BM2	CD4 + CD8	1	SL0000301
4	BM1	CD6 + CD5 + CD9	2	SL0000302
5	BM2	CD6 + CD9	2	SL0000302
6	BM1	CD2 + -CD4 + (CD6/CD7)	1	SL0000503
7	BM2	CD2 + -CD4 + (CD6/CD7)	1	SL0000503
8	BM1	-CD4 + (CD6/CD7)	1	SL0000504
9	BM2	CD8 + -CD4 + (CD6/CD7)	2	SL0000504

Vergleicht man Tabelle 4.1-6 mit Tabelle 4.1-7 ist erkennbar, dass manche Stücklisten nicht für BM1 und manche nicht für BM2 freigegeben sind. Des Weiteren erkennt man, dass sich die Codebedingungen teilweise vereinfacht haben, da durch die Baumuster manche Code vorausgesetzt oder verboten werden. Allerdings erkennt man auch, dass das System mit Baumuster-Codebedingung einen beträchtlichen Mehraufwand in der Dokumentation erfordert, da jede Stückliste, die in mehreren Baumustern für jedes davon einen eigenen Dateneintrag benötigt. So ist Tabelle 4.1-7 um 3 Zeilen länger als Tabelle 4.1-6, obwohl 3 Codebedingungen jeweils nur für ein Baumuster freigegeben sind. Dieses System ermöglicht es jedoch, einfach zu

analysieren, für welche Baumuster eine Stückliste freigegeben ist bzw. welche Stücklisten für ein Baumuster freigegeben sind.

4.2 Pre-Processing

Das Pre-Processing kann in zwei Kategorien unterteilt werden. Einerseits das Analysieren der Datensätze nach darin versteckten Information, die benötigt werden und andererseits das vorausberechnen und ablegen von bestimmten Methoden, die im Zuge der Analysen häufig benötigt werden und dadurch immer wieder berechnet werden müssen.

4.2.1 Ermitteln von Informationen aus den Datensätzen

Wie bereits Eingangs Kapitels 4.1 erwähnt, ist die Ableitung von bestimmten Informationen aus den Datensätzen möglich, jedoch mit erheblichem Mehraufwand verbunden, welcher am Beispiel der Werkskennung aus der FIN erläutert werden soll.

Um die Werkskennung aus der FIN lesen zu können, muss herausgefunden werden...

- ...an welcher Position der FIN die Werkskennung zu finden ist (vgl. Kapitel 2.2.4)
- ...ob die Kennung in der FIN und die Unternehmensinterne Bezeichnung ident sind. „Werk 1“ kann in der FIN ebenfalls mit einer „1“ oder mit anderen Ziffern bzw. Buchstaben gekennzeichnet sein. Sollten die Kennungen nicht ident sein, müssen die Bezüge hergestellt werden.

Nachdem diese Parameter feststehen, muss eine Abfrage erstellt und durchgeführt werden, welche jeden einzelnen FIN-String nach diesen untersucht und die Werkskennung entsprechend in einen Datensatz einfügt.

Neben den Rechenschritten, die für das betrachtete Bauprogramm, welches für einen PKW-Hersteller mehrere Millionen Fahrzeuge enthalten kann, durchgeführt werden müssen, ist auch ein Verständnis der Daten notwendig, um die Parameter herauszufinden.

Weitere Informationen, die aus der FIN gelesen werden können ist das Modelljahr und die Baureihe. Für beide gelten die gleichen Vorgehensweisen wie für das Ermitteln der Werkskennung aus der FIN, einzig für das Modelljahr muss die FIN mit der ISO-Norm 3779 verglichen werden.

4.2.2 Berechnung häufig verwendeter Methoden

Die Methoden, die für die Analysen häufig verwendet werden ist einerseits die Einflussanalyse der booleschen Ausdrücke (Kapitel 3.2) und andererseits die Produktauflösung (Kapitel 3.3).

Die Einflussanalyse der booleschen Ausdrücke findet in der Eigenschaft-Einfluss-Analyse (Kapitel 5.5), der Exklusivitätsanalyse (Kapitel 5.1) und der Baubarkeitsanalyse (Kapitel 5.9) Anwendung. Da das anwenden der Methode viel Zeit in Anspruch nimmt, ist es Empfehlenswert diese Berechnung für alle booleschen Ausdrücke, die in der Dokumentation vorkommen (Konfigurationsmanagement und Transformationsbeziehungen), zu Beginn der Analysen durchzuführen. Sollten sich die Datensätze ändern, ist es notwendig die Berechnung für die neuen bzw. geänderten Datensätze erneut durchzuführen.

Die Produkte werden in aufgelöster Form für die Inter-Vehicle-Commonality (Kapitel 5.2) den Exklusivitätsindex (Kapitel 5.4) und ... benötigt. Da das auflösen der Produkte besonders viel Rechenzeit beansprucht muss im Vorhinein festgelegt werden ob tatsächlich das gesamte Bauprogramm aufgelöst werden soll oder nur die zu analysierenden Mengen (Baumuster, Zeiträume, Regionen, etc.)

5 Tools, Analyse und Visualisierung

Im folgenden Kapitel werden alle Werkzeuge und die damit durchgeführten Analysen bzw. Visualisierungen, die im Zuge des dokumentationsbasierten Varianten- und Komplexitätsmanagements Verwendung finden, detailliert erläutert. Der Großteil dieser Werkzeuge greift auf die Methoden aus Kapitel 3 zurück.

5.1 Exklusivitätsanalyse

Die Exklusivitätsanalyse liefert uns Informationen darüber, welche Stücklisten bzw. Sachnummern exklusiv für einen oder mehrere Code freigegeben sind. Wenn man also bestimmte Eigenschaften von Fahrzeugen in Zukunft nicht mehr zulassen möchte, oder Zwänge wegfallen (z.B. keine Außenspiegel mehr notwendig) erhält man mit dieser Analyse Aussagen darüber, welche Stücklisten und Sachnummern dadurch nicht länger notwendig sind.

Vorweg ist dabei besonders zu beachten, dass nicht automatisch alle Sachnummern, die in einer wegfallenden Stückliste vorkommen, ebenfalls wegfallen, da diese auch in anderen Stücklisten vorkommen können, z.B. Normteile wie Schrauben und Dichtungen.

Ausgangsbasis für die Exklusivitätsanalyse ist die Wahl eines oder mehrerer Code (Tabelle 5.1-1), die man untersuchen möchte.

Tabelle 5.1-1: Code für Exklusivitätsanalyse

Code	Codebezeichnung
CD1	Code 1
CD2	Code 2
CD3	Code 3
SZC	Sitz C

Da hier die gleiche Regel wie bei der Eigenschaft-Einfluss-Analyse gilt, dass nur Code mit positivem Einfluss auf Codebedingungen betrachtet werden dürfen, benötigt man für die Exklusivitätsanalyse ebenfalls eine reduzierte Code-Einfluss-Tabelle (vgl. Tabelle 5.5-3).

Es werden also nur jene Codebedingungen herangezogen, die von den gewählten Coden positiv beeinflusst werden und diese Codebedingungen wiederum werden nur mit jenen Coden verglichen, die diese positiv beeinflussen.

Anschließend werden die Code die in der Codebedingung vorkommen, entgegen der üblichen Vorgehensweise beim Lösen der booleschen Funktionen (vgl. Kapitel 3.1), durch „FALSE“ bzw. „0“ ersetzt, dadurch wird das Sperren der Code simuliert. Nach dem Ersetzen der Code wird überprüft, ob die Funktionen theoretisch noch erfüllbar sind, dadurch erkennt man, ob die gewählten Code für die Codebedingung notwendig oder hinreichend sind. Notwendige Code sind

zwingend notwendig ob die Codebedingung zu erfüllen, hinreichende Code können die Codebedingung erfüllen, müssen allerdings nicht.

Der Unterschied zwischen notwendig und hinreichend zu positiv bestimmend und positiv beeinflussend aus Kapitel 3.2 liegt in der Kombination der Code. Ein positiv bestimmender Code ergibt in der Exklusivitätsanalyse immer einen notwendigen Code, ein positiv beeinflussender Code alleine kann nicht notwendig sein, mehrere positiv beeinflussende Code können allerdings notwendig werden.

In Tabelle 5.1-2 sind Codebedingungen, die Codebedingungen mit Ersetzten Coden und deren Erfüllbarkeit dargestellt.

Tabelle 5.1-2: Codebedingungen mit Ersetzten Coden und ausgewerteter Erfüllbarkeit

Codebedingung	Codebedingung mit Bezeichner	Erfüllbar
CD1/CD2/-CD3	0/0/-CD3	JA
(CD1/CD2)+-CD3	(0/0)+-CD3	NEIN
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	-((CD2/CD3)+-0)+-BD2+ 0+-(JD1/JD2)	NEIN
-((CD2/CD3)+-CD1)+-BD2+ (SZC/-)(JD1/JD2)	-((CD2/CD3)+-0)+-BD2+ (0/-)(JD1/JD2)	NEIN

Die Abwandlungen wurden gewählt um den Unterschied zwischen *notwendig* und *hinreichend* nochmals zu erläutern. *CDA* und *CDB* sind in der ersten Codebedingung hinreichend, da diese auch durch ein nicht vorhanden Sein von *CDC* erfüllt werden kann. Alle durch diese Codebedingung ausgelösten Stücklisten bzw. Sachnummern können somit nicht exklusiv für die Code *CDA* und *CDB* freigegeben sein.

Die zweite Codebedingung ist jedoch nicht mehr erfüllbar, da für die Erfüllung dieser das Vorhandensein von *CDA* oder *CDB* notwendig ist. Die, zu dieser Codebedingung zugehörigen Stückliste ist also exklusiv für die Code *CDA* und *CDB* freigegeben. Das gilt jedoch noch nicht für die zugehörigen Sachnummern, da diese auch in anderen Stücklisten vorkommen können.

In der dritten Codebedingung sind die Code *CDA* und *SZC* notwendig, in der vierten ist *SZC* nicht mehr notwendig, *CDA* jedoch weiterhin, deshalb ist diese Codebedingung auch nicht erfüllbar.

Alle Codebedingungen, die nicht erfüllbar sind, liefern exklusive Stücklisten. Diese Stücklisten werden in weiterer Folge in Ihre Sachnummern aufgelöst, die möglicherweise exklusiv sind.

Vergleicht man nun wie oft die Sachnummern in der Exklusivitätsanalyse vorkommen, mit der Häufigkeit in der Gesamtdokumentation (Maximalstückliste oder Alle Stücklisten) lassen sich daraus Rückschlüsse auf die Exklusivität ziehen.

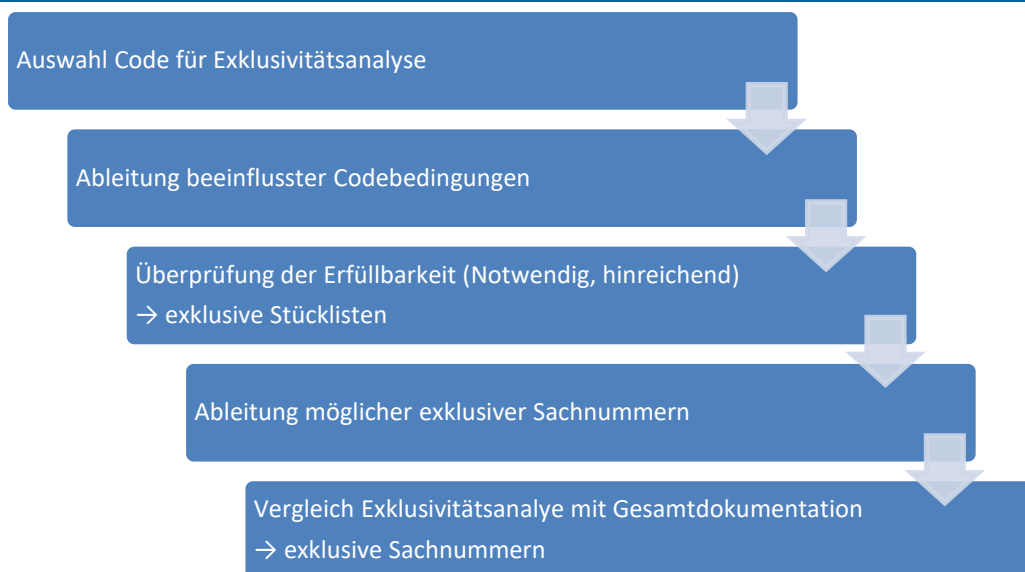
Kommt eine Sachnummer in der Gesamtdokumentation genauso oft vor wie in der Exklusivitätsanalyse, hat man durch die Code alle Stücklisten gewählt, in denen diese Sachnummer vorkommt, daraus folgt, dass diese Sachnummer exklusiv für diese Code freigegeben ist.

Unterscheidet sich die Anzahl, genauer die Häufigkeit in der Exklusivitätsanalyse ist niedriger, als die Häufigkeit in der Gesamtdokumentation, kommt die Sachnummer auch in anderen Stücklisten vor, die durch die gewählten Code nicht exklusiv werden.

Es gilt zu beachten, dass bei einem Dokumentationssystem mit Baumuster-Codebedingungen eine Stückliste, die durch Code ausgelöst wird, nicht zwingend exklusiv sein muss, da diese durch eine andere Baumuster-Codebedingung ebenfalls ausgelöst werden kann (vgl. Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.**).

In Abbildung 5.1-1 ist der schematische Ablauf der Exklusivitätsanalyse dargestellt.

Abbildung 5.1-1: Ablauf Exklusivitätsanalyse

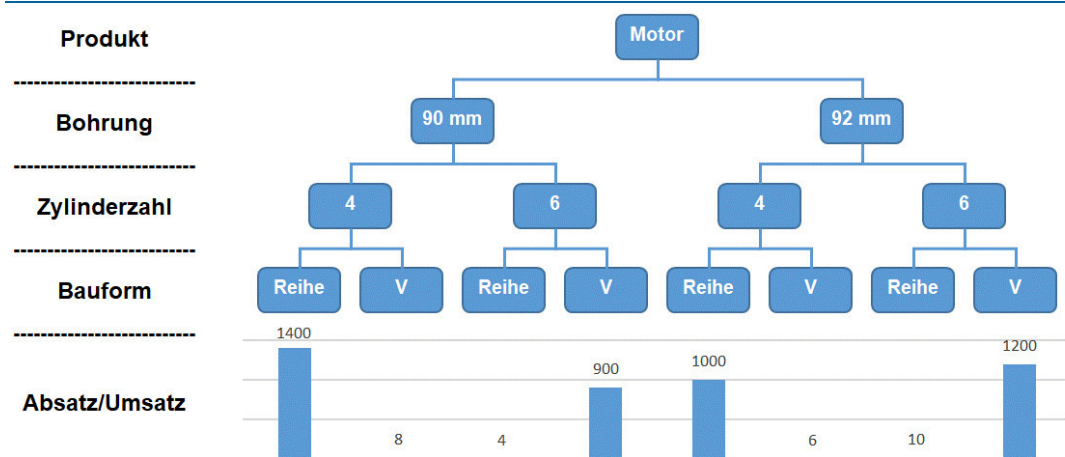


5.2 Merkmalbaum

Der Merkmalbaum nach Schuh kann dazu verwendet werden die angebotene Produktvielfalt visuell zu erfassen und mittels Analyse des Bauprogramms mit der nachgefragten externen Varianz zu vergleichen. Dadurch lassen sich nachgefragte und seltene Varianten erkennen. [5]

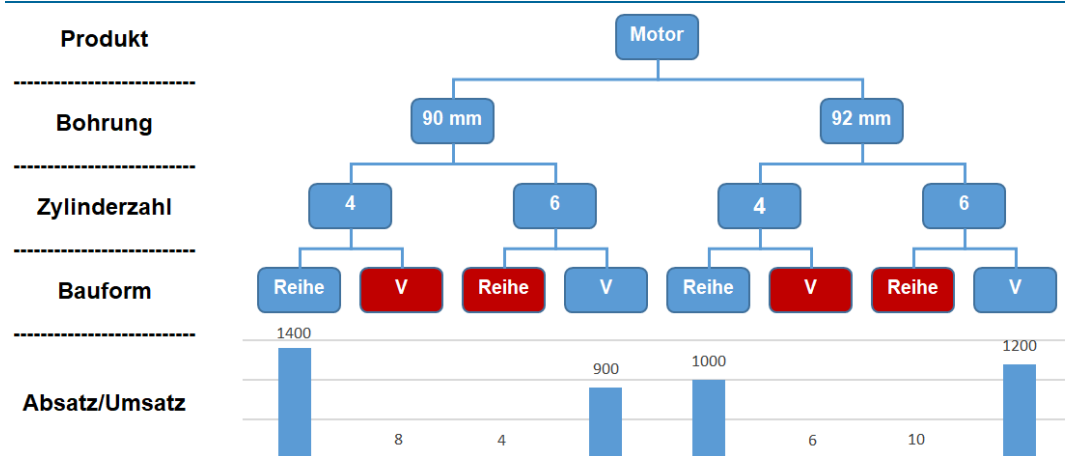
Um den Merkmalbaum zu erstellen wird ein System (Produkt, Baugruppe, Bauteil, etc.) nach seinen Merkmalen gegliedert. Jedes Merkmal wird in einer Ebene dargestellt, die unterschiedlichen Ausprägungen des Merkmals bilden die Verzweigungen des Baums und jede Variante entspricht einem Ast. Über die Informationen des Bauprogramms kann nun jede Variante mit den zugehörigen Absatz- oder Umsatzzahlen hinterlegt werden, wie in Abbildung 5.2-1 dargestellt.

Abbildung 5.2-1: Merkmalbaum mit hinterlegtem Absatz/Umsatz



Es können nun die nicht lukrativen Varianten identifiziert (in Abbildung 5.2-2 rot markiert) und entsprechend reagiert werden.

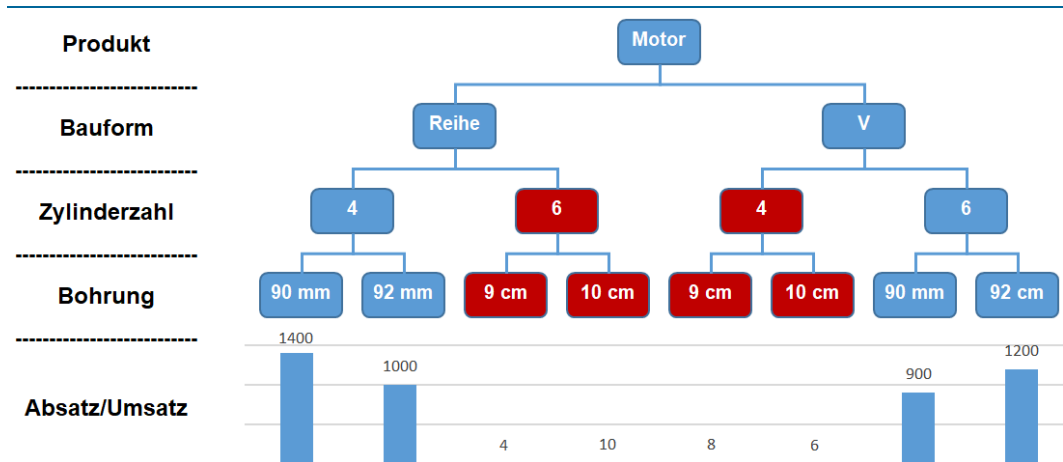
Abbildung 5.2-2: Merkmalbaum mit markierten Niedrigdrehern



Die Ebenen bzw. die Verzweigungsreihenfolge kann im Merkmalbaum umsortiert werden, um zu erkennen welche Varianten besondere Beachtung benötigen. Nach Abbildung 5.2-2 benötigen die Motorvarianten „9 cm Bohrung, 4 Zylinder, V-Form“, „9 cm Bohrung, 6 Zylinder, Reihe“, „10 cm Bohrung, 4 Zylinder, V-Form“ und „10 cm Bohrung, 6 Zylinder, Reihe“ besondere Beachtung.

Sortiert man den Baum jedoch um, wie in Abbildung 5.2-3 dargestellt, erkennt man, dass tatsächlich die Varianten „V-Form, 4 Zylinder“ sowie „Reihe, 6 Zylinder“ unabhängig von der Bohrung besondere Beachtung benötigen.

Abbildung 5.2-3: Merkmalbaum mit Niedrigdrehern, umsortiert



5.3 DSM Anwendungen

Die in Kapitel 3.5 erläuterte DSM kann zur Analyse in verschiedenen unterschiedlichen Ebenen verwendet werden. Als mögliche Domänen haben sich ergeben:

- Produkte vs. Code
- Produkte vs. Einzelteile
- Produkte vs. Stücklisten
- Stücklisten vs. Untergruppen

5.3.1 Paketbildung

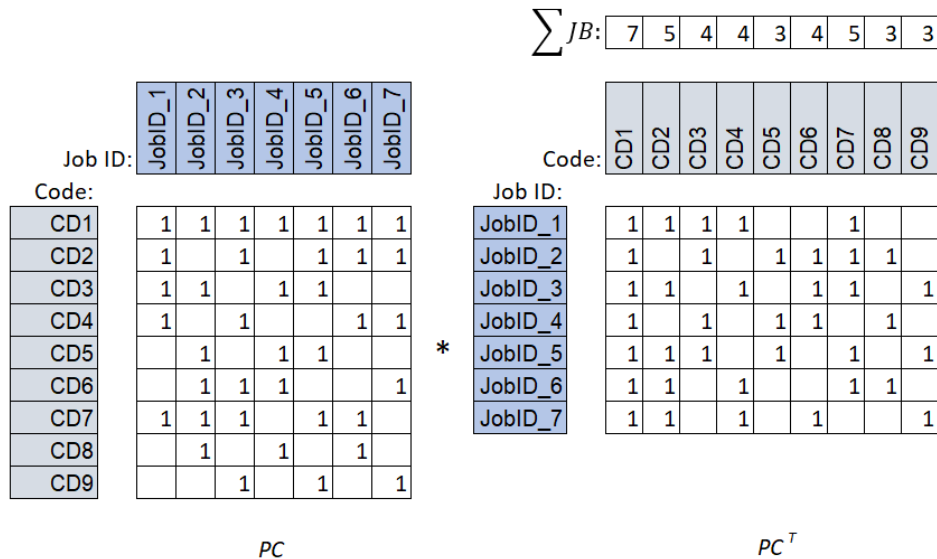
Die Paketbildung in dieser Arbeit orientiert sich an der Paketbildung wie in der Dissertation von Kesper beschrieben [1]

Ziel der Paketbildung ist es, anhand einer Code-DSM, die als Basis eine Produkt-Code-DMM verwendet, zu analysieren welche Ausstattungsmerkmale häufig mit anderen Ausstattungsmerkmalen gewählt werden.

