

Modell zur Integrationsplanung variantenreicher
Baukastenprodukte in einem Montagesystemverbund

.....

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultät
der Universität des Saarlandes

von
Lukas Hermanns

Saarbrücken
2019

Tag des Kolloquiums: 06.11.2019

Dekan: Univ.-Prof. Dr. rer. nat. Guido Kickelbick

Berichterstatter: Prof. Dr.-Ing. Rainer Müller

Prof. Dr.-Ing. Jochen Deuse

Vorsitz: Prof. Dr.-Ing Georg Frey

Akad. Mitarbeiter: Dr.-Ing. Paul Motzki

Danksagung

Diese Arbeit entstand im Rahmen meiner Tätigkeit im Fachbereich Planung Motor Montage bei der BMW Group in München sowie am Zentrum für Mechatronik und Automatisierungstechnik (ZeMA) in Saarbrücken.

Besonders bedanken möchte ich mich bei Prof. Dr.-Ing. Rainer Müller, Lehrstuhlleiter und wissenschaftlicher Geschäftsführer des ZeMA, für die wissenschaftliche Betreuung, Begleitung und Förderung. Ich bedanke mich sehr bei Prof. Dr.-Ing. Jochen Deuse, Leiter des Instituts für Produktionssysteme (IPS) der TU Dortmund, für die Übernahme des Korreferates. Für die Übernahme des Prüfungsvorsitzes bedanke ich mich herzlich bei Prof. Dr.-Ing. Georg Frey, dem Leiter des Lehrstuhls für Automatisierungs- und Energiesysteme (AES) der Universität des Saarlandes.

Fachlich und wissenschaftlich begleitet haben diese Arbeit außerdem meine Kollegen der BMW Group. Hier möchte ich mich insbesondere bei meinem internen Betreuer Dr. Nils Brunner bedanken. Einen großen Anteil am Gelingen dieser Arbeit haben auch Dr. Jan Eilers, Dr. Artur Bellmann und Raphael Gerdes durch ihr Feedback und ihre Anmerkungen zu dieser Arbeit. Bedanken möchte ich mich auch bei Marco Poitsch für die Unterstützung und Freiräume, die diese Arbeit ermöglicht haben.

Einen weiteren Dank möchte ich an unsere studentischen Mitarbeiter aussprechen, die mich während meiner Promotionszeit begleitet haben.

Vielen Dank auch an meine Eltern für die Förderung, Geduld und Motivation – sowohl während der Promotion als auch auf dem Weg dorthin.

München, Juli 2020

Zusammenfassung

Produzierende Unternehmen befinden sich in einem herausfordernden Umfeld. Um die Ansprüche der Kunden bestmöglich zu erfüllen, werden zahlreiche Produktvarianten entwickelt. Diese stellen für die Montageplanung eine Herausforderung dar.

Durch den Einsatz von Baukästen können die Auswirkungen von Variantensteigerungen teilweise reduziert werden, jedoch bringt die dadurch entstehende starke Vernetzung der Produkte eine Komplexitätssteigerung mit sich. Weiter erhöht wird die Komplexität, wenn ein Produktionsnetzwerk mit mehreren Montagesystemen geplant werden soll.

Es gibt Softwaresysteme und wissenschaftliche Ansätze, die diese Komplexität in Modellen abzubilden. Jedoch waren diese nicht in der Lage die gesamte Vernetzung von Produkt, Prozess und Ressource modellieren und deren Verknüpfungen effizient zu erfassen.

Daher wurde in dieser Arbeit ein Modell entwickelt, welches alle relevanten Zusammenhänge zwischen Produkt, Prozess und Ressource abbildet. Regelbasierte Verknüpfungen wurden verwendet, um den manuellen Pflegeaufwand zu minimieren. Weiterhin wurde eine Methodik entwickelt, die die Montageplanung unter Zuhilfenahme des Modells deutlich vereinfacht.

Die Funktionsfähigkeit des Modells konnte mithilfe eines vereinfachten, theoretischen Anwendungsfalls nachgewiesen werden. Abschließend wurde das Modell anhand eines praktischen Anwendungsfalls getestet und validiert.

Abstract

Producing companies are facing a challenging environment. To satisfy the demands of their customers they develop many product variants. Those are a challenge for the assembly planning departments.

The usage of modular products partly reduces the effects of this variant increase, but the interdependence that is created with this leads to an increased complexity. The complexity is risen further if the assembly planning is to be done for a network of assembly systems.

There are software systems and scientific approaches do implement a model for this complexity. However none of those were able to fully and efficiently include all the dependencies of product, process and resource.

In this thesis a model was developed, that included all interrelations between product, process and resource. Rule-based links were used to keep manual maintenance effort to a minimum. Furthermore a method was developed, which simplifies assembly planning by using the developed model.

The functionality of the model was shown in a simplified, theoretical use-case. Subsequent the model was tested and validated in a practical, real-life use case.

Inhaltsverzeichnis

Content

1	Einleitung	1
1.1	Problemstellung	1
1.2	Zielsetzung	3
1.3	Vorgehensweise	4
2	Grundlagen und praktischer Handlungsbedarf	7
2.1	Produktentstehungsprozess und Baukastenprodukte	7
2.2	Montage und Montageplanung	12
2.2.1	Aufgaben der Montage	12
2.2.2	Montageplanung	16
2.3	Produktanalyse für variantenreiche Baukastenprodukte	21
2.3.1	Varianten in der Produktentwicklung	21
2.3.2	Varianteninformationen in der Produktanalyse	25
2.4	Defizit in der Praxis	27
2.5	Zwischenfazit und praktischer Handlungsbedarf	30
3	Bestehende Modelle zur Planung von Montagesystemen	34
3.1	Grundüberlegungen zur Modellierung	34
3.2	Modellierungssprachen	39
3.2.1	Entity-Relationship-Model (ERM)	40
3.2.2	UML-Klassendiagramm und Objektdiagramm	41
3.2.3	EXPRESS-G	45
3.3	Datenbanken und SQL	46
3.4	Digitale Fabrik	49
3.5	Modelle zur Abbildung von Produkt, Prozess und Ressource	51
3.5.1	Modulares Informationsmanagement nach STEINWASSER (1997)	51
3.5.2	Durchgängige, rechnergestützte Planung nach JONAS (2000)	52
3.5.3	Modulare Produktionssysteme nach NEUHAUSEN (2001)	54
3.5.4	Feature-basierte Prozesskettenplanung nach FRANKE (2003)	56
3.5.5	Die virtuelle, offene Fabrik nach KLAUKE (2002)	58
3.5.6	Das Plattformkonzept für die Planung von Montagesystemen nach BRUNNER (2015)	58
3.6	Zusammenfassung und Herleitung des Handlungsbedarfs	61
4	Entwicklung des Modells zur Integrationsplanung	65
4.1	Methode der Integrationsplanung	65
4.2	Entwicklung Basis-Datenmodell	67
4.2.1	Partialmodell Produkt	69
4.2.2	Partialmodell Prozess	71
4.2.3	Partialmodell Ressource	72

4.2.4	Erläuterungen zum Basis-Datenmodell.....	74
4.3	Erweiterung des Basis-Datenmodells	76
4.3.1	Erweiterung um Anforderungen und Fähigkeiten	76
4.3.2	Erweiterung um Kapazitäten an Bauteilvarianten.....	79
4.3.3	Erweiterung um Ansprechpartner.....	81
4.4	Verwendung des Modells in der Montageplanung	83
4.4.1	Bauteilvarianten in Produktgenerationen	84
4.4.2	Bauteilvariantenentwicklung.....	85
4.4.3	Auswertung Produktkommunalität.....	87
4.4.4	Auswertung Stationskapazität.....	88
4.4.5	Analyse von Exklusivbauteilvarianten	90
4.4.6	Auswertung und Abgleich von Anforderungen und Fähigkeiten.....	91
4.5	Gesamtbetrachtung zum Modell zur Integrationsplanung	94
5	Implementierung und Auswertungen anhand der Anwendungsbeispiele ...	99
5.1	Implementierung Datenmodell	99
5.1.1	Umsetzungsalternativen.....	99
5.1.2	Detaillierung des UML-Datenmodells.....	100
5.1.3	Erstellung der Datenbank-Tabellen.....	104
5.1.4	Umsetzung Variantenregeln.....	105
5.2	Funktionale Validierung anhand einer fiktiven Kugelschreiber-Fabrik.....	107
5.2.1	Einführung in das fiktive Beispiel der Kugelschreiber-Fabrik	108
5.2.2	Befüllung des Datenmodells.....	110
5.2.3	Abfrage der Zuordnungen	112
5.2.4	Abfrage von Zeiträumen.....	116
5.2.5	Abgleich Anforderungen und Fähigkeiten	118
5.2.6	Abgleich von Kapazität.....	122
5.2.7	Kombination aller Abfragen	123
5.2.8	Analyse von Szenarien durch Änderungen im Datenmodell	125
5.3	Einsatz des Modells in der Montageplanung für Baukastenmotoren.....	132
5.3.1	Rahmenbedingungen in der Motorenmontage	133
5.3.2	Baukastenmotoren	133
5.3.3	Herausforderungen beim praktischen Einsatz des Modells	135
5.3.4	Variantenmanagement und Stationskapazität.....	138
5.3.5	Anpassung der Linienbelegung.....	143
5.3.6	Voruntersuchung Prozessabsicherung Montage.....	147
5.4	Implementierung im Rahmen des SmartF-IT Projektes.....	148
5.5	Zusammenfassung Implementierung und Anwendungsbeispiel	148
6	Zusammenfassung, Fazit und Ausblick	151
6.1	Zusammenfassung.....	151
6.2	Fazit	153
6.3	Ausblick und weiterer Forschungsbedarf	155

7	Literaturverzeichnis.....	157
8	Anhang	165

Formelzeichen und Abkürzungsverzeichnis

Formula Symbols and Abbreviations

<i>Kurzzeichen</i>	<i>Benennung</i>
B38	Dreizylinder Ottomotor (vgl. Variantencode Abschnitt 5.3.2)
B48	Vierzylinder Ottomotor (vgl. Variantencode Abschnitt 5.3.2)
CSV	Comma separates values / kommasetrennte Werte
DIN	Deutsches Institut für Normung
DMC	Data Matrix Code
EOP	End of Production / Ende der Serienproduktion
ERM	Entity-Relationship-Model
MEP	Montagesystementstehungsprozess
PEP	Produktentstehungsprozess
PKW	Personenkraftwagen
PVL	Produktionsvorläufer
REFA	Verband für Arbeitsgestaltung, Betriebsorganisation und Unternehmensentwicklung
SOP	Start of Production / Start der Serienproduktion
UML	Unified Modeling Language / vereinheitlichte Modellierungssprache
VDI	Verein deutscher Ingenieure

Abbildungsverzeichnis

List of figures

Bild 1.1:	Herausforderungen bei der Montagesystemgestaltung [FELD14, S. 5]	2
Bild 1.2:	Wissenschaftssystematik nach ULRICH [ULRI76, S. 305]	4
Bild 1.3:	Forschungsprozess nach ULRICH [ULRI81] und Aufbau der Arbeit	5
Bild 2.1:	Produktkreislauf nach STEINWASSER [STEI97, S. 4].....	7
Bild 2.2:	Produktlebenszyklus nach EIGNER [EIGN09a, S. 2]	8
Bild 2.3:	PEP-Planung nach Schneider [SCHN12, S. 159].....	9
Bild 2.4:	Montage und Montageplanung in der innerbetrieblichen Wortschöpfungskette [HOER19 in Anlehnung an SPUR86].....	10
Bild 2.5:	Kostenfestlegung und Kostenverursachung nach VDI (Daten aus [VDI87, S. 3]).....	10
Bild 2.6:	Definition der Arbeitsvorbereitung nach REFA [WIEN10, S. 195]	11
Bild 2.7:	Basisaufgaben der Montage [MUEL13, S. 705]	14
Bild 2.8:	Einsatzbereiche manueller, hybrider und automatischer Montagekonzepte nach LOTTER [LOTT12c, S. 168]	15
Bild 2.9:	Montagesystementstehungsprozess nach EILERS [EILE15, S. 9].....	16
Bild 2.10:	Planungssystematik nach KONOLD UND REGER [KONO03, S. 32]	17
Bild 2.11:	Planungssystematik nach LOTTER [LOTT92].....	18
Bild 2.12:	Planungsgrundfälle in der Fabrikplanung nach KAMPKER [KAMP11, S. 304]	20
Bild 2.13:	Fähigkeitsbasierte Integrationsplanung nach MÜLLER [MUEL15, S. 554]	20
Bild 2.14:	Merkmale von Komplexität nach RENNER [RENN07].....	24
Bild 2.15:	Einfaches Beispiel für einen Variantenbaum (eigene Darstellung in Anlehnung an [EVER89; BECK12, ZAGE06])	25
Bild 2.16:	Eingangsdaten für die Montageplanung.....	31
Bild 3.1:	Arten von Modellen nach WERNERS mit eigenen Beispielen [WERN13, S. 4].....	35
Bild 3.2:	Modelltypen der verschiedenen Autoren	36
Bild 3.3:	Modellgestützter Problemlöseprozess in Anlehnung an WERNERS [WERN13, S. 2].....	38
Bild 3.4:	Systemtechnischer Lösungsweg nach Katzy [KATZ94, S. 53]	39
Bild 3.5:	Beispiel für ein Entity-Relationship Diagramm [CHEN76, S. 19]	40
Bild 3.6:	Darstellung von Attributen im ERM [BECK12, S. 8]	40
Bild 3.7:	Beispiel für eine Klasse eines UML Klassendiagramms.....	41
Bild 3.8:	Beispiel für eine Komposition [BECK12, S. 12]	42

Bild 3.9:	Beispiel für eine Aggregation [BECK12, S. 12].....	43
Bild 3.10:	Beispiel für eine Vererbung [BECK12, S. 13]	44
Bild 3.11:	Beispiel für ein Objektdiagramm auf Basis des Klassendiagramms aus Bild 3.9	45
Bild 3.12:	Beispiel zur Verwendung von EXPRESS-G aus ISO10303-11 [ISO04]	46
Bild 3.13:	Module Produkt, Prozess und Ressource nach STEINWASSER [STEI97, S. 51].....	52
Bild 3.14:	Gesamtmodell zur rechnergestützten Planung nach JONAS [JONA00, S. 103].....	53
Bild 3.15:	Partialmodelle nach NEUHAUSEN [NEUH01, S. 59]	56
Bild 3.16:	Zusammenhänge von Produkt, Prozess und Ressource nach FRANKE [FRAN03, S. 60]	56
Bild 3.17:	Datenstruktur der Feature-basierten Prozesskettenplanung nach FRANKE [FRAN03, S. 63]	57
Bild 3.18:	Grobstruktur der „Offenen Virtuellen Fabrik“ nach KLAUKE [KLAU02, S. 55]	58
Bild 3.19:	Datenmodell für das Plattformkonzept für die Planung von Montagesystemen nach BRUNNER [BRUN15, S. 78]	59
Bild 4.1:	Fähigkeitsbasierte Montageplanung erweitert um Stellhebel der Montageplanung [MUEL15, S. 554]	66
Bild 4.2:	Beispiel für die Funktionsweise der Variantenregel.....	70
Bild 4.3:	UML-Modell des Partialmodell Produkt	71
Bild 4.4:	UML-Klassendiagramm erweitert um Prozess	72
Bild 4.5:	Partialmodell Ressource	73
Bild 4.6:	Variantenregel am Beispiel der Linienbelegung	73
Bild 4.7:	Das Basis-Datenmodell zur fähigkeitsbasierten Montageplanung [MUEL15, S. 555]	74
Bild 4.8:	Problematik bei der Zuordnung von Anforderungen und Fähigkeiten	77
Bild 4.9:	Integration von Anforderungen und Fähigkeiten ins Basis-Datenmodell	78
Bild 4.10:	Einbindung der Kapazität in das Datenmodell.....	80
Bild 4.11:	Anstellung am Beispiel der Zylinderkopf-Haube.....	80
Bild 4.12:	Integration von Ansprechpartnern in das Basis-Datenmodell.....	82
Bild 4.13:	Gesamt-Datenmodell	83
Bild 4.14:	Absolute Bauteilvielfalt zweier Produktgenerationen.....	85
Bild 4.15:	Darstellung der Bauteilvarianten-Entwicklung von der ersten Produktgeneration zur zweiten Produktgeneration (in Anlehnung an [MUEL17, S. 256])	87

Bild 4.16:	Beispielhafter zeitlicher Verlauf der Bauteilvariantenauslastung einer Station	90
Bild 4.17:	Mengen der Bauteilvarianten für zwei Mengen von Produktvarianten (Bild in Arbeit).....	91
Bild 4.18:	Zuordnung der Eingangsdaten zu den Klassen des Datenmodells	94
Bild 4.19:	Zuordnung der Auswertungen zu den Stellhebeln des Planungsprozesses	95
Bild 5.1:	Ausführliches UML Klassendiagramm zur Implementierung	103
Bild 5.2:	Aufbau des fiktiven Produktes Kugelschreiber	108
Bild 5.3:	Variantenbaum der Produktvielfalt des Kugelschreiber-Beispiels	108
Bild 5.4:	Ausgangssituation im Kugelschreiber-Szenario	109
Bild 5.5:	Variantenregel für das fiktive Kugelschreiber-Beispiel	110
Bild 5.6:	Parallelgreifer Festo DHPS-20-A als Beispiel für eine Fähigkeit mit Wertebereich	119
Bild 5.7:	Grafische Darstellung von Anforderungen und Fähigkeiten als Wert und Wertebereich (Ableichart 1)	119
Bild 5.8:	Grafische Darstellung von Anforderungen und Fähigkeiten als Wert und Werteliste (Ableichart 2)	121
Bild 5.9:	Bild 5.4 erweitert um die neue Linie mit ihren Stationen und die neuen Bauteilvarianten.....	126
Bild 5.10:	Produktvariantenvielfalt der erweiterten Kugelschreiber-Szenarios ..	127
Bild 5.11:	Herausforderungen beim Einlesen der Daten	138
Bild 5.12:	Anzahl Bauteilvarianten zweier Produktgenerationen je Bauteilart ..	139
Bild 5.13:	Entwicklung der Bauteilvariantenvielfalt	141
Bild 5.14:	Beispiel-Heatmap für einen Baukastenmotor	147
Bild 5.15:	Graph zum erweiterten Kugelschreiber-Beispiel aus Abschnitt 5.2.7	149

Tabellenverzeichnis

List of tables

Tabelle 2.1:	Beispiel für eine Strukturstückliste (in Anlehnung an [EIGN09a, S. 83-86; MOOS04, S. 431]).....	26
Tabelle 2.2:	Beispiel für eine Bauteilvariantenmatrix der Bauteilart Schwungrad ..	28
Tabelle 2.3:	Beispiel für eine Linienbelegung.....	29
Tabelle 3.1:	Abgleich der untersuchten Modelle mit den Anforderungen aus Abschnitt 2.5.....	62
Tabelle 4.1:	Übersicht der vier Kategorien zur Trendbeurteilung von Baukastenprodukten [MUEL17, S. 256]	86
Tabelle 4.2:	Heatmap für die Bauteilvariantenkommunalität verschiedener Produkte (in Anlehnung an [MUEL17, S. 256]).....	88
Tabelle 4.3:	Stationskapazitäten (in Anlehnung an [MUEL17, S. 257]).....	89
Tabelle 4.4:	Ausleitung zur Auswertung von Konflikten bei Anforderungen und Fähigkeiten.....	93
Tabelle 5.1:	Beispiele für die Verwendung des <code>LIKE</code> Befehls mit Wildcards	106
Tabelle 5.2:	Beispielhaftes Ergebnis der Abfrage aus Listing 5.5	107
Tabelle 5.3:	Beispielhaftes Ergebnis der Abfrage aus Listing 5.9	113
Tabelle 5.4:	Beispielhaftes Ergebnis der Abfrage aus Listing 5.10	114
Tabelle 5.5:	Ausschnitt des Ergebnisses aus Listing 5.11	116
Tabelle 5.6:	Ergebnis aus Listing 5.13	118
Tabelle 5.7:	Ergebnis der Abfrage aus Listing 5.14	120
Tabelle 5.8:	Abfrage der Stationskapazitäten	122
Tabelle 5.9:	Stationskapazitäten ohne (1. PG) und mit den neuen Bauteilvarianten (2.PG).....	128
Tabelle 5.10:	Ausgangssituation: Jede Linie stellt jedes Produkt her	129
Tabelle 5.11:	Erste Option: Anpassung der Linienbelegung	130
Tabelle 5.12:	Situation nach allen beschriebenen Anpassungen.....	132
Tabelle 5.13:	Variantencode für den Baukastenmotor	134
Tabelle 5.14:	Prinzipdarstellung einer Bauteilvariantenmatrix.....	136
Tabelle 5.15:	Stationsauslastung zweier Linien	142
Tabelle 5.16:	Verlagerung von Derivaten oder Produktionsvolumen innerhalb eines Produktionsnetzwerkes	144
Tabelle 5.17:	Vergleich der Varietät der Anwendungsbeispiele aus Kapitel 5	150

Listingverzeichnis

List of listings

Listing 3.1:	Beispiel für den <code>CREATE TABLE</code> Befehl.....	48
Listing 3.2:	Beispiel für den <code>INSERT INTO</code> Befehl.....	48
Listing 3.3:	Beispiel für den <code>SELECT</code> Befehl.....	49
Listing 3.4:	Beispiel für den <code>DELETE</code> Befehl.....	49
Listing 5.1:	Erstellung einer Tabelle am Beispiel von <i>tab_linien</i>	104
Listing 5.2:	Erstellung einer Verknüpfungstabelle am Beispiel von <i>Ink_prozesse_bauteilarten</i>	104
Listing 5.3:	Beispiel für den Abgleich zweier Zeichenketten mittels <code>LIKE</code> Befehl	105
Listing 5.4:	Einfache Abfrage mithilfe der Variantenregel	106
Listing 5.5:	Umsetzung der Variantenregel am Beispiel der Linienbelegung	107
Listing 5.6:	Einfügen von Daten in die Datenbank mittels SQL Query	111
Listing 5.7:	Befüllung einer Datenbank mithilfe einer anderen Datenbank.....	111
Listing 5.8:	Einlesen einer CSV-Datei.....	112
Listing 5.9:	Abfrage der Stationen je Linie	113
Listing 5.10:	Abfrage der Anzahl von Bauteilvariante je Bauteilart	114
Listing 5.11:	Abfrage aller Zusammenhänge	115
Listing 5.12:	For-Schleife zur Erstellung mehrere statischer Auswertungen in C++	116
Listing 5.13:	Abfrage zur Erstellung zeitdynamischer Abfragen in SQL.....	117
Listing 5.14:	Abfrage für Anforderungen und Fähigkeiten für Werte und Wertebereich (Abgleichart 1).....	120
Listing 5.15:	Abfrage für Anforderungen und Fähigkeiten für Werte und Werte- Listen (Abgleichart 2)	121
Listing 5.16:	Abfrage für Kapazitäten.....	122
Listing 5.17:	Abfrage aller Zusammenhänge des Modells	125

1 Einleitung

Introduction

1.1 Problemstellung

Problem

Das produzierende Gewerbe stellt einen wichtigen Wirtschaftsfaktor für den Standort Deutschland dar. Mit einem Anteil von 25% an der Bruttowertschöpfung leistet sie traditionell einen hohen Beitrag zur deutschen Wirtschaftsleistung. [DEST17]

Die produzierenden Unternehmen sind aktuell unter anderem mit zwei großen Herausforderungen konfrontiert: Zum einen mit steigendem Wettbewerb und der Entwicklung zum Käufermarkt. Zum anderen mit einem turbulenten Marktumfeld und Nachfrageschwankungen. [MUEL14, S. 124]

Durch den internationalen Wettbewerb können Kunden bei fast allen Produkten zwischen vielen Alternativen wählen. Um möglichst viele Kunden zu gewinnen oder zu behalten versuchen die Unternehmen, die Wünsche der Kunden bestmöglich zu erfüllen, was zu einer Vielzahl an Produktvarianten führt. Diese müssen sowohl in der Produktionsplanung als auch in der Produktion selbst beherrscht werden. Produktionssysteme müssen Variantenflexibel ausgelegt werden, was bei einer hohen Variantenzahl eine Herausforderung für die Produktionsplanung darstellt. [ABEL11; MUEL14, S. 124; FELD14, S. 6]

Da die Beschaffung und Installation neuer Produktionssysteme mit hohen Investitionen verbunden sind, ist es meist nicht kosteneffizient, für jede Produktvariante ein eigenes Produktionssystem zu errichten. In der Folge müssen die Produktionssysteme eine steigende Variantenvielfalt herstellen können. Ebenfalls bedingt durch den hohen Einmalaufwand für ein Produktionssystem sind produzierende Unternehmen zunehmend bestrebt, neue Produkte oder neue Produktgenerationen in bestehende Produktionssysteme zu integrieren und somit die Produktionssysteme möglichst lange zu nutzen.

Hinzu kommen Turbulenzen durch das Umfeld. Politische Entscheidungen in Form von Gesetzen, Handelsabkommen und -beschränkungen oder staatlichen Förderungen können starken Einfluss auf die Nachfrage nach bestimmten Produkten haben. [BULL09, S. 247-248]

Auch Ereignisse bei einzelnen Wettbewerbern können sich auf den gesamten Markt auswirken. Fehlerhafte Produkte eines Herstellers können die Nachfrage nach ähnlichen Produkten eines anderen Herstellers ebenfalls beeinflussen. Die Produktionssysteme müssen auf diese Nachfrageschwankungen reagieren können und entsprechend Mengenflexibel geplant sein. [FELD14, S. 6]

Bedingt durch die oben genannten Einflüsse, aber auch durch den technologischen Fortschritt stellt es zudem einen Wettbewerbsvorteil dar, wenn die Planungsdauer und

somit die Zeit bis zur Einführung eines Produktes verkürzt werden kann. [FELD14, S.5-6]

Auf die genannten Randbedingungen haben die Unternehmen kaum Einfluss, müssen jedoch mit den Folgen umgehen und eine angemessene Reaktion auf diese Folgen zeigen.

Die Montage und Inbetriebnahme stellen einen großen Teil der Wertschöpfung innerhalb des Produktionsprozesses dar. Sie ist besonders von den oben genannten Herausforderungen betroffen. Produktvarianten werden in der Montage generiert durch die Kombinationen einer Vielzahl von Bauteilvarianten mit denen häufig Prozessvarianten einhergehen. [MUEL13, S. 702]

Als wichtige Schnittstelle zwischen der Produktentwicklung und der Montage ist die Montageplanung in besonderem Maße von den Auswirkungen der Herausforderungen betroffen. In der Montage wird eine Vielzahl verschiedener Teile gefügt, die unterschiedliche Eigenschaften und Toleranzen aufweisen. [FELD14, S. 6]

Diese Herausforderungen führen zu zwei wesentlichen Anforderungen an die Montageplanung: Zum einen muss sie die Vielfalt und die hierdurch bedingten zahlreichen Wechselwirkungen beherrschen und zum anderen muss sie in der Lage sein, mit der vorherrschenden Dynamik umzugehen hier hierauf zu reagieren. Beides in Summe führt zu einer erhöhten Komplexität in der Montageplanung.



Bild 1.1: Herausforderungen bei der Montagesystemgestaltung [FELD14, S. 5]

Challenges for the assembly system design

EIGNER fasst diese Aussagen wie folgt zusammen: „Die Entwicklung, Herstellung und Vermarktung innovativer Produkte sowie die Fähigkeit, auf sich dynamisch wandelnde Märkte zu reagieren, ist eine wichtige Voraussetzung zur Aufrechterhaltung und Steigerung der Wettbewerbsfähigkeit von Unternehmen in einem globalen Umfeld.“ [EIGN09a, S. 1]

Um dies zu ermöglichen müssen Produkt, Prozess und Ressource miteinander in einen Zusammenhang gebracht werden, um gegenseitige Abhängigkeiten untersuchen zu können. Aktuell wird dieser Zusammenhang manuell und unter Verwendung von Planungswissen sowie aus diversen Datenquellen hergestellt.

Es ist kein IT Tool verfügbar, welches es ermöglicht, Produkt Prozess und Ressource ausreichend genau zu beschreiben und zu verknüpfen um hieraus die Produktanalyse durchführen oder Änderungen bewerten zu können.

Ein übergreifendes Modell, welches die Zusammenhänge in ausreichender Detaillierung beschreibt existiert ebenfalls nicht.

Die unzureichende Verknüpfung von Produkt, Prozess und Ressource ist besonders bei Baukastenprodukten mit vielen Varianten ein Problem, da hier sehr viele Abhängigkeiten existieren und somit bei der Integrationsplanung beachtet werden müssen.

1.2 Zielsetzung

Target

Im Rahmen dieser Arbeit soll untersucht werden, wie die Integrationsplanung variantenreicher Baukastenprodukte unter Berücksichtigung der steigenden Komplexität und Dynamik effizient erfolgen kann.

Sowohl in der Neuplanung als auch in der Integrationsplanung ist die Verknüpfung zwischen Produkt, Prozess und Ressource einer der ersten Schritte. Nur wenn die logischen Zusammenhänge identifiziert sind lassen sich Probleme erkennen, Lösungsansätze generieren und anschließend validieren.

Ziel dieser Arbeit ist es, die Zusammenhänge zwischen Produkt, Prozess und Ressource in einem Modell abzubilden. Mithilfe des Modells sollen sich die Auswirkungen von Änderungen an Produkt, Prozess, Ressource oder deren Zuordnung zueinander schnell gegeneinander abwägen lassen. Dabei sollen alle zuvor modellierten Einschränkungen berücksichtigt werden.

Aus den genannten Punkten lässt sich folgende Forschungsfrage ableiten:

Wie muss ein (Daten-)Modell zur Handhabung der steigenden Komplexität bei der Montagesystemplanung von Baukastenprodukten gestaltet sein?

Um den Transfer des Modells in die Praxis sicherzustellen, soll das Modell in Form einer Softwarelösung umgesetzt werden. Anhand dieser Softwarelösung sollen anschließend methodisch Planungsentscheidungen abgeleitet werden.

1.3 Vorgehensweise

Method

Die Wissenschaft lässt sich nach der Wissenschaftssystematik von ULRICH in die Formalwissenschaften und die Realwissenschaften einteilen. Formalwissenschaften haben dabei ihre eigenen Zeichensysteme und Regeln, wie beispielsweise die Philosophie oder die Mathematik. Die Realwissenschaften dagegen konzentrieren sich auf Zusammenhänge, die sich wahrnehmen lassen. [ULRI76, S. 305]

Die Realwissenschaften werden weiter unterteilt in reine Wissenschaften und angewandte Wissenschaften. Erstere versuchen mittels Modellen kleine Ausschnitte realer Zusammenhänge zu erklären. Für angewandte Wissenschaft stehen verschiedene Handlungsalternativen im Vordergrund. Die hierin entwickelten Modelle sollen den Menschen bei der Entscheidungsfindung unterstützen. [ULRI76, S. 305]

Die einzelnen Wissenschaften können nicht losgelöst voneinander betrachtet werden. Die Erkenntnisse der Formalwissenschaften bilden die Basis für die Realwissenschaften, die Grundlagenwissenschaften wiederum die Basis für die Handlungswissenschaften. [ULRI76, S. 305]

Die beschriebenen Zusammenhänge sind in Bild 1.2 dargestellt. Die vorliegende Arbeit ist den Realwissenschaften zuzuordnen. Sie verfolgt ein praktisches Ziel und ist somit den angewandten Wissenschaften zuzuordnen. Für diese Wissenschaften hat ULRICH einen Forschungsprozess beschrieben, nach dem in dieser Arbeit verfahren wird. Er verfolgt dabei den Ansatz, dass die Problemstellung nicht aus einem Theoriezusammenhang, sondern aus der Praxis kommt. Entsprechend erfolgt die Umsetzung und Validierung der entwickelten Modelle und Methoden in der Praxis. [ULRI81]

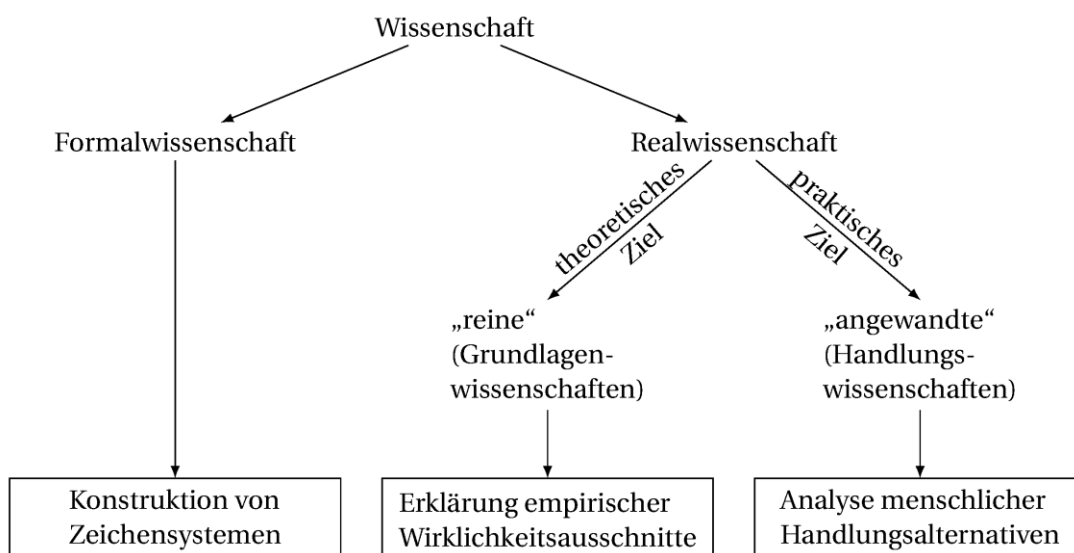


Bild 1.2: Wissenschaftssystematik nach ULRICH [ULRI76, S. 305]

Systematic of science by ULRICH

Nach ULRICH beginnt der Forschungsprozess mit der Analyse von Problemen, die eine Relevanz für die Praxis haben (vgl. Bild 1.3). Im zweiten Schritt werden Theorien und Lösungsansätze für diese Probleme in den Grundlagenwissenschaften gesucht. Im dritten Schritt werden Verfahren der Formalwissenschaften gesucht, die bei der Lösung des Problems unterstützen. [ULRI81]

In den Schritten 4 und 5 werden die relevanten Zusammenhänge ermittelt und in Regeln und Modelle überführt. Im 6. Schritt werden diese Modelle in der Praxis angewendet und es wird geprüft, ob sie korrekt sind. Im letzten Schritt lässt sich mit den Ergebnissen eine Handlungsempfehlung für die Praxis ableiten. [ULRI81]

An diesem Vorgehen von ULRICH orientiert sich der Aufbau der Arbeit (vgl. Bild 1.3). In Kapitel 2 werden die praktischen Grundlagen erläutert und hieraus eine Problemstellung und somit ein Handlungsbedarf abgeleitet. Kapitel 3 beschreibt, inwieweit in der Forschung Lösungen für den beschriebenen Handlungsbedarf existieren und welche Verfahren der Formalwissenschaften – in diesem Fall insbesondere der Logik und Informatik – zum Einsatz kommen können, um das Problem zu lösen.

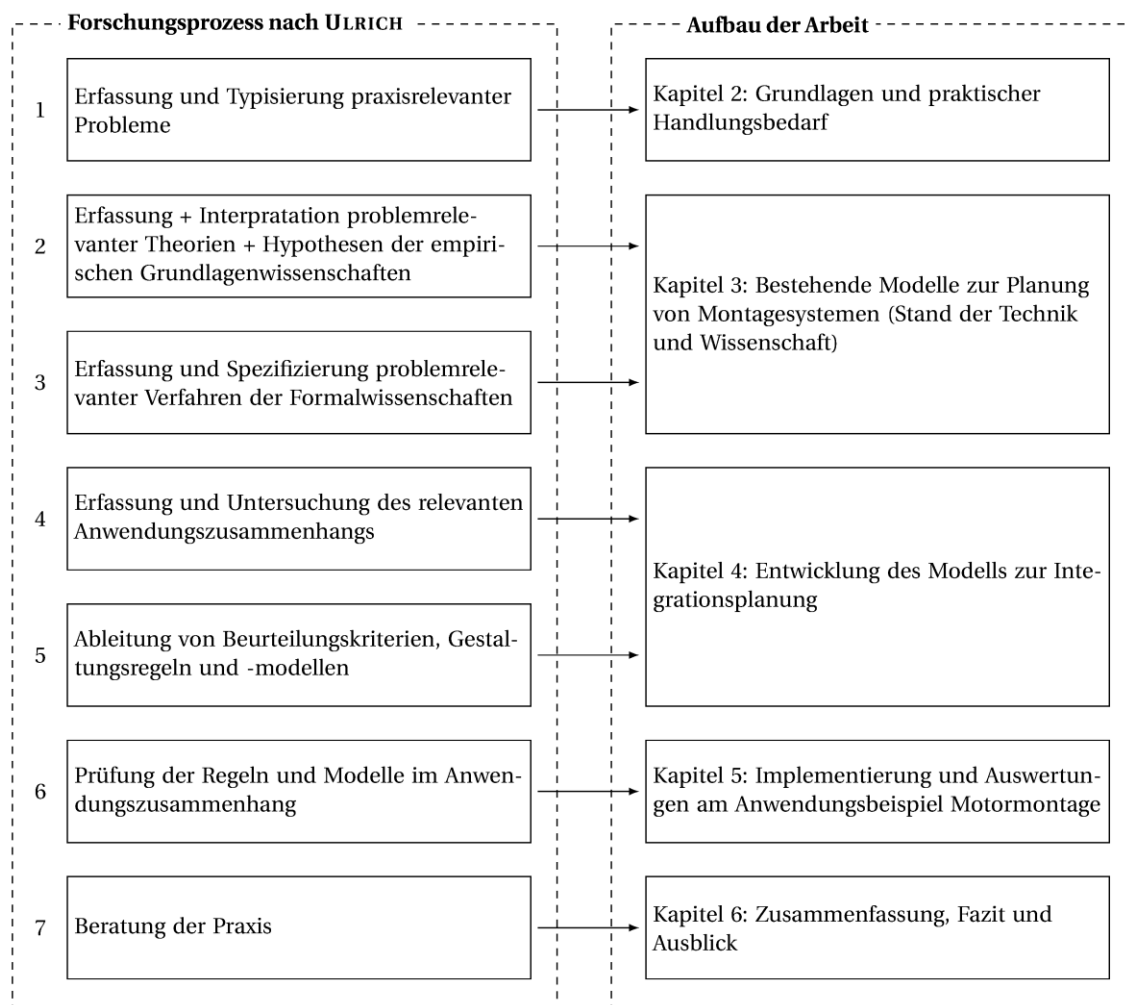


Bild 1.3: Forschungsprozess nach ULRICH [ULRI81] und Aufbau der Arbeit

Process of research by ULRICH and structure of this thesis

In Kapitel 4 wird basierend auf den Erkenntnissen der vorangegangenen Schritte im Forschungsprozess ein Modell abgeleitet. Dieses erfasst die Zusammenhänge, die für die Montageplanung relevant sind. Basierend auf diesem Modell lassen sich Kenngrößen und Entscheidungen ableiten.

In Kapitel 5 wird das Modell in Form einer Datenbank implementiert, befüllt und anschließend zur Analyse herangezogen. Diese Validierung erfolgt in zwei Schritten. Im ersten Schritt wird das Modell anhand eines einfachen Beispiels auf seine Funktionsfähigkeit geprüft. Im zweiten Schritt wird es zur Untersuchung des komplexen Anwendungsfalls der Montageplanung für PKW-Motoren herangezogen. An dieser Stelle werden Anwendungsfälle aus der Praxis mithilfe des Modells untersucht und Entscheidungen für die Montageplanung abgeleitet.

Die Forschungsmethodik nach ULRICH endet mit dem 7. Schritt, der Beratung der Praxis. In der vorliegenden Arbeit findet sich dies in Kapitel 6. Hier werden die gewonnenen Erkenntnisse zusammengefasst und kritisch beurteilt. Es wird ein Fazit gezogen, das den Nutzen für die Praxis ausweist. Weiterhin wird eine Empfehlung abgegeben, wie die Resultate dieser Arbeit in künftigen wissenschaftlichen Anstrengungen erweitert werden können.

2 Grundlagen und praktischer Handlungsbedarf

Basic and practical implications

Der Fokus der Grundlagen und des praktischen Handlungsbedarfs ist die Montageplanung. Sie ist Teil des Produktentstehungsprozesses, der in Abschnitt 2.1 erläutert wird. Planungsobjekt der Montageplanung ist die Montage, deren charakteristische Eigenschaften in Abschnitt 2.2 beschrieben sind.

Die Montageplanung wird detailliert in Abschnitt 2.2.2 beschrieben. Eine besondere Herausforderung für diese stellt die Komplexität, bedingt durch Vielfalt und Dynamik, dar (vgl. Abschnitt 2.3), die sich aktuell nicht beherrschen lassen. Hieraus ergibt sich ein Defizit in der Praxis (vgl. Abschnitt 2.4). Das Kapitel wird mit einer Zusammenfassung abgeschlossen (vgl. Abschnitt 2.5).

2.1 Produktentstehungsprozess und Baukastenprodukte

Product creation process

Ein Produktlebenszyklus umfasst alle Phasen, die ein Produkt durchläuft. Von der ersten Idee bis zur Verwertung des Produktes [EIGN09a, S.2]. Der Produktentstehungsprozess (PEP) ist Teil dieses Produktlebenszyklus [DEUS12, S. 88]. STEINWASSER und JONAS stellen den Produktlebenszyklus als Kreislauf dar (vgl. Bild 2.1). [STEI97; JONA00]

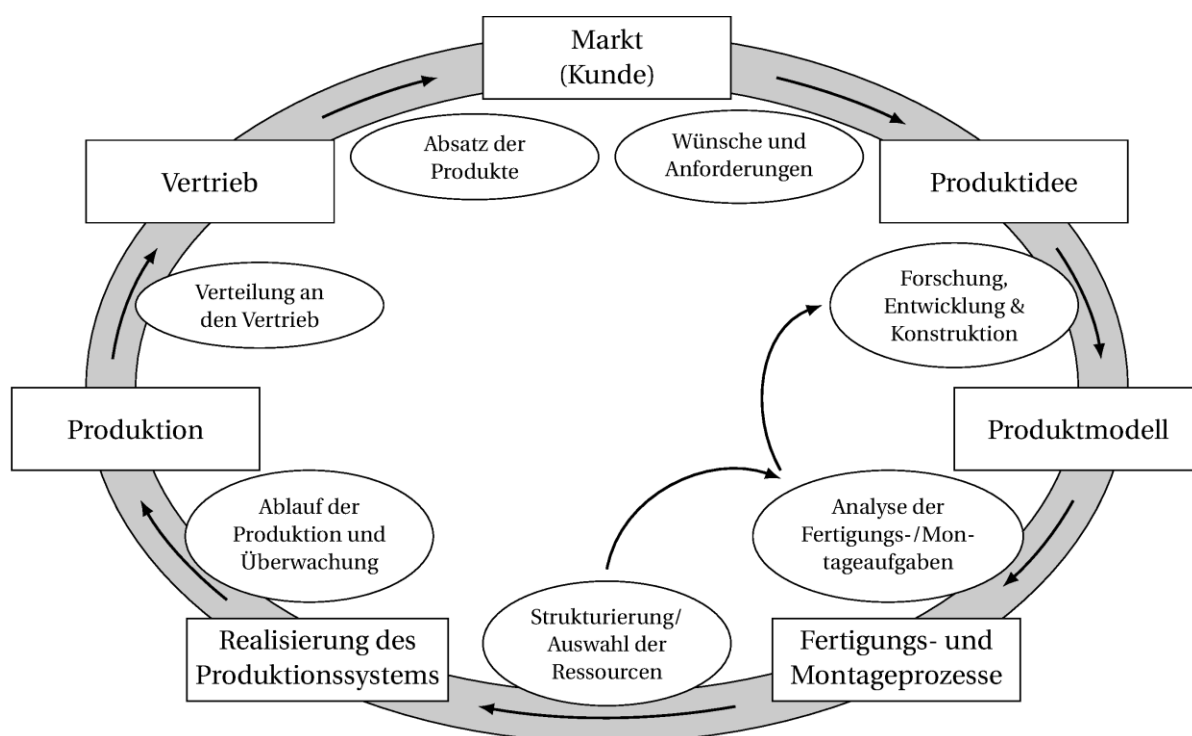


Bild 2.1: Produktkreislauf nach STEINWASSER [STEI97, S. 4]

Circle of products by STEINWASSER

Ausgehend von den Bedürfnissen des Marktes bzw. der potenziellen Kunden entsteht eine Produktidee. Durch die Forschungs- und Entwicklungsabteilung wird diese Produktidee weiterentwickelt und mithilfe der Konstruktion in ein Produktmodell überführt. Anschließend erfolgt die Produktionsplanung und Arbeitsvorbereitung, in der die Fertigungs- und Montageprozesse entwickelt werden. Als Ergebnis werden Ressourcen ausgewählt und gestaltet, die zur Realisierung des Produktionssystems führen. STEINWASSER führt hier noch eine Schleife zurück in die Konstruktion ein. Damit betont er, dass die Auswahl der Ressourcen Einfluss auf die Analyse der Montageaufgabe haben kann und diese wiederum auf die Entwicklung und Konstruktion des Produktes. Dieser Vorgang wird als Produktbeeinflussung bezeichnet. [STEI97, S.4f; JONA00]

Nachdem das Produktionssystem realisiert wurde, startet die Produktion und die Produkte werden hergestellt. JONAS unterscheidet hierbei zwischen der Fertigung, die Einzelteile herstellt und der Montage, die aus diesen Einzelteilen ein fertiges Produkt erzeugt. Dieses fertige Produkt gelangt über den Vertrieb an die Kunden. Somit ist der Kreislauf geschlossen. [STEI97, S.4f; JONA00]

EIGNER skizziert den Produktlebenszyklus dagegen als eine Abfolge von Schritten (vgl. Bild 2.2). Er beginnt mit der Definition von Anforderungen an ein Produkt. Nach der Produktplanung folgt die Entwicklung. Anschließend werden die Prozesse geplant und es wird produziert. Auf die erfolgreiche Produktion folgt der Betrieb. Ist der Betrieb des Produktes beendet wird es recycelt. [EIGN09a, S.1-2; EIGN09b, S. 247]

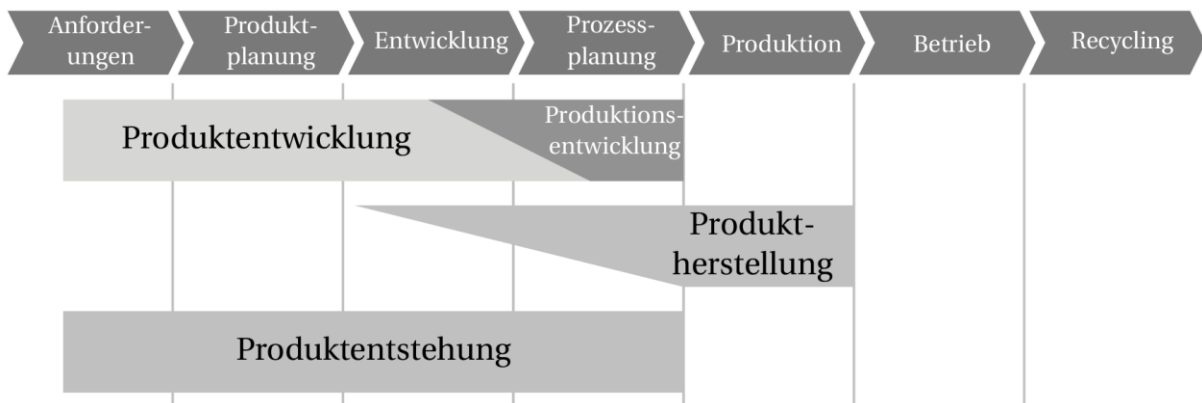


Bild 2.2: Produktlebenszyklus nach EIGNER [EIGN09a, S. 2]

Product-Lifecycle by EIGNER

Für EIGNER reicht der PEP von der Definition der Anforderungen bis hin zur Prozessplanung. Mit Ende der Prozessplanung ist er abgeschlossen. EIGNER unterteilt ihn in die Produktentwicklung und die Produktionsentwicklung. Ähnlich wie bei STEINWASSER geht auch er von einer Überlappung dieser Phasen aus und stellt somit wiederum die Produktbeeinflussung durch die Prozessplanung dar. Durch die Produktion von Vorserien und Prozessvorläufern gibt es darüber hinaus eine Verbindung zwischen Produktherstellung und Produktentstehung. Die reguläre Produktion beginnt mit dem

Start of Production (SOP) und endet mit dem *End of Production* (EOP). [EIGN09a, S.1-2; EIGN09b, S. 247; MOTU08, S. 70]

Allen Ansätzen ist gemein, dass die Prozessplanung oder Arbeitsvorbereitung ein wichtiger Teil des Prozesses ist. Sie hat Einflüsse in beide Richtungen: Sowohl auf das Produkt als auch auf die Produktion. Im Zuge des Simultaneous Engineering ist davon auszugehen, dass sich diese Überlappung weiter steigern wird. [EVER05, S. 8]

Grund hierfür ist die entstehende Zeitersparnis und Kostenersparnis durch frühe Änderungen. Durch die Zeitersparnis können die Unternehmen auf die verkürzten Innovations- und Produktlebenszyklen reagieren. Die frühe Einbindung versetzt die Produktionsentwicklung in die Lage, Einfluss auf die Produktgestalt zu nehmen um sicherzustellen, dass das Produkt montagegerecht gestaltet wird. Je früher diese Änderungen am Produkt vorgenommen werden, desto günstiger sind diese. [JONA00, S. 12; BOSS07, S. 67; EHRL07, S. 11-14]

Noch stärker sieht die Überlappung SCHNEIDER mit seinem PEP-Planung (vgl. Bild 2.3). Hierin reicht der PEP bis in die Phase der Produktherstellung. Er begründet dies damit, dass die Planung den Produkthanlauf und Produktionshochlauf, beides Teile der Produktherstellung, betreut. [SCHN12, S. 159]

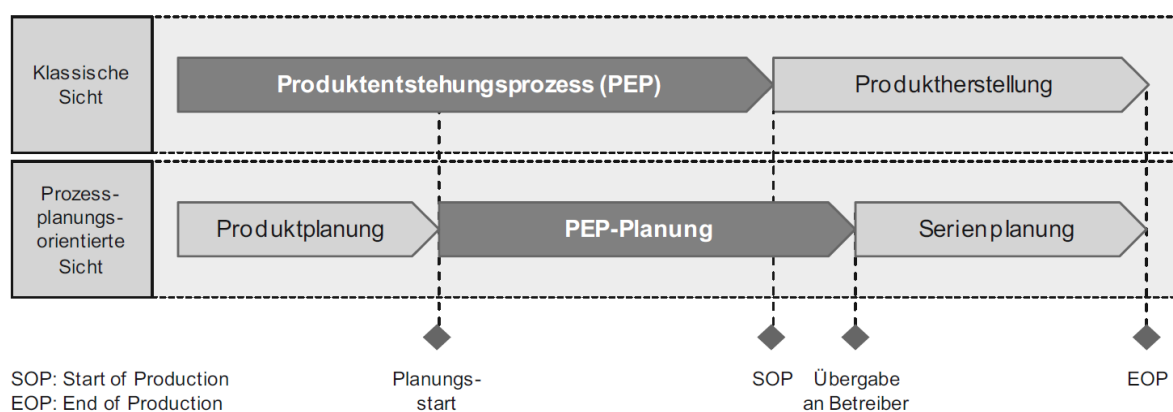


Bild 2.3: PEP-Planung nach Schneider [SCHN12, S. 159]

Predefinition and cause of costs by VDI

Die Darstellung der Informationsflüsse in der innerbetrieblichen Wortschöpfung nach SPUR verdeutlicht, dass die Montageplanung im stetigen Austausch mit vielen Unternehmensbereichen steht. Zum einen, weil sie deren Informationen für die Durchführung Ihrer Planungstätigkeit benötigt, zum anderen weil sie mit ihrer Planung diese Bereiche beeinflusst. Für eine kosteneffiziente Herstellung des Produktes ist beispielsweise eine Rückmeldung aus der Montageplanung an die Entwicklung & Konstruktion unerlässlich.

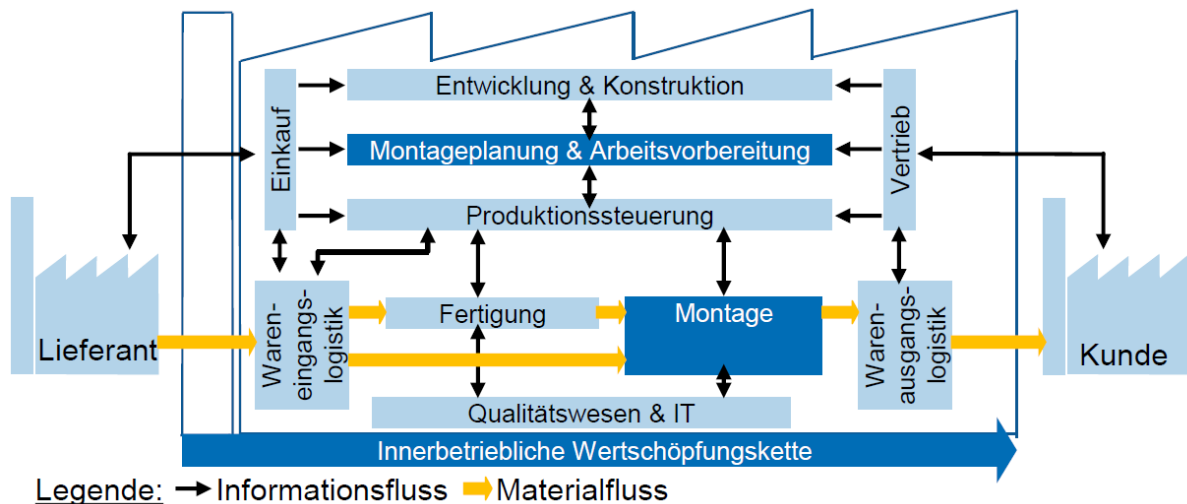


Bild 2.4: Montage und Montageplanung in der innerbetrieblichen Wertschöpfungskette
[HOER19 in Anlehnung an SPUR86]

Assembly and assembly planning in the value chain

Wird einbezogen, wann die Kosten für ein Produkt festgelegt werden, gewinnt dieser Aspekt zusätzlich an Bedeutung. In Bild 2.5 ist hierzu ein Beispiel dargestellt, welches die Festlegung und Verursachung von Kosten in ein Verhältnis setzt. So werden fast dreiviertel der Kosten bereits in der Entwicklung und Konstruktion festgelegt, die Kosten die hierbei anfallen sind jedoch sehr gering. Dagegen fallen für die Arbeitsvorbereitung und Fertigung rund ein Drittel der Kosten an, es werden jedoch nur 20% der Kosten festgelegt. Ähnliche Aussagen mit nur geringfügig abweichenden Zahlen finden sich auch bei EHRENSPIEL [EHL07, S. 11-13] und EIGNER [EIGN09a, S. 16] Bullinger. Dies macht deutlich, dass es besonders sinnvoll ist, Anforderungen der Produktion bereits in der Entwicklung zu berücksichtigen und eine enge Abstimmung zwischen Produktentwicklung und Arbeitsvorbereitung sicherzustellen.

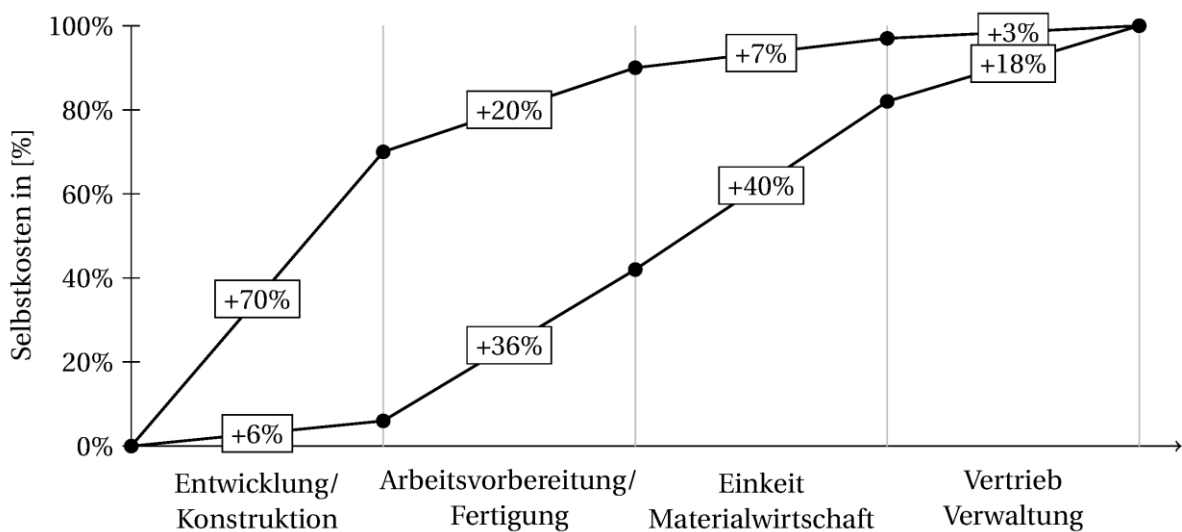


Bild 2.5: Kostenfestlegung und Kostenverursachung nach VDI (Daten aus [VDI87, S. 3])

Predefinition and cause of costs by VDI

Wichtiger Bestandteil der Arbeitsvorbereitung ist die Arbeitsplanung, in der alle Maßnahmen geplant werden, die zur Herstellung des Produktes erforderlich sind (vgl. Bild 2.6). Hierin werden entscheidende Parameter des späteren Montagesystems festgelegt. [WIEN10, S. 195-197]

Die Montage befindet sich am Ende der Wertschöpfungskette und ihr kommt an Hochlohnstandorten wie Deutschland eine Schlüsselrolle zu [MUEL09, S. 3, BREC11, S. 917]. Daher wird sie in Abschnitt 2.2 näher betrachtet.