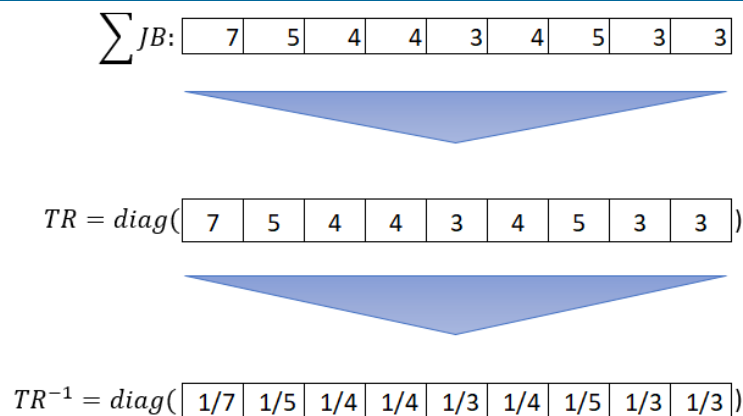
Für die aggregierte Code-DSM PKT wird also die Produkt-Code-DMM PC mit der transponierten Produkt-Code-DMM PC^T multipliziert (siehe Abbildung 5.3-1), dabei entspricht $\sum JB$ der Häufigkeit des Codes in den Jobs, also die sogenannte Takerate.

PKT liefert nun Informationen darüber, wie oft in allen vorgekommenen Produkten ein bestimmter Code mit einem anderen gewählt wurde.

Abbildung 5.3-1: DMM Multiplikation für Paketbildung



Bildet man aus der Takerate $\sum JB$ eine Diagonalmatrix TR entsprechend Abbildung 5.3-2 und multipliziert die Inverse TR^{-1} davon mit PKT erhält man die normierte, aggregierte Code-DSM PKT_{norm} wie in Abbildung 5.3-3 dargestellt.

Abbildung 5.3-2: Bildung von TR^{-1} aus der Takerate

Die normierte, aggregierte Code-DSM PKT_{norm} liefert als Ergebnis, wie oft in Prozent, also relativ gesehen zu der Takerate des betrachteten Codes, ein Code mit einem anderen in Verbindung aufgetreten ist. Der Wert in Zeile 2 und Spalte 3 besagt also, dass bei gewähltem $CD2$ zu 40% auch $CD3$ gewählt wurde

Abbildung 5.3-3: Berechnung der normierten Code-DSM

		Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9			Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9
$TR^{-1} *$	Code:		7	5	4	4	3	4	5	3	3	=	Code:		1,00	0,71	0,57	0,57	0,43	0,57	0,71	0,43	0,43
	CD1		5	5	2	4	1	2	4	1	3		CD1		1,00	1,00	0,40	0,80	0,20	0,40	0,80	0,20	0,60
	CD2		4	2	4	1	3	2	3	2	1		CD2		1,00	0,50	1,00	0,25	0,75	0,50	0,75	0,50	0,25
	CD3		4	4	1	4	0	2	3	1	2		CD3		1,00	1,00	0,25	1,00	0,00	0,50	0,75	0,25	0,50
	CD4		3	1	3	0	3	2	2	2	1		CD4		1,00	0,33	1,00	0,00	1,00	0,67	0,67	0,67	0,33
	CD5		4	2	2	2	2	4	2	2	2		CD5		1,00	0,50	0,50	0,50	0,50	1,00	0,50	0,50	0,50
	CD6		5	4	3	3	2	2	5	2	2		CD6		1,00	0,80	0,60	0,60	0,40	0,40	1,00	0,40	0,40
	CD7		3	1	2	1	2	2	2	2	3		CD7		1,00	0,33	0,67	0,33	0,67	0,67	0,67	1,00	0,00
	CD8		3	3	1	2	1	2	2	2	0		CD8		1,00	1,00	0,33	0,67	0,33	0,67	0,67	0,00	1,00
	CD9										3		CD9		1,00	1,00	0,33	0,67	0,33	0,67	0,67	0,00	1,00
			<i>PKT</i>											<i>PKT_{norm}</i>									

Bei dieser Variante der aggregierten Code-DSM werden nur unterschiedliche Produktvarianten betrachtet und es wird nicht darauf eingegangen wie oft eine Variante gekauft wurde bzw. wie groß die Bestellmenge war. Für ein Einbinden der Bestellmenge muss diese in einer Diagonalmatrix *BST* dargestellt werden (siehe Abbildung 5.3-4)

Abbildung 5.3-4: Bildung der Diagonalmatrix *BST* aus den Bestellmengen

Bestellmengen:

JobID_1	4	→	BST = diag(4	2	1	1	3	2	5)
JobID_2	2									
JobID_3	1									
JobID_4	1									
JobID_5	3									
JobID_6	2									
JobID_7	5									

BST wird nun von rechts mit *PC* multipliziert und man erhält eine gewichtete Produkt-Code DMM $PC_{gew.PKT}$, wie in Abbildung 5.3-5 dargestellt. Diese gewichtete DMM unterscheidet sich von der gewichteten für die Inter-Vehicle-Commonality und erhält dadurch einen weiteren Index.

Abbildung 5.3-5: Bildung der gewichteten Produkt-Code-DMM mittels Bestellmengen

	<table border="1" style="font-size: small;"> <tr><th>Job ID:</th><th>JobID_1</th><th>JobID_2</th><th>JobID_3</th><th>JobID_4</th><th>JobID_5</th><th>JobID_6</th><th>JobID_7</th></tr> <tr><th>Code:</th><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>CD1</td><td>1</td><td></td><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>CD2</td><td>1</td><td>1</td><td></td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>CD3</td><td>1</td><td>1</td><td></td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td>CD4</td><td>1</td><td></td><td>1</td><td></td><td></td><td>1</td><td>1</td></tr> <tr><td>CD5</td><td></td><td>1</td><td></td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td>CD6</td><td></td><td>1</td><td>1</td><td>1</td><td></td><td></td><td>1</td></tr> <tr><td>CD7</td><td>1</td><td>1</td><td>1</td><td></td><td>1</td><td>1</td><td></td></tr> <tr><td>CD8</td><td></td><td>1</td><td></td><td>1</td><td></td><td>1</td><td></td></tr> <tr><td>CD9</td><td></td><td></td><td>1</td><td></td><td>1</td><td></td><td>1</td></tr> </table> <p style="text-align: center;"><i>PC</i></p>	Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7	Code:	1	1	1	1	1	1	1	CD1	1		1		1	1	1	CD2	1	1		1	1	1	1	CD3	1	1		1	1			CD4	1		1			1	1	CD5		1		1	1			CD6		1	1	1			1	CD7	1	1	1		1	1		CD8		1		1		1		CD9			1		1		1	*	BST	=	<table border="1" style="font-size: small;"> <tr><th>Job ID:</th><th>JobID_1</th><th>JobID_2</th><th>JobID_3</th><th>JobID_4</th><th>JobID_5</th><th>JobID_6</th><th>JobID_7</th></tr> <tr><th>Code:</th><td>4</td><td>2</td><td>1</td><td>1</td><td>3</td><td>2</td><td>5</td></tr> <tr><td>CD1</td><td>4</td><td></td><td>1</td><td></td><td>3</td><td>2</td><td>5</td></tr> <tr><td>CD2</td><td>4</td><td>2</td><td></td><td>1</td><td>3</td><td></td><td></td></tr> <tr><td>CD3</td><td>4</td><td>2</td><td></td><td>1</td><td>3</td><td></td><td></td></tr> <tr><td>CD4</td><td>4</td><td></td><td>1</td><td></td><td></td><td>2</td><td>5</td></tr> <tr><td>CD5</td><td></td><td>2</td><td></td><td>1</td><td>3</td><td></td><td></td></tr> <tr><td>CD6</td><td></td><td>2</td><td>1</td><td>1</td><td></td><td></td><td>5</td></tr> <tr><td>CD7</td><td>4</td><td>2</td><td>1</td><td></td><td>3</td><td>2</td><td></td></tr> <tr><td>CD8</td><td></td><td>2</td><td></td><td>1</td><td></td><td>2</td><td></td></tr> <tr><td>CD9</td><td></td><td></td><td>1</td><td></td><td>3</td><td></td><td>5</td></tr> </table> <p style="text-align: center;"><i>PC_{gew.PKT}</i></p>	Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7	Code:	4	2	1	1	3	2	5	CD1	4		1		3	2	5	CD2	4	2		1	3			CD3	4	2		1	3			CD4	4		1			2	5	CD5		2		1	3			CD6		2	1	1			5	CD7	4	2	1		3	2		CD8		2		1		2		CD9			1		3		5
Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7																																																																																																																																																																														
Code:	1	1	1	1	1	1	1																																																																																																																																																																														
CD1	1		1		1	1	1																																																																																																																																																																														
CD2	1	1		1	1	1	1																																																																																																																																																																														
CD3	1	1		1	1																																																																																																																																																																																
CD4	1		1			1	1																																																																																																																																																																														
CD5		1		1	1																																																																																																																																																																																
CD6		1	1	1			1																																																																																																																																																																														
CD7	1	1	1		1	1																																																																																																																																																																															
CD8		1		1		1																																																																																																																																																																															
CD9			1		1		1																																																																																																																																																																														
Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7																																																																																																																																																																														
Code:	4	2	1	1	3	2	5																																																																																																																																																																														
CD1	4		1		3	2	5																																																																																																																																																																														
CD2	4	2		1	3																																																																																																																																																																																
CD3	4	2		1	3																																																																																																																																																																																
CD4	4		1			2	5																																																																																																																																																																														
CD5		2		1	3																																																																																																																																																																																
CD6		2	1	1			5																																																																																																																																																																														
CD7	4	2	1		3	2																																																																																																																																																																															
CD8		2		1		2																																																																																																																																																																															
CD9			1		3		5																																																																																																																																																																														

Analog zu der bereits beschriebenen Code-DSM lässt sich durch die Multiplikation von PC und $PC_{gew.PKT}^T$ (Abbildung 5.3-6) eine gewichtete, aggregierte Code-DSM PKT_{gew} berechnen.

Abbildung 5.3-6: Multiplikation für die gewichtete, aggregierte Code-DSM

	<table border="1" style="font-size: small;"> <tr><th>Job ID:</th><th>JobID_1</th><th>JobID_2</th><th>JobID_3</th><th>JobID_4</th><th>JobID_5</th><th>JobID_6</th><th>JobID_7</th></tr> <tr><th>Code:</th><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>CD1</td><td>1</td><td></td><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>CD2</td><td>1</td><td>1</td><td></td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>CD3</td><td>1</td><td>1</td><td></td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td>CD4</td><td>1</td><td></td><td>1</td><td></td><td></td><td>1</td><td>1</td></tr> <tr><td>CD5</td><td></td><td>1</td><td></td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td>CD6</td><td></td><td>1</td><td>1</td><td>1</td><td></td><td></td><td>1</td></tr> <tr><td>CD7</td><td>1</td><td>1</td><td>1</td><td></td><td>1</td><td>1</td><td></td></tr> <tr><td>CD8</td><td></td><td>1</td><td></td><td>1</td><td></td><td>1</td><td></td></tr> <tr><td>CD9</td><td></td><td></td><td>1</td><td></td><td>1</td><td></td><td>1</td></tr> </table> <p style="text-align: center;"><i>PC</i></p>	Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7	Code:	1	1	1	1	1	1	1	CD1	1		1		1	1	1	CD2	1	1		1	1	1	1	CD3	1	1		1	1			CD4	1		1			1	1	CD5		1		1	1			CD6		1	1	1			1	CD7	1	1	1		1	1		CD8		1		1		1		CD9			1		1		1	*			<table border="1" style="font-size: small;"> <tr><th>Code:</th><th>CD1</th><th>CD2</th><th>CD3</th><th>CD4</th><th>CD5</th><th>CD6</th><th>CD7</th><th>CD8</th><th>CD9</th></tr> <tr><th>Job ID:</th><td>4</td><td>4</td><td>4</td><td>4</td><td></td><td></td><td>4</td><td></td><td></td></tr> <tr><td>JobID_1</td><td>2</td><td></td><td>2</td><td></td><td>2</td><td>2</td><td>2</td><td>2</td><td></td></tr> <tr><td>JobID_2</td><td>1</td><td>1</td><td></td><td>1</td><td></td><td>1</td><td>1</td><td></td><td>1</td></tr> <tr><td>JobID_3</td><td>1</td><td></td><td>1</td><td></td><td>1</td><td>1</td><td></td><td>1</td><td></td></tr> <tr><td>JobID_4</td><td>3</td><td>3</td><td>3</td><td></td><td>3</td><td></td><td>3</td><td></td><td>3</td></tr> <tr><td>JobID_5</td><td>2</td><td>2</td><td></td><td>2</td><td></td><td></td><td>2</td><td>2</td><td></td></tr> <tr><td>JobID_6</td><td>5</td><td>5</td><td></td><td>5</td><td></td><td>5</td><td></td><td></td><td>5</td></tr> <tr><td>JobID_7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> <p style="text-align: center;"><i>PC_{gew.PKT}^T</i></p>	Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9	Job ID:	4	4	4	4			4			JobID_1	2		2		2	2	2	2		JobID_2	1	1		1		1	1		1	JobID_3	1		1		1	1		1		JobID_4	3	3	3		3		3		3	JobID_5	2	2		2			2	2		JobID_6	5	5		5		5			5	JobID_7									
Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7																																																																																																																																																																																
Code:	1	1	1	1	1	1	1																																																																																																																																																																																
CD1	1		1		1	1	1																																																																																																																																																																																
CD2	1	1		1	1	1	1																																																																																																																																																																																
CD3	1	1		1	1																																																																																																																																																																																		
CD4	1		1			1	1																																																																																																																																																																																
CD5		1		1	1																																																																																																																																																																																		
CD6		1	1	1			1																																																																																																																																																																																
CD7	1	1	1		1	1																																																																																																																																																																																	
CD8		1		1		1																																																																																																																																																																																	
CD9			1		1		1																																																																																																																																																																																
Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9																																																																																																																																																																														
Job ID:	4	4	4	4			4																																																																																																																																																																																
JobID_1	2		2		2	2	2	2																																																																																																																																																																															
JobID_2	1	1		1		1	1		1																																																																																																																																																																														
JobID_3	1		1		1	1		1																																																																																																																																																																															
JobID_4	3	3	3		3		3		3																																																																																																																																																																														
JobID_5	2	2		2			2	2																																																																																																																																																																															
JobID_6	5	5		5		5			5																																																																																																																																																																														
JobID_7																																																																																																																																																																																							

	$\sum JB_{gew}$: 18 15 10 12 6 9 12 5 9
--	--

Aus $PC_{gew.PKT}^T$ ergeben sich andere Werte für die Takerate $\sum JB_{gew}$, wie in Abbildung 5.3-6 ersichtlich, die für die Bildung von TR_{gew}^{-1} nach Abbildung 5.3-2 herangezogen werden müssen.

Nach den gleichen Rechenschritten wie für die normierte, aggregierte Code-DSM wird nun die normierte, gewichtete, aggregierte Code-DSM $PKT_{gew.norm}$ berechnet (siehe Abbildung 5.3-7).

Abbildung 5.3-7: Berechnung der normierten, gewichteten, aggregierten Code-DSM

	Code: CD1 CD2 CD3 CD4 CD5 CD6 CD7 CD8 CD9																				
$TR_{gew}^{-1} *$	Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9	=	Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9
	CD1	18	15	10	12	6	9	12	5	9		CD1	1,00	0,83	0,56	0,67	0,33	0,50	0,67	0,28	0,50
	CD2	15	15	7	12	3	6	10	2	9		CD2	1,00	1,00	0,47	0,80	0,20	0,40	0,67	0,13	0,60
	CD3	10	7	10	4	6	3	9	3	3		CD3	1,00	0,70	1,00	0,40	0,60	0,30	0,90	0,30	0,30
	CD4	12	12	4	12	0	6	7	2	6		CD4	1,00	1,00	0,33	1,00	0,00	0,50	0,58	0,17	0,50
	CD5	6	3	6	0	6	3	5	3	3		CD5	1,00	0,50	1,00	0,00	1,00	0,50	0,83	0,50	0,50
	CD6	9	6	3	6	3	9	3	3	6		CD6	1,00	0,67	0,33	0,67	0,33	1,00	0,33	0,33	0,67
	CD7	12	10	9	7	5	3	12	4	4		CD7	1,00	0,83	0,75	0,58	0,42	0,25	1,00	0,33	0,33
	CD8	5	2	3	2	3	3	4	5	0		CD8	1,00	0,40	0,60	0,40	0,60	0,60	0,80	1,00	0,00
	CD9	9	9	3	6	3	6	4	0	9		CD9	1,00	1,00	0,33	0,67	0,33	0,67	0,44	0,00	1,00
		PKT_{gew}											$PKT_{gew, norm}$								

$PKT_{gew, norm}$ liefert nun Informationen darüber, wie oft, relativ zu der Takerate des betrachteten Codes, ein anderer Code gewählt wurde, mit allen verkauften Produkten als Stichprobe.

In Abbildung 5.3-8 sind die Unterschiede der beiden Matrizen PKT_{norm} und $PKT_{gew, norm}$ aufgrund der unterschiedlichen Stichproben übersichtlich dargestellt, unter anderem in dem die Zellen entsprechend ihrer Werte eingefärbt wurden, was ein Erkennen von Clustern einfacher macht.

Abbildung 5.3-8: Vergleich PKT_{norm} und $PKT_{gew, norm}$

	Code: CD1 CD2 CD3 CD4 CD5 CD6 CD7 CD8 CD9																				
\neq	Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9	≠	Code:	CD1	CD2	CD3	CD4	CD5	CD6	CD7	CD8	CD9
	CD1	1,00	0,71	0,57	0,57	0,43	0,57	0,71	0,43	0,43		CD1	1,00	0,83	0,56	0,67	0,33	0,50	0,67	0,28	0,50
	CD2	1,00	1,00	0,40	0,80	0,20	0,40	0,80	0,20	0,60		CD2	1,00	1,00	0,47	0,80	0,20	0,40	0,67	0,13	0,60
	CD3	1,00	0,50	1,00	0,25	0,75	0,50	0,75	0,50	0,25		CD3	1,00	0,70	1,00	0,40	0,60	0,30	0,90	0,30	0,30
	CD4	1,00	1,00	0,25	1,00	0,00	0,50	0,75	0,25	0,50		CD4	1,00	1,00	0,33	1,00	0,00	0,50	0,58	0,17	0,50
	CD5	1,00	0,33	1,00	0,00	1,00	0,67	0,67	0,67	0,33		CD5	1,00	0,50	1,00	0,00	1,00	0,50	0,83	0,50	0,50
	CD6	1,00	0,50	0,50	0,50	0,50	1,00	0,50	0,50	0,50		CD6	1,00	0,67	0,33	0,67	0,33	1,00	0,33	0,33	0,67
	CD7	1,00	0,80	0,60	0,60	0,40	0,40	1,00	0,40	0,40		CD7	1,00	0,83	0,75	0,58	0,42	0,25	1,00	0,33	0,33
	CD8	1,00	0,33	0,67	0,33	0,67	0,67	0,67	1,00	0,00		CD8	1,00	0,40	0,60	0,40	0,60	0,60	0,80	1,00	0,00
	CD9	1,00	1,00	0,33	0,67	0,33	0,67	0,67	0,00	1,00		CD9	1,00	1,00	0,33	0,67	0,33	0,67	0,44	0,00	1,00
		PKT_{norm}											$PKT_{gew, norm}$								

Es ist erkennbar, dass bezogen auf die Produktvariante, bei gewähltem CD3 zu 50% CD2 (in Abbildung 5.3-8 schwarz umrandet) und zu 50% CD6 (in Abbildung 5.3-8 Orange umrandet) gewählt wurde. Betrachtet man allerdings die gewichtete Code-DSM ist zu erkennen, dass die Produktvarianten in denen CD3 und CD2 vorkommen öfter bestellt wurden.

Mittels Clusteranalysen (vgl. Kapitel 3.6) lassen sich 5 mögliche Extremfälle beschreiben, die auftreten können (vgl. Abbildung 5.3-9):

- Ein Code wird immer oder fast immer gewählt, egal welche Code sonst gewählt werden. Daraus lassen sich entweder bereits bestehende Zwänge erkennen, oder neue Zwänge ableiten, dieser Code wird dann zu einem Serieneckode.

- b) Ein Code wird selten oder nie gewählt, egal welcher Code sonst gewählt wurde. Hier besteht die Möglichkeit diese Code zu streichen oder Einschränkungen festzulegen.
- c) Code werden häufig miteinander gewählt, daraus lassen sich Ausstattungspakete festlegen, die immer gewählt werden müssen (z.B. das von Automobilherstellern oft angebotene „Raucherpaket“)
- d) Code werden selten gemeinsam gewählt, daraus lassen sich mögliche Einschränkungen ableiten
- e) Ein Code wird häufig mit einem anderen gewählt, umgekehrt jedoch selten. Daraus lassen sich Zwänge bzw. Einschränkungen ableiten. So z.B. wird bei gewähltem CD8 zwingend CD7 gewählt, wenn allerdings CD7 gewählt wird, sollte CD8 vermieden werden und stattdessen CD2 gewählt werden, wenn es eine funktionale Alternative darstellt. (vgl. Werte aus Abbildung 5.3-8)

Es sei an dieser Stelle nochmal erwähnt, dass immer kritisch hinterfragt werden muss, warum diese Cluster entstanden sind. Ein Cluster für mögliche Ausstattungspakete kann als Ursache die Code, die durch ein Baumuster ausgelöst werden, haben.