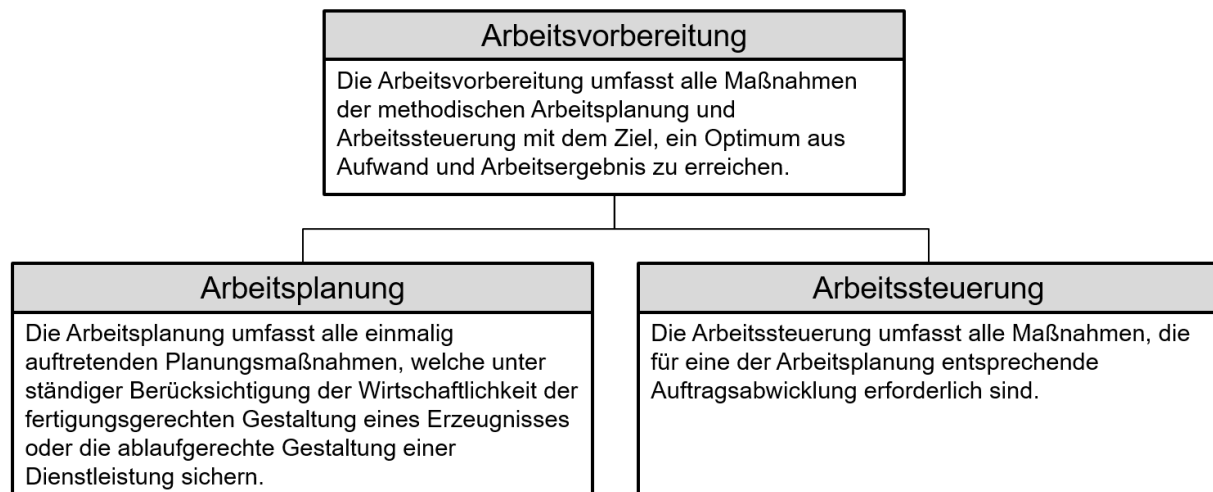


Bild 2.6: Definition der Arbeitsvorbereitung nach REFA [WIEN10, S. 195]

Definition of the term "Arbeitsvorbereitung" by REFA

Im Fokus dieser Arbeit stehen sogenannten Baukastenprodukte, die im Folgenden kurz vorgestellt werden sollen. Ziel von Baukastenprodukten ist es, eine Vielzahl an Produktvarianten mit einer geringen Zahl an Bauteilvarianten darstellen zu können. Der Effekt hiervon ist, dass eine steigende Komplexität in der Produktvariantenvielfalt nicht zu einem ebenso großen Anstieg der Komplexität in der Bauteilvariantenvielfalt führt. [SCHU11]

Die Bauteilvarianten müssen hierzu einem modularen Aufbau folgen und so gestaltet werden, dass sie in möglichst vielen Produktvarianten eingesetzt werden können. Sie werden also anstatt für ein bestimmtes Produkt für eine übergreifende Funktion entwickelt. [BRAE11, S. 106, PAHL07, S. 662, SCHU11]

Hierin liegt ein zentraler Unterschied zwischen dem Baukastenansatz und dem Plattformansatz, der ebenfalls im Zusammenhang mit Komplexität und Variantenvielfalt zu Einsatz kommt. In einer Plattform werden Gemeinsamkeiten verschiedener Produkte zusammengefasst und dann um spezifische Elemente erweitert. In einem Baukasten werden mehrere Bausteine zusammengesetzt und durch die Kombination entstehen die Produktvarianten. [PAHL07, S. 686]

Daraus folgt, dass die Bauteilvarianten nicht mehr speziell für einen Anwendungsfall entwickelt werden. Vielmehr werden die Anforderungen des Produktspektrums untersucht und anschließend ermittelt, inwieweit sich dieses durch weniger Module abdecken lässt. Daher müssen zunächst die benötigten Funktionen ermittelt und die diversen Anforderungen sinnvoll zu Modulen zusammengefasst werden. [BRAE11, S. 106, PAHL07, S. 662]

Hieraus ergeben sich Vorteile in vielen Unternehmensbereichen. Die Entwicklung kann sich auf die Gestaltung weniger Module konzentrieren. Diese Module werden in höheren Stückzahlen benötigt, was Skaleneffekte für den Einkauf ermöglicht. Zudem erhöht der Einsatz erprobter Module die Robustheit der Produkte. Der Vertrieb kann durch die verschiedenen Module dem Kunden eine gewisse Freiheit bei der Produktgestaltung einräumen. [BRAE11, S. 106; SCHU12, S. 322; SCHU14, S. 314; PAHL07, S. 678-679]

Für die weiteren Auswirkungen solcher Baukastenprodukte auf die Produktentwicklung, sowie die Chancen und Herausforderungen die für Einkauf und Vertrieb mit ihnen einhergehen, sein auf entsprechende Literatur verwiesen. Wichtig für diese Arbeit ist der Aspekt der produktübergreifenden Verwendung von Bauteilvarianten.

Beispiele für Baukastenprodukte lassen sich in vielen Industrien finden, besonders verbreitet sind sie jedoch im Maschinen und Anlagenbau sowie in der Automobilindustrie. [SCHU10; BRAE11, S. 106]

2.2 Montage und Montageplanung

Assembly planning

2.2.1 Aufgaben der Montage

Tasks of assembly

Der Begriff der *Montage* ist für diese Arbeit von großer Bedeutung. Daher wird dieser Begriff im Folgenden näher erläutert. VDI2815 definiert die Montage als den „Zusammenbau von Teilen und/oder Gruppen zu Erzeugnissen oder zu Gruppen höherer Erzeugnisebenen in der Produktion“ [VDI78].

Ein wichtiger Bestandteil der Montage ist das Verfahren Fügen. DIN8593 definiert es als „das auf Dauer angelegte Verbinden oder sonstige Zusammenbringen von zwei oder mehr Werkstücken geometrisch bestimmter Form oder von ebensolchen Werkstücken mit formlosem Stoff“ [DIN03].

Wichtig in ihrer Definition ist der Aspekt des Verbindens von Bauteilen. Die Definition des VDI betont die steigende Komplexität, die mit einem fortschreitenden Montageprozess einhergeht. WARNECKE hat diese beiden Definitionen zusammengefasst, indem er die Aufgabe der Montage umschrieb als „Einzelteile zusammensetzen, um somit ein Produkt zu erzeugen. Dieses muss eine höhere Komplexität aufweisen und

eine vorgegebene Funktion besitzen“. Er bezieht zusätzlich das finale Ziel der Montage, die Schaffung eines Produktes, mit ein. Weiterhin legt WARNECKE fest, dass dieses Produkt neben der höheren Komplexität, die der VDI ebenfalls fordert, auch eine vorgegebene Funktion erfüllen muss. [WARN75]

Zum besseren Verständnis und zur Abgrenzung der Montage gegen andere Fertigungsverfahren wird in DIN 8593 die Hauptgruppe des Fügens in 9 Gruppen unterteilt. [DIN03]

1. Zusammensetzen
2. Füllen
3. Anpressen/ Einpressen
4. Fügen durch Urformen
5. Fügen durch Umformen
6. Fügen durch Schweißen
7. Fügen durch Löten
8. Kleben
9. Textiles Fügen

Die verschiedenen Gruppen des Fügens unterscheiden sich durch die Art, wie die Werkstücke zusammengehalten werden. Beim An- oder Einpressen ist das Prinzip des Zusammenhalts beispielsweise der Kraftschluss, wogegen beim Fügen durch Ur- oder Umformen ein Formschluss hergestellt werden muss. Eine wichtige Eigenschaft, die mit der Fügemethode festgelegt wird, ist, ob die Bauteile zerstörungsfrei getrennt werden können. [DIN03]

Der VDI stellt fest, dass das Fügen alleine nicht ausreicht, um die Montage vollständig zu beschreiben. Nach MÜLLER kommen als weitere Operationen das Handhaben, Inbetriebnehmen sowie Hilfsprozesse und Sonderoperationen hinzu [MUEL13, S. 704]. LOTTER kommt zu einer ähnlichen Einteilung: Neben dem Fügen sieht er das Handhaben, Kontrollieren, Justieren und Sonderoperationen als relevant an. [LOTT12a, S. 2]

Handhaben wird vom VDI definiert als „das Schaffen [...] einer vorgegebenen räumlichen Anordnung von geometrisch bestimmten Körpern [...]“ [VDI90, S.2]. Nach JÜNEMANN UND SCHMIDT beschreibt das Handhaben „das gezielte räumliche Manipulieren von Gegenständen“ [JUEN00, S. 185]. Es beschreibt, wie die verschiedenen beteiligten Werkstücke einer Fügeoperation zueinander finden. Dazu gehören einerseits das Zuführen zum Montageprozess und andererseits das Transportieren des entstehenden montierten Produktes. [VDI90; MUEL11]

Die Definition nach WERNECKE fordert, dass das Ergebnis der Montage eine Funktion erfüllt. Hierzu werden die meisten Produkte am Ende des Montageprozesses in Betrieb genommen. Dazu zählt das Justieren, also der Abgleich mit einem Messgerät

gegenüber einem Ziel. Dieses stellt sicher, dass die vorgegebenen Toleranzen eingehalten werden. Der zweite Schritt einer Inbetriebnahme ist das Parametrieren. Viele moderne Produkte sind mit elektronischen Steuergeräten ausgestattet. Bevor sich solche Produkte verwenden lassen, müssen die Steuergeräte parametriert werden. Dabei werden dem Steuergerät wichtige Attribute des Produktes mitgeteilt. In einem digitalen Thermometer kann beispielsweise voreingestellt werden, ob die Temperatur in Grad Celsius oder Grad Fahrenheit ausgegeben werden soll. [MUEL11]

Ist die korrekte Montage des Produktes sichergestellt und die Parametrierung durchgeführt, so erfolgt die finale Kontrolle in Form einer Funktionsprüfung [EILE15, S. 7]. Beispielsweise wird am Bandende einer Motormontage der Motor geprüft, indem Drehmoment durch einen externen Elektroantrieb in die Kurbelwelle eingeleitet wird. Anschließend werden alle Sensoren des Motors abgefragt und die Plausibilität der Rückgabewerte überprüft. Alle Aktoren des Motors werden angesteuert und verstellt und wiederum wird die Plausibilität der Rückgabewerte überwacht. Verbraucht ein elektrischer Steller beispielsweise zu viel Strom zum Durchführen einer Stelloperation, dann deutet dies auf ein Klemmen hin.

Die beschriebenen Basisaufgaben der Montage sind in Bild 2.7 dargestellt. Die Hilfsprozesse und die Sonderoperationen sind für die erfolgreiche Durchführung einer Montage zwar wichtig, stehen in dieser Arbeit aber nicht weiter im Fokus und werden daher nicht näher vorgestellt.

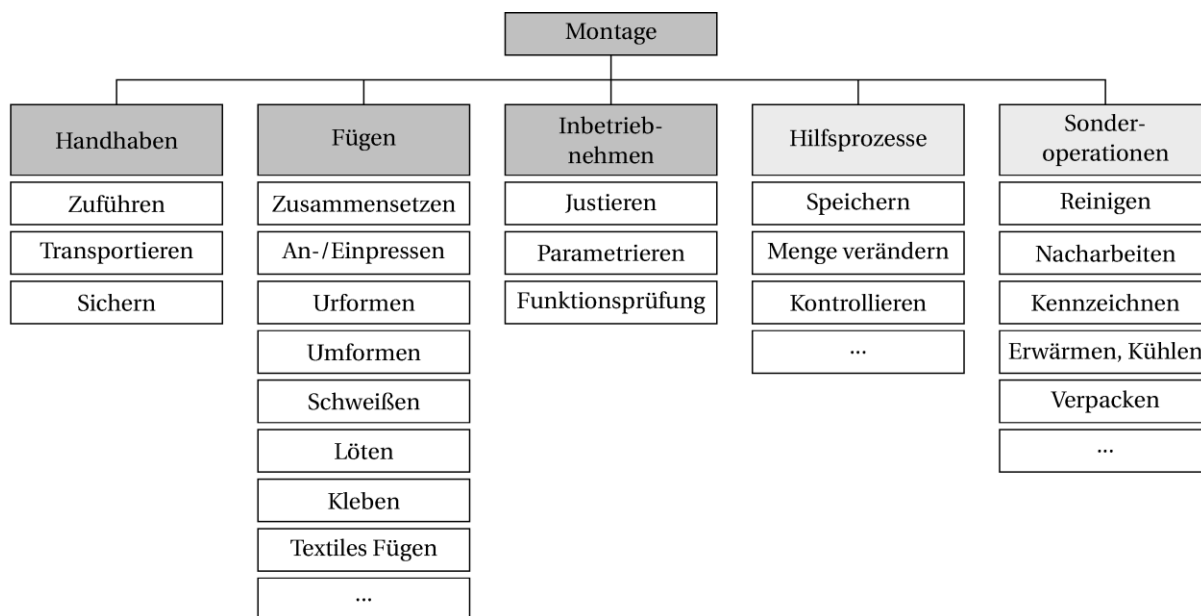


Bild 2.7: Basisaufgaben der Montage [MUEL13, S. 705]

Basic tasks of assembly

Grundsätzlich lässt sich eine Montageaufgabe auf drei unterschiedlichen Arten erledigen [LOTT12c]:

1. Manuell Montage
2. Hybride Montage
3. Vollautomatisierte Montage

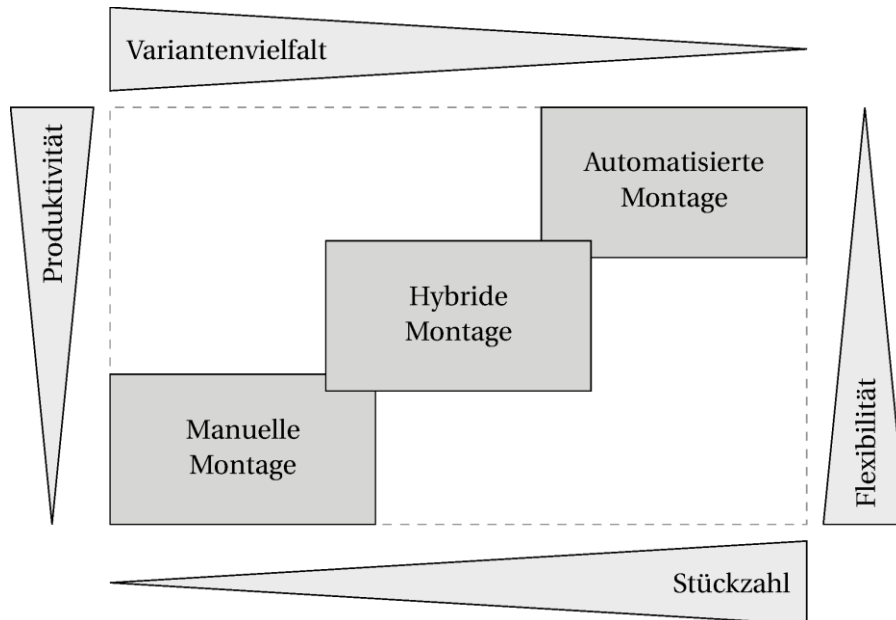


Bild 2.8: Einsatzbereiche manueller, hybrider und automatischer Montagekonzepte nach LOTTER [LOTT12c, S. 168]

Area of operation for manual, hybrid and automatic assembly concepts by LOTTER

LOTTER hat hierfür Einsatzbereiche definiert. Die jeweils gegenläufigen Anforderungen Variantenvielfalt und Stückzahl sowie Produktivität und Flexibilität spannen eine Fläche auf (vgl. Bild 2.8). Links unten in dieser Fläche befindet sich die manuelle Montage. Diese ist durch eine hohe Flexibilität und eine hohe mögliche Variantenvielfalt gekennzeichnet, aber auch durch geringe Produktivität und Stückzahl. [LOTT12c]

Diametral gegenüber steht die automatisierte Montage. Variantenvielfalt und Flexibilität sind gering, dafür lassen sich hohe Stückzahlen und eine sehr gute Produktivität erreichen. Die beiden Extrema manuelle Montage und automatisierte Montage sind nicht gegeneinander abgegrenzt: Es sind beliebige Mischformen vorstellbar und auch vielfach vorhanden. Diese wird als hybride Montagen bezeichnet. [HUCK90; LOTT12c]

Zu beachten ist hierbei, dass es sich um eine Prinzipdarstellung handelt. Mit fortschreitender Automatisierungs- und Programmierungstechnik kann sich sowohl die Lage als auch die Größe der Flächen verändern. Ein Beispiel hierfür kann die Mensch-Roboter-Kollaboration sein. Die Tendenzen, die in Bild 2.8 dargestellt sind bleiben jedoch weiterhin bestehen.

2.2.2 Montageplanung

Assembly planning

Die Montageplanung beschreibt die Prozessplanung (vgl. Abschnitt 2.1) für die Montage (vgl. Abschnitt 2.2).

Planen ist in VDI 2385 definiert als „Suche nach einer realisierbaren Lösung einer Aufgabe in befristeter Zeit mit vorgegebenem Kostenaufwand unter Berücksichtigung aller wesentlichen Faktoren und Einflussgrößen“ [VDI89, S. 2]. Eine sehr ähnliche Definition hat das REFA, indem es das Planen als „systematisches Suchen und Festlegen von Zielen sowie im Vorbereiten von Aufgaben, deren Durchführung zum Erreichen der Ziele erforderlich sind“ beschreibt. [REFA84]

Neben diesen allgemeinen Definitionen für den Begriff „Planen“ gibt es auch Definitionen, die sich mit dem Planen von Produktions- bzw. Montagesystemen auseinandersetzen. MÜLLER spricht hierbei auch von der Montagesystemgestaltung. Nach MÜLLER besteht „die zentrale Aufgabe der Montagesystemgestaltung [...] in der sinnvollen Auswahl, Zusammenstellung und Vernetzung einzelner Prozesse, basierend auf den Anforderungen des zu montierenden Produktes.“ [MUEL13, S. 704]

Sie ist verantwortlich für die „optimale Gestaltung und wirtschaftliche Umsetzung von Investitionsvorhaben für die Produktion“ [SPUR97, S. 20] Dazu gehören die Konzeption und Auswahl von Produktionsmitteln und deren Lieferanten. Die Koordination dieser Lieferanten und die Terminplanung ist ebenfalls Umfang der Montageplanung. Zudem überwacht sie den Aufbau der Anlagen in der Fabrik und deren Inbetriebnahme. EILERS fasst diese Vorgänge zum Montagesystementstehungsprozess (MEP) zusammen (vgl. Bild 2.9). [SPUR97, S. 20, EILE15 S. 9]

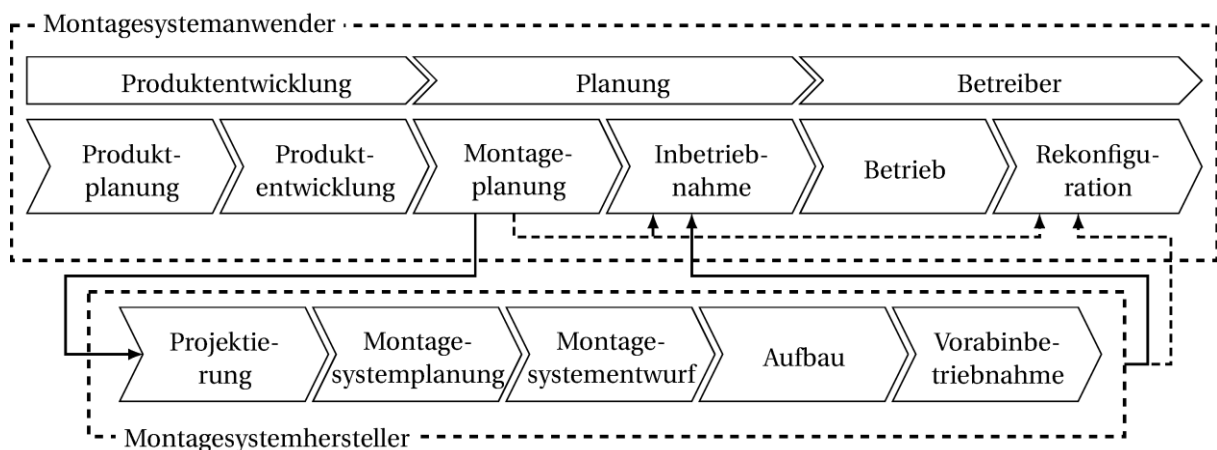


Bild 2.9: Montagesystementstehungsprozess nach EILERS [EILE15, S. 9]

Assembly system creation process by EILERS

Im MEP verfeinert er den Schritt der Montageplanung zu Projektierung, Montagesystemplanung, Montagesystementwurf (Konstruktion), Aufbau und Vorabinbetriebnahme. Diese Schritte werden sowohl bei der erstmaligen Planung eines Montagesystems durchlaufen, als auch bei der Rekonfiguration eines bestehenden Montagesystems. [EILE15, S. 9]

Eine weitere Methode zur Planung von Montagesystemen die Planungssystematik nach KONOLD UND REGER (vgl. Bild 2.10). Diese gibt einen groben Leitfaden anhand der 5 Planungsphasen: Aufgabenstellung, Grobplanung, Feinplanung, Realisierung und Montagesystemanlauf. [KONO03, S. 32]

In der Phase Aufgabenstellung werden die Prämissen für das Planungsprojekt, also Ausgangssituation und Ziele, Verantwortliche und Termine definiert. In der Grobplanungsphase werden basierend auf der Aufgabenstellungen Eckwerte für das entstehende Montagesystem definiert. Es wird die Ausbringung ermittelt, das Montagesystem wird gegliedert und Arbeitsabläufe werden grob definiert. Hieraus wird ein Personalbedarf ermittelt, der wieder eine grobe Berechnung der Wirtschaftlichkeit ermöglicht. Gegebenenfalls werden mehrere Alternativen nach dieser Vorgehensweise entwickelt. [KONO03, S. 32]



Bild 2.10:Planungssystematik nach KONOLD UND REGER [KONO03, S. 32]

Systematic planning by KONOLD UND REGER

Wurde im Rahmen der Grobplanung ein wirtschaftliches Montagesystem-Grobkonzept definiert so folgt die Phase der Feinplanung. In dieser werden die vorher grob definierten Arbeitsinhalte im Detail ausgeplant. Hieraus lassen sich Spezifikationen generieren, die dann in eine Ausschreibung einfließen. Zudem lässt sich die grobe Personalbedarfsschätzung noch validieren. Zum Ende der Feinplanung wird erneut die Wirtschaftlichkeit des Montagesystemkonzeptes überprüft. [KONO03, S. 32]

Lässt sich die Wirtschaftlichkeit wiederum bestätigen kann das Montagesystem realisiert werden. Hierzu werden Lieferanten beauftragt das Montagesystem zu installieren und das künftige Personal wird trainiert. Die Arbeitsplätze werden mittels MTM Sekundengenau mit Arbeitsinhalten belegt. [KONO03, S. 32]

Nach Abschluss der Realisierung des Montagesystems erfolgt der Anlauf. Da es sich bei Montagesystemen in der Regel um einen Verbund von Sondermaschinen handelt ist nicht davon auszugehen, dass diese sofort nach der Installation fehlerfrei und voll leistungsfähig sind. Die Anlaufphase dient dazu, diese Fehler zu finden und zu beseitigen. Sind die im Rahmen der Beschaffung vereinbarten Kennzahlen (bspw. Anlagenverfügbarkeit und Taktzeit) erreicht, wird die Endabnahme an den Lieferanten erteilt. [KONO03, S. 32]

Auch LOTTER stellt in seiner Planungssystematik (vgl. Bild 2.11) die Montage in den Fokus. Er unterteilt die Montageplanung in 11 Schritte. Im ersten Schritt werden allgemeine Anforderungen an die zu planende Montagelinie ermittelt, zum Beispiel welches Produkt in welcher Stückzahl hergestellt werden soll. [LOTT92, HART12, S.368]

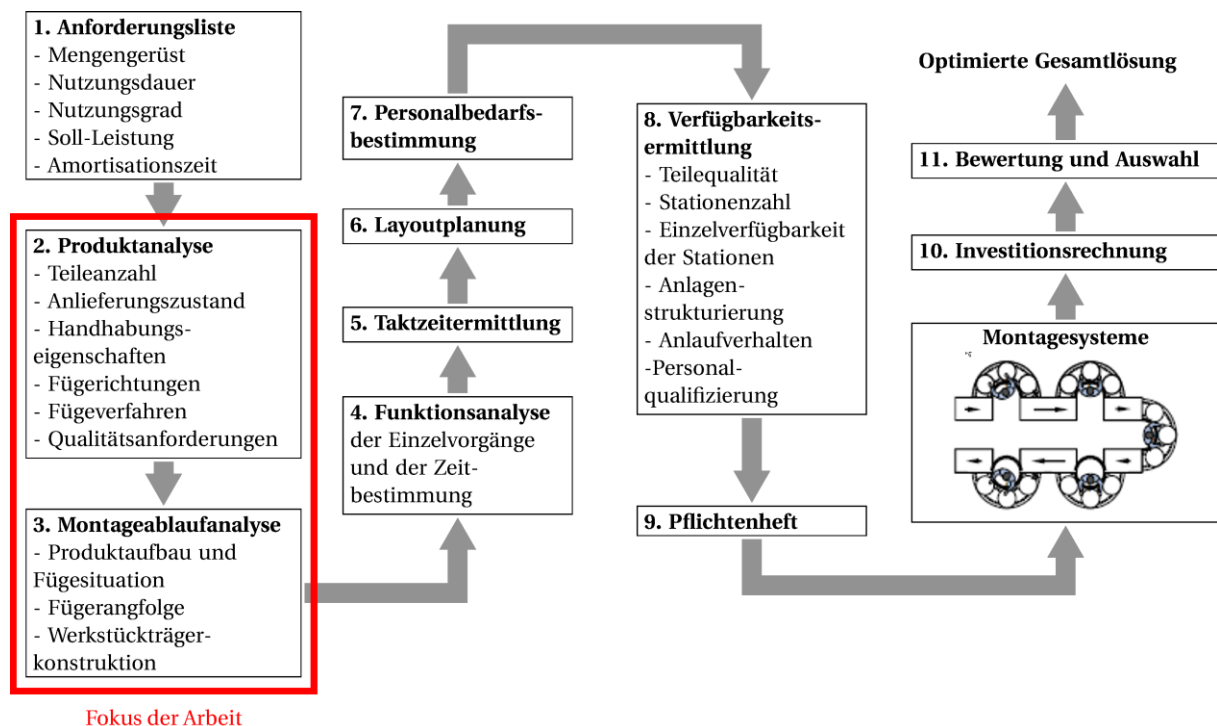


Bild 2.11:Planungssystematik nach LOTTER [LOTT92]

Systematic planning by LOTTER

In der darauf folgenden Produktanalyse werden die Bestandteile des Produktes und deren Zusammenwirken untersucht. Es wird ermittelt, welche Teile zu verbauen sind, wie diese beschaffen sind und angeliefert werden. Die Möglichkeiten zum Handhaben und Fügen, der Automatisierungsgrad sowie Anforderungen an die Qualität werden bestimmt. [HART12, S. 368]

Mit den Informationen aus den ersten beiden Schritten lässt sich bestimmen, wie die Montage ablaufen muss: Im Schritt Montageablaufanalyse wird die Montagereihenfolge bestimmt. Im vierten Schritt, der Funktionsanalyse, kommt ein Zeitermittlungsverfahren zum Einsatz, um für die nötigen Arbeitsschritte die Soll-Zeiten zu ermitteln. Basierend auf den angestrebten Produktionszahlen lässt sich ein Kundentakt bzw. Taktzeit bestimmen. In Kombination mit den Soll-Zeiten lässt sich die Anzahl an Arbeitsstationen ermitteln. Diese haben wiederum Einfluss auf das Layout, das im sechsten Schritt der Systematik festgelegt wird. Sind Montageablauf, Arbeitsstationen und das Layout geplant, kann bestimmt werden, wie viel Personal zum Betreiben des Gesamtsystems benötigt wird. An dieser Stelle verweist LOTTER auch auf automatisierte Arbeitsstationen, ohne jedoch zu erläutern, in welchen Schritt und wie der Automatisierungsgrad zu ermitteln ist. [HART12, S. 368-371]

Die Verfügbarkeitsermittlung reichert die Ergebnisse der vorausgegangenen Schritte an. So wird für die Bauteile eine Anlieferqualität bestimmt. Für die Stationen im Layout wird eine Verfügbarkeit berechnet, die Hochlaufkurve und die Qualifikation des Personals werden bestimmt. Mit diesen Größen lässt sich bestimmen, ob das geplante Montagesystem die gewünschte Stückzahl in der gewünschten Qualität liefern kann. Wenn nicht, müssen ein oder mehrere der vorausgegangenen Schritte wiederholt werden. [HART12, S. 371]

Insgesamt wird der oben beschriebene Prozess in mehreren Iterationen durchlaufen und je Iteration verfeinert. Ist das Gesamtergebnis stimmig, wird das Ergebnis aller Schritte in ein Pflichtenheft geschrieben und somit das Montagesystem spezifiziert. Basierend auf dem Pflichtenheft können Angebote eingeholt werden, die die Basis für eine Investitionsrechnung bilden. In der Regel werden mehrere Angebote eingeholt und mehrere Investitionsrechnungen durchgeführt, sodass am Ende noch entschieden werden muss, welche Variante umgesetzt werden soll. [HART12, S. 371-372]

ECKSTEIN beschreibt einen sehr ähnlichen Ansatz. Zusätzlich betont er jedoch die enge und frühzeitige Vernetzung zur Produktentwicklung, um die integrierte Produkt- und Prozessgestaltung sicherzustellen. Produkte und Prozesse beeinflussen sich gegenseitig. Zur Gestaltung des optimalen Montagesystems ist daher eine enge Abstimmung zwischen Montageplanung und Entwicklung erforderlich. [ECKS13]

JONAS führt als ersten Schritt eines Montageplanungsprozesses die Produktbeeinflussung an [JONA00, S. 13]. Auch HESSE beschreibt die Beeinflussung des Produktes hin zu einer montagerechten Gestalt als eine zentrale Aufgabe der Montageplanung [HESS12, S.9ff].

Die vorgestellten Planungsvorgehen beziehen sich auf die Neuplanung von Montagesystemen. Um hohe Investitionen zu vermeiden und kürzere Produktlebenszyklen darstellen zu können, gewinnt allerdings die Rekonfigurationsplanung, Integrationsplanung oder Umplanung zunehmend an Bedeutung. Als besondere Herausforderung treten hier die bestehenden Produktionsmittel und die mit ihnen verbundenen Restriktionen für die Montageplanung auf. [KAMP11, S. 304, KARL14, S. 3-4]

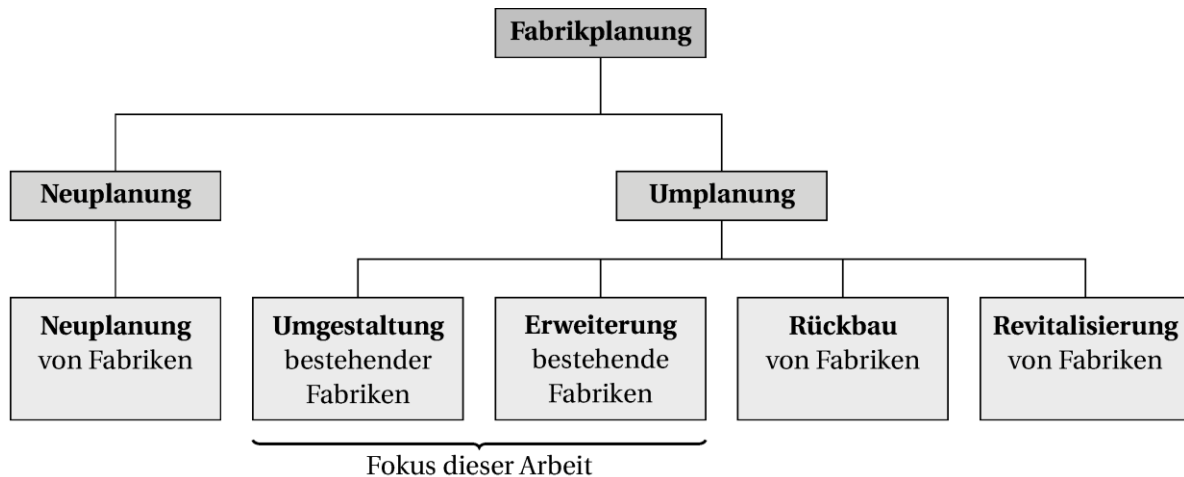


Bild 2.12: Planungsgrundfälle in der Fabrikplanung nach KAMPKER [KAMP11, S. 304]

Basic cases in factory planning by KAMPKER

KAMPKER unterteilt die Umplanung in vier weitere Untergruppen (vgl. Bild 2.12): Die Umgestaltung, bei der die Anlagen modernisiert werden und die Erweiterung, in der die Fabrik neuen, gestiegenen Anforderungen angepasst wird. Weiterhin beschreibt er den Rückbau, der dazu dient, installierte Überkapazitäten abzubauen und die Revitalisierung, in der stillgelegte Fabriken wieder reaktiviert werden. [KAMP11, S. 304]

Der häufigsten Planungsfälle sind Umgestaltungen und Erweiterungen. Daher sollen diese beiden in dieser Arbeit im Fokus stehen. Für die hier betrachteten Fälle und Problemstellungen sei angenommen, dass der Auslöser für eine Umplanung bzw. Erweiterung eine Änderung an einem Produkt oder Bauteil ist.

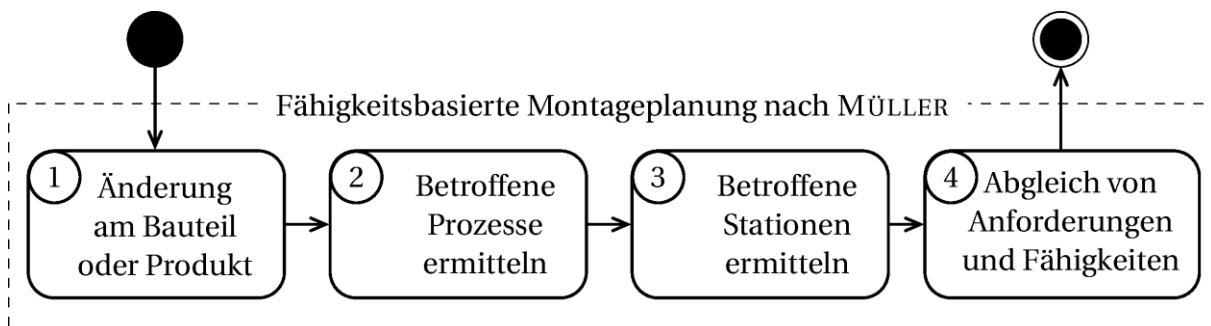


Bild 2.13: Fähigkeitenbasierte Integrationsplanung nach MÜLLER [MUEL15, S. 554]

Capability-based integration-planning by MÜLLER

MÜLLER beschreibt ein Modell, das besonders für die Integrationsplanung geeignet ist. Ausgehend von einer Änderung an einem Bauteil oder Produkt (vgl. Schritt 1, Bild 2.13) werden die Prozesse ermittelt, die von der Änderung betroffen sind (vgl. Schritt 2, Bild 2.13). Zu diesen Prozessen lassen sich wiederum alle Montagestationen ermitteln, auf denen diese ausgeführt werden (vgl. Schritt 3, Bild 2.13). Im vierten Schritt wird verglichen, ob die geänderten Anforderungen des Bauteils oder Produktes mit den Fähigkeiten der in der Station installierten Betriebsmittel erfüllt werden können (vgl. Schritt 4, Bild 2.13). Ist dies der Fall, sind keine weiteren Aktionen der Montageplanung erforderlich. Lassen sich die Anforderungen jedoch nicht erfüllen, so muss die Montageplanung aktiv werden. [MUEL17, S. 553]

Wie oft die Montageplanung aktiv wird hängt demnach von der Änderungsrate der Bauteile oder Produkte ab. Wie in Abschnitt 2.1 beschrieben, überschneiden sich Produktentwicklung und Produktionsplanung zunehmend. Dies führt auch dazu, dass die Montageplanung beginnt, wenn das Produkt noch eine geringe Reife hat. Folglich wird sich das Produkt noch sehr häufig ändern, bis ein seine endgültige Form erreicht. Als grobe Orientierungswerte kann von 1-10 kleineren Änderungen je Woche ausgegangen werden und 1-2 großen Änderungen pro Jahr, wobei von einer großen Änderung mehrere Produktvarianten und Bauteilvarianten betroffen sind.

Kürzere Innovationszyklen und somit kürzere Produktlebenszyklen kommen noch verstärkend hinzu, da auch diese zu Änderungen an den Produkten führen. TÜCKS UND EILERS sprechen in diesem Zusammenhang von der Dimension Dynamik. [TUEC11, S. 4]

Neben dieser Dimension gibt es nach TÜCKS UND EILERS noch eine weitere Dimension: Die Vielfalt. [TUEC11, S. 4]. Diese steigt zunehmend an. Der Grund hierfür liegt darin, dass die verschiedenen Hersteller im Wettbewerb immer stärker versuchen, die individuellen Kundenwünsche möglichst gut zu erfüllen und somit möglichst viele Kunden zu gewinnen oder zu behalten. Für die Montageplanung sind die Vielfalt und die daraus resultierende Komplexität von entscheidender Bedeutung und wird daher im Folgenden näher betrachtet. [TUEC11, S. 4; LOTT12a, S. 3, SCHU05, S. 9; SCHU10, S. 3].

2.3 Produktanalyse für variantenreiche Baukastenprodukte

2.3.1 Varianten in der Produktentwicklung

Variants in product development

Die Variantenvielfalt hat großen Einfluss auf die Montageplanung. Zunächst muss definiert werden, was unter einer Variante zu verstehen ist. GROSSE-HEITMEYER UND WIENDAHL definieren Variantenvielfalt als „die Anzahl der unterschiedlichen Ausführungsformen eines Teiles, einer Baugruppe oder eines Produktes“ [GROS04, S. 7]. Auch FIRCHAU beschreibt Variantenvielfalt als „die Anzahl und die Verschiedenheit der

Varianten eines Bauteils, einer Baugruppe oder eines Produktes“ [FIRC02, S. 12]. DIN199 definiert eine Variante als einen „Gegenstand ähnlicher Form und/oder Funktion mit einem in der Regel hohen Anteil identischer Gruppen oder Teile“ [DIN02]. Während GROSSE-HEITMEYER UND WIENDAHL und FIRCHAU den Unterschied als Kriterium für die Definition von Varianten heranzieht, sieht die DIN199 die Ähnlichkeit als ausschlaggebend. Daraus lässt sich folgern: Varianten sind sich ähnlich aber nicht identisch.

Produktvarianten werden von BUCHHOLZ definiert als „[...] Elemente einer Produktart, in der die Merkmale der zugehörigen Produkte Ähnlichkeit [...] besitzen, aber sich die einzelnen Varianten in geringem Maße voneinander unterscheiden.“ Sie stellt darüber hinaus fest, dass die Unterscheidungskriterien, die zur Differenzierung von Varianten herangezogen werden, subjektiv sind und sich an dem betrachteten Anwendungsfall orientieren. [BUCH12, S. 14] Die Begriffe Modell und Derivat werden oft synonym verwendet.

Die Ursache für steigende Variantenvielfalt liegt in deren positiven Effekten: Unternehmen sind bestrebt, die Kundenwünsche besser zu erfüllen oder durch neue Varianten neue Kunden zu erreichen und auch in Nischen vorzudringen. Schlussendlich sollen diese Bemühungen zu einem gesteigerten Absatz führen. Ein weiterer Grund für die Generierung zusätzlicher Varianten sind Einsparungen am Produkt, die sich bei gewissen Bauteilen generieren lassen. Für einige Länder oder Märkten können die Anforderungen geringer sein, sodass in diesen günstigere Bauteile verwendet werden, als in anderen. [FIRC02, S. 1, RENN07, S. 28]

Diesen positiven Effekten in Einkauf und Vertrieb stehen große Herausforderungen und Nachteile in anderen Unternehmensbereichen gegenüber. Für die Planung und den Betrieb von Montagesystemen sind dies folgende:

Durch die steigende Variantenanzahl sinken die Losgrößen. Es werden kaum mehr gleiche Produkte produziert, fast jedes Produkt ist einzigartig. Durch die Schwankungen der Nachfrage sind die Montageanlagen entweder überbelegt oder unausgelastet. Anlagen und Prozesse, die nur für spezielle Varianten benötigt werden, sind meist unausgelastet. [PILL99, S. 14]

Die Montageanlagen sollen möglichst universell einsetzbar sein, um eine große Bandbreite an Produktvarianten abzudecken. Sie müssen demnach eine hohe Flexibilität aufweisen. Dies macht die Anlagen und somit auch deren Planung aufwendiger und komplexer. Gleichzeitig sind mit solchen Anlagen höhere Investitionen verbunden. [PILL99, S. 14]

Für die Mitarbeiter in den Kommissionierungs- und Montagebereichen besteht die Gefahr, dass es wegen der vielen ähnlichen Teile zu Verwechslungen kommt. Um dies auszuschließen, müssen die Teile öfter kontrolliert werden, beispielsweise durch optische Messverfahren oder Scannen eines Barcode oder DMC (Data Matrix Code).

Auch hier wirken sich die geringeren Losgrößen sowie ein großer Nachfrageunterschied zwischen den Produktvarianten aus: Es gibt Varianten, die von den Mitarbeitern nur sehr selten hergestellt werden. Infolge dessen bleibt der Lerneffekt aus und das Risiko, bei einer Variante einen Fehler zu machen, steigt. [ALDE06, S. 224-225; PILL99, S. 14; WIEN04, S. 10-11]

Die steigende Variantenvielfalt führt zudem zu steigender Komplexität. Jedoch muss unterschieden werden, ob die Variantenvielfalt lediglich zu mehr Bauteilvarianten in der Anstellung an der Linie führt oder sich diese auch auf den Montageprozess auswirkt. BRUNNER spricht hier von montagerelevanten Produktmerkmalen [BRUN15, S. 108]. Ein Beispiel hierfür kann der Fahrersitz eines PKW sein. Hier kann es viele Bauteilvarianten geben, beispielsweise bedingt durch das Material und Farbe des Bezuges sowie elektrische Verstellmöglichkeiten und die Sitzheizung. Der Montageprozess des Sitzes kann bei geschickter Produktgestaltung variantenübergreifend identisch erfolgen, in dem die montagerelevanten Produktmerkmale gleich gestaltet werden. Die Komplexität findet sich dann nur in der Handhabung und Bereitstellung des Bauteils, jedoch nicht im Montageprozess selbst.

Die Komplexität wird weiter erhöht durch die Dynamik, mit der sich die Elemente verändern. [PILL99, S. 5f]. SCHULTE definiert Komplexität als „Anzahl und Diversität der Systemelemente sowie deren Verknüpfungen“. [SCHU95, S. 758]

RENNER fasst die Merkmale von Komplexität nach PATZAK [PATZ82, S. 23] und LINDEMANN [LIND09, S. 10] zusammen (vgl. Bild 2.14). Demnach zeichnet sich Komplexität durch eine hohe Varietät an Elementen aus. Diese setzt sich zusammen aus deren Anzahl und aus der Vielfalt der Arten dieser Elemente. Je mehr Elemente ein System beinhaltet und je unterschiedlicher diese sind, desto höher ist die Komplexität. [PATZ82, S. 22-24]

Die Art der Verbindung dieser Elemente miteinander hat ebenfalls einen Einfluss auf die Komplexität. Ähnlich wie bei der Varietät sind auch hier die Anzahl und die Artenvielfalt relevant. Je mehr Abhängigkeiten es gibt und je vielfältiger diese sind, desto komplexer ist ein System. [PATZ82, S. 22-24]

Neben diesen beiden Einflussgrößen nennt LINDEMANN die (Eigen-)Dynamik des Systems als Beschreibungsgröße für Komplexität. Weiterhin steht ein System immer in Wechselwirkung mit seiner Umwelt und unterliegt somit Einflüssen von außen. Auch diese tragen zur Komplexität eines Systems bei (vgl. Bild 2.14). [LIND09, S. 10]

Aufgrund der Komplexität sind die Zusammenhänge mit einfachen Mitteln nicht nachvollziehbar. Es lassen sich keine Auswirkungen von Entscheidungen ermitteln. Dies führt wiederum dazu, dass keine optimalen Entscheidungen getroffen werden. [EHRL07, S. 27-31]

SNOWDEN führt zur Definition der Komplexitätsbegriffs u.A. ein weiteres Kriterium an: Die Vorhersagbarkeit. Sowohl komplizierte als auch komplexe Systeme verfügen nach

seiner Definition über zahlreiche Wechselwirkungen, aber erst wenn die Systemelemente oder die Beziehungen soweit ansteigen, dass keine Vorhersagbarkeit mehr gegeben ist, spricht er von einem komplexen System. Andersfalls handelt es sich nach der Definition von SNOWDEN um ein kompliziertes System. [SNOW05]

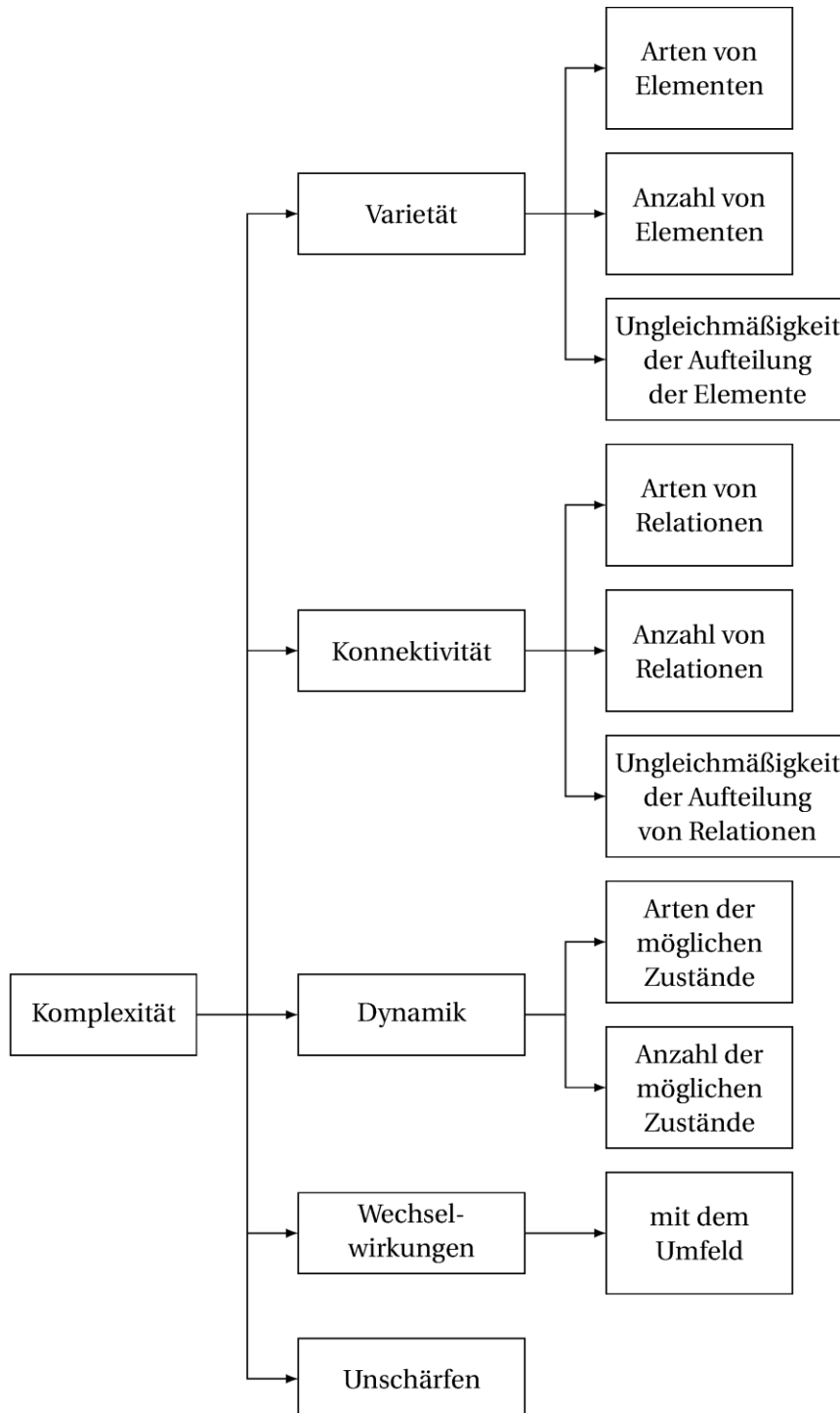


Bild 2.14: Merkmale von Komplexität nach RENNER [RENN07]

Features of complexity by RENNER

2.3.2 Varianteninformationen in der Produktanalyse

Information on variants in the product analysis

Nach der ausführlichen Betrachtung der Grundlagen zum Thema Vielfalt in Abschnitt 2.3 sollen die Auswirkungen auf die Praxis näher untersucht werden.

Wie in Abschnitt 2.2.2 beschrieben ist die Produktanalyse einer der ersten Schritte in der Montageplanung. Der Montageplaner benötigt eine Übersicht der Produkte, die hergestellt werden sollen. [DIET13, S. 325] Diese Übersicht lässt sich mithilfe eines Variantenbaumes darstellen (vgl. Bild 2.15). Dabei werden ausgehend von einer Basis die Äste des Baumes mit Ausprägungen zu den Eigenschaften belegt. Durch die Kombination der verschiedenen Ausprägungen ergibt sich die Baumstruktur. [ZAGE06, S. 46-48]

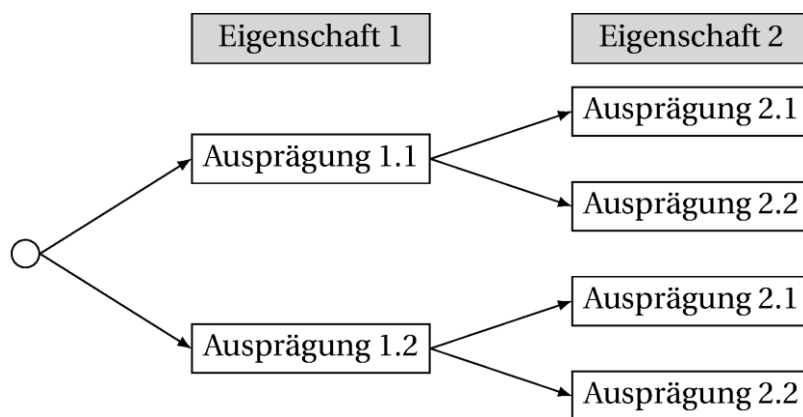


Bild 2.15: Einfaches Beispiel für einen Variantenbaum (eigene Darstellung in Anlehnung an [EVER89; BECK12, ZAGE06])

Simple example of a variant tree

Dies erscheint zunächst einfach, stellt aber in der Praxis oft die erste Hürde dar. Die verschiedenen Produkte laufen zu verschiedenen Zeitpunkten an, zwischen erstem und letztem SOP einer Produktgeneration liegen meist mehrere Jahre. Dazwischen können sich viele Einflussgrößen ändern, was entweder zu neuen Produktvarianten oder zur Streichung geplanter Produktvarianten führen kann. Hierüber zu jederzeit eine Gesamtübersicht zu generieren ist eine Herausforderung.

Ebenfalls sehr wichtig ist die Information, aus welchen Bauteilen die Produktvarianten bestehen. Insbesondere ist entscheidend, welche Bauteilvarianten in welchen Produktvarianten eingesetzt werden.

Klassischerweise werden Produkte als Stücklisten dargestellt. [BUCH12, S. 26-29] Eine gängige Darstellungsform für eine solche Stückliste ist eine Strukturstückliste. Ein Beispiel hierfür ist in Tabelle 2.1 dargestellt. [EIGN09a, S. 83-86] Eine Die Montageplanung muss jedoch viele Produkte in ein Montagesystem einplanen und somit

mehrere Stücklisten gleichzeitig berücksichtigen. In der frühen Phase sind Stücklisten zudem kaum vorhanden, da sie sehr detailliert und somit aufwendig zu erstellen sind.

Tabelle 2.1: Beispiel für eine Strukturstückliste (in Anlehnung an [EIGN09a, S. 83-86; MOOS04, S. 431])

Example of a structured bill-of-material

Fahrrad	Menge
Lenkstange	1
Klingel	1
Griff	2
Stange	1
Vorderrad	1
Felge	1
Speiche	36
Vorderradnabe	1
Hinterrad	
Felge	1
Speiche	36
Hinterradnabe	1

Unabhängig von der gewählten Form der Darstellung der Bauteil- und Produktvarianten ist für den Planer wichtig, dass er schnell erkennen kann, welche Produktvarianten sich Bauteilvarianten teilen. Aus dieser Information lässt sich ableiten, welche Produkte sich sehr ähnlich sind und somit ähnliche Betriebsmittel erfordern, und welche Produkte große Unterschiede aufweisen und somit mehr Planungsaufwand erfordern.

Hinzu kommt, dass der Planer ermitteln muss, wie viele verschiedene Bauteilvarianten sich aus den verschiedenen Produktvarianten ergeben. Im Rahmen seiner Planung muss er diese mit den vorhandenen Kapazitäten in der Logistik und am Montageband abgleichen. Unter Kapazitäten sei in dieser Arbeit die maximal mögliche Anzahl verschiedener Bauteilvarianten verstanden, die in der Logistik oder der Montage gehandhabt, bereitgestellt oder verarbeitet werden können. Sie kann sich zum Beispiel aus

dem Platz an der verbauenden Montagestation oder der verfügbaren Lagerfläche in der Logistik ergeben.

Sind diese Informationen bekannt, kann die Montageplanung eine grobe Abschätzung von Aufwand und erforderlichen Investitionen erarbeiten.

Für die weiteren Analysen durch die Montageplanung benötigt sie Daten über die Anforderungen der Produkte und Bauteilvarianten. Anforderungen sind in diesem Zusammenhang alle Eigenschaften der Bauteilvarianten, die einen Einfluss auf den Montageprozess haben, also beispielsweise die Anzahl und Art der Schrauben mit der zwei Bauteile aneinander befestigt werden.

Basierend auf diesen Informationen, lassen sich die weiteren Schritte der Planungsvorgehen (vgl. Abschnitt 2.2.2) vollziehen. Bei den meisten Planungen in dem für diese Arbeit betrachteten Anwendungsfall handelt es sich um Integrations- oder Umplanungen in denen bereits Strukturen und Betriebsmittel vorhanden sind oder erste Strukturen und Betriebsmittel festgelegt wurden. [SCHU06, S. 167]

2.4 Defizit in der Praxis

Deficit in praxis

Im betrachteten Anwendungsfall liegen alle zur Planung erforderlichen Informationen vor. Jedoch werden die Informationen losgelöst voneinander erstellt und sind somit nicht kompatibel, durchgängig und eindeutig.

Informationen über Bauteilvarianten je Bauteilart werden im betrachteten Anwendungsfall in Form einer Matrix dargestellt. Die Matrix enthält alle für diese Bauteilart relevanten Produkt-Eigenschaften. Auf der Ordinate sind dabei verschiedene Grundmotortypen aufgetragen. Die Abszisse kann je nach relevanten Eigenschaften für jede Bauteilart angepasst werden (vgl. graue Bereich in Tabelle 2.2).

In die Felder der Matrix trägt der Bauteilentwickler beliebige Zeichenketten ein (vgl. weiße Bereich in Tabelle 2.2). Sind die Zeichenketten zweier Felder identisch, bedeutet dies, dass in beiden Produktvarianten die gleiche Bauteilvariante verbaut wird. Jede Zeichenkette entspricht somit einer Bauteilsachnummer.

Dieses Prinzip ist geeignet, um schnell einen Überblick über alle Bauteilvarianten einer Bauteilart zu erhalten. Gäbe es diese Übersicht nicht, müssten alle Stücklisten aller Produktvarianten miteinander verglichen und konsolidiert werden.

Tabelle 2.2: Beispiel für eine Bauteilvariantenmatrix der Bauteilart Schwungrad*Example of a product matrix*

Verbrennungsart	Zylinderzahl	Manuell	Automatik
Otto	3-Zylinder	Schwungrad Variante 1	Schwungrad Variante 2
	4 Zylinder	Schwungrad Variante 3	Schwungrad Variante 2
..

Nachteil ist, dass die Matrix keinen einheitlichen Aufbau hat. Die Bezeichnungen der Merkmals-Ausprägungen unterscheiden sich von Bauteilart zu Bauteilart. Weiterhin ist bei nicht befüllten Feldern innerhalb der Matrix unklar, ob keine Bauteilvariante benötigt wird oder ob das Feld noch nicht befüllt wurde.

Für eine neue Produktgeneration wurden zudem die Matrizen neu erstellt. Somit existiert für jede Produktgeneration eine eigene Matrix. Wenn eine Bauteilvarianten-Bezeichnung in beiden Tabellen auftaucht, ist nicht eindeutig definiert, ob die Bauteilvariante tatsächlich Produktgenerationsübergreifend identisch ist oder rein zufällig eine gleiche Bezeichnung gewählt wurde.