Abbildung 5.3-9: Mögliche Extremfälle bei der Paketbildung

a)	Code: CD1	b)	Code: CD8	c)	Code: CD2 CD4			
	Code:		Code:		Code:			
	CD1	1,00	CD1	0,28	CD2	1,00	0,80	
	CD2	1,00	CD2	0,13	CD4	1,00	1,00	
	CD3	1,00	CD3	0,30				
	CD4	1,00	CD4	0,17	d)	Code: CD3 CD4		
	CD5	1,00	CD5	0,50		Code:		
	CD6	1,00	CD6	0,33		CD3	1,00	0,40
	CD7	1,00	CD7	0,33		CD4	0,33	1,00
	CD8	1,00	CD8	1,00	e)	Code: CD7 CD8		
	CD9	1,00	CD9	0,00		Code:		
						CD7	1,00	0,33
						CD8	0,80	1,00

Die Überlegungen der Paketbildung können auch auf eine Variante mit den Domänen Produkt vs. Stückliste angewendet werden. Führt man die Berechnungen durch, lässt sich erkennen, welche Stückliste mit welcher wie oft verbaut wurde. Dabei kann besonderes Augenmerk auf jene Stücklisten gelegt werden, die unabhängig von den restlichen Stücklisten selten gewählt wurden. Zusammen mit der Inter-Stücklisten-Commonality lässt sich daraus ableiten, ob eine seltene Stückliste mit einer häufig gewählten Stückliste zusammengeführt werden kann.

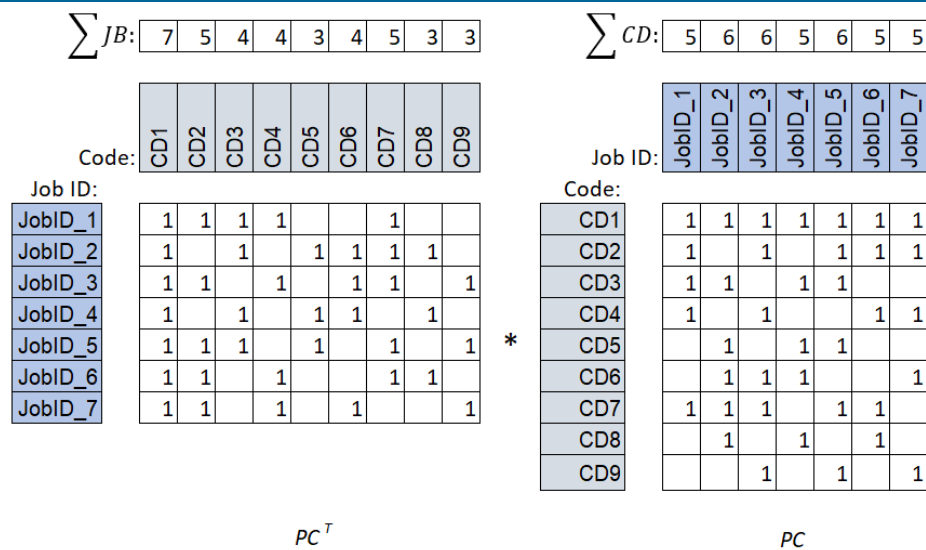
5.3.2 Inter-Vehicle-Commonality

Die Inter-Vehicle-Commonality (IVC) ist eine DSM, in der ersichtlich ist wie ähnlich Produkte untereinander sind. Sie kann sowohl Codebasiert aus Vertriebsicht, als auch Sachnummernbasiert aus Produktionssicht, dargestellt werden. Für ein einfaches Verständnis wird im nachfolgenden Kapitel nur auf die Codebasierte IVC eingegangen, die Rechenschritte und Überlegungen gelten analog für die Sachnummernbasierte IVC.

Ausgangsbasis ist eine DMM PC auf Basis von Produkten aus dem Bauprogramm und Codes, genannt Produkt-Code-DMM. Diese DMM wird auch bei der Codeanalyse für die Serienausstattung bzw. Paketbildung verwendet, vgl. Kapitel 5.3.1

Für die IVC wird die Matrix in umgekehrter Reihenfolge ausgerechnet, also statt $PC * PC^T$ wird $PC^T * PC$ gerechnet (Abbildung 5.3-10). In dieser Abbildung ist analog zur Paketbildung $\sum JB$ die Häufigkeit des Codes in den Jobs, also die sogenannte Takerate, zusätzlich stellt $\sum CD$ die Anzahl der Code in einem Job dar.

Abbildung 5.3-10: IVC Multiplikation



Im ersten Schritt erhält man die Anzahl an gleichen Codes (IVC in Abbildung 5.3-12) multipliziert man diese IVC von links mit der Inversen der Diagonalmatrix AC (Abbildung 5.3-11), ähnlich der normierten Code-DSM (Kapitel 5.3.1), erhält man die normierte IVC (IVC_{norm} in Abbildung 5.3-12), die besagt, wieviel Prozent der Code eines Produktes in einem anderen vorkommen.

Abbildung 5.3-11: Bildung des Vektors AC und TR

$$\sum CD: \begin{array}{|c|c|c|c|c|c|c|} \hline 5 & 6 & 6 & 5 & 6 & 5 & 5 \\ \hline \end{array}$$

$$AC = \text{diag}(\begin{array}{|c|c|c|c|c|c|c|} \hline 5 & 6 & 6 & 5 & 6 & 5 & 5 \\ \hline \end{array})$$

$$AC^{-1} = \text{diag}(\begin{array}{|c|c|c|c|c|c|c|} \hline 1/5 & 1/6 & 1/6 & 1/5 & 1/6 & 1/5 & 1/5 \\ \hline \end{array})$$

Während die IVC symmetrisch ist, ist die normierte IVC, aufgrund der Tatsache, dass Jobs aus unterschiedlich vielen Codes bestehen, nicht symmetrisch, die Diagonale ist dabei immer mit einer 1 besetzt.


Abbildung 5.3-12: IVC und normierte IVC

$$AC^{-1} * \begin{array}{c} \text{Job ID:} \\ \begin{array}{|c|c|c|c|c|c|c|} \hline \text{JobID_1} & \text{JobID_2} & \text{JobID_3} & \text{JobID_4} & \text{JobID_5} & \text{JobID_6} & \text{JobID_7} \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|} \hline \text{JobID_1} & 5 & 3 & 4 & 2 & 4 & 4 & 3 \\ \hline \text{JobID_2} & 3 & 6 & 3 & 5 & 4 & 3 & 2 \\ \hline \text{JobID_3} & 4 & 3 & 6 & 2 & 4 & 4 & 5 \\ \hline \text{JobID_4} & 2 & 5 & 2 & 5 & 3 & 2 & 2 \\ \hline \text{JobID_5} & 4 & 4 & 4 & 3 & 6 & 3 & 3 \\ \hline \text{JobID_6} & 4 & 3 & 4 & 2 & 3 & 5 & 3 \\ \hline \text{JobID_7} & 3 & 2 & 5 & 2 & 3 & 3 & 5 \\ \hline \end{array} \\ \text{IVC} \end{array} = \begin{array}{c} \text{Job ID:} \\ \begin{array}{|c|c|c|c|c|c|c|} \hline \text{JobID_1} & \text{JobID_2} & \text{JobID_3} & \text{JobID_4} & \text{JobID_5} & \text{JobID_6} & \text{JobID_7} \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|} \hline \text{JobID_1} & 1,00 & 0,60 & 0,80 & 0,40 & 0,80 & 0,80 & 0,60 \\ \hline \text{JobID_2} & 0,50 & 1,00 & 0,50 & 0,83 & 0,67 & 0,50 & 0,33 \\ \hline \text{JobID_3} & 0,67 & 0,50 & 1,00 & 0,33 & 0,67 & 0,67 & 0,83 \\ \hline \text{JobID_4} & 0,40 & 1,00 & 0,40 & 1,00 & 0,60 & 0,40 & 0,40 \\ \hline \text{JobID_5} & 0,67 & 0,67 & 0,67 & 0,50 & 1,00 & 0,50 & 0,50 \\ \hline \text{JobID_6} & 0,80 & 0,60 & 0,80 & 0,40 & 0,60 & 1,00 & 0,60 \\ \hline \text{JobID_7} & 0,60 & 0,40 & 1,00 & 0,40 & 0,60 & 0,60 & 1,00 \\ \hline \end{array} \\ \text{IVC}_{norm} \end{array}$$

In unserem Beispiel kann man in Abbildung 5.3-12 erkennen, dass *JobID_7* mit *JobID_3* Code gemeinsam hat. *JobID_7* besteht insgesamt aus 5 Codes und ist zu 100% in *JobID_3* enthalten, *JobID_3* ist jedoch nur zu 83% in *JobID_7* enthalten, da nur 5 der 6 Code von *JobID_3* in *JobID_7* vorkommen.

Analog zur Paketbildung kann auch eine gewichtete IVC berechnet werden, indem man mit *TR* aus Abbildung 5.3-13 die Rechenoperationen $TR * PC = PC_{gew.IVC}$ (Abbildung 5.3-14), $AC_{gew}^{-1} * IVC_{gew} = IVC_{gew,norm}$ (Abbildung 5.3-15) und $PC^T * PC_{gew.IVC} = IVC_{gew}$ (Abbildung 5.3-16) durchführt.

Abbildung 5.3-13: Bildung der Takerate Diagonalmatrix

$$\sum JB: \begin{bmatrix} 7 & 5 & 4 & 4 & 3 & 4 & 5 & 3 & 3 \end{bmatrix}$$


$$TR = \text{diag} \left(\begin{bmatrix} 7 & 5 & 4 & 4 & 3 & 4 & 5 & 3 & 3 \end{bmatrix} \right)$$

Die gewichtete Product-Code-DMM lässt sich auch zusätzlich um die Bestellmengen erweitern, dazu muss sie zusätzlich von rechts mit der Bestellmengenmatrix aus Abbildung 5.3-4 multipliziert werden (vgl. Kapitel 5.3.1)

Abbildung 5.3-14: Berechnung PC_{gew}

$$\sum CD: \begin{bmatrix} 5 & 6 & 6 & 5 & 6 & 5 & 5 \end{bmatrix} \quad \sum CD_{gew}: \begin{bmatrix} 25 & 26 & 28 & 21 & 27 & 24 & 23 \end{bmatrix}$$

	Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7
TR *	Code:							
	CD1	1	1	1	1	1	1	1
	CD2	1		1		1	1	1
	CD3	1	1		1	1		
	CD4	1		1			1	1
	CD5		1		1	1		
	CD6		1	1	1			1
	CD7	1	1	1		1	1	
	CD8		1		1		1	
CD9			1		1		1	

$$=$$

	Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7
PC _{gew.IVC}	Code:							
	CD1	7	7	7	7	7	7	7
	CD2	5		5		5	5	5
	CD3	4	4		4	4		
	CD4	4		4			4	4
	CD5		3		3	3		
	CD6		4	4	4			4
	CD7	5	5	5		5	5	
	CD8		3		3		3	
CD9			3		3		3	

Wie an der gewichteten Produkt-Code-DMM ersichtlich, verschiebt sich der Schwerpunkt von „Anzahl an Codes in einem Job“, zu „wie oft wurden die Codes in diesem Job sonst noch gewählt“. Vergleicht man $JobID_1$ mit $JobID_2$ erkennt man, dass zwar $\sum CD_1 = \sum CD_4$ gilt, allerdings in PC_{gew} . $\sum CD_{gew.1} > \sum CD_{gew.4}$ gilt.

Abbildung 5.3-15: Berechnung IVC_{gew}

	Code:		Job ID:
Job ID:	Code:	Job ID:	Code:
JobID_1	CD1	JobID_1	CD1
JobID_2	CD2	JobID_2	CD2
JobID_3	CD3	JobID_3	CD3
JobID_4	CD4	JobID_4	CD4
JobID_5	CD5	JobID_5	CD5
JobID_6	CD6	JobID_6	CD6
JobID_7	CD7	JobID_7	CD7
	CD8		CD8
	CD9		CD9

	PC^T		$PC_{gew.IVC}$
JobID_1	1	JobID_1	7
JobID_2	1	JobID_2	5
JobID_3	1	JobID_3	4
JobID_4	1	JobID_4	4
JobID_5	1	JobID_5	3
JobID_6	1	JobID_6	4
JobID_7	1	JobID_7	5
	1		3

Die gewichtete Matrix AC_{gew} lässt sich aus $PC_{gew.IVC}$ ableiten, vgl. Abbildung 5.3-11

Abbildung 5.3-16: Berechnung $IVC_{gew,norm}$

	Job ID:		Job ID:
JobID_1	JobID_1	JobID_1	JobID_1
JobID_2	JobID_2	JobID_2	JobID_2
JobID_3	JobID_3	JobID_3	JobID_3
JobID_4	JobID_4	JobID_4	JobID_4
JobID_5	JobID_5	JobID_5	JobID_5
JobID_6	JobID_6	JobID_6	JobID_6
JobID_7	JobID_7	JobID_7	JobID_7

	IVC_{gew}		$IVC_{gew,norm}$
JobID_1	25	JobID_1	1,00
JobID_2	16	JobID_2	0,62
JobID_3	21	JobID_3	0,75
JobID_4	11	JobID_4	0,52
JobID_5	21	JobID_5	0,78
JobID_6	21	JobID_6	0,88
JobID_7	16	JobID_7	0,70

Gemäß der Gewichtung nach oft gewählten Codes, führt die gewichtete und genormte IVC zu höheren Werten als die normierte IVC, wenn die gleichen Code auch oft gewählt wurden bzw. zu geringeren, wenn die Code selten gewählt wurden, wie man an der Gegenüberstellung IVC_{norm} und $IVC_{gew,norm}$ in Abbildung 5.3-17 anhand der Kombinationen $JobID_2 \leftrightarrow JobID_4$ und $JobID_5 \leftrightarrow JobID_6$ sehen kann. $JobID_2 \leftrightarrow JobID_4$ haben in der normierten IVC Übereinstimmungen von 83%, in der gewichteten, normierten jedoch nur 81%. $JobID_5 \leftrightarrow JobID_6$ haben in der normierten IVC eine Übereinstimmung von 50% jedoch eine gewichtete Übereinstimmung von 63%. Die gewichtete IVC liefert also auch Informationen zur Exklusivität (vgl. Kapitel 5.1), da ein Produkt, dass in der gewichteten IVC mit allen anderen Fahrzeugen eine geringe Übereinstimmung hat, exklusiv sein muss.

Abbildung 5.3-17: Gegenüberstellung IVC_{norm} und $IVC_{gew,norm}$

Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7
JobID_1	1,00	0,60	0,80	0,40	0,80	0,80	0,60
JobID_2	0,50	1,00	0,50	0,83	0,67	0,50	0,33
JobID_3	0,67	0,50	1,00	0,33	0,67	0,67	0,83
JobID_4	0,40	1,00	0,40	1,00	0,60	0,40	0,40
JobID_5	0,67	0,67	0,67	0,50	1,00	0,50	0,50
JobID_6	0,80	0,60	0,80	0,40	0,60	1,00	0,60
JobID_7	0,60	0,40	1,00	0,40	0,60	0,60	1,00

IVC_{norm}

≠

Job ID:	JobID_1	JobID_2	JobID_3	JobID_4	JobID_5	JobID_6	JobID_7
JobID_1	1,00	0,64	0,84	0,44	0,84	0,84	0,64
JobID_2	0,62	1,00	0,62	0,81	0,73	0,58	0,42
JobID_3	0,75	0,57	1,00	0,39	0,71	0,75	0,82
JobID_4	0,52	1,00	0,52	1,00	0,67	0,48	0,52
JobID_5	0,78	0,70	0,74	0,52	1,00	0,63	0,56
JobID_6	0,88	0,63	0,88	0,42	0,71	1,00	0,67
JobID_7	0,70	0,48	1,00	0,48	0,65	0,70	1,00

$IVC_{gew,norm}$

Eine genormte IVC mit allgemein hohen Werten bedeutet also, dass die betrachteten Jobs untereinander sehr ähnlich sind, im Gegensatz dazu bedeutet eine genormte IVC mit niedrigen Werten, außer natürlich in der Diagonale, da diese immer 1 sein müssen, dass die Produkte sehr stark voneinander abweichen.

Diese Überlegung führt dazu, dass eine Sachnummernbasierte, normierte IVC im Idealfall hohe Werte aufweist, da dies auf eine niedrige interne Varianz rückschließen lässt, eine Codebasierte lässt auf die externe Varianz rückschließen.

Vergleicht man also eine Codebasierte mit einer Sachnummernbasierten IVC, stellen niedrige Werte in der Codebasierten und hohe in der Sachnummernbasierten ein Optimum dar, da dies bedeutet hohe externe Varianz bei gleichzeitig niedriger interner Varianz (vgl. Kapitel 2.1.7).

Durch Clusterbildung in der IVC lassen sich Produkte mit hohen Gemeinsamkeiten einfach erkennen und anschließend kann analysiert werden, wodurch diese Cluster entstanden sind (z.B. ähnliche Regionen, etc.). Abbildung 5.3-18 zeigt die IVC eines Produktionsprogramms, das bereits auf Layouts reduziert wurde. Diese IVC ist in Abbildung 5.3-19 für den Grenzwert 0,9 geclustert dargestellt.

Abbildung 5.3-18: IVC für Layouts eines Produktionsprogramms

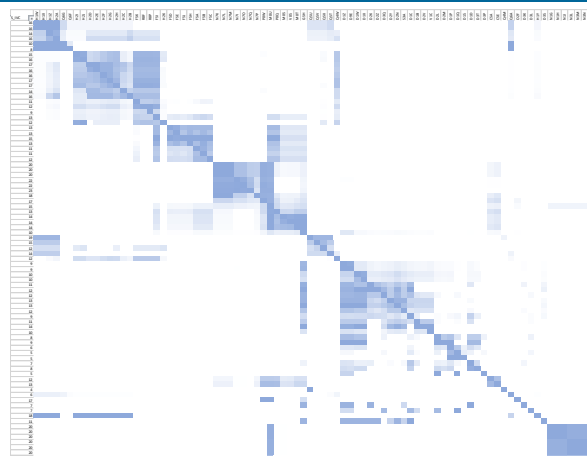
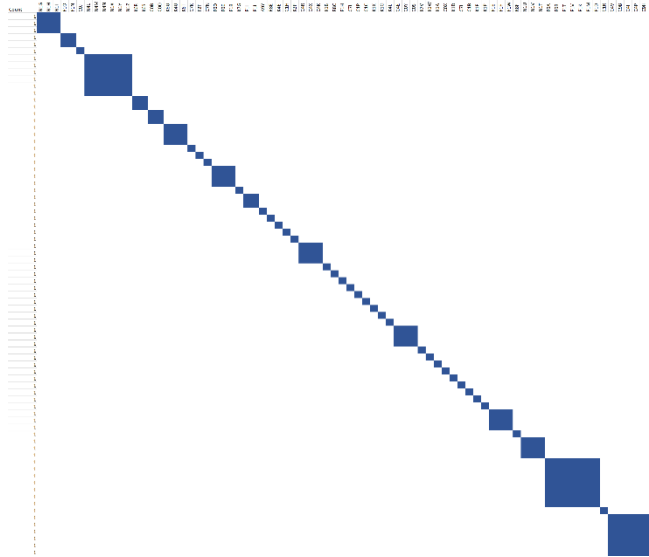


Abbildung 5.3-19: Geclusterte IVC aus Abbildung 5.3-18



5.3.3 Inter-Stücklisten-Commonality

Die Inter-Stücklisten-Commonality (ISC) funktioniert in der Berechnung wie die IVC (vgl. Kapitel 5.3.2), daher wird auf die Rechenschritte nicht näher eingegangen.

Als Ausgangsbasis dient eine DMM mit den Domänen Stücklisten und der ersten untergeordneten Ebene der Stücklisten (dies können entweder Baugruppen und somit Stücklisten oder Einzelteile sein) wie in Abbildung 5.3-20 dargestellt.

Abbildung 5.3-20: Stücklisten-DMM

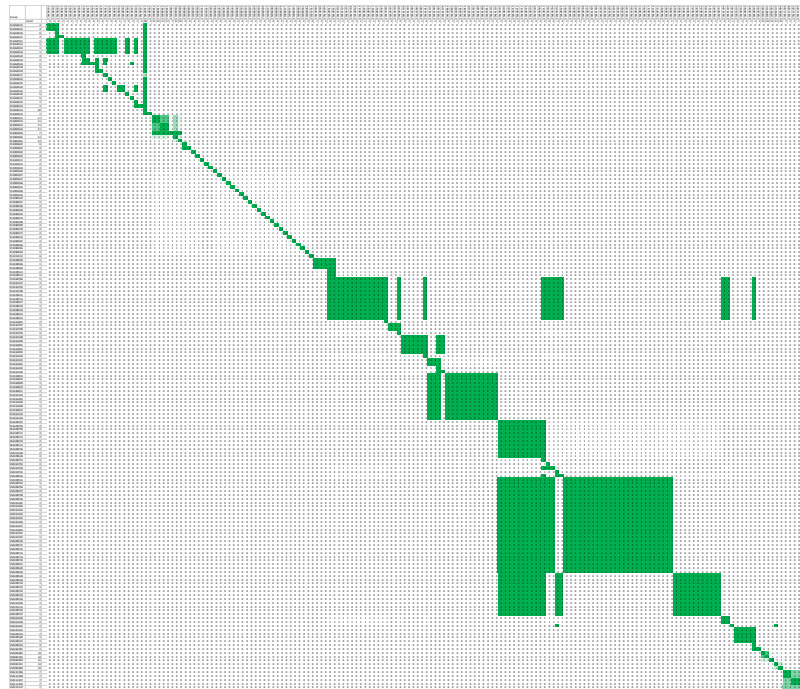
	SL1	SL2	SL3	SL4	SL5	SL6	SL7
ET1	1	1		1		1	
ET2		1		1		1	
ET3	1		1		1		1
ET4			1			1	1
ET5					1		
SL1					1		1
SL2						1	
SL3	1						

Aus der gewichteten normierten ISC lässt sich erkennen wie ähnlich Stücklisten zueinander sind und aufgrund der Gewichtung auch die Exklusivität dieser.

Abbildung 5.3-21 zeigt einen Ausschnitt einer ISC für alle Stücklisten, die für ein bestimmtes Baumuster freigegeben sind. Die grünen Bereiche zeigen eine Übereinstimmung von 100%,

welche in weiterer Folge untersucht werden müssen, um zu erkennen, warum diese so eine hohe Überdeckung haben.

Abbildung 5.3-21: ISC für Stücklisten, die für ein BM freigegeben sind



5.4 Exklusivitätsindex

Der Exklusivitätsindex (EI) ist eine Abwandlung der gewichteten, normierten IVC, dabei werden die gewichteten Werte eines tatsächlichen Produkts mit einem gleich konfigurierten, fiktivem Produkt verglichen, das bei jedem Code die maximale Takerate besitzt, ein solches fiktives Produkt ist anhand der JobID_1 aus PC_{gew} (Abbildung 5.3-14) in Abbildung 5.4-1 dargestellt.