In der Matrix werden von der Produktentwicklung nur wenige ausgewählte Bauteilvarianten dargestellt. Dies sind die Produkte, an denen das Management die grundsätzliche Entwicklungsrichtung festlegt. Basierend auf diesen werden die weiteren Produkte nach denselben Grundsätzen entwickelt. Problem hierbei ist, dass die Produktentwicklung nur ausgewählte Produktvarianten an das Management berichtet. In der Montageplanung müssen jedoch alle Produktvarianten berücksichtigt werden.

Aus der Kombinatorik aller Bauteilvariantenmatrizen lassen sich viele Produktvarianten generieren. Jedoch werden den Kunden nicht alle Kombinationsmöglichkeiten angeboten. So wird beispielsweise nicht jede Motorvariante auch als Hybrid angeboten. Um sicherzustellen, dass nur Produktvarianten geplant werden, die auch real existieren, muss ein Variantenbaum in die Betrachtung einbezogen werden. Ein solcher liegt für den Anwendungsfall zwar vor, jedoch lassen sich nicht alle Eigenschaften aus den Bauteilvariantenmatrizen im Variantenbaum wiederfinden. Dies ist für eine exakte Montageplanung jedoch erforderlich.

In engem Zusammenhang mit dem Variantenbaum sollte die Linienbelegung stehen. Sie legt fest, welche Produkte in welcher Linie hergestellt werden. Mithilfe der Linienbelegung, des Variantenbaumes und der Bauteilvariantenmatrix lässt sich ermitteln, welche Bauteilvarianten an welchen Linien benötigt werden.

Die Linienbelegung ist als Tabelle verfügbar, ähnlich der beispielhaften Darstellung aus Tabelle 2.3. In ihr werden die Produktvarianten jedoch auf einer sehr ungenauen

Ebene deklariert. Dies führt dazu, dass teilweise zu viele Produktvarianten in eine Linie eingeplant werden. Das wiederum bedingt höhere Investitionen, die dann nicht genutzt werden und somit eine Fehlleistung darstellen. Weiterhin sind die Stückzahlen auf Jahresebene zusammengefasst. Es lässt sich nicht bestimmen, ob die Stückzahl ab Jahresbeginn oder erst zum Jahresende geliefert werden muss. Für die Planung von Umbauten und die Steuerung von Lieferanten und die Ermittlung der Auslastung der Kapazitäten ist diese Information jedoch von großer Bedeutung.

Bauteilvariantenmatrix, Variantenbaum und Linienbelegung müssen in hoher Auflösung zur Verfügung stehen, um Fehler in der Montageplanung zu vermeiden. In der Praxis fragen die Montageplaner die fehlenden Informationen bei den zuständigen Ansprechpartnern an. Problem hierbei ist, dass die Information nicht zentral abgelegt ist, sondern für jeden Planer einzeln. Zudem kommt es häufig vor, dass sich die Informationen unterscheiden, je nachdem, welche Ansprechpartner zu welchem Zeitpunkt angefragt wurden. Somit planen die Planer auf Basis verschiedener Ausgangssituationen. In späteren Planungsphasen lässt sich nicht mehr oder nur noch mit großen Aufwand nachvollziehen, was der Planstand für die letzte Planung war.

Tabelle 2.3: Beispiel für eine Linienbelegung

Example of assembly line allocation

Linie	Produkt	Stückzahl			
		2016	2017	2018	...
Linie A	Produkt 1	200.000	180.000	160000	...
Linie A	Produkt 2	0	20.000	30.000	...
Linie B	Produkt 1	0	20.000	140.000	...
...

Die Prozesse und Stationen der Montage sind in jedem Unternehmen in irgendeiner Form beschrieben. Im betrachteten Anwendungsfall passt die Genauigkeit und Beschreibung nicht zu den anderen Informationen, sodass auch hier keine direkte Verknüpfung möglich ist.

Selbst wenn alle Informationen vorliegen, ist ein großer Teil der Arbeit des Planers das Kombinieren der vorhandenen Informationen. Aus der Linienbelegung werden die zu bauenden Produkte für eine Linie ermittelt. Zu diesen Produkten müssen jeweils alle Bauteilvarianten ermittelt werden. Da die Linienbelegung zu verschiedenen Zeitpunkten unterschiedlich ist, muss dies für jeden Zeitpunkt separat durchgeführt werden. Bei einer großen Anzahl an Produkt- und Bauteilvarianten ist das beschriebene Vorgehen mit sehr hohem manuellem Aufwand verbunden und zudem fehleranfällig.

Anschließend muss der Planer bestimmen, in welchen Stationen die Bauteilarten an Prozessen beteiligt sind und abgleichen, ob sich die Prozesse mit den ermittelten Bauteilvarianten durchführen lassen. Dazu müssen alle bekannten Produkthanforderungen und alle Ressourcenfähigkeiten abgeglichen werden. Weiterhin muss bestimmt werden, an welcher Station die Bauteilart verbaut wird und somit von der Logistik ange stellt werden muss.

Wie in Abschnitt 2.2.2 beschrieben, findet ein permanenter Austausch zwischen der Montageplanung und der Produktentwicklung statt, um das Planungsergebnis inkrementell zu verbessern und ein Optimum zwischen Montageplanung und Produktentwicklung sicherzustellen. Dies führt dazu, dass der Planungszyklus sehr häufig durchlaufen werden muss.

2.5 Zwischenfazit und praktischer Handlungsbedarf

Interim conclusion and practical implications

Basierend auf den in Kapitel 1 beschriebenen Herausforderungen wurden in diesem Kapitel die Auswirkungen auf den Produktentstehungsprozess untersucht. Dieser muss vor dem Hintergrund der steigenden Dynamik (vgl. Abschnitt 1.1) und, um auf Schwankungen reagieren zu können, immer häufiger und schneller durchlaufen werden.

Dies führt zu einer Überlappung zwischen Produktentwicklung und Montageplanung, die die Vorteile der Zeitersparnis und der guten Beeinflussbarkeit des Produktes mit sich bringt. Ebenso ist sie aber auch eine Herausforderung, da die Montageplanung bzw. die Produktanalyse zu ihrem Start auf einem unfertigen Produkt basiert und somit häufig angepasst werden muss.

Da die Montage für einen Großteil der Wertschöpfung verantwortlich ist, die Kosten für ein Produkt aber zu beinahe dreiviertel bereits durch die Entwicklung festgelegt werden (vgl. Bild 2.5), ist eine Überlagerung und Abstimmung zwischen diesen beiden Bereichen besonders sinnvoll. Der Montageplanung kommt hierbei die Aufgabe als Bindeglied zu: Sie muss einerseits das Produkt so beeinflussen, dass es montagegerecht gestaltet wird. Andererseits muss sie Investitionen tätigen, um die Produktion für die Herstellung des neuen Produktes zu befähigen. In der Praxis ist dies eine schwierige Aufgabe, da durch zu starke Einschränkungen seitens der Montageplanung wichtige Innovation im Bereich der Produktentwicklung unterbunden werden können und durch zu große Änderung des Produktes hohe Investitionen erforderlich werden können.

Für die Gestaltung einer Montage gibt es sehr viele Möglichkeiten, was das Finden der optimalen Lösung schwierig macht. Zum einen muss für alle Bauteile festgelegt werden, wie diese im Prozess gehandhabt werden können. Weiterhin muss entschieden werden, wie die Teile gefügt und nach vollendeter Montage in Betrieb genommen

werden können (vgl. Bild 2.7). Zum anderen muss für jeden dieser Vorgänge festgelegt werden, ob dieser manuell, automatisiert oder als Hybrid durchgeführt wird (vgl. Bild 2.8).

Die Integrationsplanung wird im aktuellen Stand der Technik nur am Rande betrachtet. Die meisten Vorgehensweisen zur Montageplanung beschreiben eine Neuplanung. Im Anwendungsfall dieser Arbeit steht die Integration neuer Produktvarianten in bestehende Montagesysteme im Vordergrund. Als Basis für die weitere Betrachtung dient die fähigkeitsbasierte Montageplanung nach MÜLLER (vgl. Bild 2.13).

Eine besondere Herausforderung für die Montageplanung stellt Beherrschung der Komplexität aufgrund der Dynamik und Vielfalt dar (vgl. Bild 2.14). Die Vielzahl an Produktvarianten muss durch die Montageplanung hergestellt und abgesichert werden. Die vielen zu berücksichtigenden Eingangsgrößen und deren Auswirkungen erschweren die Produktanalyse deutlich. Beispielhaft sind diese Zusammenhänge in Bild 2.16 dargestellt.

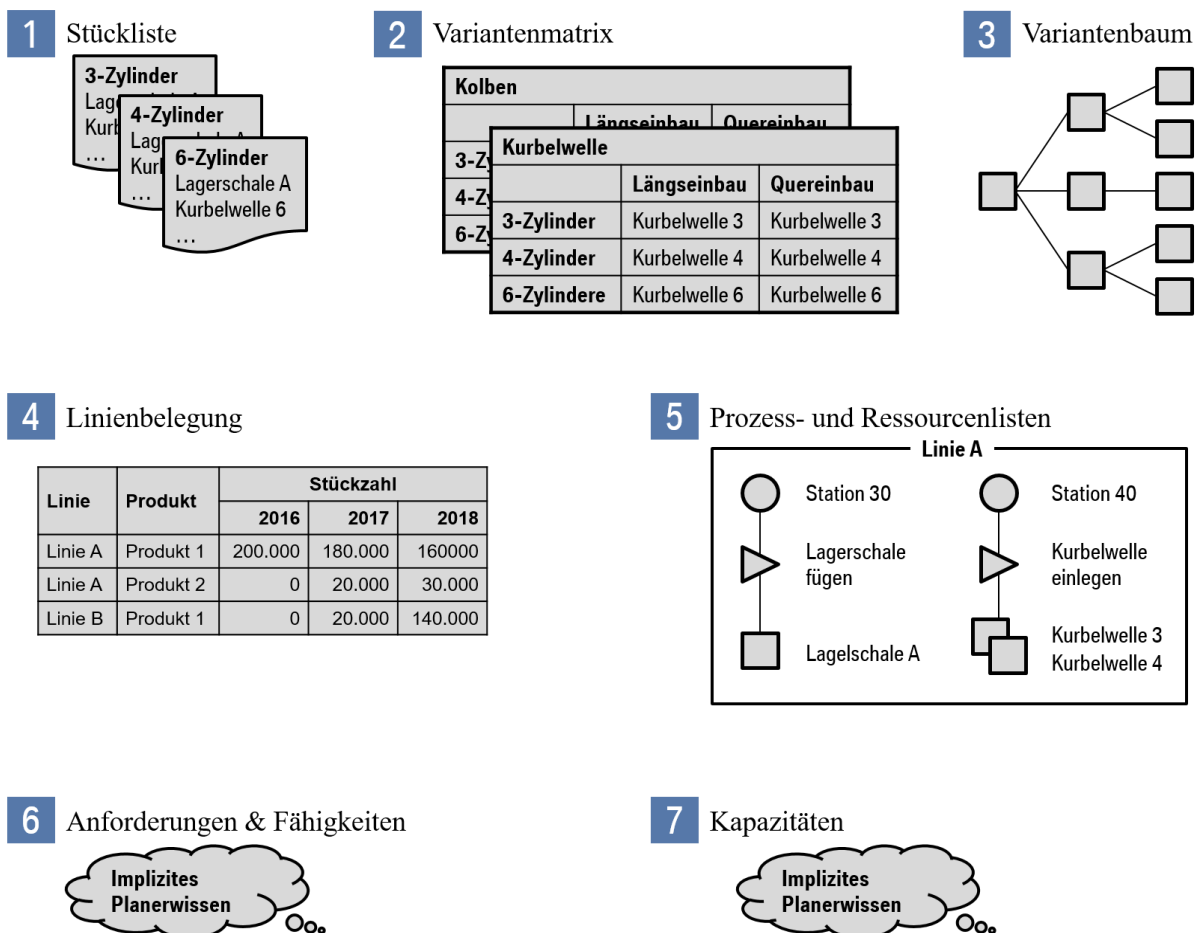


Bild 2.16: Eingangsdaten für die Montageplanung
Input data for assembly planning

In der Praxis liegen die viele Daten, die zum Treffen von Planungsentscheidungen nötig sind meist vor. Allerdings sind Sie unabhängig voneinander in verschiedenen Systemen abgebildet und sind einer gemeinsamen, vernetzten Auswertung nicht zugänglich. Im Zuge ihrer Arbeit stellen die Montageplaner zwar einen Zusammenhang zwischen allen vorliegenden Daten her, jedoch haben sie keine Möglichkeit, diese Zusammenhänge zu dokumentieren. Dies führt einerseits dazu, dass sich Planungsentscheidungen nur schwer rekonstruieren lassen und andererseits dazu, dass bei Änderungen große Teile des Planungsprozesses erneut durchlaufen werden müssen, um zu prüfen, ob sich auf Basis der Daten eine andere Lösung als Optimum ergibt oder ob die bisherige Lösung noch möglich ist. Da die Zusammenhänge zwischen den Daten manuell durch den Planer hergestellt werden müssen, ist dies mit hohem Zeit- und Kostenaufwand verbunden. Anforderungen und Fähigkeiten sowie Kapazitäten liegen meist nur als implizites Planerwissen vor und sind somit einer systematischen, rechnergestützten Analyse ebenfalls nicht zugänglich.

Aus diesen Feststellungen lässt sich der praktische Handlungsbedarf ableiten:

1. Verknüpfung Produkt, Prozess und Ressource:

Das Modell muss die Verknüpfung von Produkt, Prozess und Ressource bzw. deren Unterklassen gemäß der Abhängigkeiten in der Realität mit Fokus auf die Montageplanung abbilden können.

2. Logik über Verknüpfungen / Redundanzfreiheit:

Alle Abhängigkeiten sollen über Verknüpfungen realisiert werden. Angelegte Verknüpfungen müssen erhalten bleiben, es werden keine Kopien oder redundante Daten angelegt. Änderungen der Daten oder Verknüpfungen stehen somit im gesamten Modell zur Verfügung.

3. Integration des Baukastenansatzes:

Produkte die nach dem Baukastenansatz entwickelt wurden, weisen eine starke Ähnlichkeit und Vernetzung auf, da die Module in vielen Produkten übergreifend eingesetzt werden. Das Modell muss diese Logik abbilden und verwalten können.

4. Regelbasierte Verknüpfung / Variantenvielfalt:

Die Darstellung der hohen Produkt- und Bauteilvariantenvielfalt muss mithilfe von Regeln und nicht über direkte Verknüpfung umgesetzt werden.

5. Anforderungen/Fähigkeiten:

Die Erfassung und der automatische Abgleich von Anforderungen und Fähigkeiten muss möglich sein.

6. Montagesystemverbund:

Die Abbildung eines gesamten Montagesystemverbundes muss in einem Modell erfolgen. Dabei soll jedes einzelne Montagesystem flexibel mit Produkten belegt werden können ...

7. Dynamik:

... und die Belegung sich über die Zeit ändern können.

8. Kommunikation:

Die Kommunikation zwischen den verschiedenen Schnittstellen soll erleichtert werden.

9. Analysen:

Auswertungen und Analysen, aus denen sich konkrete Handlungsvorschläge ableiten lassen und die somit eine Optimierung im praktischen Einsatz in der Montageplanung darstellen, müssen bereitgestellt werden.

10. Fehleridentifikation

Fehler im Modell – bspw. durch fehlende Daten – sollten sich automatisiert identifizieren lassen.

3 Bestehende Modelle zur Planung von Montagesystemen

Existing models for the planning of assembly systems

Für eine effiziente Integrationsplanung wird ein Modell benötigt, das die beschriebenen Daten der Produkte, Prozesse und Ressourcen sowie deren Verknüpfung umfasst und diese für die Montageplanung nutzbar macht. In diesem Kapitel werden vorhandene Lösungsansätze aus dem Stand der Wissenschaft vorgestellt und untersucht.

Zunächst werden dabei Grundüberlegungen zur Modellierung angestellt, in denen die Grundlagen zu Modellen aufgezeigt werden (vgl. Abschnitt 3.1). Modelle lassen sich mithilfe von Modellierungssprachen beschreiben. Die zwei verbreitetsten Sprachen werden in Abschnitt 3.2 vorgestellt.

Der Überbegriff für die Vernetzung, Verwaltung und Verwendung von Daten im Rahmen der Produktion und Produktionsplanung ist *digitale Fabrik*. Diese wird in Abschnitt 3.4 näher erläutert. Es existieren bereits sehr konkrete Ansätze und Modelle zur Planung von Montagesystemen, die in Abschnitt 3.5 vorgestellt werden.

Umgesetzt werden Datenmodelle häufig in Datenbanken, daher werden Datenbanken und eine Sprache zur Steuerung und Abfrage von Datenbanken in Abschnitt 3.3 vorgestellt.

In Abschnitt 3.6 werden die Defizite in der Forschung zu Lösung des in Kapitel 2 beschriebenen Problems dargelegt und ein Fazit zum Stand der Technik gezogen.

3.1 Grundüberlegungen zur Modellierung

Basic thoughts about modelling

Ein Modell hat nach STACHOWIAK drei Hauptmerkmale: Das Abbildungsmerkmal, das Verkürzungsmerkmal und das Pragmatische Merkmal. [STAC73, S. 131-134]

Das Abbildungsmerkmal besagt, dass ein Modell immer ein Abbild von etwas ist. STACHOWIAK spricht hier von Originalen. Es gibt keine Einschränkung, was dieses Original sein muss. Es kann ein reales Objekt oder auch ein rein fiktiver Zusammenhang sein. Es kann auf Naturgesetzen oder auf von Menschen erdachten Regeln beruhen. Alles, was sich in irgendeiner Form erfassen lässt, kann als Original für ein Modell verwendet werden. [STAC73, S. 131-132; RICH09, S. 7-8, TOEL10, S. 9]

In einem Modell sind niemals alle Eigenschaften und Zusammenhänge des Originals abgebildet. Dies bezeichnet STACHOWIAK als Verkürzungsmerkmal. Ein Modell vereinfacht das Original und vermeidet Attribute, die keinen oder nur einen sehr geringen Einfluss auf das Ziel des Modells haben. Somit ist auch festgelegt, dass jedes Modell ein Ziel haben muss und dass das Ziel entscheidend für die Gestaltung des Modells ist. Dies nennt STACHOWIAK das pragmatische Merkmal. Für STACHOWIAK ist nicht nur

das zu modellierende Original relevant, „sondern auch, für wen, wann und wozu [...] das Modell ist“. [STAC73, S. 133; RICH09, S. 7-8; TOEL10, S. 10]

BECKER fasst die Merkmale von STACHOWIAK zusammen zu einer Definition des Modellbegriffs: „Modelle sind immaterielle Repräsentationen eines Originals für Zwecke eines Subjekts. Modelle sind Abstraktionen. Sie beziehen die Dinge ein, die für den Zweck des Subjektes notwendig sind, und lassen die Dinge außen vor, die dem Zweck nicht dienlich sind“ [BECK12, S. 1]. WERNERS verwendet eine ähnliche Definition: „Ein *Modell* ist ein zweckorientiertes, ggf. vereinfachtes Abbild eines Ausschnitts der Realität, welches hinsichtlich der interessierenden Zusammenhänge *strukturähnlich* oder *strukturgleich* ist“ [WERN13, S.3]. Eine kürzere Definition erarbeitet RICHTER: „Ein Modell ist eine verkürzte Abbildung eines Originals, deren Verwendung einem untersuchenden Subjekt hinsichtlich seiner Erkenntnisgewinnung zweckmäßig erscheint“ [RICH09, S. 8].

In diesen Definitionen lassen sich die Merkmale nach STACHOWIAK wiederfinden. Alle Definitionen sind sich in ihren Grundgedanken sehr ähnlich, daher kann davon ausgegangen werden, dass der Modellbegriff in der Wissenschaft einheitlich verstanden wird.

Bei allen betrachteten Autoren betrachteten Autoren drei Modelltypen auf: Beschreibungsmodelle, Erklärungsmodelle und Entscheidungsmodelle. Beispielhaft ist in Bild 3.1 die Darstellung nach WERNERS inklusive einer Erklärung und einem Beispiel zu sehen.

Ein Beschreibungsmodell dient vor allem dazu, einen bestimmten Zustand festzuhalten. Es gibt meist einen bestimmten Fokus, wie beispielsweise die Aufstellung der Kosten für eine Montagelinie (vgl. Bild 3.1). Ein Beschreibungsmodell dient dazu, einen Überblick über eine Situation zu bekommen. Die erfassten Zusammenhänge liegen meist in der Vergangenheit und müssen sich mit Sinnesorganen oder Messinstrumenten wahrnehmen lassen. [SCHW97, S. 5; WERN13, S. 3]

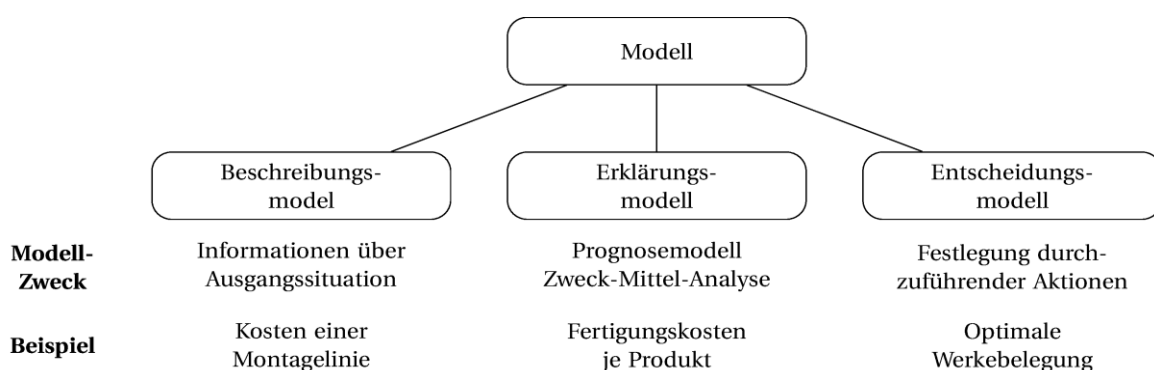


Bild 3.1: Arten von Modellen nach WERNERS mit eigenen Beispielen [WERN13, S. 4]

Types of models by WERNER with own examples

Erklärungsmodelle haben einen in die Zukunft gerichteten Fokus. Sie stellen Zusammenhänge her, aus denen sich Aussagen für künftige Ereignisse ableiten lassen. Häufig werden sie genutzt, um geplante Aktivitäten auf ihre Auswirkungen hin zu untersuchen – im Sinne einer *was wäre wenn* Analyse. [SCHW97, S. 5-7; WERN13, S. 3]

Einen Schritt weiter gehen Entscheidungsmodelle. Sie geben konkrete Handlungsanweisungen. Dabei kommen meist Optimierungsverfahren zum Einsatz. [SCHW97, S. 7-8] Ein praktischer Anwendungsfall für solche Modelle ist die Werkebelegung (vgl. Bild 3.1). Große Autohersteller haben mehrere Werke auf der ganzen Welt. Auf die Entscheidung, welches Produkt in welchem Werk hergestellt wird, wirken zahlreiche Einflüsse. Die Nachfrageprognosen der Märkte, Lohnkosten, Kosten für die Anpassung der Betriebsmittel und Logistikkosten für den Transport der Materialien in das Werk und der fertigen Produkte zu den Abnehmern sind nur einige der Faktoren. Ein Entscheidungsmodell kann all diese Faktoren einbeziehen und eine optimale Belegung der Werke errechnet.

Ähnliche Darstellungen wie die von WERNERS aus Bild 3.1 sind bei zahlreichen weiteren Autoren zu finden. In Bild 3.2 sind die Modellarten wichtiger Autoren aufgezeigt und den Autoren zugeordnet. Manche Autoren fügen noch weitere Modellarten hinzu. Es ist aber zu erkennen, dass in der Forschung weitestgehend Einigkeit über Einteilung der Modelle in Modellarten herrscht.

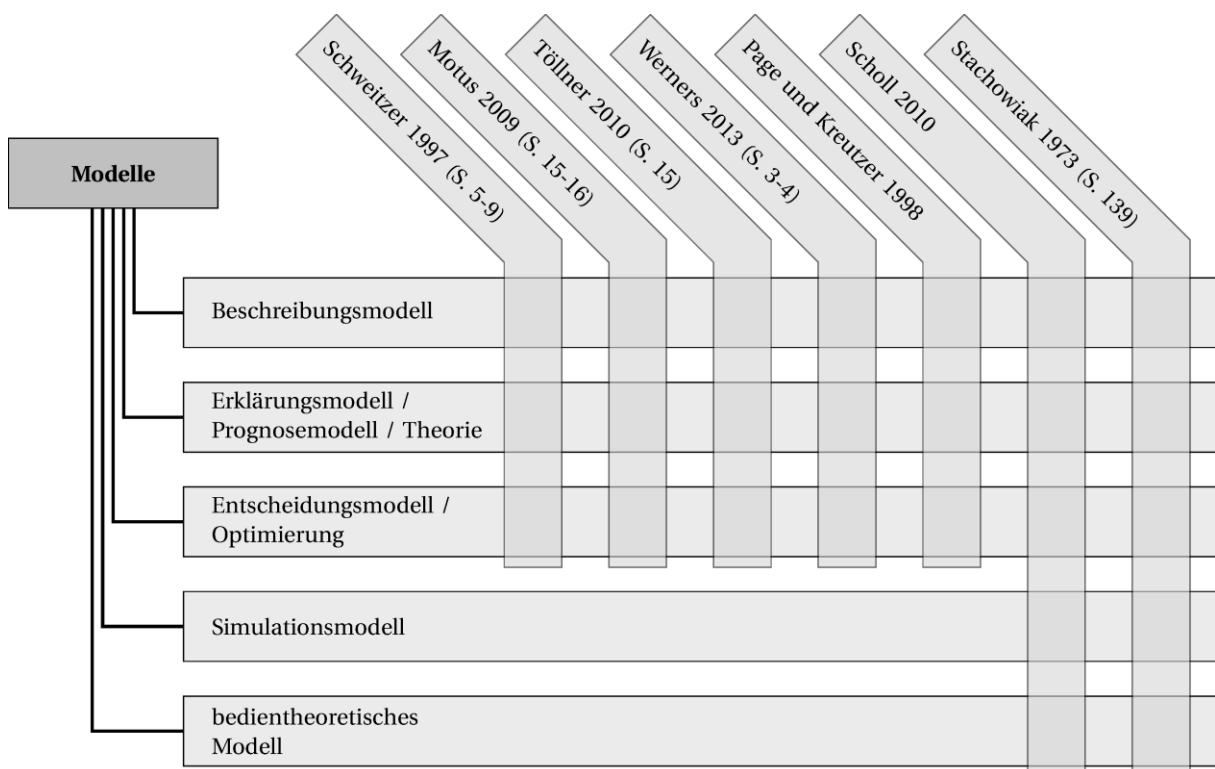


Bild 3.2: Modelltypen der verschiedenen Autoren

Types of models by the different authors

Es gibt darüber hinaus viele weitere Arten von Modellen aus anderen Forschungsgebieten. Modelle lassen sich nach vielen weiteren Parametern gliedern, bspw. über ihr zeitliches Verhalten (statisch, dynamisch) oder die Art ihrer Formulierung (graphisch, technisch/physikalisch, semantisch, ...). Für nähere Informationen zur Definition und Fein-Gliederung von Modellen sei auf entsprechende Literatur, bspw. [STAC73], [RICH09], [SCHW97], [TOEL10]. [STAC73], [RICH09], [SCHW97], [TOEL10].

Aufbauend auf der Definition von Modellen und der Einordnung in verschiedene Modelltypen, wird im Folgenden der Ablauf zur Erstellung eines Modells beschrieben. Um ein auf Algorithmen basierendes Modell für die Lösung eines realen Problems einsetzen zu können, werden üblicherweise mehrere Schritte durchlaufen. Dazu muss das in der Realität auftretende Problem analysiert werden und in ein verbales Modell oder Wortmodell überführt werden. Hierin werden das Problem und zugehörigen Zusammenhänge verbal beschrieben. [WERN13; BOSS92]

Im zweiten Schritt wird ein mathematisches Modell erstellt. In diesem werden die Zusammenhänge aus Schritt 1 in eine mathematische Form überführt. BOSSEL spricht hier von Wirkbeziehungen, die analysiert werden und logischer Deduktion, in der die sprachliche Beschreibung in eine formale, logische Beschreibung überführt wird. [WERN13; BOSS92, S. 50-53]

Dieses in Logiken beschriebene Modell lässt sich in Algorithmen überführen, die als Basis für eine Software-Lösung dienen. Ist das Modell in Software umgesetzt und wurde mit dieser eine Lösung ermittelt, lässt sich diese auf das mathematische Modell übertragen. Die Lösung des mathematischen Modells lässt sich wiederum in eine Lösung für das verbale Modell überführen. Das Ergebnis der IT muss also unter Berücksichtigung der Wirkzusammenhänge in eine verbale Aussage überführt werden. Aus dieser lässt sich dann die Lösung des realen Problems ableiten. [WERN13; BOSS92, S. 50-53]

Die beschriebenen Schritte werden nicht einmalig sequentiell durchlaufen. Vielmehr werden die Schritte iterativ durchlaufen. So kann bei der Aufstellung von logischen und mathematischen Zusammenhängen auffallen, dass das Problem an bestimmten Stellen des verbalen Modells noch nicht ausreichend beschrieben ist. Folglich wird das Problem weiter analysiert und das verbale Modell verbessert. Ebenso ist es denkbar, dass die mithilfe des Modells ermittelte Lösung nicht zufriedenstellend ist und die Schritte erneut durchlaufen werden müssen. [WERN13, S. 3]

Die beschriebenen Schritte sind in Bild 3.3 dargestellt. Die Abbildung ist in Anlehnung an WERNERS entstanden und wurde um die iterativen Schleifen, die im Zuge der Modellentwicklung zu durchlaufen sind, ergänzt. [WERN13, S. 2-3]

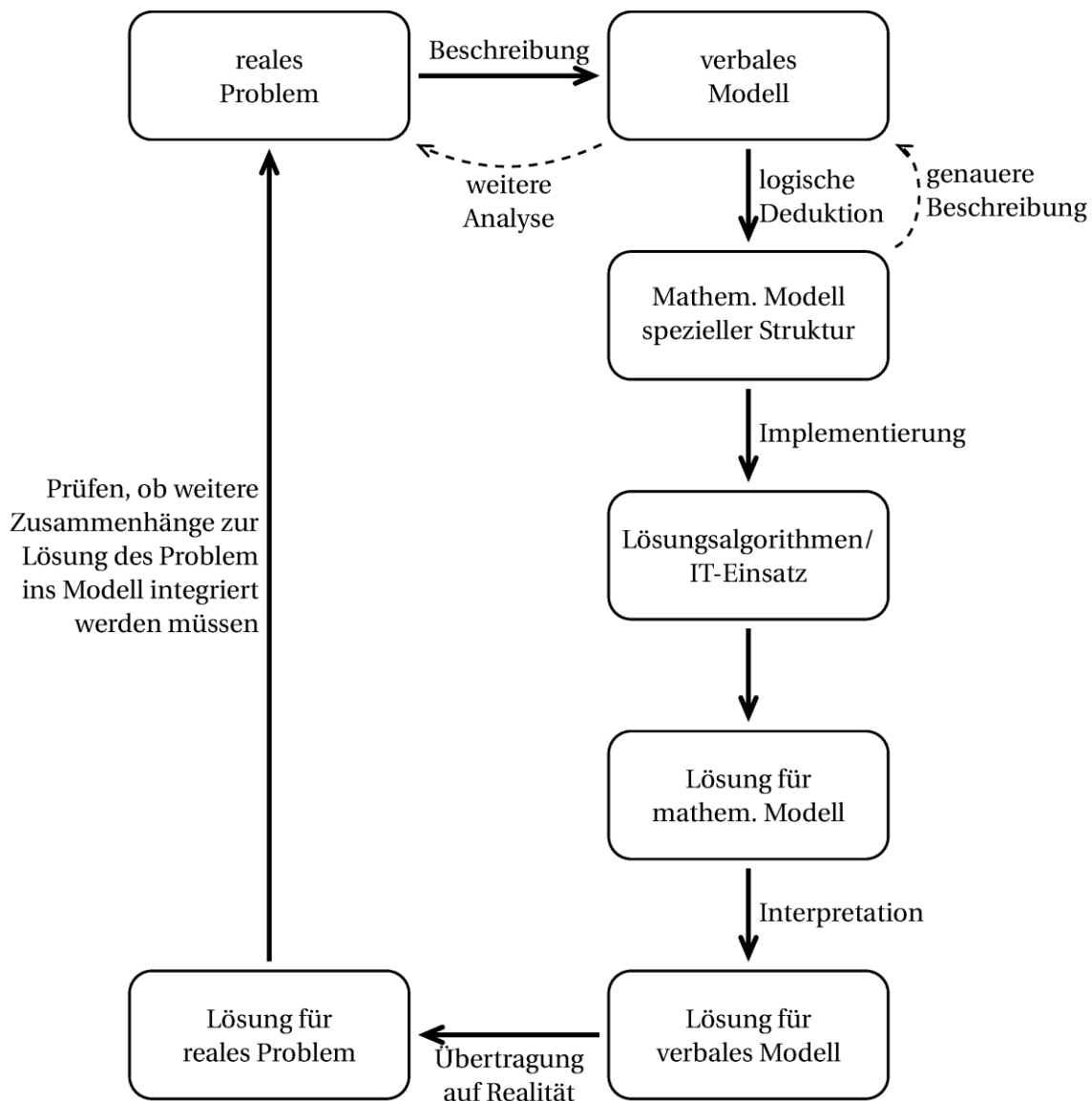


Bild 3.3: Modellgestützter Problemlöseprozess in Anlehnung an WERNERS [WERN13, S. 2]
Process of solving problems with models in imitation of WERNERS

KATZY definiert einen ähnlichen Prozess, jedoch legt er den Fokus auf produzierende Unternehmen (vgl. Bild 3.4). [KATZ94, S.52-57]

Durch eine beliebige Anforderung wird der Prozess gestartet – oder angeregt. In der Phase der Modellbildung wird versucht, die entscheidenden Einflussgrößen auf das System sowie deren Zusammenhänge und Wechselwirkungen zu ermitteln. Wie zu Beginn von Abschnitt 3.1 beschrieben, muss ein Modell einerseits die Wirklichkeit vereinfachen und dazu Elemente vernachlässigen. Andererseits müssen alle maßgebenden Einflüsse in das Modell einbezogen werden. Genau diese Entscheidungen werden im Rahmen der Modellbildung getroffen. Die Daten werden strukturiert und anschließend wird ein Modell erstellt, welches den aktuellen Zustand darstellt. [KATZ94, S.54]

In der Modellexploration werden zunächst kleine Versuche mit dem Modell durchgeführt, um dessen korrekte Funktionsweise zu überprüfen. Im Anschluss werden verschiedene Alternativen mithilfe des Modells auf ihre Konsequenzen hin untersucht und bewertet (vgl. Bild 3.4). [KATZ94, S.54]

Die Phase der Entscheidung ähnelt der Übertragung auf die Realität aus Bild 3.3. Hier werden die Resultate aus der Modellexploration herangezogen, um eine Entscheidung zu treffen und in der Realität umzusetzen. Wie auch der Prozess von WERNERS (Bild 3.3) muss auch der Prozess von KATZY nicht streng in dieser Reihenfolge durchgeführt werden. Es kann jederzeit zwischen den Stufen gesprungen werden. Die Kenntnisse aus der Modellexploration können beispielsweise genutzt werden, um das Modell im Rahmen der Modellbildung weiter zu verbessern, zu verfeinern oder damit Probleme in der Realität zu lösen. [KATZ94, S.54]

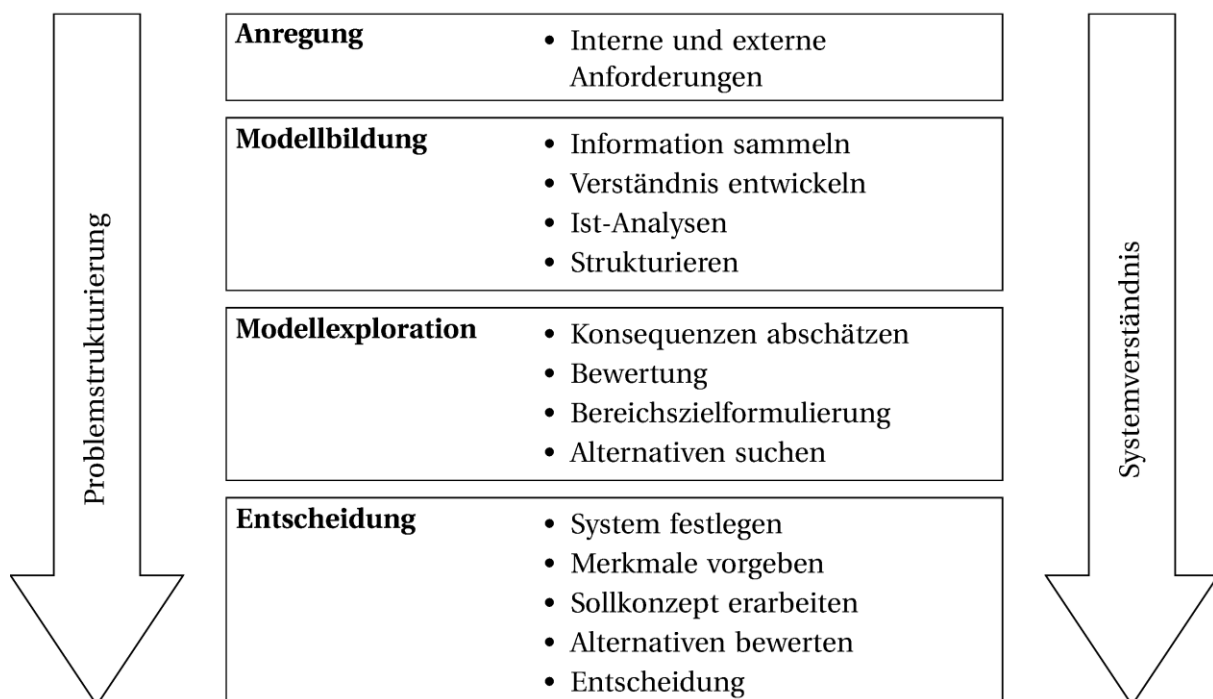


Bild 3.4: Systemtechnischer Lösungsweg nach Katzy [KATZ94, S. 53]

Systematic solution process by Katzy

3.2 Modellierungssprachen

Modelling languages

In Abschnitt 3.1 wurde allgemein hergeleitet, was ein Modell charakterisiert und welche Grundsätze bei der Erarbeitung eines Modells zu beachten sind. Weiterhin wurde erläutert, welche Arten von Modellen es gibt und wie Probleme mithilfe von Modellen gelöst werden. Im Modell, das in dieser Arbeit entwickelt wird, sollen Daten modelliert werden. Dafür gibt es nach BECKER zwei wesentliche Modellierungssprachen: Entity-Relationship-Diagramme (ERM) und UML-Klassendiagramme, die in den folgenden

beiden Abschnitten 3.2.1 und 3.2.2 vorgestellt werden. [BECK12] Weiterhin zu nennen ist die Produktdaten-Modellierungssprache EXPRESS-G, die in ISO 10303-11 definiert und z.B. von NEUHAUSEN zu Modellierung verwendet wird.

3.2.1 Entity-Relationship-Model (ERM)

Entity-Relationship-Model (ERM)

Das Entity-Relationship-Model wurde erstmals 1976 von CHEN vorgestellt. Es verfolgt das Ziel, Zusammenhänge zwischen Daten effizient darzustellen. Nach CHEN werden mittels ERM Entitäten – also reale oder fiktive Einheiten – und deren Zusammenhänge modelliert und grafisch dargestellt. [CHEN76, S.9ff]

Als einfaches Beispiel nennt er den Zusammenhang zwischen Projekten und Mitarbeitern. Ein Mitarbeiter kann in mehreren Projekten tätig sein und einem Projekt können mehrere Mitarbeiter zugewiesen werden. Dieser Zusammenhang wird m:n-Verknüpfung genannt. Mithilfe des ERM lassen sich diese Zusammenhänge leicht darstellen (vgl. Bild 3.5). [CHEN76, S. 19]

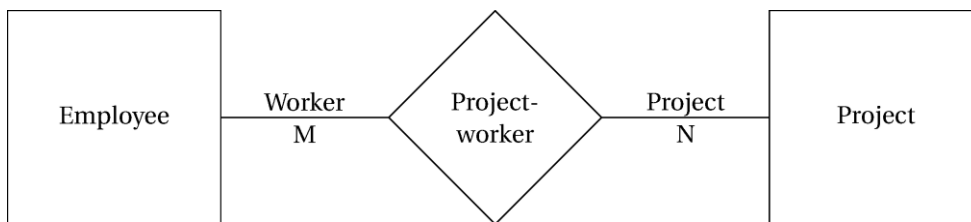


Bild 3.5: Beispiel für ein Entity-Relationship Diagramm [CHEN76, S. 19]

Example of an Entity-Relationship-Diagram

Die grundsätzliche Beschreibung des ERM von CHEN wurde mit den Jahren erweitert bis zu ihrer heutigen Form wie sie von BECKER beschrieben wird. Unter anderem werden Attribute in Form von Ellipsen direkt an den Entitäten und den Beziehungen dargestellt (vgl. Bild 3.6). Für weitere Elemente im modernen ERM sei auf BECKER verwiesen. [BECK12, S. 4-9]

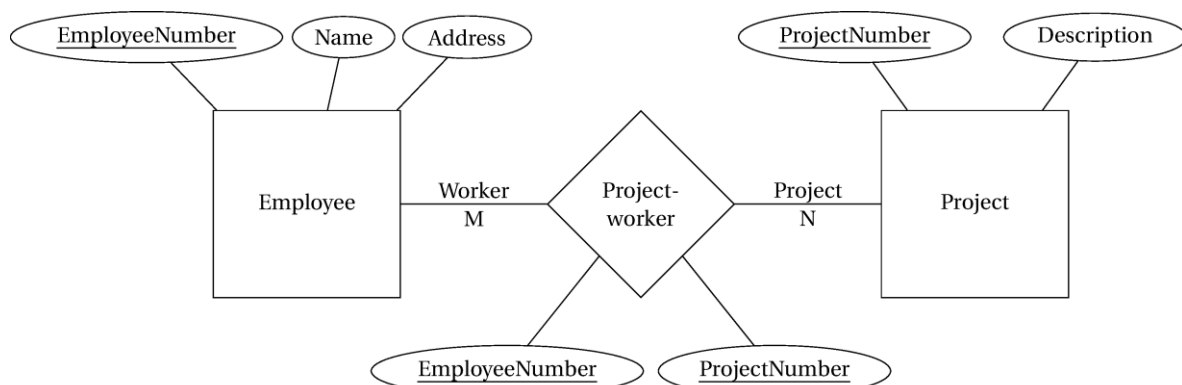


Bild 3.6: Darstellung von Attributen im ERM [BECK12, S. 8]

Description of attributes in ERM

3.2.2 UML-Klassendiagramm und Objektdiagramm

UML class diagram and object diagram

Eine weitere Notation zur grafischen Abbildung von Modellen ist die Unified-Modeling-Language (UML). Auch sie dient zur Entwicklung und Dokumentation von Elementen und deren Zusammenhängen. UML fasst verschiedene Methoden der objekt-orientierten Modellierung zusammen und vereinheitlicht diese. Dabei lassen sich mittels UML sowohl statische Strukturen als auch dynamische Verhaltensweisen darstellen. [RUMB05, S. 3-4, 12-13; ISO05, S. 2, S.15]

In der UML Notation sind zahlreiche Diagrammtypen beschrieben. Für diese Arbeit sind davon das Klassendiagramm und das Objektdiagramm relevant. Für die weiteren Diagrammtypen sei auf entsprechende Literatur verwiesen. [vgl. ISO05, S.4; RUMB05]

In einem Klassendiagramm sind Klassen und deren Beziehungen zueinander dargestellt [RUMB05, S. 28; ISO05, S. 201]. Klassen werden verwendet, um reale oder imaginäre Objekte zu modellieren. Diese Objekte können physischer Natur sein, wie bspw. ein Verbrennungsmotor, oder abstrakt wie bspw. ein Bankkonto oder eine Arbeitsauftrag. [RUMB05, S. 47]

Eine Klasse wird durch ein Rechteck dargestellt, das in drei vertikal untereinanderliegende Bereiche eingeteilt ist. Im obersten Bereich steht der Klassenname. Dieser muss im gesamten Klassendiagramm eindeutig sein. Er wird zentriert und fett gedruckt dargestellt. [RUMB05, S. 28, S. 50, S. 215; ISO05 S. 32]

Im zweiten Bereich sind die Eigenschaften, die zu der Klasse gehören, dargestellt. Diese werden auch Attribute genannt. Zu einer Klasse Motor könnte beispielsweise das Attribut Leistung gehören. Vor dem Attribut wird per Plus- oder Minussymbol die Sichtbarkeit des Attributs definiert [RUMB05, S. 113]. Darüber hinaus gibt es noch *protected* und *package* als Optionen für die Sichtbarkeit. Diese sind für die Arbeit nicht relevant. Interessierte seien auf entsprechende Literatur wie bspw. [BECK12] verwiesen. Hinter dem Namen des Attributes folgt ein Doppelpunkt und danach der Typ des Attributes, also Integer für eine Ganzzahl, String für eine Zeichenkette usw. [RUMB05, S. 28, S. 50; BECK12, S. 12]

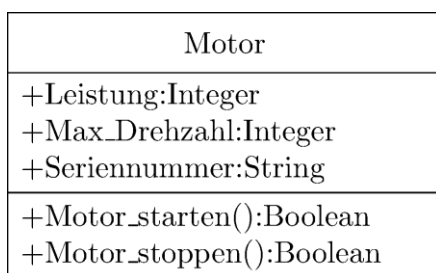


Bild 3.7: Beispiel für eine Klasse eines UML Klassendiagramms

Example for a class of a UML class diagram

Im dritten, untersten Bereich werden die Methoden der Klasse definiert. Dies sind Aktionen, die die Klasse ausführen kann. Beim Verbrennungsmotor könnte eine Funktion zum Beispiel *Motor_starten* heißen. Diese enthielte die notwendigen Schritte zum Starten des Motors. Für Methoden wird die Sichtbarkeit nach demselben Prinzip dargestellt, wie bei Attributen. Die Rückgabewerte der Methoden stehen – ebenfalls wie bei Attributen – nach einem Doppelpunkt der auf den Methodennamen und die Parameterliste folgt. [BECK12, S. 9] Ein Beispiel für eine Klasse ist in Bild 3.7 dargestellt.

Es gibt die Möglichkeit, ein Klassendiagramm verkürzt darzustellen. In der verkürzten Variante wird nur der Klassenname angegeben. Die Attribute und Methoden werden nicht aufgeführt. In dieser verkürzten Variante liegt der Fokus auf den Beziehungen der Klassen zueinander. [RUMB05, S. 28]

Zur Verknüpfung von Klassen gibt es mehrere Möglichkeiten. Folgende vier sind für diese Arbeit relevant und werden daher näher betrachtet:

1. Assoziation,
2. Komposition,
3. Aggregation und
4. Vererbung.

Eine Assoziation ist eine allgemeine Beschreibung einer Verbindung zwischen zwei Klassen. Die Beziehung ist nicht gerichtet und somit kann auch keine Aussage getroffen werden, ob eine Klasse Kenntnis von der verbundenen Klasse hat. Dennoch lässt sich mittels einer Assoziation festlegen, dass die Objekte in einer Beziehung zueinander stehen. Komposition und Aggregation verfeinern diese Beziehungsinformation und sind somit genauere Beschreibungen einer Assoziation. [RUMB05, S. 53-57]

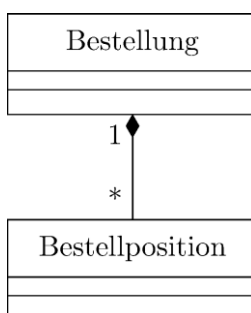


Bild 3.8: Beispiel für eine Komposition [BECK12, S. 12]

Example of a composition

Mittels Kompositionen lassen sich Zusammenhänge modellieren, die nicht ohneinander existieren können. Entsprechend können die Klassen einer Komposition nur gemeinsam existieren. Diese Beziehung wird daher *starke Beziehung* genannt. In der Regel wird mit Erstellung der Oberklasse die Unterklasse mit angelegt. Im Beispiel von BECKER, das in Bild 3.8 dargestellt ist, kann die Bestellung nur existieren, wenn

mindestens eine Bestellposition enthalten ist. Entsprechend gilt, dass mit Löschung der Bestellung auch die Bestellpositionen gelöscht werden und eine Bestellposition nur in einer Bestellung existieren kann. [RUMB05 S. 56f, S. 264-270; ISO05 S. 238-241; BECK12, S. 11-12]

Um eine schwache Beziehung zwischen Klassen zu beschreiben wird eine Aggregation verwendet. Im Gegensatz zur Komposition können die beiden verbundenen Klassen unabhängig voneinander existieren und somit auch unabhängig voneinander erstellt werden. Die Aggregation lässt sich umschreiben mit *verwendet ein*. [RUMB05 S. 56f, S. 164-168; BECK12 S. 12]

Im Beispiel von BECKER (vgl. Bild 3.9) kann eine Filiale existieren, ohne Mitarbeiter. Ebenso kann der Mitarbeiter ohne die Filiale existieren. Für die Instanziierung, also das Anlegen eines konkreten Objektes einer Klasse, bedeutet dies, dass die Instanz des Mitarbeiters losgelöst von der Filiale erstellt werden kann. Zu einem beliebigen Zeitpunkt wird ein Verweis von der Filiale auf den Mitarbeiter erstellt, sodass beide Klassen verknüpft sind. Die Filiale *verwendet* anschließend den Mitarbeiter. Diese Verbindung lässt sich wieder auflösen, sodass der Mitarbeiter nicht mehr zur Filiale zugewiesen ist. Ebenso ist es möglich, dass mehrere Instanzen von Filiale auf denselben Mitarbeiter zugreifen. [BECK12, S. 10-12]

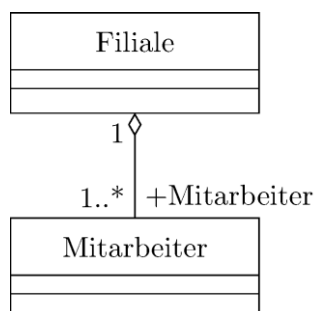


Bild 3.9: Beispiel für eine Aggregation [BECK12, S. 12]

Example of an aggregation

Vorteile dieser Möglichkeiten der Verknüpfung ist, dass das Modell sehr genau beschrieben werden kann. Die Beziehung der Klassen zueinander und Abhängigkeit der Klassen voneinander lassen sich über die verschiedenen Verknüpfungsarten implementieren.

Als zusätzliche Information der Verknüpfung zwischen zwei Klassen wird die Multiplizität angegeben. Diese beschreibt, wie viele Elemente der einen Klasse der anderen Klasse zugewiesen werden können. Im Beispiel aus Bild 3.9 bedeutet dies, dass einer Filiale beliebig viele Mitarbeiter zugewiesen werden können. Symbolisiert wird dies durch die 1 am Beginn der Verknüpfung und den Ausdruck 1..* am Ende. [ISO05, S. 234; RUMB05, S. 28, 53-54]

Multiplizitäten können mehrere Gestalten annehmen. Entweder sind sie als konstante Zahl angegeben oder beschreiben einen Zahlenbereich. Bereiche werden in der Form $x..y$ angegeben, wobei x der Startwert und y der Endwert ist. Ist der Endwert beliebig wird dies durch einen Stern beschrieben. [RUMB05 S. 52-54]

Um die Beziehung umsetzen zu können, muss die Zielklasse ein Attribut enthalten, das auf die Quellklasse verweist. Wie in Bild 3.9 zu sehen erfolgt dies über eine Bezeichnung, die an die Pfeilspitze geschrieben wird. Im Beispiel erhält die Filiale demnach ein Attribut Mitarbeiter. Eine mögliche Implementierung wäre eine Liste, in der alle zugewiesenen Mitarbeiter gespeichert werden. [BECK12, S. 10-11]

Eine weitere Art der Beziehung zwischen Klassen ist die Vererbung, auch Generalisierung genannt. Bei einer Vererbung oder Generalisierung übernehmen die Unterklassen dabei die Attribute und Methoden der übergeordneten Klasse. Im Beispiel von BECKER aus Bild 3.10 erben die Klassen Geschäftskunde und Privatkunde die Attribute KundenNr und Name sowie die Methode ermittleAnzahlEinkäufe von der Klasse Kunde. [BECK12, S. 13; RUMB05 S. 57-59]

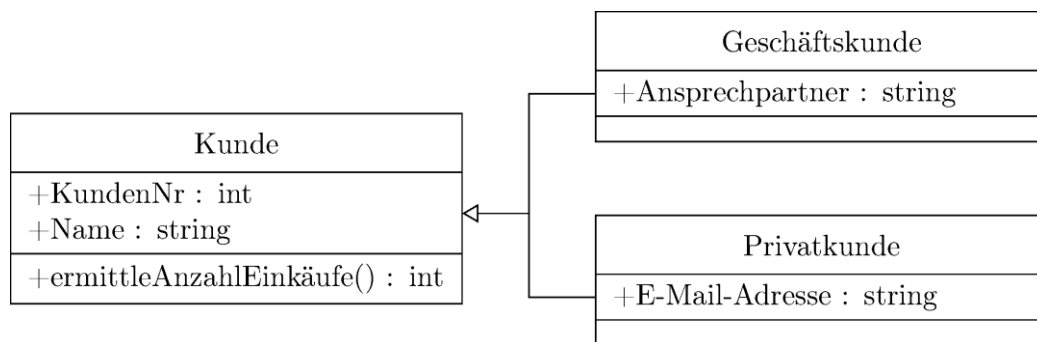


Bild 3.10: Beispiel für eine Vererbung [BECK12, S. 13]

Example of a generalization

Neben diesen gibt es noch einige weitere Elemente in Klassendiagrammen. Der bisher vorgestellte Umfang reicht jedoch zur Vorstellung der in der Literatur bestehenden Modelle aus. Daher wird auf die weiteren Elemente nicht eingegangen. Für eine vollständige Übersicht der Elemente, die in der UML verwendet werden, sei auf [RUMB05] verwiesen.

Ein Klassendiagramm ist eine allgemeine Beschreibung über die Zusammenhänge mehrerer Klassen. Um eine konkrete Ausprägung dieser Zusammenhänge zu zeigen eignen sich Objektdiagramme. Sie zeigen anhand eines Beispiels, wie das Klassenmodell funktioniert und welche Zustände es haben kann. Außerdem kann gezeigt werden, wie es sich über die Zeit oder bei bestimmten Aktionen ändert. Formell gesprochen ist ein Objektdiagramm eine Instanz eines Klassendiagramms zu einem bestimmten Zeitpunkt. [ISO05, S.201; RUMB05, S. 67]

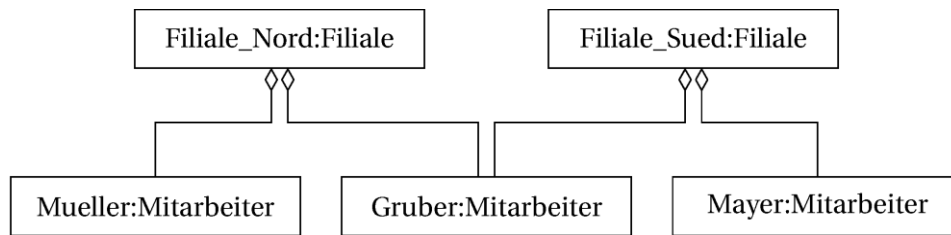


Bild 3.11: Beispiel für ein Objektdiagramm auf Basis des Klassendiagramms aus Bild 3.9
Example of an object diagram based on the class diagram of Bild 3.9

In Bild 3.11 wurde das Klassendiagramm aus Bild 3.9 in ein Objektdiagramm überführt. In diesem Beispiel gibt es zwei Filialen, von denen jede zwei Mitarbeiter hat. Der Mitarbeiter Müller arbeitet in der Filiale Nord und der Mitarbeiter Mayer in der Filiale Süd. Der Mitarbeiter Gruber fungiert als Springer und ist somit beiden Filialen zugeordnet. Da der Mitarbeiter aber nur einmal in der Datenstruktur angelegt werden soll, sind per Assoziation beide Filialen mit ihm verbunden.

Je nach gewünschtem Verhalten muss in der Modellentwicklung also entschieden werden, welche Art der Verknüpfung zu verwenden ist.

3.2.3 EXPRESS-G

EXPRESS-G ist eine Möglichkeit zur grafischen Darstellung der in der Programmiersprache EXPRESS modellierten Zusammenhänge. EXPRESS wiederum ist Bestandteil der ISO 10303, die das STEP Format definiert. STEP steht dabei für „Standard for the Exchange of Product model data“ und beschreibt ein Format zum Austausch von Produktdaten. [ISO04]

Ähnlich wie im ERM oder in der UML werden Formen und Linien genutzt, um die Beziehungen von Elementen zueinander zu beschreiben. Bild 3.12 zeigt ein Beispiel für die Verwendung von EXPRESS-G. Ein weiteres Beispiel ist das Datenmodell von NEUHAUSEN, welches in Abschnitt 3.5.3 vorgestellt wird. [ISO04]

Außer zur Analyse des Datenmodells von NEUHAUSEN finden EXPRESS-G in dieser Arbeit keine weitere Anwendung. Interessierte seien daher für weitere Details zur Notation auf entsprechende Literatur und die ISO-Norm verwiesen.