Abbildung 5.4-1: Gewichtete Konfiguration verglichen mit fiktivem Produkt

$\sum CD_{gew}: 18$	$\sum CD_{gew.fiktiv}: 28$		
Job ID: JobID_1	Job ID: Job_Fiktiv		
Code:	Code:		
CD1	6	CD1	7
CD2		CD2	
CD3	5	CD3	7
CD4		CD4	
CD5		CD5	
CD6		CD6	
CD7	3	CD7	7
CD8		CD8	
CD9	4	CD9	7

Bildet man nun den Quotienten aus der gewichteten Anzahl der Code des tatsächlichen Produktes $\sum CD_{gew}$ und der gewichteten Anzahl der Code des fiktiven Produktes $\sum CD_{gew.fiktiv}$, erhält man die gewichtete prozentuelle Übereinstimmung der beiden Produkte:

$$UE_i = \frac{\sum CD_{gew.i}}{\sum CD_{gew.fiktiv.i}}$$

Da das fiktive Produkt mit einer maximalen Takerate der Code das Optimum darstellt, sind hohe Übereinstimmungswerte besser, da wir aber an der Exklusivität EX des Produktes interessiert sind, wird der Gegenwert der Übereinstimmung herangezogen. $EX_i = 1 - UE_i$

Es lässt sich aus Abbildung 5.4-1 herauslesen, dass $\sum CD_{gew.fiktiv}$ auch über die Anzahl der Code in einem Job $\sum CD$ multipliziert mit der Anzahl der betrachteten Jobs $\#JB$ berechnen lässt, es gilt also:

$$\sum CD_{gew.fiktiv.i} = \sum CD_i * \#JB_i$$

Daraus folgt unmittelbar für die Exklusivität:

$$EX_i = 1 - \frac{\sum CD_{gew.i}}{\sum CD_i * \#JB}$$

Nachdem bei der Berechnung des Exklusivitätsindex eines tatsächlich gebauten Produktes dieses mit dem restlichen Bauprogramm verglichen werden soll, es allerdings selbst Bestandteil dieses Bauprogramms ist, muss die Formel für die Berechnung entsprechend um die eigenen Werte reduziert werden. Dies führt dazu, dass der höchste Wert für den EI unabhängig von der Probengröße „1“ betragen kann. Aus $\sum CD_{gew.i}$ muss der eigene Beitrag abgezogen werden, der genau $\sum CD_i$ entspricht, für das korrigierte $\sum CD_{gew.fiktiv.i}$ gilt $\sum CD_i * (\#JB - 1)$ somit folgt für die Berechnung des EI:

$$EI_i = 1 - \frac{\sum CD_{gew.i} - \sum CD_i}{\sum CD_i * (\#JB - 1)}$$

In den folgenden Abbildungen ist der EI , sowie EX zum Vergleich, anhand eines Bauprogramms mit $\#JB = 4$ und unterschiedlichen Konfigurationen dargestellt.

Abbildung 5.4-2 zeigt extreme Beispiele für die Berechnung der Exklusivität, einerseits ein Beispiel mit keiner Wiederholrate der Code (in Abbildung 5.4-2 links), jedes Produkt verwendet also genau einen Code und jeder Code wird nur in einem Produkt verwendet.

Es ist zu erkennen, dass EX trotz maximaler Exklusivität nur den Wert 0,75 annimmt. Das liegt daran, dass bei maximaler Exklusivität $\sum CD_{gew.i} = \sum CD_i$ gilt und somit $EX_i = 1 - \frac{1}{\#JB}$. Im Gegensatz dazu erreicht EI bei maximaler Exklusivität den Wert 1.

Abbildung 5.4-2: EI Berechnung, extreme Beispiele

<p>EI: 1,00 1,00 1,00 1,00 EX: 0,75 0,75 0,75 0,75</p>	<p>EI: 0,00 0,00 0,00 0,00 EX: 0,00 0,00 0,00 0,00</p>																																																		
$\sum CD_{gew}$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">1</td><td style="width: 20px; height: 20px; text-align: center;">1</td><td style="width: 20px; height: 20px; text-align: center;">1</td><td style="width: 20px; height: 20px; text-align: center;">1</td></tr></table>	1	1	1	1	$\sum CD_{gew}$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">16</td><td style="width: 20px; height: 20px; text-align: center;">16</td><td style="width: 20px; height: 20px; text-align: center;">16</td><td style="width: 20px; height: 20px; text-align: center;">16</td></tr></table>	16	16	16	16																																										
1	1	1	1																																																
16	16	16	16																																																
$\sum CD$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">1</td><td style="width: 20px; height: 20px; text-align: center;">1</td><td style="width: 20px; height: 20px; text-align: center;">1</td><td style="width: 20px; height: 20px; text-align: center;">1</td></tr></table>	1	1	1	1	$\sum CD$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">4</td><td style="width: 20px; height: 20px; text-align: center;">4</td><td style="width: 20px; height: 20px; text-align: center;">4</td><td style="width: 20px; height: 20px; text-align: center;">4</td></tr></table>	4	4	4	4																																										
1	1	1	1																																																
4	4	4	4																																																
$\sum JB$: <table style="display: inline-table; border-collapse: collapse; margin-left: 10px;"> <tr> <td style="border: none; padding-right: 5px;">$\sum JB$:</td> <td style="border: none; padding-right: 5px;">JobID_1</td> <td style="border: none; padding-right: 5px;">JobID_2</td> <td style="border: none; padding-right: 5px;">JobID_3</td> <td style="border: none; padding-right: 5px;">JobID_4</td> </tr> <tr> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">CD1</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">CD2</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;"></td> </tr> <tr> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">CD3</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;"></td> </tr> <tr> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">CD4</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;"></td> </tr> </table>	$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4	1	CD1			1	1	CD2		1		1	CD3	1			1	CD4	1			$\sum JB$: <table style="display: inline-table; border-collapse: collapse; margin-left: 10px;"> <tr> <td style="border: none; padding-right: 5px;">$\sum JB$:</td> <td style="border: none; padding-right: 5px;">JobID_1</td> <td style="border: none; padding-right: 5px;">JobID_2</td> <td style="border: none; padding-right: 5px;">JobID_3</td> <td style="border: none; padding-right: 5px;">JobID_4</td> </tr> <tr> <td style="border: none; padding-right: 5px;">4</td> <td style="border: none; padding-right: 5px;">CD1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">4</td> <td style="border: none; padding-right: 5px;">CD2</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">4</td> <td style="border: none; padding-right: 5px;">CD3</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">4</td> <td style="border: none; padding-right: 5px;">CD4</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> </table>	$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4	4	CD1	1	1	1	4	CD2	1	1	1	4	CD3	1	1	1	4	CD4	1	1	1
$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4																																															
1	CD1			1																																															
1	CD2		1																																																
1	CD3	1																																																	
1	CD4	1																																																	
$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4																																															
4	CD1	1	1	1																																															
4	CD2	1	1	1																																															
4	CD3	1	1	1																																															
4	CD4	1	1	1																																															

In Abbildung 5.4-2 rechts ist die andere Extremvariante eines Beispiels dargestellt, indem alle Produkte die exakt gleiche Konfiguration aufweisen. Aufgrund des Maximums an Wiederholrate ist somit $EX = EI = 0$.

Das Beispiel in Abbildung 5.4-3 links ist so gewählt, dass 3 Produkte komplett identisch sind und nur ein Produkt durch maximale Exklusivität hervorsteht. Da nun ein exklusives Produkt im Bauprogramm vorhanden ist, können die anderen 3 Produkte nicht mehr $EI = 0$ erreichen, für das Produkt mit maximaler Exklusivität wird allerdings immer gelten $EI = 1$.

Abbildung 5.4-3 rechts zeigt ein Beispiel mit zufällig verteilter Konfiguration, darin ist zu erkennen, dass jenes Produkt mit exklusivem Code auch einen höheren EI aufweist, als die anderen Produkte.

Abbildung 5.4-3: EI Berechnung, weitere Beispiele

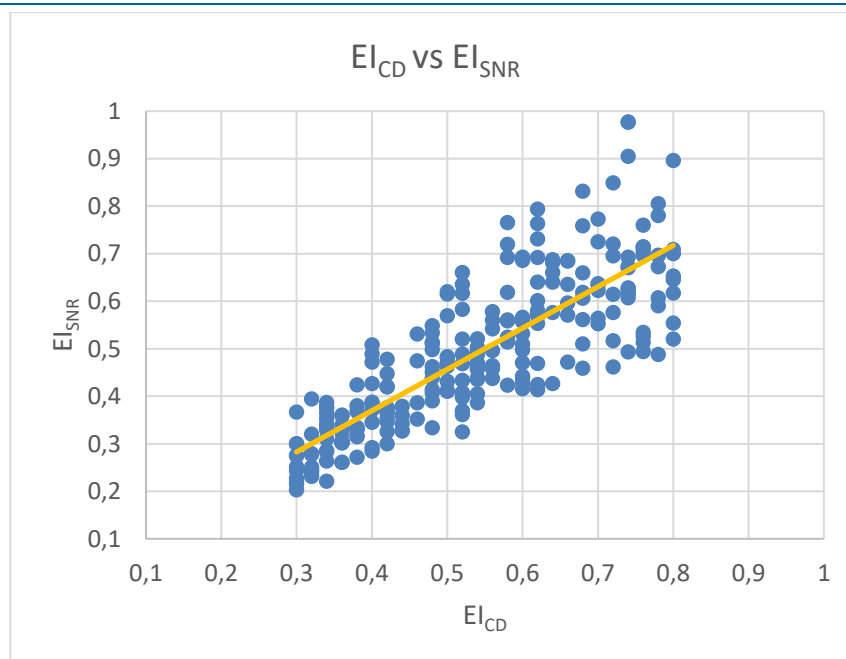
<p>EI: 0,33 0,33 0,33 1,00 EX: 0,25 0,25 0,25 0,75</p>	<p>EI: 0,33 0,33 0,50 0,44 EX: 0,25 0,25 0,38 0,33</p>																																																		
$\sum CD_{gew}$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">9</td><td style="width: 20px; height: 20px; text-align: center;">9</td><td style="width: 20px; height: 20px; text-align: center;">9</td><td style="width: 20px; height: 20px; text-align: center;">1</td></tr></table>	9	9	9	1	$\sum CD_{gew}$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">6</td><td style="width: 20px; height: 20px; text-align: center;">6</td><td style="width: 20px; height: 20px; text-align: center;">5</td><td style="width: 20px; height: 20px; text-align: center;">8</td></tr></table>	6	6	5	8																																										
9	9	9	1																																																
6	6	5	8																																																
$\sum CD$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">3</td><td style="width: 20px; height: 20px; text-align: center;">3</td><td style="width: 20px; height: 20px; text-align: center;">3</td><td style="width: 20px; height: 20px; text-align: center;">1</td></tr></table>	3	3	3	1	$\sum CD$: <table border="1" style="display: inline-table; border-collapse: collapse;"><tr><td style="width: 20px; height: 20px; text-align: center;">2</td><td style="width: 20px; height: 20px; text-align: center;">2</td><td style="width: 20px; height: 20px; text-align: center;">2</td><td style="width: 20px; height: 20px; text-align: center;">3</td></tr></table>	2	2	2	3																																										
3	3	3	1																																																
2	2	2	3																																																
$\sum JB$: <table style="display: inline-table; border-collapse: collapse; margin-left: 10px;"> <tr> <td style="border: none; padding-right: 5px;">$\sum JB$:</td> <td style="border: none; padding-right: 5px;">JobID_1</td> <td style="border: none; padding-right: 5px;">JobID_2</td> <td style="border: none; padding-right: 5px;">JobID_3</td> <td style="border: none; padding-right: 5px;">JobID_4</td> </tr> <tr> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">CD1</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">3</td> <td style="border: none; padding-right: 5px;">CD2</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">3</td> <td style="border: none; padding-right: 5px;">CD3</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">3</td> <td style="border: none; padding-right: 5px;">CD4</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> </table>	$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4	1	CD1			1	3	CD2	1	1	1	3	CD3	1	1	1	3	CD4	1	1	1	$\sum JB$: <table style="display: inline-table; border-collapse: collapse; margin-left: 10px;"> <tr> <td style="border: none; padding-right: 5px;">$\sum JB$:</td> <td style="border: none; padding-right: 5px;">JobID_1</td> <td style="border: none; padding-right: 5px;">JobID_2</td> <td style="border: none; padding-right: 5px;">JobID_3</td> <td style="border: none; padding-right: 5px;">JobID_4</td> </tr> <tr> <td style="border: none; padding-right: 5px;">2</td> <td style="border: none; padding-right: 5px;">CD1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">CD2</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;"></td> </tr> <tr> <td style="border: none; padding-right: 5px;">4</td> <td style="border: none; padding-right: 5px;">CD3</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;">1</td> </tr> <tr> <td style="border: none; padding-right: 5px;">2</td> <td style="border: none; padding-right: 5px;">CD4</td> <td style="border: none; padding-right: 5px;">1</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">1</td> </tr> </table>	$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4	2	CD1	1		1	1	CD2		1		4	CD3	1	1	1	2	CD4	1		1
$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4																																															
1	CD1			1																																															
3	CD2	1	1	1																																															
3	CD3	1	1	1																																															
3	CD4	1	1	1																																															
$\sum JB$:	JobID_1	JobID_2	JobID_3	JobID_4																																															
2	CD1	1		1																																															
1	CD2		1																																																
4	CD3	1	1	1																																															
2	CD4	1		1																																															

Analog zur IVC, kann der EI ebenfalls sowohl Codebasiert (EI_{CD}), als auch Sachnummernbasiert (EI_{SNR}) berechnet werden. Ein hoher, codebasierter EI ist nicht zwingend negativ, da eine exklusive Konfiguration nicht zwingend eine komplexere Produktion bedeutet, es ist allerdings davon auszugehen, dass mit steigendem EI_{CD} auch EI_{SNR} steigt.

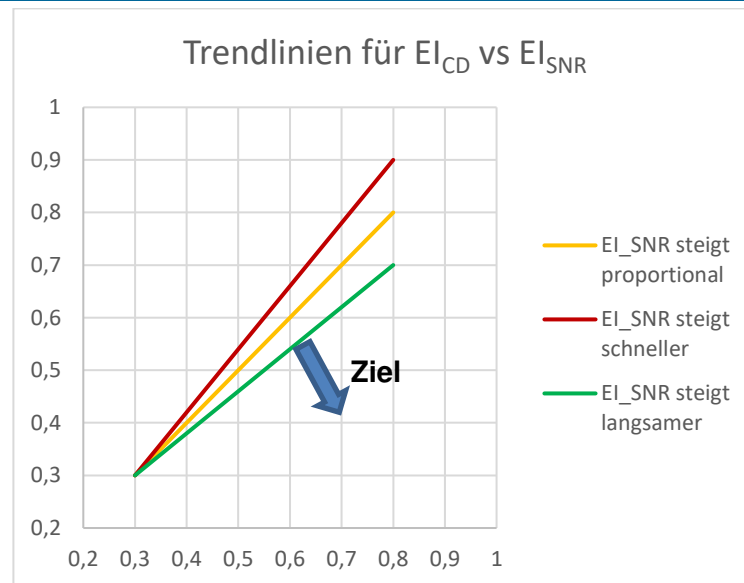
Stellt man EI_{CD} und EI_{SNR} in einem Punktdiagramm gegenüber (Abbildung 5.4-4) kann man eine gewisse Streuung erkennen und, dass ein hoher EI_{CD} nicht zwingend einen hohen EI_{SNR} bedeutet, allerdings lässt sich eine Trendlinie (gelb) generieren, die zeigt, dass ein hoher EI_{CD} im Schnitt auch einen hohen EI_{SNR} verursacht.

Abhängig von der Gestaltung der Produktstruktur können die Trendlinien unterschiedlich schnell steigen, wie in Abbildung 5.4-5 dargestellt. Ist die Steigung der Trendlinie größer als 1, bedeutet das der EI_{SNR} steigt schneller als der EI_{CD} . Das ist der ungünstigste Fall, da daraus schließen lässt, dass mit geringer Konfigurationsänderung kundenseitig, mehr unterschiedliche Sachnummern benötigt werden.

Abbildung 5.4-4: Gegenüberstellung EI_{CD} und EI_{SNR}



Ist die Steigung der Trendlinie kleiner als 1 bedeutet das hingegen, dass bei einer kundenseitigen Konfigurationsänderung die Anforderung an unterschiedliche Sachnummern nicht so hoch ist.

Abbildung 5.4-5: Trendlinien für Gegenüberstellung von EI_{CD} und EI_{SNR} 

Aus den obigen Überlegungen lässt sich erkennen, dass das Ziel eine möglichst flache Trendlinie sein sollte. Diese lässt sich einerseits über Konfigurationsregeln beeinflussen indem Konfigurationen, die einen hohen EI_{SNR} Wert haben, nicht zugelassen werden, andererseits durch Verwendung bzw. Verbesserung eines Baukastensystems mit Mehrfachverwendung der Bauteile.

5.5 Eigenschaft-Einfluss-Analyse

Aus den Transformationsbeziehungen in der Produktarchitektur wissen wir, welche Stücklisten durch die Codebedingungen ausgelöst werden. Dadurch lässt sich im Nachhinein analysieren, welche Code bzw. Codegruppen die Treiber der internen Variantenvielfalt im Produktprogramm sind bzw. welche Eigenschaften wieviel Einfluss auf die Stücklisten haben, daher die Bezeichnung Eigenschaft-Einfluss-Analyse (EEA). Dazu wird die, von den jeweiligen Coden/Codegruppen, beeinflussten Menge an Stücklisten und in weiterer Folge Sachnummern analysiert. Jene Code/Codegruppen, die eine große Anzahl an Sachnummern beeinflussen, sind demnach auch Treiber der internen Variantenvielfalt.

Für die Auswahl bzw. Entwicklung der Algorithmen ist dabei darauf zu achten, dass „not“ Codebedingungen (z.B. -CD3) nicht als Einfluss auf eine Sachnummer zu betrachten sind. Diese Bedingungen kommen unter anderem zustande, wenn für eine Standardausführung kein eigener Code gewählt werden muss. Als Beispiel sei hier ein Standardsitz erwähnt, der immer eingebaut wird, wenn nicht der Code für einen anderen Sitz ausgewählt wurde vgl. Tabelle 5.5-1.

Tabelle 5.5-1: Beispiel Codebedingung für Eigenschaft-Einfluss-Analyse

Stücklistennummer	Stücklistenbezeichnung	Codebedingung	Codebedingung Übersetzung
LS000101	Standardsitz	-(SZB/SZC/SZD)	NOT (Sitz B OR Sitz C OR Sitz D)
LS032402	BsplisteA	CD1/CD2/-CD3	CD1 OR CD2 OR NOT CD3
LS058335	BsplisteB	-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	NOT ((CD2 OR CD3) AND NOT CD1) AND NOT BD2 AND SZC AND NOT(JD1 OR JD2)

Für das zweite Beispiel aus Tabelle 5.5-1 gilt, dass CD1 und CD2 einen Einfluss auf die Stückliste BsplisteA haben, C3 jedoch nicht.

In einem ersten Schritt müssen also die Codebedingungen der Stücklisten, bezüglich ihrer Code und deren Einflussart analysiert werden (vgl. Kapitel 3.2), das Ergebnis dieser Analyse von Tabelle 5.5-1 ist in Tabelle 5.5-2 dargestellt

Tabelle 5.5-2: Code-Einfluss-Tabelle

Codebedingung	Code	Einfluss
-(SZB/SZC/SZD)	SZB	-2
-(SZB/SZC/SZD)	SZC	-2
-(SZB/SZC/SZD)	SZD	-2
CD1/CD2/-CD3	CD1	1
CD1/CD2/-CD3	CD2	1
CD1/CD2/-CD3	CD3	-1
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	CD2	-1
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	CD3	-1
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	CD1	1
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	BD2	-2
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	SZC	2
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	JD1	-2
-((CD2/CD3)+-CD1)+-BD2+ SZC+-(JD1/JD2)	JD2	-2

Da nur Code mit positivem Einfluss eine Stückliste aktiv auslösen können, wird die Auswertung aus Tabelle 5.5-2 auf die Code mit positivem Einfluss reduziert (siehe Tabelle 5.5-3).

Tabelle 5.5-3: Reduzierte Code-Einfluss-Tabelle

Codebedingung	Code	Einfluss
CD1/CD2/-CD3	CD1	1
CD1/CD2/-CD3	CD2	1
-((CD2/CD3)+-CD1)+-BD2+ SZC+- (JD1/JD2)	CD1	1
-((CD2/CD3)+-CD1)+-BD2+ SZC+- (JD1/JD2)	SZC	2

Abschließend müssen die vom jeweiligen Code beeinflussten Codebedingungen gezählt werden (siehe Tabelle 5.5-4).

Tabelle 5.5-4: Anzahl positiv beeinflusster Codebedingungen

Code	Anzahl positiv beeinflusster Codebedingungen
CD1	2
CD2	1
SZC	1

Über die beeinflussten Codebedingungen erkennt man direkt die beeinflussten Stücklisten, daraus lassen sich wiederum die beeinflussten Sachnummern ableiten.

Auch hier gilt, dass abhängig von der Struktur des Dokumentationssystems reine Codebedingungen oder Baumuster-Code-Kombinationen betrachtet werden können bzw. müssen.

In Abbildung 5.5-1 sind die Schritte der Eigenschaft-Einfluss-Analyse schematisch dargestellt.