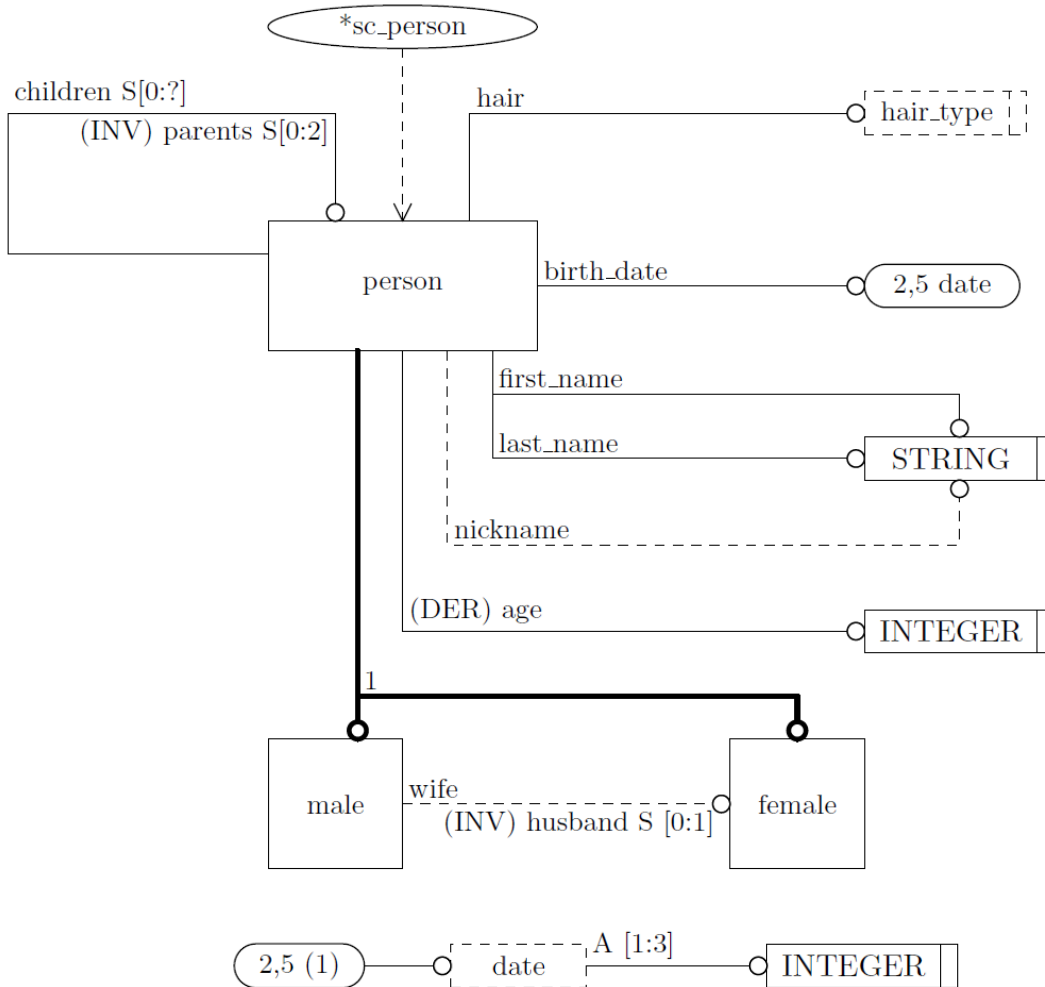


Bild 3.12: Beispiel zur Verwendung von EXPRESS-G aus ISO10303-11 [ISO04]

Example for the use of EXPRESS-G from ISO 10303-33

3.3 Datenbanken und SQL

Databases and SQL

Die in Abschnitt 3.2 vorgestellten Modellierungssprachen dienen dazu, Systemelemente mit ihren Eigenschaften und Zusammenhängen zu modellieren. Die in diesen Modellierungssprachen beschriebenen Modelle lassen sich auf verschiedene Arten umsetzen, beispielsweise als Programm oder in Form einer Datenbank. [MOOS04, S. 1-53]

Datenbanken lassen sich nach der Art, wie ihre Daten abgelegt werden, klassifizieren. In hierarchischen Datenbanken sind die Daten baumartig abgelegt, mit einem Wurzelknoten der sich in Äste aufteilt. In Netzwerkdatenbanken sind sie in einem Netz angeordnet. Relationale Datenbanken speichern Daten in Form von Tabellen. Bei objektorientierten Datenbanken werden Objekte gespeichert, wie sie aus der objektorientierten Programmierung bekannt sind. Universell objektorientierte Datenbanken sind

eine Verknüpfung von relationalen und objektorientierten Datenbanken. XML Datenbanken legen ihre Daten in Form von Dokumenten ab. [MOOS04, S. 125-126]

Für die Umsetzung als Programm wäre eine höhere Programmiersprache wie C++ oder Java denkbar. Da eine Hauptaufgabe jedoch das Erfassen, Verwalten und Ausgeben von Daten ist, sind Datenbanken eine gute Alternative. Hierbei können sowohl bewährte Datenbanksprachen wie SQL, als auch neuere, nicht-relationale (noSQL) Ansätze wie Graphdatenbanken eingesetzt werden.

Ziel dieser Arbeit ist die Entwicklung eines Modells, nicht die Entwicklung einer Software oder Datenbank-Struktur. Allerdings soll das Modell im praktischen Einsatz getestet werden, was eine Implementierung des Modells erforderlich macht. Deshalb werden im Folgenden die Grundlagen zu Datenbanken betrachtet.

Datenbanken werden heute an vielen Stellen eingesetzt. Hauptgründe hierfür sind: [MOOS04, S. 121-125]:

1. Die Daten werden unabhängig von einem Programm gespeichert und verwaltet und können von mehreren Programmen genutzt werden.
2. Die Integrität und Korrektheit der Daten kann durch die Verwendung einer Datenbank unterstützt werden, es existieren Kontrollmechanismen.
3. Die Daten werden an einer Stelle – und nur dort – verwaltet. Somit entstehen keine redundanten Daten.
4. Durch regelmäßige Backups und einen geschützten Zugriff auf die Daten kann deren Sicherheit gewährleistet werden.

Für diese Arbeit werden relationale Datenbanken betrachtet. Da die Beschreibung des Modells umsetzungsneutral erfolgt, ist eine Umsetzung in einer anderen Datenbankart ebenfalls möglich. Moos definiert eine Relation wie folgt:

„Eine Relation ist eine Menge gleichartiger Tupel. Die Gesamtheit der Tupel wird dem Betrachter in Tabellenform dargeboten. Ein Tupel entspricht bei der tabellenartigen Darstellung einer Tabellenzeile. Eine Tabellenspalte entspricht einem Attribut.“ [MOOS04, S. 64]

Damit sich jedes Tupel in der Datenbank eindeutig identifizieren lässt, wird ihm ein sogenannter Primärschlüssel zugewiesen. Dieser ist eindeutig, unveränderlich und wird nicht neu vergeben. Aufgrund dieser Eigenschaften des Primärschlüssels eignet er sich, um Daten explizit miteinander zu verknüpfen. Der Verweis auf einen Primärschlüssel wird Fremdschlüssel genannt. [MOOS04, S. 65; SCHU04, S. 62-63, S. 75-77; KRE10, S. 89-92]

Zur Abfrage von Datenbanken wurde 1970 von CODD die Datenbanksprache Structured Query Language (SQL) entwickelt. Mithilfe dieser Sprache lassen sich die Daten einer Datenbank anzeigen, verändern oder entfernen. [MOOS2004, S. 126; KRE10, S. 27] Neben SQL gibt es noch weitere Datenbanksprachen wie bspw.

In einer neuen Datenbank müssen zunächst Tabellen angelegt werden. Dies geschieht mit dem `CREATE TABLE` Befehl. In Listing 3.1 folgt auf diesen Befehl noch der Zusatz `IF NOT EXISTS`, der vor dem Befehl prüft, ob die Tabelle schon existiert. Der `CREATE TABLE` Befehl wird dann nur ausgeführt, falls die gewünschte Tabelle noch nicht existiert. [KREI10, S. 207]

Auf diesen Befehl folgt der Name der Tabelle. Dieser kann frei gewählt werden. Dem Namen folgt in Klammern die Parameterliste. In einer tabellarischen Ausgabe entsprechen diese den Spalten der Tabelle. Jeder Parameter besteht aus einem Namen gefolgt von dem Typ des Parameters. Die ID wird zusätzlich als Primärschlüssel gekennzeichnet – dies ist an dem Zusatz `PRIMARY KEY` zu erkennen. [KREI10, S. 40]

```
CREATE TABLE IF NOT EXISTS tab_linien (  
    ID_Linie INTEGER PRIMARY KEY,  
    Bezeichnung TEXT,  
    Standort TEXT  
);
```

Listing 3.1: Beispiel für den `CREATE TABLE` Befehl

Example of the `CREATE TABLE` command

Die neu angelegte Tabelle enthält noch keine Daten. Diese lassen sich über den Befehl `INSERT INTO` einfügen (vgl. Listing 3.2). Auf den Befehl folgt der Name der Tabelle, in die Daten eingefügt werden sollen. Dem Tabellennamen folgen die Parameter, die befüllt werden sollen. Hierbei muss nicht zwangsläufig die Reihenfolge eingehalten werden, in der die Parameter bei der Erstellung der Tabelle (vgl. Listing 3.1) aufgeführt wurden. [KREI10, S. 46-47]

```
INSERT INTO tab_linien(ID_Linie, Bezeichnung, Standort)  
VALUES (1, „Band A“, "München");
```

Listing 3.2: Beispiel für den `INSERT INTO` Befehl

Example for the `INSERT INTO` Command

Ist die Datenbank mit Informationen gefüllt, so sollen die Daten angezeigt werden. Dies geschieht mittels `SELECT` Befehl (vgl. Listing 3.3). Auf den Befehl folgen die Spalten, die ausgegeben werden sollen. Da Spaltennamen in verschiedenen Tabellen identisch sein können, wird den Spaltennamen der Tabellennamen vorangestellt. Tabellennamen und Spaltennamen werden durch einen Punkt getrennt. Mittels `AS` Befehl kann für die entstehende Ausgabespalte ein neuer Name festgelegt werden. [KREI10, S. 62-76]

Nachdem alle auszugebenden Spalten definiert sind, werden über den `FROM` Befehl die zu betrachtenden Tabellen beschrieben, hier `tab_linien`. Über den `LEFT JOIN` Befehl wird diese Tabelle mit der Tabelle `Stationen` verknüpft. Als Basis für die Verknüpfung dient die `ID_Linie`. Welche Spalte für die Herstellung der Verknüpfung verwendet werden soll, legt der `USING`-Befehl fest. Damit dieser funktioniert muss diese Spalte in beiden Tabellen die gleiche Bezeichnung haben. Ist die nicht der Fall oder

sollen Regeln zum Verknüpfen definiert werden, kommt der `ON`-Befehl, gefolgt von einer Bedingung für die Verknüpfung zum Einsatz. [KREI10, S. 62-76]

Die zu betrachtende Datengrundlage steht damit fest. Sie kann jedoch noch gefiltert, sortiert und gruppiert werden. Der `WHERE` Befehl legt Bedingungen fest, nach denen die Daten gefiltert werden. Mittels `ORDER BY` kann eine Sortierreihenfolge angegeben werden. `GROUP BY` kann genutzt werden, um mehrere Elemente zusammenzufassen. [KREI10, S. 62-76]

```
SELECT
    tab_linien.Bezeichnung AS Linienbezeichnung,
    tab_stationen.Bezeichnung AS Stationsbezeichnung
FROM tab_linien
LEFT JOIN tab_stationen USING (ID_Linie)
WHERE tab_linien.Standort = "München"
ORDER BY tab_linien.Bezeichnung, tab_stationen.Bezeichnung;
```

Listing 3.3: Beispiel für den `SELECT` Befehl

Example of the `SELECT` Command

Mithilfe des `DELETE` Befehls lassen sich Daten aus der Datenbank löschen (vgl. Listing 3.4). Der Aufbau des `DELETE` Befehls entspricht dem des `SELECT` Befehls. Beim `DELETE` Befehl werden die Daten, auf die die angegebenen Bedingungen zutreffen, gelöscht. [KREI10, S. 48-49]

```
DELETE FROM tab_linien WHERE Standort="München";
```

Listing 3.4: Beispiel für den `DELETE` Befehl

Example of the `DELETE` Command

Um die in den Listings beschriebenen Befehle ausführen zu können, wird ein Datenbankmanagement-System benötigt. Ein Beispiel für ein eingebettetes, relationales Datenbankmanagement-System ist SQLite. Anders als viele andere Datenbankmanagement-Systeme ist es ohne Server lauffähig. Das SQLite Paket benötigt nur wenig Speicherplatz und enthält trotzdem alle benötigten Bibliotheken. Zudem verbraucht SQLite im Betrieb relativ wenige Ressourcen. [KREI10, S.1ff]

Da SQLite als Open Source Software verwendbar ist, kann es beliebig angepasst werden. Zudem ist es als *public domain* veröffentlicht. Programmierer können SQLite also zu jedem Zweck verwenden, selbst für kommerzielle Produkte. [KREI10, S. 1-6]

3.4 Digitale Fabrik

Digital factory

Der übergreifende Begriff für die Vernetzungen von Informationen aus der Produktentwicklung und der Produktion ist die *Digitale Fabrik*. In der VDI 4499 wird der Begriff wie folgt definiert:

„Die Digitale Fabrik ist der Oberbegriff für ein umfassendes Netzwerk von digitalen Modellen, Methoden und Werkzeugen, u. a. der Simulation und 3D-Visualisierung, die durch ein durchgängiges Datenmanagement integriert werden. Ihr Ziel ist die ganzheitliche Planung, Evaluierung und laufende Verbesserung aller wesentlichen Strukturen, Prozesse und Ressourcen der realen Fabrik in Verbindung mit dem Produkt.“ [VDI08, S. 3]

Mit Blick auf die Montageplanung verfolgt die digitale Fabrik das Ziel, die Qualität zu steigern, die Kosten zu verringern und die Planungsdauer zu reduzieren. Hierzu müssen alle beteiligten Schnittstellen von Produktion und Entwicklung einbezogen werden. [KUEH, S. 2; LAND13, S. 112; VDI08, S. 3]

In leicht geänderter Form lässt sich die Definition des VDI bei anderen Autoren finden (vgl. [KUEH06], [LAND13]), sodass davon ausgegangen werden kann, dass diese Definition unumstritten ist. Dennoch legt jeder Autor eigene Schwerpunkte bei seiner Arbeit zur digitalen Fabrik.

LANDHERR unterteilt die digitale Fabrik in 6 Methoden: [LAND13, S. 114-116]

1. Modellierung
2. Simulation
3. Optimierung
4. Visualisierung
5. Dokumentation
6. Kommunikation

Für diese Arbeit ist insbesondere die Methode der Modellierung interessant. In dieser wird nach LANDHERR gedanklich untersucht, welche Abhängigkeiten und Wechselwirkungen sich aus den realen Gegebenheiten ergeben. Ziel der Modellierung ist es, diese im Modell zu erfassen. [LAND13, S. 114]

Basierend auf dem Modell können die weiteren Methoden der digitalen Fabrik nach LANDHERR angewendet werden. So können die Daten und Zusammenhänge als Eingangsgröße für eine Simulation dienen oder für ein Optimierungsverfahren eingesetzt werden. Ebenso können die Daten visualisiert werden. [LAND13, S. 115]

Auch KÜHN beschreibt als Ziel der Digitalen Fabrik, Produkte, Prozesse und Betriebsmitteln in Modellen abzubilden. Basierend auf diesen Modellen lassen sich Verbesserungsmaßnahmen ableiten und deren Effekte im Modell erproben. [KUEH06, S. 2]

Ein Vorteil hierdurch ist, dass sich erste Untersuchungen bereits in frühen Phasen der Produktentwicklung durchführen lassen und dass mögliche Fehler oder Probleme bereits früh identifiziert werden können. Da die Untersuchung rein virtuell erfolgen kann, fallen keine Kosten für Hardwareaufbauten an. [KUEH06, S. 2f]

Die Problemstellung und Ziele dieser Arbeit passen zu denen der digitalen Fabrik. Allen Ansätzen und Untersuchungen der digitalen Fabrik ist gemein, dass sie ein Modell als Basis für die weiteren Methoden benötigen. Um ein Ziel dieser Arbeit – die Abbildung von Zusammenhängen zwischen Produkt, Prozess und Ressource – zu erreichen, müssen im Folgenden Modelle untersucht werden, die genau solche Zusammenhänge abbilden.

3.5 Modelle zur Abbildung von Produkt, Prozess und Ressource

Models of product, process and resource

Basis für die Verarbeitung von Daten im Rahmen der digitalen Fabrik bildet ein Datenmodell. Dieses beschreibt, welche Daten der Modellautor für relevant hält und in welchem Zusammenhang diese nach seiner Auffassung stehen. Diverse Autoren haben sich in den letzten 20 Jahren mit der Modellierung produktionsrelevanter Informationen beschäftigt. Die Basis für viele Modelle bildet der Ansatz nach Steinwasser (vgl. Abschnitt 3.5.1). Jonas greift dessen Grundstruktur auf und leitet daraus sein „Modell zur durchgängigen und rechnergestützten Planung“ ab (vgl. Abschnitt 3.5.2). Die weiteren Modelle der Abschnitte 3.5.3 bis 3.5.6 bauen auf diesen beiden Modellen auf und erweitern sie jeweils um anwendungsfallsspezifische Lösungen.

3.5.1 Modulares Informationsmanagement nach STEINWASSER (1997)

Bereits in seiner Arbeit von 1997 beschreibt STEINWASSER die Relevanz der Zusammenhänge von Produkt, Prozess und Ressource. Entsprechend teilt er sein Modell in diese drei Module ein (vgl. Bild 3.13). Das Modul *Produkt* enthält die Baugruppen und Bauteile sowie deren Geometrie und weitere charakteristische Eigenschaften. Außerdem enthält das Produktmodul prozessrelevante Informationen. Prozesse sind laut STEINWASSER umsetzungsneutrale Beschreibungen von Vorgängen. Sie stellen die Verbindung zwischen Produkten und Ressourcen her. Die Ressourcen beschreiben die Operationen, die zum Herstellen der Produkte verfügbar sind. [STEI97]

Der Vernetzung der drei Module kommt eine wichtige Rolle zu, weil die Planung und somit das Modell dynamisch sind. „Wichtig in diesem Zusammenhang ist jedoch, dass die drei Module Produkt, Ressource und Prozess in Ihrer Gesamtheit ein Informationssystem bilden, welches nicht statisch ist, sondern durch den Planungsvorgang selbst erzeugt, modifiziert und korrigiert wird.“ [STEI97, S. 131]

Auch für STEINWASSER sind der Rückfluss und die enge Kopplung mit der Produktentwicklung wichtig, um somit zur montagegerechten Produktgestaltung beizutragen. (vgl. Abschnitt 2.2.2). [STEI97, S. 118]

Das Ziel des Modells von STEINWASSER ist die funktionale und geometrische Gestaltung von Arbeitsstationen [STEI97, S. 132f]. Produktionsbezogene Restriktionen kön-

nen in seinem Modell nicht abgebildet werden. Einflussfaktoren wie eine große Produkt- und Bauteilvielfalt oder die Komplexität, die durch die Betrachtung eines Montagesystemverbundes entsteht, werden nicht betrachtet.

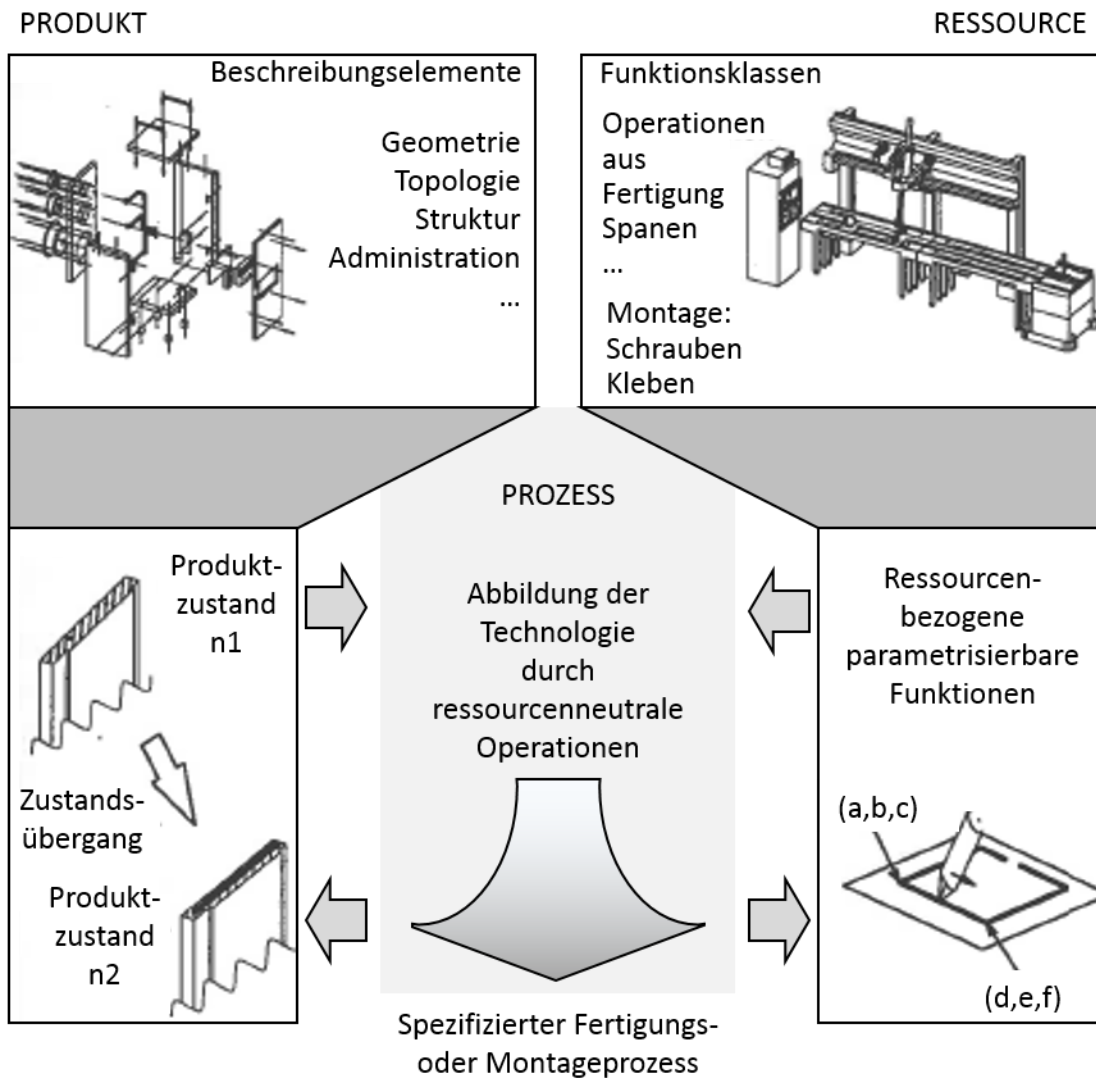


Bild 3.13: Module Produkt, Prozess und Ressource nach STEINWASSER [STEI97, S. 51]

Modules product, process and resource by STEINWASSER

3.5.2 Durchgängige, rechnergestützte Planung nach JONAS (2000)

JONAS greift diese Dreiteilung von STEINWASSER auf. Er erweitert sie um das Modul Verbindungsinformation (vgl. Bild 3.14).

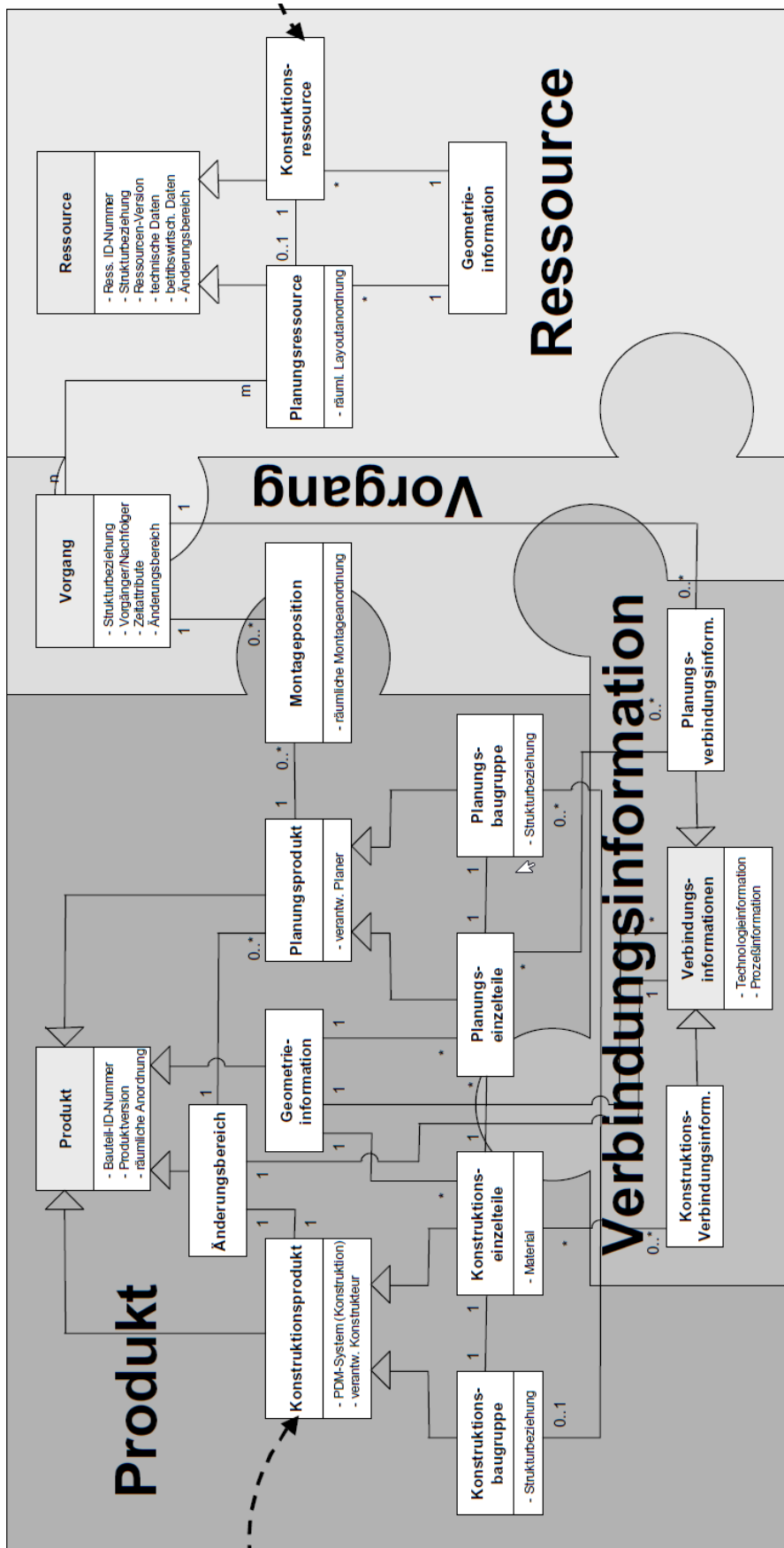


Bild 3.14: Gesamtmodell zur rechnergestützten Planung nach JONAS [JONA00, S. 103]

Complete model of continuous, computer-based planning by JONAS

JONAS betont die Relevanz eines durchgängigen Datenmodells für die Konstruktion und die Planung, um ein ideales Ergebnis zu erzielen. Ein nicht einheitliches und nicht durchgängiges Datenmodell führt zu redundanten Daten und mehreren Datenständen. Einerseits erfolgt die Planung dadurch nicht immer basierend auf dem neuesten Informationsstand, andererseits muss Arbeit in den Abgleich der Daten investiert werden.

Innerhalb der Module beschreibt JONAS mithilfe eines Klassendiagramms die Zusammenhänge. Dabei setzt er überwiegend auf Vererbungen und allgemeine Assoziationen (vgl. Bild 3.14).

Das Produkt bildet die Schnittstelle von Konstruktion und Planung (vgl. Abschnitt 2.1). Von der allgemeinen Klasse Produkt erben das Konstruktionsprodukt und das Planungsprodukt. Mithilfe der Klasse Konstruktionsprodukt wird ein Bezug zur Produktentwicklung hergestellt. Anstatt diese Daten zu kopieren und mit den (redundanten) Kopien zu arbeiten, werden mittels IDs Verknüpfung zwischen den Daten hergestellt. Die Klassen Konstruktionsbaugruppe und Konstruktionseinzelteil erben wiederum von der Klassen Konstruktionsprodukt und helfen dabei, eine Bauteilstruktur aufzubauen. Mittels der Klasse Geometrieinformation werden die Bauteile im Raum angeordnet (vgl. Bild 3.14).