Abbildung 5.5-1: Ablauf Eigenschaft-Einfluss-Analyse

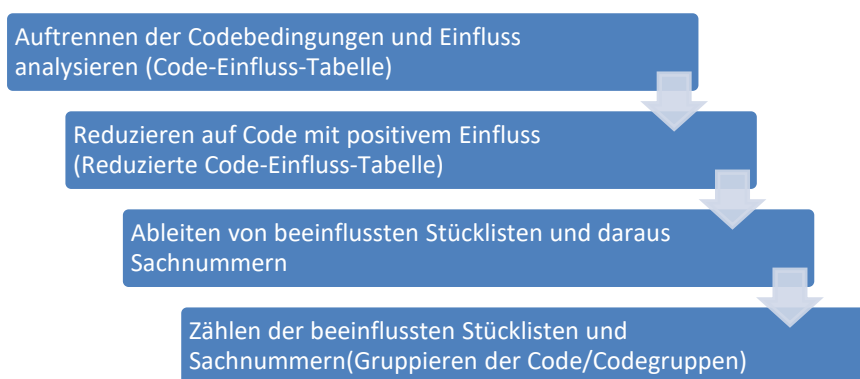


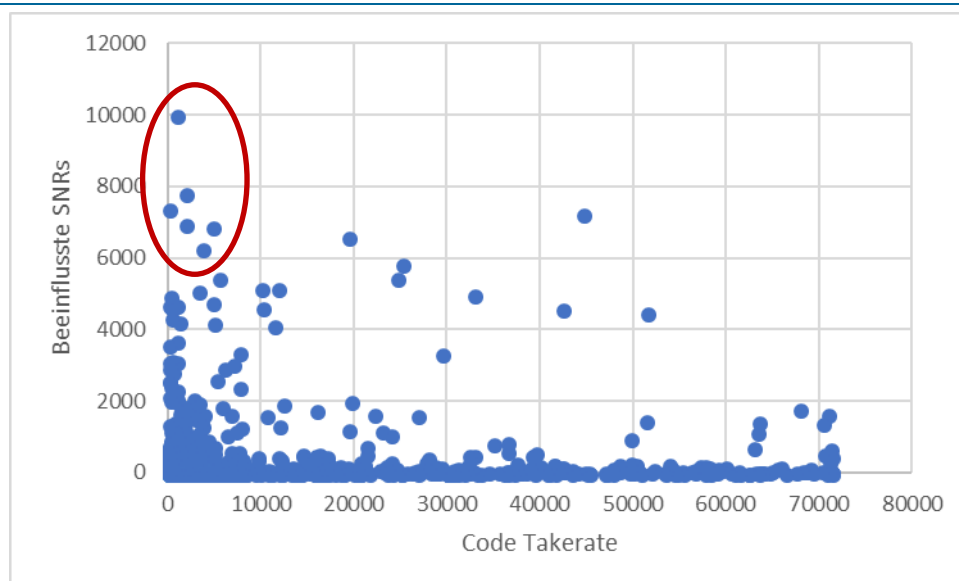
Tabelle 5.5-5 zeigt einen Ausschnitt einer ausgewerteten Eigenschaft-Einfluss-Analyse für das Fahrerhaus eines Nutzfahrzeugherstellers. Darin ist zu erkennen, dass alleine der Code FHL1 für das lange Fahrerhaus einen Einfluss auf fast 3000 Sachnummern hat. Neben den Codes der Gruppe Fahrerhaus-Länge haben auch die Codes der Gruppe Fahrerhaus-Boden einen Einfluss auf eine große Anzahl von Sachnummern.

Tabelle 5.5-5: Variantentreibende Code

Codegruppe	Codegruppe Benennung	Code	Code Benennung	#beeinflusster Sachnummern
1010101	Fahrerhaus-Länge	FHL1	Fahrerhaus lang	2941
1010102	Fahrerhaus-Boden	FHB1	Tunnelhöhe mittel und Staufach	1975
1010102	Fahrerhaus-Boden	FHB2	Tunnelhöhe niedrig	1866
1010102	Fahrerhaus-Boden	FHB3	Tunnelhöhe hoch	1748
1010101	Fahrerhaus-Länge	FHL2	Fahrerhaus mittel	1679
1010102	Fahrerhaus-Boden	FHB4	Tunnelhöhe mittel	1595
2110202	Luftzuführung	LTZ1	Luftzuführung hinten, aufgesetzt	1283
1010101	Fahrerhaus-Länge	FHL3	Fahrerhaus klein	1249
2002321	Fahrerhaus Aufsitzhöhe	FHA1	Aufsetzhöhe hoch	1073
2931032	Fahrerhaus-Dach	FHD	Dachvariante CW Optimiert	1029

Das Ergebnis einer EEA kann mit der Takerate des jeweiligen Codes verglichen werden und liefert dadurch eine Darstellung wie Variantentreibend ein Code ist und wie oft er von einem Kunden gewählt wurde. In Abbildung 5.5-2 ist diese Gegenüberstellung für das Produktionsprogramm des Jahres 2017 eines Nutzfahrzeugherstellers dargestellt. Dabei sind die Code im rot markierten Bereich näher zu untersuchen, da diese sehr treibend für die Interne Variantenvielfalt sind, jedoch von Kunden nur selten gewählt wurden.

Abbildung 5.5-2: Code Takerate vs. beeinflusste Sachnummern



Die Variantentreiberanalyse nach Codegruppen liefert ein ähnliches Bild. Neben den „not“ Bedingungen ist hier bei der Entwicklung der Logarithmen auch darauf zu achten, dass beeinflusste Sachnummern nicht doppelt gezählt werden dürfen, da nur unterschiedliche Sachnummern relevant sind. Daraus folgt, dass die von Codes beeinflussten Sachnummern nicht über die Codegruppen aufsummiert werden dürfen, da davon auszugehen ist, dass z.B. die Code FHL1 und FHL2 auch gleiche Stücklisten und somit Sachnummern auslösen.

Tabelle 5.5-6: Variantentreibende Codegruppen

Codegruppe	Codegruppe Benennung	#beeinflusster Sachnummern	#beeinflusster Sachnummern aufsummiert
1010101	Fahrerhaus-Länge	3919	5869
1010102	Fahrerhaus-Boden	3806	8244
2931032	Fahrerhaus-Dach	2563	4263
2002321	Fahrerhaus Aufsitzhöhe	1801	3344
2110202	Luftzuführung	1792	3126

In Tabelle 5.5-6 ist ein Ausschnitt der Variantentreibenden Codegruppen anhand der gleichen Daten wie in Tabelle 5.5-5 dargestellt, in der letzten Spalte sind die Werte zu sehen, die man erhalten würde, wenn man aufsummiert. Zu erkennen ist, dass Fahrerhaus-Länge deutlich mehr unterschiedliche Sachnummern beeinflusst als Fahrerhaus-Boden. Durch Aufsummieren würde sich das Ergebnis umdrehen.

Die EEA kann auch verwendet um einen bestimmten Entwicklungsumfang (z.B. Abgasrohr) zu analysieren. Dazu werden alle für diesen Entwicklungsumfang relevanten Stücklisten danach untersucht, von welchen Coden bzw. Codegruppen sie beeinflusst werden. Dadurch lässt sich erkennen welche Eigenschaften (Code bzw. Codegruppen) direkt von dem Entwicklungsumfang betroffen sind und für die Entwicklung berücksichtigt werden müssen.

Tabelle 5.5-7 zeigt einen Ausschnitt einer solchen Auswertung für den Entwicklungsumfang Abgasrohr

Tabelle 5.5-7: Beeinflussende Codegruppen für den Entwicklungsumfang Abgasrohr

Codegruppe_ID	Codegruppe Benennung	#Beeinflusste Stücklisten
CG1	Motor	281
CG2	Abgasstufe des Motors	231
CG3	Abgasanlage	230
CG4	Fahrerhauslänge	128

5.6 Layouts

Fahrzeughersteller geben häufig an, dass keines ihrer Produkte identisch zu einem anderen ist. Das bedeutet rein auf Codebasis kann kein Fahrzeug als Hochdreher oder Niedrigdreher betrachtet werden, da alle die Wiederholrate 1 haben.

Das Ziel der Methode ist es, die vollständigen Codeketten der Produkte auf diese Code bzw. Codegruppen zu reduzieren die für einen bestimmten, zu betrachtenden Umfang relevant sind.

Dazu können entweder Layoutbildende Code entweder durch eine EEA (vgl. Kapitel 5.5). identifiziert oder aus Erfahrung gewählt werden

Die EEA wird mit allen Coden, die für den Umfang relevant sind, durchgeführt und liefert als Ergebnis Codegruppen, die viele Sachnummern beeinflussen und solche, die wenige Sachnummern beeinflussen.

Wählt man z.B. alle freigegebenen Code erkennt man, dass diese Codegruppen mit hohem Einfluss in den meisten Fällen layoutbildend sind, da z.B. Code, die den Radstand, die Radformel, den Aufbau oder das Fahrerhaus beschreiben mehr und umfangreichere Stücklisten beeinflussen, als Code, welche die Position des Feuerlöschers oder das Radio beschreiben. Die Codegruppen, die aus der EEA gewählt werden, können in folgende Kategorien unterteilt werden.

Aus der Codegruppe muss...

- genau ein Code vorkommen ($\#CD = 1$)
- einer oder mehrere Code vorkommen ($1 \leq \#CD \leq n$)
- kein oder ein Code vorkommen ($0 \leq \#CD \leq 1$)
- keiner oder mehrere Code vorkommen ($0 \leq \#CD \leq n$)

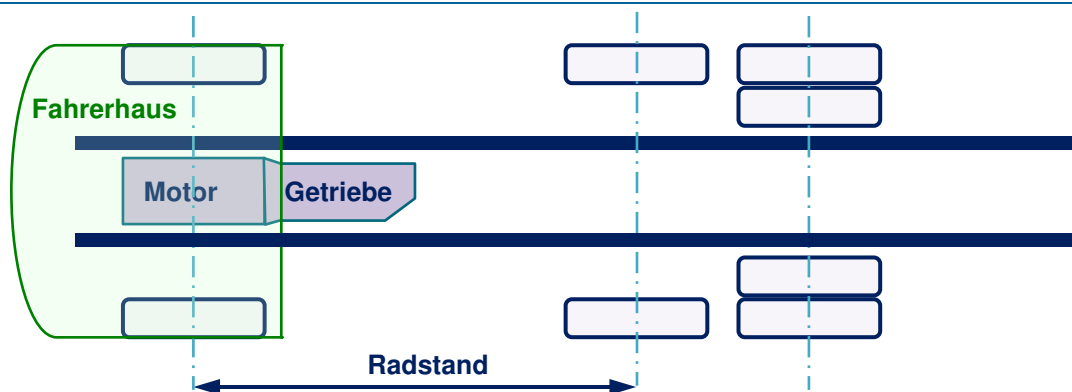
Tabelle 5.6-1 zeigt ein mögliches Beispiel an gewählten Codegruppen für die Wahl eines Layouts für ein Nutzfahrzeug, bei dem es sich um ein sehr einfaches Layout handelt, da es nur Codegruppen der Kategorie 1 enthält, aber noch nicht ausreicht um ein Nutzfahrzeug vollständig zu beschreiben, da unter anderem der Rahmenüberhang, Ausführung der Hinter- bzw. Vorderachse, der Tank usw. fehlt.

Tabelle 5.6-1: Codegruppen für ein einfaches Nutzfahrzeug Layout

Codegruppe		
Antrieb	Fahrgestell	Fahrerhaus
Motor	Radformel	Länge
Getriebe	Radstand	Breite
	Gewichtsklasse	Höhe
	Terrain	Bodenvariante
	Fahrgestellaufbau	

Das Layout zu Tabelle 5.6-1 ist in Abbildung 5.6-1 schematisch dargestellt.

Abbildung 5.6-1: Schematische Darstellung eines einfachen Nutzfahrzeug-Layouts



Bei der Auswahl der Codegruppen bzw. Code ist bereits Verständnis der Daten und der Technik notwendig, um zu erkennen welche Codegruppen trotz hohem Einfluss nicht layoutbildend sind, welche Codegruppen vollständig betrachtet werden müssen und aus welchen Codegruppen man einzelne Code streichen kann. Aufgrund der unterschiedlichen Detaillierungsgrade bzw. des unterschiedlichen Aufbaus der Dokumentationssysteme kann das sehr stark variieren.

Sind die relevanten Code/Codegruppen ausgewählt werden sie, wie eingangs erwähnt, mit dem Bauprogramm verglichen. Dazu werden alle Produkte aus dem Bauprogramm, die einen oder mehrere der gewählten Code in ihrer Codekette haben, zusammengefasst.

Aufgrund der Dokumentationsarchitektur und deren Wachstum kann es vorkommen, dass Code nicht in der Codegruppe vermerkt sind, in der sie sein sollten, was dazu führen kann, dass unvollständige Layouts in dem Bauprogramm gefunden werden (z.B. ein Fahrzeug ohne Radstand oder Motor). In diesem Fall ist es notwendig, die EEA detaillierter zu betrachten und die fehlenden Code zu finden um ausschließlich vollständige Layouts zu finden.

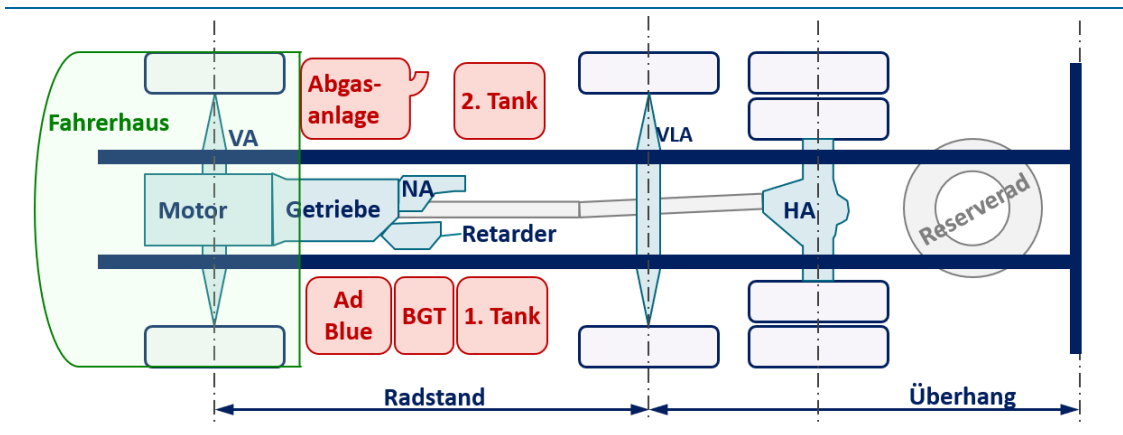
Diese Layouts können in unterschiedlichen Detaillierungsgraden dargestellt werden, je höher dieser ist, desto mehr Layouts mit niedriger Stückzahl werden gefunden. Beispielhafte Codegruppen für ein detaillierteres Layout sind in Tabelle 5.6-2 dargestellt.

Tabelle 5.6-2: Codegruppen für ein detailliertes Nutzfahrzeug Layout

Codegruppe			
Antrieb	Fahrgestell	Fahrerhaus	Anbauten
Motor	Radformel	Länge	Abgasanlage
Getriebe	Radstand	Breite	Tank
Nebenantriebe	Gewichtsklasse	Höhe	Batteriegeräteträger (BGT)
Retarder	Terrain	Bodenvariante	Reserverad
Hinterachse (HA)-Ausführung	Fahrgestellaufbau	Aufsetzhöhe	
Vorderachse (VA)-Ausführung	Rahmenüberhang		
Bremsen	Vorlaufachse (VLA)		
	Nachlaufachse (NLA)		

Bei den Codegruppen in Tabelle 5.6-2 sind bereits Codegruppen aller 4 Kategorien dargestellt, so können aus der Gruppe Tank einer oder mehrere Code ausgewählt werden (Auf jeden Fall 1 Tank und zusätzlich eventuell 2. Tank oder AdBlue-Tank), aus der Codegruppe Reserverad kein Code oder mehrere Code und aus der Kategorie Nachlaufachse ein Code oder kein Code. Das schematische Layout ist in Abbildung 5.6-2 dargestellt.

Abbildung 5.6-2: Schematische Darstellung eines detaillierten Nutzfahrzeug-Layouts



In welche Kategorien die Codegruppen fallen, hängt davon ab wie das Dokumentationssystem aufgebaut ist. So können z.B. auch Code dokumentiert sein die als Bezeichnung „Kein Reserverad“ beinhalten, somit muss aus der Codegruppe Reserverad auf jeden Fall ein Code vorkommen.

Mit einer Mengenbetrachtung der gebauten Layouts auf Basis aller freigegeben Code lässt sich eine ABC-Analyse durchführen und in weiterer Folge kann darauf eingegangen werden, wie mit den Layouts der Kategorie C umgegangen werden soll.

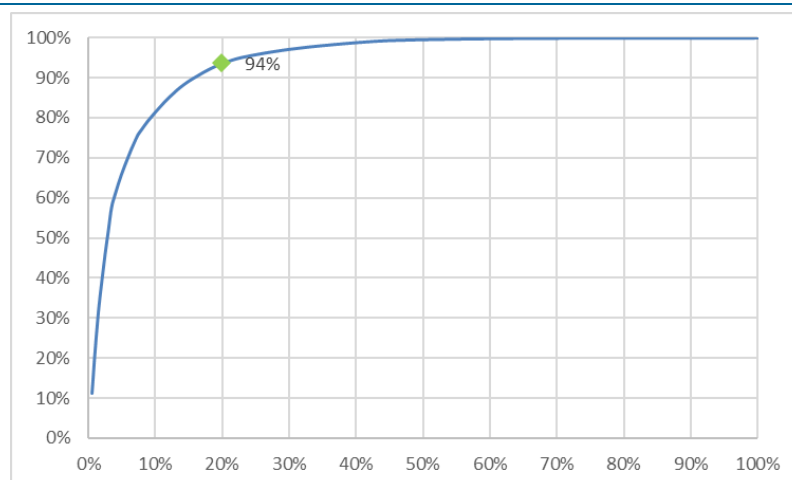
Tabelle 5.6-3 zeigt einen Ausschnitt der Layouts für die Codegruppen aus Tabelle 5.5-7

Tabelle 5.6-3: Layouts für Entwicklungsumfang Abgasrohr

Motor	Abgasstufe	Abgasanlage	Fahrerhaus Länge	# Fahrzeuge
Motor 1	Euro 6	Auslass unten rechts	Lang	31912
Motor 1	Euro 6b	Auslass unten rechts	Lang	31820
Motor 1	Euro 6 mit Diagnose	Auslass unten rechts	Lang	26939
Motor 1	Euro 5	Auslass unten	Lang	20679
Motor 1	Euro 6	Auslass rechts aussen	Lang	18874
Motor 1	Euro 6 mit Diagnose	Auslass rechts aussen	Lang	16886
Motor 1	Euro 6b	Auslass rechts aussen	Lang	16447
Motor 2	Euro 6b	Auslass rechts aussen	Lang	8441
Motor 2	Euro 6	Auslass rechts aussen	Lang	7676
Motor 2	Euro 6 mit Diagnose	Auslass rechts aussen	Lang	6874

Abbildung 5.6-3 zeigt die ABC-Analyse für die Layouts des Entwicklungsumfangs Abgasrohr. Insgesamt decken 205 unterschiedliche Layouts eine Stückzahl größer als 200.000 ab. Wie im Diagramm zu erkennen ist, kann mit 20% (41 Layouts) der Layouts 94% des Bedarfs abgedeckt werden.

Abbildung 5.6-3: ABC-Analyse der Layouts für den Entwicklungsumfang Abgasrohr



5.7 Abschichtung

Bei der Abschichtung handelt es sich um eine Optimierung der verwendeten Sachnummern und der daraus baubaren Produkte. Eingangsgrößen sind dabei alle Produkte in aufgelöster Form.

Ziel der Abschichtung ist es eine Hochlaufkurve zu generieren, die mit möglichst geringer Anzahl an Sachnummern eine möglichst große Anzahl an Produkten ermöglicht. Anhand dieser Hochlaufkurve lässt sich eine ABC-Analyse durchführen.

Vergleicht man diese ABC-Analyse mit jener der Layoutbildenden Code, erkennt man, dass diese eine deutlich steilere Kurve liefert.

Die Schwierigkeit bei der ABC-Analyse mittels Abschichtung besteht darin, dass man zwar weiß, welche Sachnummern in der Kategorie C sind und bei Streichung dieser Sachnummern nur wenig Produkte nicht baubar werden. Im Zuge dieser Arbeit ist es jedoch nicht gelungen Regeln zu finden, um Code oder Codebedingungen zu sperren, wodurch diese Sachnummern in Zukunft nicht mehr bearbeitet werden, andere Sachnummern, die durch gleiche oder ähnliche Codebedingungen ausgelöst werden jedoch weiter betrachtet werden können.

Umgekehrt liefert eine ABC-Analyse nach Codetakerate zwar Code, die selten genommen werden und Streichpotenzial bieten, allerdings liefert eine Exklusivitätsanalyse für diese Code nur eine geringe Anzahl an Stücklisten, die dadurch eingespart werden können.

5.8 Baubarkeitsprüfung

Bei der Baubarkeitsprüfung (BBP) wird eine Codekette mit den Codebedingungen aus dem Konfigurationsmanagement (vgl. Kapitel 4.1.3.1), die durch die Code aus der Codekette ausgelöst werden, verglichen.

Sind alle booleschen Funktionen „TRUE“ ist die Konfiguration laut Dokumentation baubar, aufgrund von Fehlern in der Dokumentation kann es allerdings sein, dass sie technisch nicht baubar ist.

Sind nicht alle booleschen Funktionen „TRUE“ werden all jene, die „FALSE“ sind analysiert und zurückgegeben. Daraus lässt sich herauslesen, warum die Konfiguration nicht baubar ist.

Es kann zwischen zwei BBP-Typen unterschieden werden:

BBP Typ 1:

Vergleich einer vollständigen Codekette mit den Konfigurationsbedingungen.

Ist das Ergebnis der BBP nicht „TRUE“, müssen die verletzten Codebedingungen analysiert werden. Diese können entweder „FALSE“ zurückliefern da ein Code fehlt, oder ein Code, der in Konflikt mit der Codebedingung steht, teil der Konfiguration ist. Dieser BBP Typ kann unter anderem dazu verwendet eine unvollständige Codekette zu vervollständigen, allerdings führt dies nicht immer zu baubaren Konfigurationen.

Es kann vorkommen, dass eine BBP eines bereits gebauten Fahrzeugs „FALSE“ ist. Das ist in den meisten Fällen darauf zurückzuführen, dass das Fahrzeug mit einem anderen Dokumentationsstand konfiguriert wurde und dadurch zum Zeitpunkt des Konfigurierens andere Codebedingungen gültig waren. Es kann allerdings auch daran liegen, dass Fehler in der Dokumentation vorliegen (falsche Codekette im Bauprogramm dokumentiert, Konfigurationsmanagement fehlerhaft).

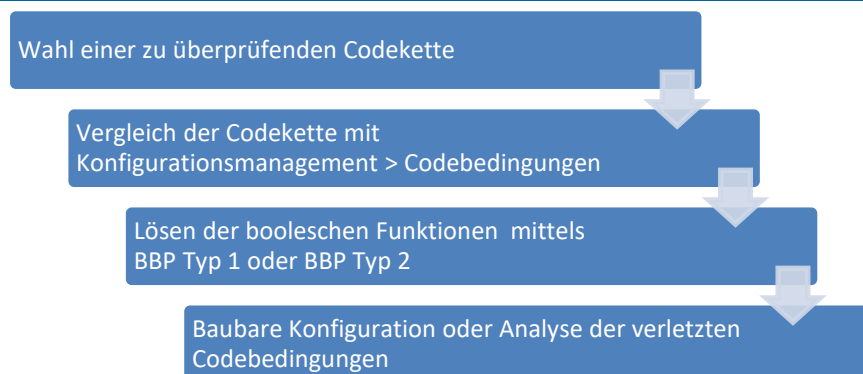
BBP Typ 2:

Vergleich einer Teilcodekette mit den Konfigurationsbedingungen. Dabei liegt das Interesse daran, ob die gewählte Codekette theoretisch baubar ist z.B. ein Motor mit einer bestimmten Tonnage, unabhängig von der Wahl des Getriebes oder der Radformel.

Dieser Typ kann nur dann „FALSE“ sein, wenn die Konfiguration in Konflikt mit negativ beeinflussenden oder negativ bestimmenden (vgl. Kapitel 3.2) Coden steht, also die überprüfte Konfiguration explizit nicht baubar ist.

In Abbildung 5.8-1 ist nochmals der Ablauf einer BBP schematisch dargestellt.

Abbildung 5.8-1: Schematischer Ablauf der Baubarkeitsprüfung



5.9 Baubarkeitsanalyse

Wie in Kapitel (EINLEITUNG) erwähnt stehen produzierende Unternehmen, mit einer hohen Anzahl an Konfigurationsmöglichkeiten, vor dem Problem, dass sie nicht wissen wie groß die externe Variantenvielfalt tatsächlich ist.

Im Zuge dieser Arbeit wurde eine BBA entwickelt, welche aus gegebenen Codegruppen und Coden die tatsächlich baubaren Varianten ermittelt. Die Entwicklungsschritte (Versionen) und Überlegungen, die zur Weiterentwicklung führten, werden in diesem Kapitel erläutert.

Folgende Randbedingungen werden für die BBA angenommen:

- Die zu überprüfende Varianten wird durch eine Codekette beschrieben, diese besteht aus einer Anzahl k an Coden
- Jeder Code ist aus einer zugehörigen Codegruppe CG_k und wird benannt als $CG_{k.1}$ bis $CG_{k.n(k)}$

- Jede Codegruppe besteht aus einer endlichen Anzahl $\#CG_k$ an Coden

Für die Erklärung der BBA werden die Eingangsgrößen aus Tabelle 5.9-1 verwendet

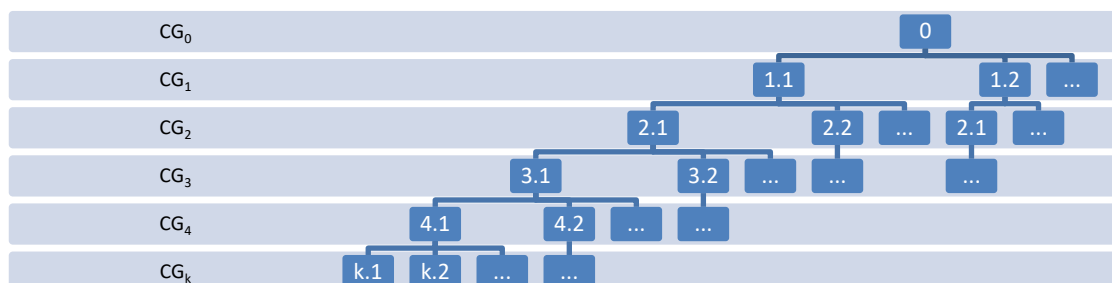
Tabelle 5.9-1: Codegruppen mit Anzahl der Code

Codegruppe	Benennung	$\#CG_k$
CG ₁	Motor	5
CG ₂	Getriebe	26
CG ₃	Radformel	18
CG ₄	Radstand	61
CG ₅	Fahrerhaus Breite	3
CG ₆	Fahrerhaus Länge	3
CG ₇	Fahrerhaus Boden	6
CG ₈	Fahrerhaus Dach	7
CG ₉	Fahrerhaus Würfel	31
CG ₁₀	Rahmenüberhang	40

BBA-Version 1

In der ersten Version der BBA werden alle möglichen Varianten auspermutiert und anschließend einer BBP unterzogen. Dabei gilt zu beachten, dass ein BBP-Typ 2 angewendet werden muss, da wir an der Baubarkeit der gewählten Code/Codegruppen interessiert sind, unabhängig davon, welche Code zusätzlich anfallen könnten.

Abbildung 5.9-1: Permutation Baumstruktur



Für die Permutation kann die Variantenvielfalt als Baumstruktur betrachtet werden, jede Ebene entspricht einer Codegruppe und die Codekette wird Ebene für Ebene zusammengesetzt.

$$CG_{1,1} \rightarrow CG_{1,1}; CG_{2,1} \rightarrow \dots \rightarrow CG_{1,1}; CG_{2,1}; CG_{3,1}; \dots; CG_{9,1}; CG_{10,1}$$

Ist die Codekette vollständig (in diesem Fall, wenn sie aus 9 Coden besteht) wird sie abgespeichert und es wird eine Ebene zurückgesprungen, in diesem Fall zu Ebene 9 $CG_{1,1}; CG_{2,1}; CG_{3,1}; \dots; CG_{8,1}$ und der nächste Code aus Ebene 10 an die Kette angehängt, zu $CG_{1,1}; CG_{2,1}; CG_{3,1}; \dots; CG_{9,1}; CG_{10,2}$. Ist eine Codegruppe (Ebene) vollständig durchlaufen wird 2 Ebenen zurückgesprungen und die Permutation an dieser Stelle fortgesetzt.

In diesem Beispiel:

$$CG_{1,1}; CG_{2,1}; CG_{3,1}; \dots; CG_{8,1}; CG_{9,1}; CG_{10,40} \rightarrow CG_{1,1}; CG_{2,1}; CG_{3,1}; \dots; CG_{8,1} \rightarrow CG_{1,1}; CG_{2,1}; CG_{3,1}; \dots; CG_{8,1}; CG_{9,2} \rightarrow CG_{1,1}; CG_{2,1}; CG_{3,1}; \dots; CG_{8,1}; CG_{9,2}; CG_{10,1}$$

Auf diese Weise wird die komplette Variantenvielfalt durchpermutiert, bis schließlich die letzte Codekette abgespeichert wird, in diesem Bsp. $CG_{1,5}; CG_{2,26}; CG_{3,18}; \dots; CG_{8,7}; CG_{9,31}; CG_{10,40}$

Unter den gewählten Eingangsgrößen aus Tabelle 5.9-1 ergibt sich die Variantenvielfalt $\#VR_k$ in einer Ebene k nach Kapitel 2.3.1 zu:

$$\#VR_k = \prod_{t=1}^k \#CG_t$$

Die theoretische Variantenvielfalt durch reines Auspermutieren der 10 Codegruppen führt zu:

$$\#VR_{10} = \prod_{t=1}^{10} \#CG_t = 6,69 * 10^{10}$$

Sind alle möglichen Kombinationen auspermutiert, wird für jede einzelne Codekette eine BBP durchgeführt. Die, im Zuge dieser Arbeit, verfügbaren technischen Hilfsmittel benötigten für eine BBP ca. 0,5 Sekunden. Für die Variantenvielfalt dieses Beispiels würde die reine Rechenzeit ca. 1000 Jahre betragen, was zur Weiterentwicklung der BBA führte

BBA-Version 2

Aus dem Konfigurationsmanagement (vgl. Kapitel 4.1.3.1) ist bekannt, dass nicht jeder Code für jedes Baumuster freigegeben ist. Werden also die Eingangsgrößen auf die jeweiligen, für die einzelnen Baumuster freigegebenen, Code reduziert kann jedes Baumuster separat auspermutiert werden. Tabelle 5.9-2 zeigt, anhand der Eingangsgrößen aus Tabelle 5.9-1, einen Ausschnitt aus einer Pivot-Darstellung der freigegebenen Code pro Baumuster und Codegruppe sowie die daraus resultierende Variantenvielfalt.

Tabelle 5.9-2: Code, freigegeben für Baumuster und theoretische Variantenvielfalt

BM	#CG ₁	#CG ₂	#CG ₃	#CG ₄	#CG ₅	#CG ₆	#CG ₇	#CG ₈	#CG ₉	#CG ₁₀	#VR ₁₀
BM1	4	7	1	11	2	3	4	5	15	10	5544000
BM2	3	4	1	4	2	3	4	5	15	5	432000
BM3	2	3	1	2	2	2	4	5	10	1	9600
BM4	4	7	1	8	2	3	4	5	15	8	3225600
BM5	3	6	1	7	2	3	4	5	16	6	1451520

BM6	3	7	1	4	2	3	4	5	15	6	907200
BM7	4	6	1	8	2	3	4	5	15	9	3110400
BM8	3	2	1	2	1	1	1	2	2	2	96
BM9		2	1	1	1	1	1	3	2	1	12 (0)
BM10	3	4	1	1	2	1	3	4	7	1	2016
...

Wie in Tabelle 5.9-2 erkennbar ist, ergeben sich durch die Filterung Baumuster mit einer hohen Variantenvielfalt, mit einer niedrigen Variantenvielfalt und Baumuster, für die aus manchen Codegruppen kein Code freigegeben ist. In letzterem Fall gilt es zu hinterfragen, warum aus einer Gruppe kein Code freigegeben ist, es hat sich gezeigt, dass entweder tatsächlich keiner Freigegeben ist, oder die Dokumentation (noch) nicht vollständig ist, wenn z.B. ein neues Baumuster eingeführt wurde.

Die gesamte Variantenvielfalt für das begleitende Beispiel ergibt sich zu

$$\sum_{t=1}^{85} \#VR_{BMt} = 8,89 * 10^7$$

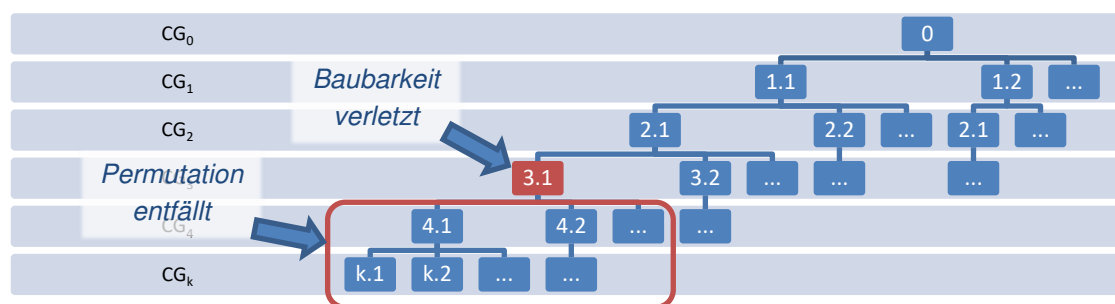
und schließlich einer Berechnungsdauer von ca. 3 Jahren. Neben einer deutlich reduzierten Berechnungsdauer, die für schnelle Analysen noch immer zu hoch ist, ermöglicht die Filterung ausgewählte Baumuster zu permutieren und zu analysieren.

BBA-Version 3

Ebenfalls aus dem Konfigurationsmanagement ist bekannt, dass Code nicht beliebig kombiniert werden dürfen und somit nicht alle Permutationen baubar sein können. Wenn also frühzeitig erkannt wird, dass bestimmte Kombinationen nicht baubar sind kann die aktuelle Permutation in der entsprechenden Ebene abgebrochen werden und alle Permutationen, die in der Codekette noch kommen würden, müssen nicht mehr auspermutiert werden und folglich muss auch keine BBP für diese Kombinationen durchgeführt werden.

Anhand der Baumstruktur zu sehen, wenn in einer bestimmten Ebene abgebrochen wird, entfallen alle nachkommenden Äste hinter einer verletzten BBP, in Abbildung 5.9-2 rot markiert.

Abbildung 5.9-2: Entfallene Permutationen durch Abbruch in einer Ebene



In der Permutation sieht die Abfolge wie folgt aus:

$$CG_{1,1}; CG_{2,1} \rightarrow CG_{1,1}; CG_{2,1}; CG_{3,1} \rightarrow \text{Baubarkeit verletzt!} \rightarrow CG_{1,1}; CG_{2,1}; CG_{3,2} \rightarrow$$

$CG_{1,1}; CG_{2,1}; CG_{3,2}; CG_{4,1} \rightarrow CG_{1,1}; CG_{2,1}; CG_{3,2}; CG_{4,1}; \dots; CG_{9,1}; CG_{10,1}$ unter der Voraussetzung, dass die folgenden Kombinationen baubar sind.

Wie viele Permutationen sich durch frühzeitiges Erkennen einsparen lassen lässt sich durch folgende Überlegung berechnen:

Wird die BBP in der letzten Ebene verletzt, lassen sich 0 Permutationen einsparen, wird sie in der vorletzten Eben verletzt, entfallen $\#CG_k$ Kombinationsmöglichkeiten, wird sie noch eine Ebene früher verletzt entfallen $\#CG_{k-1} * \#CG_k$ Kombinationsmöglichkeiten, daraus folgt für die entfallenen Permutationen über die Ebene v in der die BBP verletzt wird:

$$\text{Anzahl entfallener Permutationen: } \prod_{t=v+1}^k \#CG_t \text{ für } v < k$$

Es gilt zu beachten, dass für das frühzeitige Erkennen der BBP-Verletzungen die BBP in die Permutation integriert werden muss, das bedeutet, dass nach jedem weiteren Code eine BBP durchgeführt wird.

Ebene 1 muss noch nicht überprüft werden, da die Code bereits auf jene reduziert sind, die für ein Baumuster freigegeben sind. In Ebene 2 folgt die erste BBP, die Anzahl an BBP ergibt sich aus der Anzahl an möglichen Kombinationen in dieser Ebene also zu $\#BBP_2 = \#VR_2$ die Anzahl an BBP für eine Permutation mit 3 Ebenen ergibt sich zu $\#BBP_3 = \#VR_3 + \#BBP_2 = \#VR_3 + \#VR_2$ daraus ergibt sich die Anzahl an BBP für k Ebenen:

$$\#BBP_k = \sum_{t=2}^k \#VR_t$$

Ein erster Blick auf die Formel lässt erkennen, dass theoretisch gilt $\#BBP_k > \#VR_k$ allerdings kann $\#VR_k$ durch die Integration der BBP stark reduziert werden, sodass gilt $\#VR_{k.Version3} \ll \#VR_{k.Version2} \rightarrow \#BBP_{k.Version3} \ll \#BBP_{k.Version2} = \#VR_{k.Version2}$

Wie viele BBP sich durch frühzeitiges Erkennen einsparen lassen, lässt sich ähnlich zu den entfallenen Permutationen und den anfallenden BBP berechnen. Grundsätzlich entfallen alle BBP die ab der Verletzung noch anfallen würden. Wird die BBP erst in der letzten Ebene verletzt ergeben sich keine Einsparungen, wird sie in der vorletzten Ebene verletzt erspart man sich $\#CG_k$ Kombinationsmöglichkeiten für die eine BBP durchgeführt werden muss, noch eine Ebene früher erspart man sich $\#CG_{k-1} * \#CG_k$ Kombinationsmöglichkeiten die einer BBP unterzogen werden und jene aus der vorletzten Ebene also ergibt sich die Ersparnis an BBPs aus der vorletzten Ebene zu $\#CG_{k-1} * \#CG_k + \#CG_{k-1}$ noch eine Eben früher ist die Ersparnis $\#CG_{k-2} * \#CG_{k-1} * \#CG_k + \#CG_{k-2} * \#CG_{k-1} + \#CG_{k-2}$. Allgemein lässt sich die Anzahl ersparter BBPs für eine beliebige Ebene v in der abgebrochen wird folgendermaßen berechnen:

$$\text{Anzahl ersparter BBPs: } \sum_{t=v+1}^k \prod_{s=v+1}^t \#CG_s \text{ für } v < k$$

Tabelle 5.9-3 zeigt für das BM4 aus Tabelle 5.9-2 die theoretische Anzahl an Kombinationsmöglichkeiten, theoretischen BBP, sowie die entfallenen Kombinationsmöglichkeiten und eingesparten BBPs bei verletzter BBP pro Ebene.

Tabelle 5.9-3: #VR und #BBP pro Ebene; Entfall #VR und Ersparnis #BBP abhängig von Ebene mit BBP-Verletzung

k	1	2	3	4	5	6	7	8	9	10
#CG _k	4	7	1	8	2	3	4	5	15	8
#VR _k	4	28	28	224	448	1344	5376	26880	403200	3225600
#BBP _k	0	28	56	280	728	2072	7448	34328	437528	3663128
#VR entfall	806400	115200	115200	14400	7200	2400	600	120	8	0
#BBP erspart	915782	130825	130824	16352	8175	2724	680	135	8	0

Wie an Tabelle 5.9-2 erkennbar, wären für eine BBA-Version 2 für das BM4 3.440.640 BBPs mit einer Rechenzeit von ca. 40 Tagen notwendig. Damit BBA-Version 3 tatsächlich schneller ein Ergebnis liefert, müssen genügend Abbruchkriterien gefunden werden sodass gilt $\#BBP_{erspart} > \#BBP_{10} - \#VR_{10}$. Es ist weiters erkennbar, dass der Effekt der Einsparung größer ist, wenn er in einer niedrigen Ebene entdeckt wird.

Unter der Annahme, dass für dieses Baumuster, aufgrund der Kombination von 2 Codegruppen, nur die Hälfte der theoretischen Konfigurationsmöglichkeiten tatsächlich baubar sind, erhält man die in Tabelle 5.9-4 dargestellten Einsparungen an BBPs, abhängig von der Ebene in der die BBP verletzt wurde. Es gilt dabei zu beachten, dass in Ebene 1 noch keine Verletzung auftreten kann.

Tabelle 5.9-4: Ersparnis #BBP bei Hälfte baubar, abhängig von entdeckter Ebene

Ebene BBP verletzt	1	2	3	4	5	6	7	8	9	10
#CG _k	4	7	1	8	2	3	4	5	16	8
#VR _k	4	28	28	224	448	1344	5376	26880	403200	3225600
#BBP verletzt	0	14	14	112	224	672	2688	13440	201600	1612800
#BBP erspart	0	1831550	1831536	1831424	1831200	1830528	1827840	1814400	1612800	0

Aufgrund der Berechnung von #VR und #BBP liegt die Überlegung nahe, dass durch Sortieren der Codegruppen der Anstieg von #VR und #BBP beeinflusst werden kann.

Im Folgenden wird der Anstieg von #VR (Abbildung 5.9-3) und #BBP (Abbildung 5.9-4) entsprechend der Sortierungen der CG in Tabelle 5.9-5 gezeigt.

Tabelle 5.9-5: #VR und #BBP, abhängig von der Permutationsreihenfolge

Ebene	1	2	3	4	5	6	7	8	9	10
#CG optimal	1	2	3	4	4	5	7	8	8	15
#VR	1	2	6	24	96	480	3360	26880	215040	3225600
#BBP	0	2	8	32	128	608	3968	30848	245888	3471488
#CG zufällig	4	7	1	8	2	3	4	5	15	8
#VR	4	28	28	224	448	1344	5376	26880	403200	3225600
#BBP	0	28	56	280	728	2072	7448	34328	437528	3663128
#CG ungünstig	15	8	8	7	5	4	4	3	2	1
#VR	15	120	960	6720	33600	134400	537600	1612800	3225600	3225600
#BBP	0	128	1152	8320	44160	187520	760960	2481280	5921920	9362560

Abbildung 5.9-3: Anstieg von #VR, abhängig von der Permutationsreihenfolge

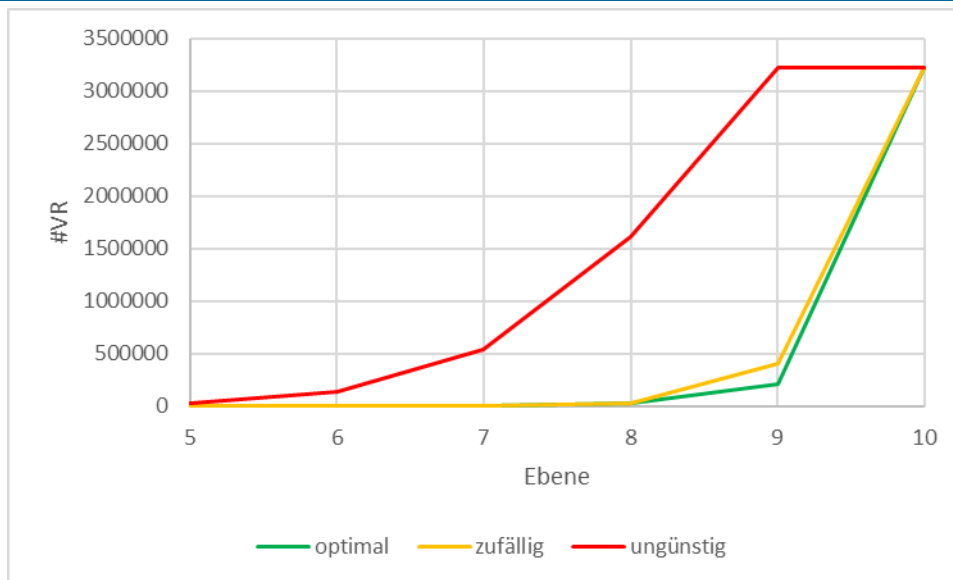
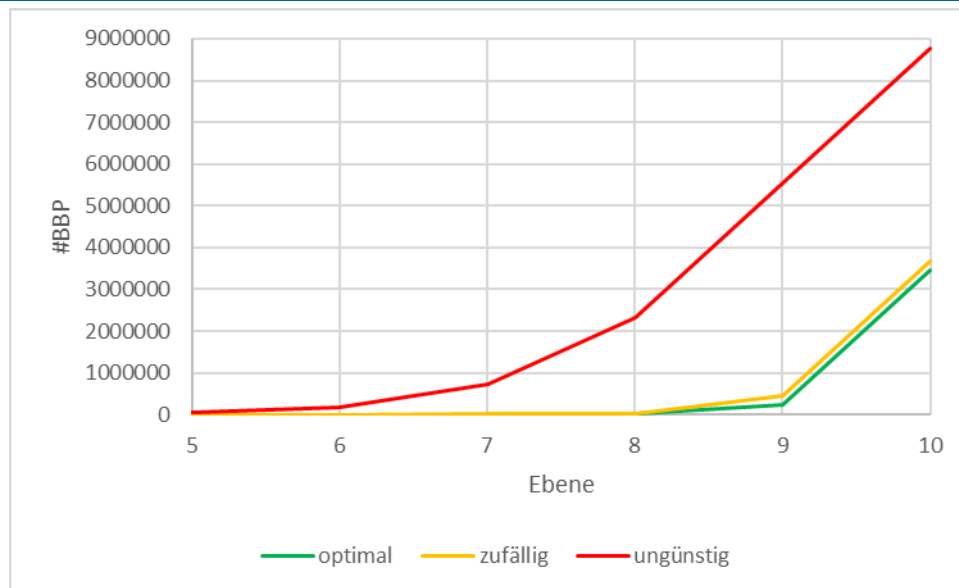


Abbildung 5.9-4: Anstieg von #BBP, abhängig von der Permutationsreihenfolge

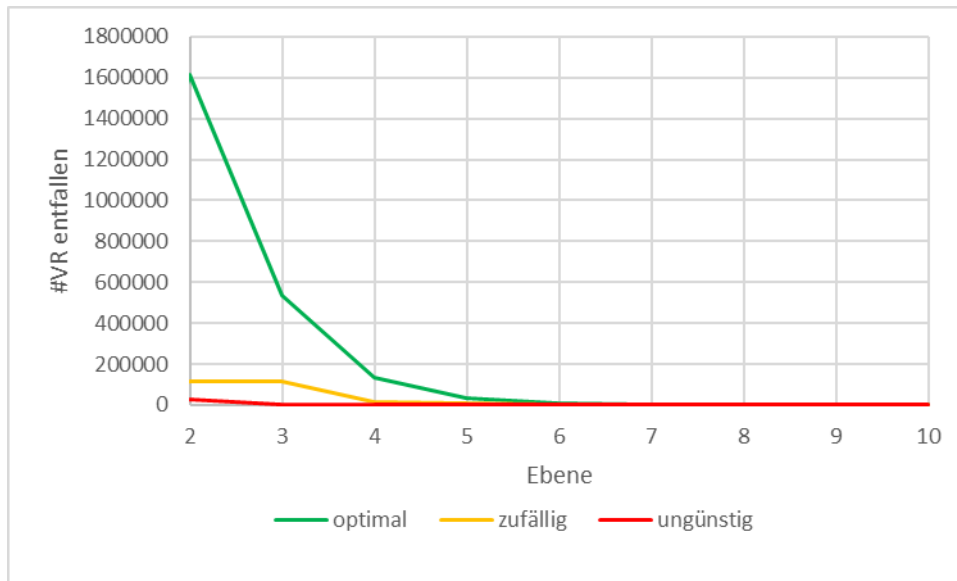


Analog zu dem Anstieg von #VR und #BBP lässt sich der Entfall an #VR und die Einsparung an #BBP berechnen, die Ergebnisse sind in Tabelle 5.9-6 und Abbildung 5.9-5 bzw. Abbildung 5.9-6 dargestellt.

Tabelle 5.9-6: Entfallene #VR und eingesparte #BBP, abhängig von Permutationsreihenfolge

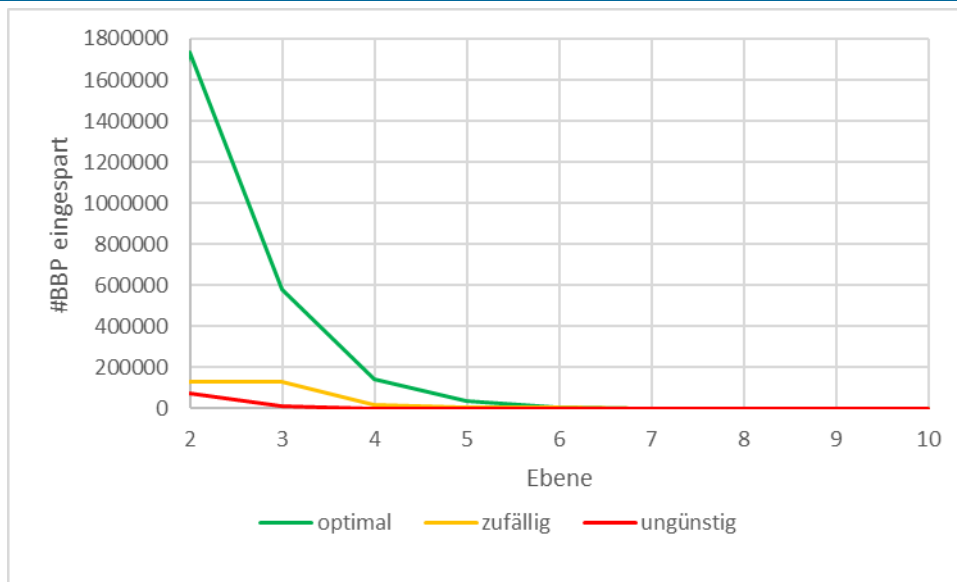
Ebene	1	2	3	4	5	6	7	8	9	10
#CG optimal	1	2	3	4	4	5	7	8	8	15
#VR	0	1612800	537600	134400	33600	6720	960	120	15	0
#BBP	0	1735743	578580	144644	36160	7231	1032	128	15	0
#CG zufällig	4	7	1	8	2	3	4	5	15	8
#VR	0	115200	115200	14400	7200	2400	600	120	8	0
#BBP	0	130825	130824	16352	8175	2724	680	135	8	0
#CG ungünstig	15	8	8	7	5	4	4	3	2	1
#VR	0	26880	3360	480	96	24	6	2	1	0
#BBP	0	73144	9142	1305	260	64	15	4	1	0

Abbildung 5.9-5: #VR entfallen, abhängig von Permutationsreihenfolge



Eine optimale Sortierung sorgt also nicht nur dafür, dass insgesamt weniger BBPs anfallen, sondern auch, dass bei frühzeitigem Abbrechen der Permutation mehr BBPs wegfallen

Abbildung 5.9-6: #BBP eingespart, abhängig von Permutationsreihenfolge



BBA-Version 4

Während in der BBA-Version 3 die Permutationsebenen nach aufsteigender #CG geordnet werden, wird in Version 4 zusätzlich untersucht, wie sich die Code und folglich die Codegruppen gegenseitig beeinflussen, mit dem Ziel #BBP zu reduzieren.

Dabei helfen folgende Überlegungen:

- Es sind lediglich die Codebedingungen aus den Randbedingungen interessant, die auch von den Coden der Permutation beeinflusst werden
- Da für die BBA eine BBP-Typ 2 verwendet wird, können nur jene Codebedingungen einen Abbruch verursachen, die durch einen Code aus der Permutationsauswahl negativ beeinflusst werden (vgl. Kapitel 5.8)

Es werden also in einem ersten Schritt die Codebedingungen aus dem Konfigurationsmanagement mit dem Ergebnis der Einflussanalyse der booleschen Ausdrücke verglichen und zusätzlich, einerseits um die Codegruppen der jeweiligen Code ergänzt und andererseits auf die Datensätze reduziert, die in Spalte Einfluss einen Wert kleiner „0“ haben. Tabelle 5.9-7 zeigt einen Ausschnitt aus den oben genannten Informationen.

Tabelle 5.9-7: Auslösende und beeinflussende Code für BBA

CG Ausl.	Code Ausl.	CDBED	CG Beinfl.	Code Beinfl.	Beinfl.
CG9	CD9.1	–CD5.1	CG5	CD5.1	-2
CG9	CD9.1	–(CD6.2/CD6.3)	CG6	CD6.2	-2
CG9	CD9.1	–(CD6.2/CD6.3)	CG6	CD6.3	-2
CG9	CD9.2	–(CD7.1/ CD7.3/ CD7.5)	CG7	CD7.1	-2
CG9	CD9.2	–(CD7.1/ CD7.3/ CD7.5)	CG7	CD7.3	-2
CG9	CD9.2	–(CD7.1/ CD7.3/ CD7.5)	CG7	CD7.5	-2
CG9	CD9.3	–(CD8.4/ CD8.5/ CD8.6/ CD8.7)	CG8	CD8.4	-2
CG9	CD9.3	–(CD8.4/ CD8.5/ CD8.6/ CD8.7)	CG8	CD8.5	-2
CG9	CD9.3	–(CD8.4/ CD8.5/ CD8.6/ CD8.7)	CG8	CD8.6	-2
CG9	CD9.3	–(CD8.4/ CD8.5/ CD8.6/ CD8.7)	CG8	CD8.7	-2
CG7	CD7.1	CD6.3 + CDX.1 – CD1.3	CG1	CD1.3	-2
CG10	CD10.6	–((CDX.1/CDX.2/CDX.3) + (CD4.1/CD4.2/CD4.3))	CG4	CG4.1	-1
CG10	CD10.6	–((CDX.1/CDX.2/CDX.3) + (CD4.1/CD4.2/CD4.3))	CG4	CG4.2	-1
CG10	CD10.6	–((CDX.1/CDX.2/CDX.3) + (CD4.1/CD4.2/CD4.3))	CG4	CG4.3	-1

Es gibt nun folgende Informationen die schnell aus Tabelle 5.9-7 gelesen werden können:

- Ein Datensatz hat in der Spalte Einfluss den Wert „-2“:
Die Kombination dieser beiden Code führt zwingend zu einem Abbruch (vgl. Kapitel 3.2), folglich müssen die beiden Codegruppen des Datensatzes in der Permutation nach vorne gereiht werden

- Eine Codegruppe kommt weder in „CG Ausl.“ noch in „CG Beeinfl.“ vor:

Diese Codegruppe hat weder einen negativen Einfluss, noch wird sie negativ beeinflusst. Daraus folgt, dass diese Codegruppe keinen Abbruch auslösen kann, sie wird in der Permutation nach hinten gereiht

- Der Datensatz hat in der Spalte Einfluss den Wert „-1“:

Da dieser Wert besagt, dass der entsprechende Code zu einem Abbruch führen könnte, müssen diese Fälle gesondert untersucht werden. Sicher ist, dass für einen Abbruch mindestens 2 Code und somit mindestens 3 Codegruppen (Auslösender Code und mindestens 2 Code für Abbruch) benötigt werden.

Treten nur Datensätze mit dem auslösenden Code und einer beeinflussenden Codegruppe auf, wird diese Codebedingung keinen Abbruch auslösen, folglich können diese Codegruppen für die Permutation nach hinten verschoben werden (Unter der Voraussetzung, dass sie nicht durch eine andere Codebedingung einen Abbruch auslösen).

Treten mehrere beeinflussende Codegruppen mit einer Codebedingung auf, muss überprüft werden, ob diese ausreichen um eine BBP zu verletzen. Ist dies der Fall, werden die Codegruppen in der BBA für die BBP herangezogen, andernfalls werden sie ebenfalls nach hinten verschoben.

Die Analyse der Tabelle 5.9-7 nach oben genannten Punkten ermöglicht eine Neuordnung der Codegruppen für die BBA und kann in zwei Gruppen unterteilt werden:

- Codegruppen lösen eine Verletzung der BBP aus:

Aus den Erkenntnissen der BBA-Version 1 bis 4 werden diese Codegruppen nach vorne gereiht, die Reihung innerhalb dieser Gruppe kann nach aufsteigenden #CG oder nach anderen Kriterien optimiert werden.

- Codegruppen werden die BBP nicht verletzen:

Diese Codegruppen werden in der BBA nach hinten gereiht und für diese muss keine BBP mehr durchgeführt werden

Ist für die BBA ein qualitatives Ergebnis gefordert, wird die erste Gruppe mit eingebundener BBP analysiert, ab der ersten Ebene mit einer Codegruppe aus der zweiten Gruppe wird keine BBP mehr benötigt und es kann einfach auspermutiert werden.

Ist nur ein quantitatives Ergebnis gefordert, reicht es aus eine BBA für die Codegruppen der ersten Gruppe durchzuführen und die Anzahl der baubaren Varianten mit dem Produkt der restlichen #CG zu multiplizieren.

Für die Daten in Tabelle 5.9-7 ergeben sich die beiden Gruppen wie folgt:

Gruppe 1 enthält die Codegruppen CG1, CG5, CG6, CG7, CG8 und CG9.

Gruppe 2 enthält die Codegruppen CG2 und CG3 (kommen beide nicht in der Tabelle vor) sowie CG4 und CG10 (kommen in der Tabelle vor, allerdings nur mit Einfluss „-1“ und CGX fehlt für Abbruch)

In diesem speziellen Beispiel ist weiters zu erkennen, dass CG9 von sehr vielen Codegruppen mit „-2“ beeinflusst wird, daher wird diese Codegruppe, obwohl sie eine hohe Anzahl an Coden

hat, nach vorne gereiht um früh zu Abbrüchen zu führen, danach kommen alle Codegruppen, die CG9 beeinflussen nach aufsteigendem #CG sortiert, die restlichen CG aus Gruppe 1 nach aufsteigendem #CG und schließlich die Codegruppen aus Gruppe 2 (ohne BBP) daraus ergibt sich eine Sortierung wie in Tabelle 5.9-8.

Tabelle 5.9-8: Sortierung der Codegruppen nach ihrer Einflussart

Ebene	1	2	3	4	5	6	7	8	9	10
BBP	Gruppe 1						Gruppe 2			
CG	CG9	CG5	CG6	CG8	CG7	CG1	CG3	CG2	CG10	CG4
#CG	15	2	3	5	4	4	1	7	8	8
#VR	15	30	90	450	1800	7200	7200	50400	403200	3225600
#BBP	0	30	120	570	2370	9570	9570	9570	9570	9570

Aus Tabelle 5.9-8 ersichtlich kann durch verschieben der Codegruppen und differenzieren zwischen der Gruppe 1 und Gruppe 2 die Anzahl der nötigen BBP merklich reduziert werden, auch wenn es nur wenige Abbrüche geben sollte. Allerdings ist ein hohes Verständnis der Datensätze, sowie aufwändiges pre-processing notwendig.

In Tabelle 5.9-9 sind nochmals die Anforderungen und Eigenschaften der unterschiedlichen BBA-Versionen zusammengefasst.

Tabelle 5.9-9: Gegenüberstellung der unterschiedlichen BBA Versionen

BBA	Anforderungen	Eigenschaften
Version 1	<ul style="list-style-type: none"> ■ BBP Typ 2 ■ Kein Verständnis der Daten notwendig ■ Codegruppen mit zugehörigen Coden 	<ul style="list-style-type: none"> ■ Komplette Auspermutation mit anschließender BBP ■ Lange Rechenzeiten bei großer Variantenvielfalt
Version 2	<ul style="list-style-type: none"> ■ Anforderungen von Version 1 ■ Konfigurationsmanagement mit unterschiedlicher Freigabe für Baumuster 	<ul style="list-style-type: none"> ■ Auspermutation Baumusterbezogen ■ Einzelne Baumuster betrachtbar ■ Reduzierte Rechenzeit im Vergleich zu Version 1, abhängig von Menge an freigegebenen Coden pro Baumuster ■ Lange Rechenzeiten bei großer Variantenvielfalt
Version 3	<ul style="list-style-type: none"> ■ Anforderungen von Version 1 oder Version 2, aber ■ Verständnis der Daten notwendig um Effizienz abschätzen zu können ■ Integration von BBP in Permutation (kompliziertere Algorithmen) 	<ul style="list-style-type: none"> ■ Durch integrierte BBP Abbruch während Permutation möglich ■ Codegruppen aufsteigend nach #CG geordnet ■ Nur effizienter als Version 1 oder 2 wenn genügend BBP-Verletzungen in frühen Ebenen auftreten
Version 4	<ul style="list-style-type: none"> ■ Anforderungen von Version 3, aber ■ Sehr hohes Verständnis der Daten notwendig um gegenseitiges Beeinflussen der Codegruppen analysieren zu können ■ Code-Einfluss Analyse ■ Integration von BBP und Sortieralgorithmen in Permutation 	<ul style="list-style-type: none"> ■ Nur noch BBP in Permutationsebenen, in denen Verletzungen auftreten können ■ In allen Fällen schnellste Version, aber ■ Sehr kompliziert in Anwendung, Programmierung und Pre-Processing

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Ziel der Arbeit war es, herauszufinden wie Anhand einer bestehenden Produktdokumentation aus PDM- und ERP-Systemen Varianten- und Komplexitätsmanagement möglich ist.

Es sollte untersucht werden wie die in der Literatur beschriebenen Analysen in der Praxis angewendet werden können, welche Methoden als Grundlage für die Analysen notwendig sind, welche zusätzliche Analysen mit diesen Methoden möglich sind, ob sich aus den Analysen geeignete Indizes für die Bewertung ableiten lassen und welche mögliche Maßnahmen sich aus den Analysen ergeben können.

Es hat sich gezeigt, dass die theoretischen Grundlagen speziell für die Anwendung in der Automobilindustrie adaptiert werden müssen, da dort in der Praxis andere Notationen bzw. Begriffe verwendet werden (vgl. Kapitel 2.3.2 und Kapitel 2.1.5) und, dass sich in der Literatur unterschiedliche Auffassungen von Komplexität, Variantenvielfalt und Varianz vorfinden.

Aufgrund der unterschiedlichen vorherrschenden Dokumentationssysteme, in Form von verschiedenen Stücklisten (Gleichteilstückliste, Plus-Minus-Stückliste, Variantenstückliste, Maximalstückliste, vgl. Kapitel 2.2.2), Konfigurationsbedingungen (positiv Vercodung bzw. negativ Vercodung) etc. wurde versucht eine möglichst allgemeine Beschreibung der Methoden niederzuschreiben, die für alle Dokumentationssysteme gültig ist.

Als größte Herausforderung für das Dokumentationsbasierte Varianten- und Komplexitätsmanagement hat sich die Verknüpfung zwischen Produktstruktur auf Code und Sachnummernbasis gezeigt, da einfach zu analysierende Granularitäten (z.B. ABC-Analyse auf Basis der Code-takerate) geringe Effekte auf die Sachnummernvielfalt haben. Die Ergebnisse der Analysen auf Sachnummernbasis können zwar hohe Effekte auf die Sachnummernvielfalt haben, um diese Effekte zu erzielen sind jedoch Codeseitig Maßnahmen erforderlich, welche noch nicht eindeutig definiert werden konnten.

Folgende Methoden für Schwerpunktlegerung und qualitativen Ergebnissen wurden identifiziert:

- Identifikation näher zu betrachtender Umfänge
 - ABC-Analyse (Kapitel 3.7)
 - DSM Anwendungen (Kapitel 5.3)
 - Eigenschaft-Einfluss-Analyse (Kapitel 5.5)
 - Merkmalbaum (Kapitel 5.2)
- Analysen mit Quantitativen und Qualitativen Ergebnissen, welche direkten Einfluss auf die Variantenvielfalt haben
 - Exklusivitätsanalyse (Kapitel 5.1)
 - Baubarkeitsanalyse (Kapitel 5.9)

- Reduzierung der Produktbeschreibenden Codekette auf Layoutbildende Code, um Stückzahlen dieser Layouts zu ermitteln(Kapitel 5.6)

Weiters wurden die theoretischen Grundlagen erarbeitet, nach welchen Verfahren die theoretische Baubarkeit und somit externe Variantenvielfalt eines Produktprogramms analysiert werden kann (Kapitel 5.9). Diese BBA funktioniert in der aktuellen Version allerdings nur für Systeme mit negativer Vercodung. In einem weiteren Schritt muss die BBA für andere Dokumentationssysteme angepasst werden.

6.2 Ausblick

In diesem Kapitel werden mögliche Maßnahmen erläutert, die aus den Ergebnissen der Analysen getroffen bzw. abgeleitet werden können.

6.2.1 Aktiv/Passiv Unterteilung

Unter aktiven Produkten werden hochgängige Produkte verstanden, also jene Produkte, die in einer ABC-Analyse in die Kategorie A bis B fallen. Passive Produkte hingegen sind Produkte der Kategorie B bis C, wo genau die Grenze gezogen wird, hängt von der Managementstrategie ab.

Bei der Unterteilung nach aktiven und passiven Produkten werden diese durch unterschiedliche Prozesse abgefertigt, was unter anderem zu einer Reduktion der Teilevielfalt am aktiven Produktionsprozess, mit folgenden Nebeneffekten führt:

- Entlastung der Logistik
- Erhöhung der Wiederholrate in der Montage → keine Unterbrechung der Lernkurve
- Reduzierung der Nacharbeit
- Weniger Stillstand der Montage

Passive Produkte können zum Teil am aktiven Produktionsprozess partizipieren (aktiver Umfang der passiven Produkte) werden zu einem geeigneten Zeitpunkt aus dem aktiven Prozess in den passiven umgeleitet und dort fertiggestellt.

Die Unterteilung zwischen aktiv und passiv läuft dynamisch ab. Das bedeutet, erfüllen passive Produkte die Kriterien für aktive Produkte, werden sie dem aktiven Prozess zugeordnet und umgekehrt.

Neben den unterschiedlichen Produktionsprozessen kann auch im Änderungsprozess differenziert werden. Als ein Beispiel sei hier genannt, dass Facelift-Änderungen in einem ersten Schritt nur auf das aktive Produktprogramm ausgerollt werden, auf das passive nur bei Bedarf.

Es haben sich folgende Kriterien ergeben, nach denen die Grenze zwischen aktiv und passiv gelegt werden kann:

- Unterscheidung auf Ebene der Layoutbildenden Code (vgl. Kapitel 5.5).

Durch die Layoutbildenden Code ist es möglich, einfach zwischen Hoch- und Niedrigdrehern zu unterscheiden. Es können z.B. alle Layouts, die innerhalb eines bestimmten Zeitraums nur einmal gebaut wurden in Zukunft als passive Layouts festgelegt werden.

- Unterscheidung nach Exklusivitätsindex (vgl. Kapitel 5.4)

Es wird ein bestimmter Grenzwert für den internen Exklusivitätsindex festgelegt, ab dem die Produkte passiv sind.

- Unterscheidung nach Abschichtung (vgl. Kapitel 5.7)

Festlegen eines Prozentsatzes, bei dem die Grenze gezogen wird, welche Sachnummern passiv gesetzt sind, und welche aktiv bleiben. Der Prozentsatz bezieht sich in diesem Fall auf die „Reduktion“ der Produkte, nicht der Sachnummern.

Die Unterscheidung nach Layoutbildenden Coden liefert Produkte bzw. Codeketten, die passiv gesetzt werden. Durch die Exklusivitätsanalyse (vgl. Kapitel 5.1) für diese Codeketten lassen sich die passiven Sachnummern herausfinden.

Die Unterscheidung nach Abschichtung liefert direkt passive Sachnummern.

Die Unterscheidung nach Exklusivitätsindex kann sowohl Sachnummern als auch Code liefern, die passiv gesetzt werden abhängig davon, ob man ihn Codebasiert oder Sachnummernbasiert durchführt.

Werden nun neue Produkte konfiguriert, kann anhand der Produktauflösung festgestellt werden, ob passive Stücklisten für die Produktion notwendig sind, und der notwendige Prozess ausgelöst werden.

Vergleicht man die 3 Unterscheidungsmöglichkeiten miteinander, erkennt man, dass bei gleicher Anzahl an passiv gesetzten Produkten die Menge an passiven Sachnummern stark variiert.

Die Unterscheidung nach Layoutbildenden Coden liefert die kleinste Menge an passiven Sachnummern, da ein exklusives Produkt auf Codeseite nicht zwingend ein exklusives auf Sachnummerneben bedeutet. Gleichzeitig kann es dazu führen, dass einige Sachnummern bald wieder in den aktiven Prozess übernommen werden müssen.

Wird nach dem Exklusivitätsindex unterschieden, zeigt sich eine deutlich größere Menge an passiven Sachnummern, da beim EI bereits darauf eingegangen wird, ob eine Sachnummer niedrigdrehend ist.

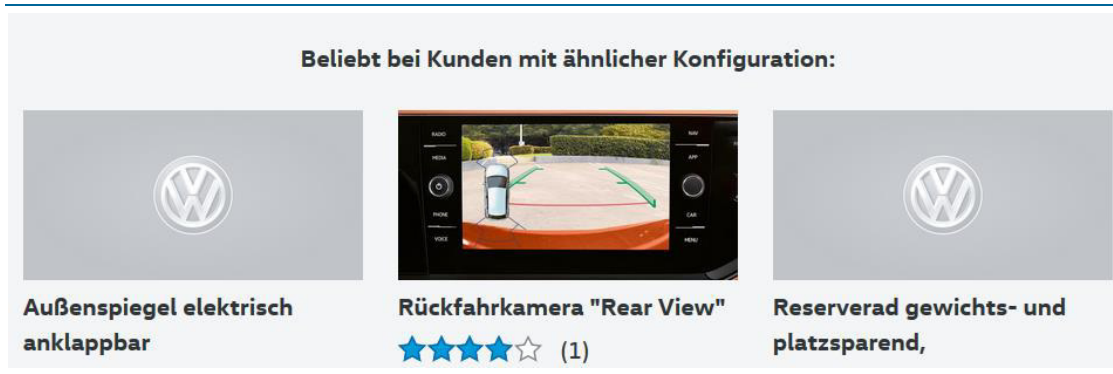
Den größten Effekt erzielt man nach Unterscheidung durch die Abschichtung, da die Abschichtung darauf ausgelegt ist möglichst viele Sachnummern einzusparen, bei gleichzeitig möglichst wenig beeinflussten Produkten.

6.2.2 Geführte Konfiguration

Bei der geführten Konfiguration wird während dem Konfigurationsprozess verglichen, welche bereits bestellten Produkte die gleichen Code gewählt haben und es werden automatisch Code für die weitere Konfiguration vorgeschlagen, die häufig gewählt wurden.

Es handelt sich dabei um eine Methode, die bereits in vielen Industrien verwendet wird, unter anderem von dem Onlinehändler Amazon, sie hat allerdings auch in die Automobilindustrie Einzug gehalten wie Abbildung 6.2-1 von dem Online Konfigurator der Volkswagen AG zeigt.

Abbildung 6.2-1: Geführte Konfiguration im Online-Konfigurator der Volkswagen AG [36]



Ziel der geführten Konfiguration ist es möglichst viele hochdrehende Produkte im Bestellprozess zu erzeugen. Wird die geführte Konfiguration gemeinsam mit der Unterscheidung zwischen aktiven und passiven Produkten angewendet, führt er zu einer Reduzierung der bestellten passiven Produkte, zugunsten der aktiven Bestellmenge.

6.2.3 Indizes für die Bewertung

Für die Bewertung der Produktkomplexität haben sich aus den in dieser Arbeit verwendeten Analysen folgende Bewertungsindizes herauskristallisiert:

- Anzahl verkaufter unterschiedlicher Layouts:

Auf Basis der Layoutbildenden Code wird analysiert wie viele unterschiedliche Layouts in einer Betrachtungsperiode produziert wurden. Anhand dieser Zahl können Ziele gesetzt werden, wie viele es in Zukunft werden sollen und entsprechend Maßnahmen getroffen werden. Es ist vor allem ein Maß dafür, wie gut die geführte Konfiguration funktioniert, da das Ziel sein sollte möglichst wenig aktive Layouts mit einer hohen zugrundeliegenden Stückzahl zu erhalten.
- Anzahl Sachnummern

Die Anzahl der Sachnummern gibt Auskunft über die interne Variantenvielfalt.
- Steigung Exklusivitätsindexgerade

Die Steigung der Geraden des EI-SNR über EI-Code gibt Auskunft über die Qualität der zugrundeliegenden Produktarchitektur, inwiefern das Unternehmen in der Lage ist eine hohe externe Variantenvielfalt mit einer geringen internen Variantenvielfalt abzubilden.

6.2.4 Künstliche Intelligenz

Mit der Methode der künstlichen Intelligenz (KI) wäre es möglich zu erkennen, welche Code bzw. Code-Kombinationen gesperrt werden müssen, um einen möglichst großen Einsparungseffekt bei den Sachnummern zu erhalten.

7 Verzeichnisse

7.1 Literaturverzeichnis

- [1] H. Kesper, *Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden*, 1. Aufl. München: Verl. Dr. Hut, 2012.
- [2] U. Lindemann, R. Reichwald, und M. F. Zäh, Hrsg., *Individualisierte Produkte: Komplexität beherrschen, in Entwicklung und Produktion*. Berlin: Springer, 2006.
- [3] W. Kersten, *Vielfaltsmanagement: integrative Lösungsansätze zur Optimierung und Beherrschung der Produkte und Teilevielfalt*. München: TCW Transfer-Centrum, 2002.
- [4] S. Schlott, „Wahnsinn mit Methode“, *Automobil-Produktion*, Bd. Heft 1, Nr. Heft 1, S. 38–42, Jän-2005.
- [5] G. Schuh und M. Riesener, *Produktkomplexität managen: Strategien - Methoden - Tools*, 3., Vollständig überarbeitete Auflage. München: Hanser, 2018.
- [6] VDI, „Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte“. Beuth, 1993.
- [7] J. Neuhausen, „Methodik zur Gestaltung modularer Produktionssysteme für Unternehmen der Serienproduktion“, RWTH Aachen, Aachen, 2001.
- [8] U. Lindemann, *Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden*, 3., Korrigierte Aufl. Berlin: Springer, 2009.
- [9] *DIN 199-1:2002-03 Technische Produktdokumentation – CAD Modelle, Zeichnungen und Stücklisten – Teil 1: Begriffe*. Berlin: Beuth, 2003.
- [10] A. Fraß, *Automobilzuliefererindustrie in Deutschland*, 1. Aufl. München: Hampp, 2012.
- [11] J. Feldhusen und K.-H. Grote, Hrsg., *Pahl/Beitz Konstruktionslehre*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [12] „Duden | Pro- dukt- port- fo- lio | Rechtschreibung, Bedeutung, Definition“. [Online]. Verfügbar unter: <https://www.duden.de/rechtschreibung/Produktportfolio>. [Zugegriffen: 14-Feb-2018].
- [13] „Definition: Produktportfolio | Onpulson-Wirtschaftslexikon“. [Online]. Verfügbar unter: <http://www.onpulson.de/lexikon/produktportfolio/>. [Zugegriffen: 14-Feb-2018].
- [14] C. Homburg, *Grundlagen des Marketingmanagements: Einführung in Strategie, Instrumente, Umsetzung und Unternehmensführung*, 3., Überarb. und erw. Aufl. Wiesbaden: Springer Gabler, 2012.
- [15] H. Jonas, D. Krause, und J. Gausemeier, *Eine Methode zur strategischen Planung modularer Produktprogramme*, 1. Aufl. Hamburg: TuTech Innovation, 2014.
- [16] G. W. Mair, *Sicherheitsbewertung von Composite-Druckgasbehältern: Potential statistischer Methoden jenseits aktueller Vorschriften*. Berlin Heidelberg: Springer Vieweg, 2016.
- [17] M. Kropik, *Produktionsleitsysteme in der Automobilfertigung*. Berlin: Springer, 2009.
- [18] G. Schuh, W. Stölzle, und F. Straube, Hrsg., *Anlaufmanagement in der Automobilindustrie erfolgreich umsetzen: ein Leitfaden für die Praxis*. Berlin: Springer, 2008.
- [19] K. Ehrlenspiel, A. Kiewert, und U. Lindemann, *Kostengünstig entwickeln und konstruieren: Kostenmanagement bei der integrierten Produktentwicklung*, 5., Bearb. Aufl. Berlin: Springer, 2005.

- [20] H. Seeger, *Design technischer Produkte, Produktprogramme und -systeme: Industrial design engineering*, 2., Bearb. und erw. Aufl. Berlin: Springer, 2005.
- [21] R. Eberle, „Varianten-Controlling in der Automobil-Industrie Rudolf Eberle“, *Control. Manag.*, Bd. 44, Nr. 6, S. 343–350, Nov. 2000.
- [22] M. Oestreich und O. Romberg, *Keine Panik vor Statistik! Erfolg und Spaß im Horrorfach nichttechnischer Studiengänge*, 4., Aktualisierte Aufl. Wiesbaden: Springer Spektrum, 2012.
- [23] F. Klug und Springer-Verlag GmbH, *Logistikmanagement in der Automobilindustrie Grundlagen der Logistik im Automobilbau*. Berlin: Springer Berlin, 2017.
- [24] C.-A. Hoffmann, T. Vietor, U. Dombrowski, und J. Axmann, *Methodik zur Steuerung modularer Produktbaukästen*. Wiesbaden: Springer, 2018.
- [25] C. Zenner, *Durchgängiges Variantenmanagement in der Technischen Produktionsplanung*. Saarbrücken: LFT, Univ, 2006.
- [26] M. Eigner und R. Stelzer, *Produktdatenmanagement-Systeme: ein Leitfaden für Product Development und Life Cycle Management*. Berlin Heidelberg New York Barcelona Hongkong London Mailand Paris Tokio: Springer, 2001.
- [27] S. Geier, *Demand Fulfillment bei Assemble-to-Order-Fertigung: Analyse, Optimierung und Anwendung in der Computerindustrie*. Wiesbaden: Springer-Gabler, 2014.
- [28] Europäische Kommission, *VERORDNUNG (EU) Nr. 19/2011 DER KOMMISSION vom 11. Januar 2011 über die Typp Genehmigung des gesetzlich vorgeschriebenen Fabrik-schildes und der Fahrzeug-Identifizierungsnummer für Kraftfahrzeuge und Kraftfahrzeug-anhänger zur Durchführung der Verordnung (EG) Nr. 661/2009 des Europäischen Parla-ments und des Rates über die Typp Genehmigung von Kraftfahrzeugen, Kraftfahrzeuganhängern und von Systemen, Bauteilen und selbstständigen technischen Einheiten für diese Fahrzeuge hinsichtlich ihrer allgemeinen Sicherheit*. 2011.
- [29] ISO 3779:2009, *Road vehicles - Vehicle identification number (VIN) - Content and struc-ture*. .
- [30] G. Schuh, *Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2006.
- [31] E. Westkämper und C. Löffler, *Strategien der Produktion: Technologien, Konzepte und Wege in die Praxis*. Berlin Heidelberg: Springer Vieweg, 2016.
- [32] J. Ponn und U. Lindemann, *Konzeptentwicklung und Gestaltung technischer Produkte: systematisch von Anforderungen zu Konzepten und Gestaltlösungen*, 2. Aufl., [Gründlich überarb.]. Berlin: Springer, 2011.
- [33] C. Meinel und M. Mundhenk, *Mathematische Grundlagen der Informatik: mathematisches Denken und Beweisen; eine Einführung*, 4., Überarb. Aufl. Wiesbaden: Vieweg + Teubner, 2009.
- [34] M. Danilovic und T. R. Browning, „Managing complex product development projects with design structure matrices and domain mapping matrices“, *Int. J. Proj. Manag.*, Bd. 25, Nr. 3, S. 300–314, Apr. 2007.
- [35] M. S. Maurer, *Structural awareness in complex product design*, 1. Aufl. München: Verl. Dr. Hut, 2007.
- [36] „Der neue Polo - Der Volkswagen Konfigurator“. [Online]. Verfügbar unter: [https://konfigurator.volkswagen.at/cc-at/de_AT_VW/V/equipment/046/AW143V04/5M5M/EL/9S0@9WTUQ5@PF2@6YD@7Y8@WL1PD5@\\$T2@W317X2@PJ6FC1@WW3\\$BJ/@](https://konfigurator.volkswagen.at/cc-at/de_AT_VW/V/equipment/046/AW143V04/5M5M/EL/9S0@9WTUQ5@PF2@6YD@7Y8@WL1PD5@$T2@W317X2@PJ6FC1@WW3$BJ/@). [Zugegriffen: 26-Feb-2018].

7.2 Abbildungsverzeichnis

Abbildung 2.1-1: Variantentreiber der Windschutzscheibe	9
Abbildung 2.1-2: Interne vs. externe Komplexität[5]	14
Abbildung 2.2-1: Funktionsstruktur	15
Abbildung 2.2-2: Produktstruktur.....	15
Abbildung 2.2-3: Produktarchitektur [11].....	16
Abbildung 2.2-4: METUS-Raute[11].....	16
Abbildung 2.2-5: Beispiel eines Gozinto-Graphen	17
Abbildung 2.2-6: Stücklistendarstellungen	18
Abbildung 2.2-7: Vollständige Auflösung der Varianten P1, P2 und P3	19
Abbildung 2.2-8: Gleichteilstückliste der Varianten P1, P2 und P3	20
Abbildung 2.2-9: Plus-Minus-Stückliste und Variantenstückliste der Varianten P1, P2 und P3	20
Abbildung 2.2-10: Variantenleisten und Variantenausprägungen P1 und P3	21
Abbildung 2.2-11: Maximalstückliste mit Konfiguration von Variante P3	21
Abbildung 2.2-12: PDM-System, IT-Infrastruktur [11].....	24
Abbildung 2.2-13: Verknüpfung von PDM- und ERP-Systemen [11].....	25
Tabelle 2.3-1: Wahrheitstabelle einfacher, verknüpfter Aussagen	27
Tabelle 2.3-2: Wahrheitstabelle verknüpfter komplexer Aussagen	28
Tabelle 3.1-1:Gegenüberstellung Notationen der booleschen Operatoren	30
Tabelle 3.2-1: Wahrheitstabelle von $CD1 + -CD2 + (CD3/-CD4)$	32
Tabelle 3.2-2: Beschreibung der Einflussarten von Coden auf boolesche Funktionen	32
Tabelle 3.2-3: Codebedingung mit beeinflussenden Coden und Einflussart	33
Tabelle 3.2-4: Ausgeführte Einflussanalyse der booleschen Ausdrücke	33
Abbildung 3.3-1: Methodenablauf Produktauflösung	34
Abbildung 3.4-1: Produkt-Code-DMM und Produkt-Komponenten-DMM.....	35
Abbildung 3.5-1: DSM zur Darstellung von Eigenschaftenkombinationen [1]	35
Abbildung 3.5-2: Aggregierte DSM nach Kesper [1].....	36
Abbildung 3.6-1: Clusterbildung	38
Abbildung 3.7-1: ABC-Analyse in Säulenform	39
Abbildung 3.7-2: ABC-Analyse in aggregierter Form.....	39
Tabelle 4.1-1: Mögliche Dokumentation eines Produktionsprogramms	41
Tabelle 4.1-2: Beispiel für eine Codeauflistung	42
Tabelle 4.1-3: Codebedingungen für gewählte Code	44
Tabelle 4.1-4: Baumuster und Paketcode mit auslösenden Coden.....	44
Tabelle 4.1-5: Codebedingungen für gewählte Baumuster-Code-Kombinationen	45
Tabelle 4.1-6: Transformationsbeziehung mit Codebedingungen	46
Tabelle 4.1-7: Transformationsbeziehung mit Baumuster-Codebedingungen	46

Tabelle 5.1-1: Code für Exklusivitätsanalyse	49
Tabelle 5.1-2: Codebedingungen mit Ersetzen Coden und ausgewerteter Erfüllbarkeit	50
Abbildung 5.1-1: Ablauf Exklusivitätsanalyse	51
Abbildung 5.2-1: Merkmalbaum mit hinterlegtem Absatz/Umsatz	52
Abbildung 5.2-2: Merkmalbaum mit markierten Niedrigdrehern	52
Abbildung 5.2-3: Merkmalbaum mit Niedrigdrehern, umsortiert	53
Abbildung 5.3-1: DMM Multiplikation für Paketbildung	54
Abbildung 5.3-2: Bildung von TR^{-1} aus der Takerate	54
Abbildung 5.3-3: Berechnung der normierten Code-DSM	55
Abbildung 5.3-4: Bildung der Diagonalmatrix BST aus den Bestellmengen	55
Abbildung 5.3-5: Bildung der gewichteten Produkt-Code-DMM mittels Bestellmengen	56
Abbildung 5.3-6: Multiplikation für die gewichtete, aggregierte Code-DSM	56
Abbildung 5.3-7: Berechnung der normierten, gewichteten, aggregierten Code-DSM	57
Abbildung 5.3-8: Vergleich PKT_{norm} und $PKT_{gew. norm}$	57
Abbildung 5.3-9: Mögliche Extremfälle bei der Paketbildung	58
Abbildung 5.3-10: IVC Multiplikation	59
Abbildung 5.3-11: Bildung des Vektors AC und TR	60
Abbildung 5.3-12: IVC und normierte IVC	60
Abbildung 5.3-13: Bildung der Takerate Diagonalmatrix	61
Abbildung 5.3-14: Berechnung PC_{gew}	61
Abbildung 5.3-15: Berechnung IVC_{gew}	62
Abbildung 5.3-16: Berechnung $IVC_{gew, norm}$	62
Abbildung 5.3-17: Gegenüberstellung IVC_{norm} und $IVC_{gew, norm}$	63
Abbildung 5.3-18: IVC für Layouts eines Produktionsprogramms	63
Abbildung 5.3-19: Geclusterte IVC aus Abbildung 5.3-18	64
Abbildung 5.3-20: Stücklisten-DMM	64
Abbildung 5.3-21: ISC für Stücklisten, die für ein BM freigegeben sind	65
Abbildung 5.4-1: Gewichtete Konfiguration verglichen mit fiktivem Produkt	65
Abbildung 5.4-2: EI Berechnung, extreme Beispiele	67
Abbildung 5.4-3: EI Berechnung, weitere Beispiele	67
Abbildung 5.4-4: Gegenüberstellung EI_{CD} und EI_{SNR}	68
Abbildung 5.4-5: Trendlinien für Gegenüberstellung von EI_{CD} und EI_{SNR}	69
Tabelle 5.5-1: Beispiel Codebedingung für Eigenschaft-Einfluss-Analyse	70
Tabelle 5.5-2: Code-Einfluss-Tabelle	70
Tabelle 5.5-3: Reduzierte Code-Einfluss-Tabelle	71
Tabelle 5.5-4: Anzahl positiv beeinflusster Codebedingungen	71
Abbildung 5.5-1: Ablauf Eigenschaft-Einfluss-Analyse	71
Tabelle 5.5-5: Variantentreibende Code	72
Abbildung 5.5-2: Code Takerate vs. beeinflusste Sachnummern	72

Tabelle 5.5-6: Variantentreibende Codegruppen	73
Tabelle 5.5-7: Beeinflussende Codegruppen für den Entwicklungsumfang Abgasrohr	73
Tabelle 5.6-1: Codegruppen für ein einfaches Nutzfahrzeug Layout	75
Abbildung 5.6-1: Schematische Darstellung eines einfachen Nutzfahrzeug-Layouts	75
Tabelle 5.6-2: Codegruppen für ein detailliertes Nutzfahrzeug Layout.....	76
Abbildung 5.6-2: Schematische Darstellung eines detaillierten Nutzfahrzeug-Layouts	76
Tabelle 5.6-3: Layouts für Entwicklungsumfang Abgasrohr.....	77
Abbildung 5.6-3: ABC-Analyse der Layouts für den Entwicklungsumfang Abgasrohr	77
Abbildung 5.8-1: Schematischer Ablauf der Baubarkeitsprüfung	79
Tabelle 5.9-1: Codegruppen mit Anzahl der Code.....	80
Abbildung 5.9-1: Permutation Baumstruktur	80
Tabelle 5.9-2: Code, freigegeben für Baumuster und theoretische Variantenvielfalt	81
Abbildung 5.9-2: Entfallene Permutationen durch Abbruch in einer Ebene.....	82
Tabelle 5.9-3: #VR und #BBP pro Ebene; Entfall #VR und Ersparnis #BBP abhängig von Ebene mit BBP-Verletzung.....	84
Tabelle 5.9-4: Ersparnis #BBP bei Hälfte baubar, abhängig von entdeckter Ebene.....	84
Tabelle 5.9-5: #VR und #BBP, abhängig von der Permutationsreihenfolge.....	85
Abbildung 5.9-3: Anstieg von #VR, abhängig von der Permutationsreihenfolge	85
Abbildung 5.9-4: Anstieg von #BBP, abhängig von der Permutationsreihenfolge.....	86
Tabelle 5.9-6: Entfallene #VR und eingesparte #BBP, abhängig von Permutationsreihenfolge.....	86
Abbildung 5.9-5: #VR entfallen, abhängig von Permutationsreihenfolge	87
Abbildung 5.9-6: #BBP eingespart, abhängig von Permutationsreihenfolge	87
Tabelle 5.9-7: Auslösende und beeinflussende Code für BBA	88
Tabelle 5.9-8: Sortierung der Codegruppen nach ihrer Einflussart	90
Tabelle 5.9-9: Gegenüberstellung der unterschiedlichen BBA Versionen.....	91
Abbildung 6.2-1: Geführte Konfiguration im Online-Konfigurator der Volkswagen AG [36].....	95