Da die Planung eine andere logische Struktur desselben Produktes hat, führt JONAS das Planungsprodukt ein und detailliert dieses mittels der Klassen Planungseinzelteil und Planungsbaugruppe. Somit lassen sich verschiedene Szenarien planen, ohne dass die tatsächlichen Konstruktionsprodukte verändert werden müssen. Mittels einer Assoziation sind die Planungsbaugruppen und die Konstruktionsbaugruppen verknüpft (vgl. Bild 3.14).

Neben der räumlichen Gestalt legt die Konstruktion fest, wie verschiedene Bauteile miteinander verbunden werden sollen. Beispiele hierfür führt die DIN 8593 an (vgl. Abschnitt 2.2). Diese Verbindungsinformationen müssen für jedes Bauteile beschrieben werden und stehen im Klassendiagramm daher einerseits in einer Beziehung mit den Klassen aus der Konstruktion und andererseits den Klassen aus der Planung. Außerdem sind sie mit der Klasse Vorgang verknüpft.

Die Klasse Vorgang beschreibt die Schritte, die zum Herstellen des Produktes notwendig sind. In ihr sind Strukturbeziehungen und Reihenfolgen definiert. Außerdem werden hierin Zeitattribute angegeben, um eine spätere Abtaktung zu ermöglichen. Um die Vorgänge durchführen zu können, werden Bauteile benötigt. Die hierzu erforderliche Verknüpfung erfolgt über die Klasse Montageposition, die beschreibt, wo sich Planungsprodukte im Verhältnis zum Prozess befinden.

3.5.3 Modulare Produktionssysteme nach NEUHAUSEN (2001)

NEUHAUSEN verfolgt mit seinem Modell das Ziel, ein flexibles Montagesystem zu planen. Flexibel definiert er als „die Senkung der Wechselwirkung zwischen bestimmten

Segmenten, Produktionslinien, Betriebsmitteln und Produktionsprozessen“. [NEUH01, S. 55]

Da sich Änderungen am Produkt meist auf die Produktion auswirken, muss auch in seinem Modell beides abgebildet werden. Weiterhin sieht er das Produktprogramm, also die Produktvarianten und deren zur produzierende Stückzahlen, und die Bewertung, also die Evaluierung mehrerer Alternativen, als relevant für sein Vorhaben. Entsprechend teilt er sein Modell in diese vier Partialmodelle ein: Produkt, Produktprogramm, Produktion und Bewertung (vgl. Bild 3.15). [NEUH01, S. 55-58]

Auffällig bei NEUHAUSEN im Vergleich zu den anderen genannten Autoren ist, dass einige Klassen in mehreren Partialmodellen auftauchen und somit Schnittmengen zwischen den Klassen entstehen (vgl. [NEUH01, S. 58-59]).

Das Partialmodell Produktprogramm besteht aus Produkttypen und deren Produktmerkmalen. Die Produkttypen beschreiben verschiedene Varianten eines Produktes, die sich anhand der Produktmerkmale voneinander unterscheiden lassen. Im Partialmodell sind alle möglichen Kombinationen der Ausprägungen von Produkteigenschaften und somit das gesamte Produktportfolio abgebildet. Hinzu kommt die Änderungsfrequenz von Produktmerkmalen – ein wichtiges Kriterium für Entscheidungen über die Flexibilität (vgl. Bild 3.15). [NEUH01, S. 56-64]

Im Partialmodell Produkt tauchen die Produktmerkmale ebenfalls auf. Hier dienen sie zur Definition von Produktkomponenten, also den Bauteilen des Produktes. Dieses Partialmodell beschreibt also die physische Gestalt des Produktes, die sich aus den im Produktprogramm definierten Anforderungen ergibt. Die Klasse *teilweise zusammengesetztes Produkt* besteht aus diesen Produktkomponenten und beschreibt das Produkt entlang des Produktionsprozesses bis zum fertigen Produkt. [NEUH01, S. 56-64]

Die Klasse *Produktionsprozess*, angesiedelt in den Partialmodellen *Produktion* und *Bewertung*, verarbeitet die *Produktkomponenten* und das *teilweise zusammengesetzte Produkt*, um somit ein neues *teilweise zusammengesetztes Produkt* zu schaffen. Der Produktionsprozess selbst unterliegt einer Reihenfolge, die über mögliche Vorgänger und Nachfolger festgelegt wird. Aus dieser Information lässt sich ein Vorranggraph erstellen. Weiterhin wird der Produktionsprozess definiert durch die Klasse *Prozessmerkmal*. Der Ansatz hierbei ist identisch zum Produkt: Ein Produkt wird über seine Produktmerkmale definiert und differenziert, ein Prozess anhand seiner Prozessmerkmale. „Diese Prozessmerkmale geben verbal den produktbezogenen Unterschied zwischen den Produktionsprozessen an.“ [NEUH01, S. 57-69]

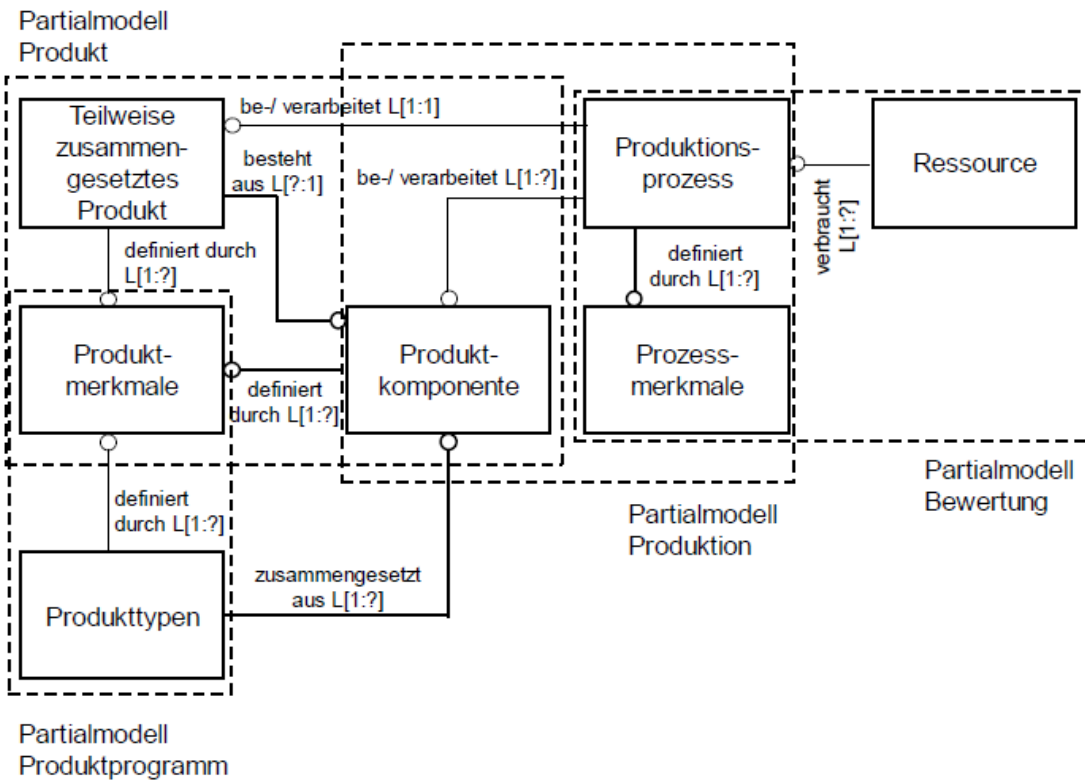


Bild 3.15: Partialmodelle nach NEUHAUSEN [NEUH01, S. 59]

Partial models by NEUHAUSEN

Das Partialmodell Bewertung überschneidet sich bei den Klassen Produktionsprozess und Prozessmerkmale mit dem Partialmodell Produktion. Für die Bewertung kommt die Klasse *Ressource* hinzu. Diese ermittelt den Verbrauch von Ressourcen welcher Natur auch immer und ermöglicht somit eine quantitative Bewertung verschiedener Alternativen. [NEUH01, S. 56-70]

3.5.4 Feature-basierte Prozesskettenplanung nach FRANKE (2003)

Auch FRANKE verwendet als Basis für ihre Überlegungen die Einteilung in Produkt, Prozess und Ressource von STEINWASSER. [FRAN03, S. 59–60]

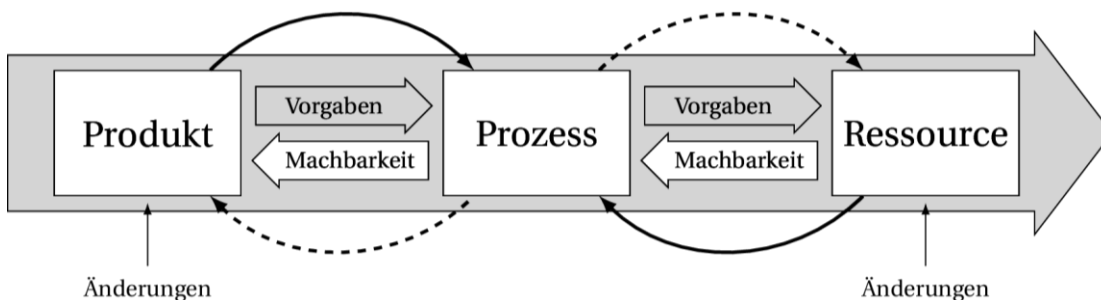


Bild 3.16: Zusammenhänge von Produkt, Prozess und Ressource nach FRANKE [FRAN03, S. 60]

Relationship of product, process and resource by FRANKE

In ihrem Modell legt FRANKE den Fokus auf die Gestaltung des Prozessplans. Dieser besteht aus Objekten der Klasse *Element im Prozessplan* [FRAN03, S. 62]. Für jedes dieser Elemente kann festgelegt werden, ob es sich um einen wertschöpfenden, einen nicht wertschöpfenden oder einen Prüfprozess handelt. Zudem lassen sich die Elemente in Fertigung und Montage gliedern. Die Montage wird wieder spezifiziert durch den *Montageprozess*, die *Ressource* und die *Baugruppe* (vgl. Bild 3.17). Die Verbindung zwischen den Einzelteilen einer Baugruppe und dem Montageprozess erfolgt über sogenannte *Assembly Features*. [FRAN03, S. 60–61]

Sie verwendet dabei die Definition des Feature Begriffs aus VDI 2218: „Features sind informationstechnische Elemente, die Bereiche von besonderem (technischen) Interesse von einzelnen oder mehreren Produkten darstellen.“ [FRAN03, S. 74; VDI03]

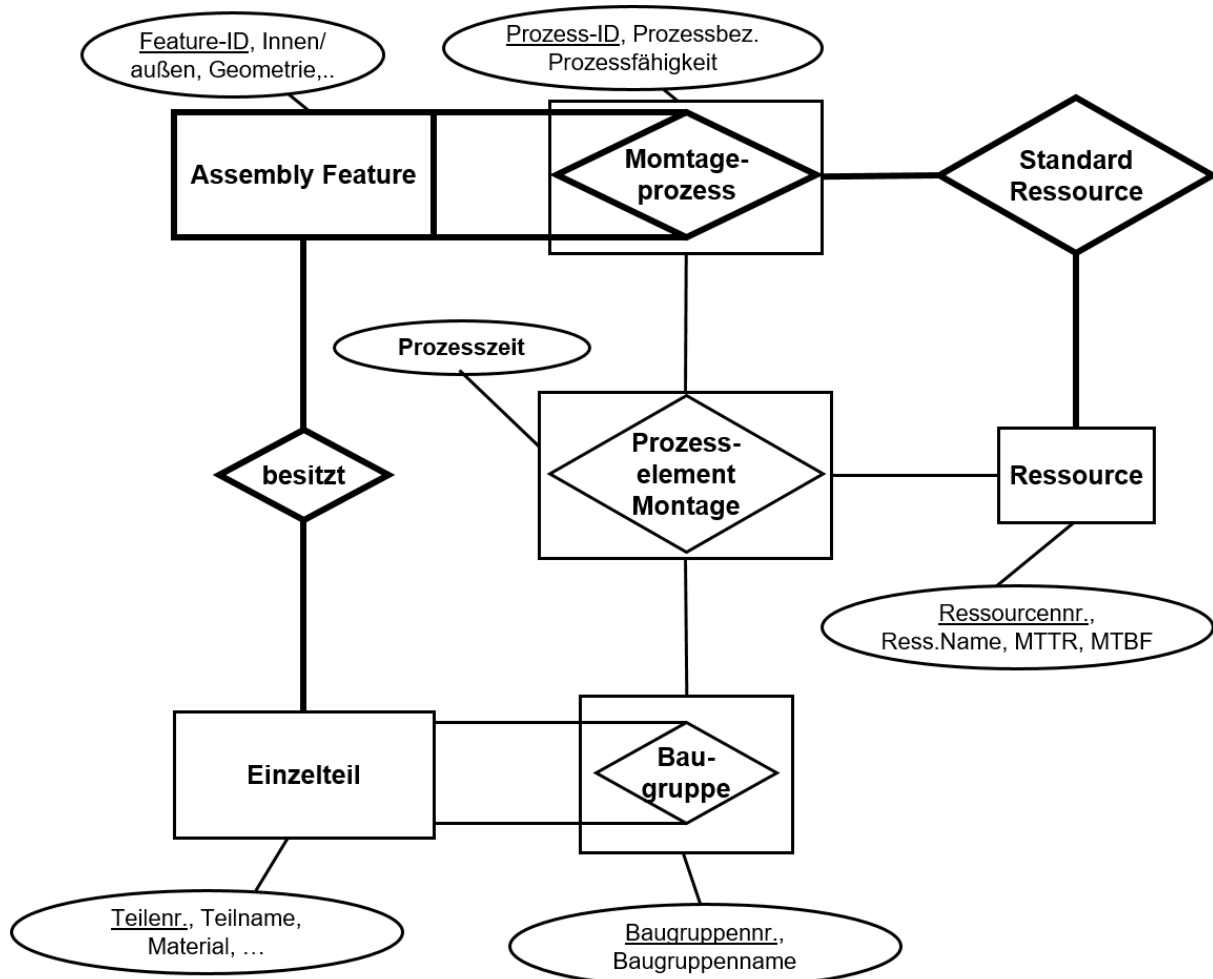


Bild 3.17: Datenstruktur der Feature-basierten Prozesskettenplanung nach FRANKE [FRAN03, S. 63]

Data structure for the feature based planning of process chains by FRANKE

Mithilfe der *Assembly Features* lässt sich prüfen, ob bspw. ein bestimmter Fügeprozess zum Fügen zweier Bauteile herangezogen werden kann. [FRAN03, S. 52–63]

3.5.5 Die virtuelle, offene Fabrik nach KLAUKE (2002)

KLAUKE lehnt ihre Basisklassen an den Ansatz von STEINWASSER an und teilt ihr Modell in drei Basisklassen Produkt, Prozess und Ressource. Jedoch bleibt sie bei der inhaltlichen Beschreibung dieser Klassen auf einem hohen Abstraktionsniveau und ordnet den Basisklassen Schlagworte zu.

Der von KLAUKE beschriebene Ansatz umfasst jedoch weit mehr als die bloße Verknüpfung von Produkt, Prozess und Ressource.



Bild 3.18: Grobstruktur der „Offenen Virtuellen Fabrik“ nach KLAUKE [KLAU02, S. 55]

Basic structure of the “Open Virtual Factory” by KLAUKE

In Ihrem Modell bildet sie ebenfalls Verwaltungsinformationen wie Änderungsprotokolle und Benutzerrechte ab. Hinzu kommen methodische Aspekte wie Ergonomieanalysen, Simulationen und allgemeine Planungsdaten wie Kalkulationen und Schichtmodelle (vgl. Bild 3.18). [KLAU02, S. 56-57]

Ein Klassendiagramm des Modells ist in Ihrer Arbeit nicht enthalten und kann somit nicht für eine nähere Analyse herangezogen werden. [KLAU02, S. 139]

3.5.6 Das Plattformkonzept für die Planung von Montagesystemen nach BRUNNER (2015)

Auch BRUNNER unterteilt sein Modell in die Partialmodelle *Produkt*, *Prozess* und *Ressource* (vgl. Bild 3.19). Ähnlich wie JONAS fügt er die Verbindungsinformation – er bezeichnet diese als *Verknüpfungsdetails* – als ein weiteres Partialmodell hinzu. [BRUN15, S. 76-79]

Die Struktur der Partialmodelle *Produkt*, *Prozess* und *Ressource* ähnelt der der vorher genannten Autoren. Er führt *Bauteile* und *Baugruppen* ein, verwendet die Klassen *Prozess* und *Montageschritt* zur Festlegung der Abläufe und eine Klasse *Betriebsmittel* zur Beschreibung der Ressourcenseite (vgl. Bild 3.19). Ähnlich wie JONAS arbeitet er mit einigen Vererbungen. [BRUN15, S. 71-79]

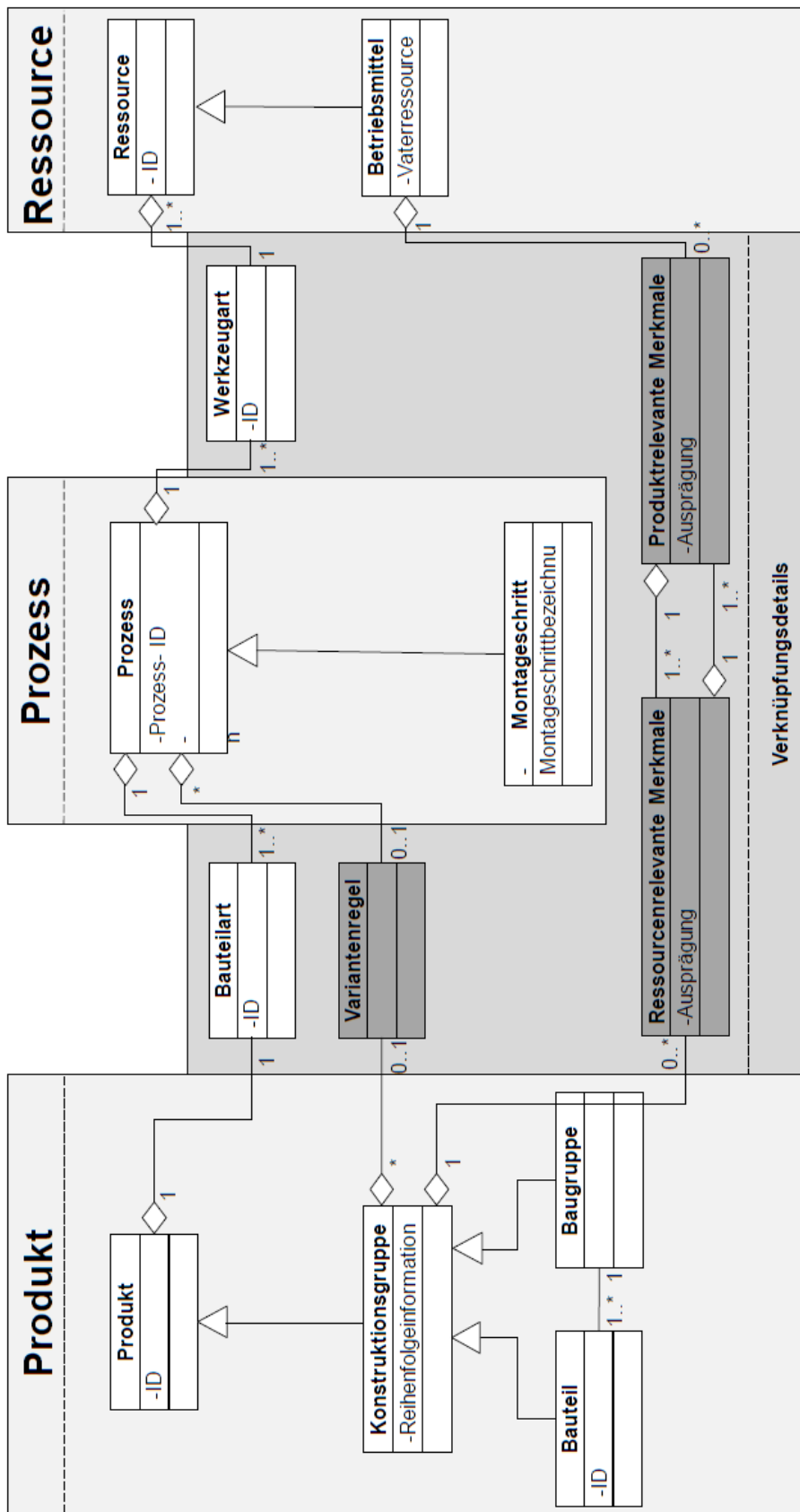


Bild 3.19: Datenmodell für das Plattformkonzept für die Planung von Montagesystemen nach BRUNNER [BRUN15, S. 78]

Data model for the platform concept for assembly planning by BRUNNER

Den Verknüpfungsansatz übernimmt er ebenfalls. Sein Ansatz basiert auf der Ableitung von einem generischen Arbeitsplan in einen produktspezifischen Arbeitsplan. Der produktspezifische Arbeitsplan wird wiederum in einen werksbezogenen Arbeitsplan überführt. [BRUN15, S. 80]

Hervorzuheben ist sein Ansatz der Verbindungsinformationen. *Produkt* und *Prozess* werden über eine neue Klasse *Bauteilart* verknüpft. Da die *Konstruktionsgruppe* die Attribute der Klasse *Produkt* erbt und von dieser wiederum die Klassen *Bauteil* und *Baugruppe* erben, lassen sich alle vier Klassen mit einer *Bauteilart* versehen und hierüber mit *Prozessen* verknüpfen. [BRUN15, S. 76-79]

Zusätzlich führt BRUNNER als Verknüpfungsklasse die *Variantenregel* ein. Über diese können bestimmte Varianten gezielt mit Prozessen verknüpft werden. Ist ein bestimmter Prozessschritt nur für einen Teil der Produkte erforderlich, so kann dies mittels der Variantenregel festgelegt werden. [BRUNN15, S. 76-79]

Analog zur *Bauteilart*, die *Produkt* und *Prozess* verknüpft, werden mithilfe der *Werkzeugart* *Prozess* und *Ressource* verknüpft. Da die Klasse *Betriebsmittel* von der Klasse *Ressource* erbt, lässt sich für jedes *Betriebsmittel* eine *Werkzeugart* festlegen und somit eine Verknüpfung zu möglichen Prozessen herstellen. [BRUN15, S. 76-79]

FRANKE führt zur Verknüpfung von *Prozess* und *Bauteilen* die *Assembly Features* ein (vgl. Abschnitt 3.5.4). BRUNNER führt mit seinen beiden Klassen *ressourcenrelevante Merkmale* und *produktrelevante Merkmale* ähnlich bezeichnete Klassen ein, verwendet diese jedoch, um *Konstruktionsgruppen* und *Betriebsmittel* miteinander zu verknüpfen. [BRUN15, S. 77-79]

Die *Konstruktionsgruppe* bzw. eine ihrer Kinderklassen verfügen demnach über *ressourcenrelevante Merkmale*. BRUNNER nennt hier das Beispiel von Bohrungen im Bauteil, die zu dessen Indexierung, also der festen Positionierung in der Montageanlage, verwendet werden können. Gibt es hierzu ein passendes produktrelevantes Merkmal, so lässt sich die Montageoperation durchführen. Im Beispiel der Indexierung wären das beispielsweise Zentrierstifte in einer Montageanlage, die in die Bohrung eintauchen. [BRUN15, S. 62, 77-78]

Die *produktrelevanten Merkmale* der Montageressourcen beschreiben dabei die in der Montageanlage installierte Flexibilität. Wann immer sich ein ressourcenrelevantes Merkmal einer Konstruktionsgruppe innerhalb dieser Flexibilität befindet, ist die Herstellbarkeit gegeben. [BRUN15, S. 62-63]

BRUNNER verwendet die Features, um Aussagen bzgl. der Standardisierungseignung zu treffen. Ein Ansatz, die Informationen für eine Aussage zur Montierbarkeit zu nutzen existiert nicht.

3.6 Zusammenfassung und Herleitung des Handlungsbedarfs

Summary and deduction of implications

Um komplexe Sachverhalte – wie die Ausplanung eines Montagesystemverbundes bei hoher Variantenvielfalt – einer Analyse zugänglich zu machen, werden Modelle verwendet. Modelle zeichnen sich durch drei Merkmale aus: Das Abbildungsmerkmal, das Verkürzungsmerkmal und das pragmatische Merkmal. Darüber hinaus sind bei der Erstellung eines Modells gewisse Grundsätze zu beachten.

Es gibt verschiedene Arten von Modellen – für das in dieser Arbeit beschriebene Problem eignet sich vor allem ein Prognose- oder Erklärungsmodell. Für die Erstellung von Modellen gibt es Prozesse, die iterativ durchlaufen werden können (vgl. Bild 3.3 und Bild 3.4).

Zur eindeutigen Beschreibung und Dokumentation von Modellen gibt es verschiedene Modellierungssprachen. Die gängigsten sind das Entity-Relationship-Model und UML-Klassendiagramm.

Werden Daten der Produktion in einem Modell abgebildet, gehört die betrachtete Thematik in das Forschungsfeld der digitalen Fabrik. Die Literatur hierzu bleibt oft auf einer generischen Beschreibungsebene, ohne konkrete Details zu hinterlegten Modellen offenzulegen. Einige Autoren beschreiben Modelle zur Verknüpfung von Produkt, Prozess und Ressource, wenn auch mit abweichenden Zielsetzungen. Der Abgleich der Modelle aus Abschnitt 3.5 mit den in Abschnitt 2.5 beschriebenen Anforderungen ist in Tabelle 3.1 dargestellt.

STEINWASSER beschreibt die logische Verknüpfung zwischen Produkt, Prozess und Ressource. Jedoch bleibt er auf dieser Ebene und geht nur beim Produkt geringfügig auf die darunterliegende Objektebene ein. Ein Baukastenansatz oder eine regelbasierte Variantenlogik, die gemäß Abschnitt 2.5 erforderlich sind, finden sich bei ihm nicht, ebenso wenig wie die Betrachtung eines Verbundes von Montagesystemen oder deren Dynamik. Dennoch bilden seine Ansätze die Basis für alle weiteren Modelle in diesem Bereich.

JONAS entwickelt in seiner Arbeit ein vollständiges Modell und zeigt die Zusammenhänge aller Klassen. Jedoch werden in seinem Modell viele Verknüpfungen zwischen ähnlichen Elementen angelegt. Idealerweise erfolgt dies automatisch, basierend auf Regeln. Da keine Regel zur automatischen Generierung dieser Verknüpfungen gezeigt wird, muss davon ausgegangen werden, dass diese manuell angelegt werden müssen. Bei den Varianten spricht JONAS als einziger Autor von einer einfachen Systematik die auf einer Zahlenlogik basiert. Er generiert für jede Bauteilvariante eine eindeutige Nummer, die die Bauteilvariante mit einbezieht. Jedoch kommen hierbei ebenfalls keine Regeln zum Tragen, was wiederum dazu führt, dass die Verknüpfungen zu diesen Nummern aufwendig und einzeln erfolgen müssen. Anforderungen von Produkten und Fähigkeiten von Ressourcen (vgl. Abschnitt 2.3.2) gibt es in dieser

Form nicht, jedoch stellt er mit den Verbindungsinformationen ein ähnliches Konstrukt vor. Konkrete Ausprägungen werden hierin jedoch nicht abgelegt und ein Abgleich erfolgt ebenfalls nicht.

Tabelle 3.1: Abgleich der untersuchten Modelle mit den Anforderungen aus Abschnitt 2.5
Matching analyzed models with requirements from section 2.5

10. Fehleridentifikation											
9. Analyse											
8. Kommunikation											
7. Dynamik											
6. Montagesystemverbund											
5. Anforderungen/Fähigkeiten											
4. Variantenvielfalt / Regelbasiert											
3. Baukastenansatz											
2. Logik über Verknüpfungen / Redundanzfrei											
1. Verknüpfung P-P-R											
STEINWASSER		◐	◑	○	○	◑	○	○	○	○	○
JONAS		●	◑	○	◑	◑	○	○	○	○	○
NEUHAUSEN		●	◑	○	○	◑	●	○	○	◑	○
FRANKE		●	◑	○	○	◑	○	○	○	◑	○
BRUNNER		●	◑	●	◑	◑	◑	○	○	◑	○

Auch NEUHAUSEN gliedert sein Modell in die Partialmodelle nach dem Vorbild von STEINWASSER. Die Verknüpfungen sind so angelegt, dass weitestgehend ohne redundante Daten gearbeitet werden kann. Merkmale werden zwar im Modell erfasst, allerdings an keiner Stelle für einen automatischen Abgleich oder eine Bewertung herangezogen. Der praktische Einsatz des Modells in der Montageplanung wird nur im Ansatz vorgestellt. Konkrete Ergebnisse des Modells und deren Implikationen für die Planung werden nicht vorgestellt.

FRANKE legt mit ihrem Ansatz der *Assembly Features* einen Schwerpunkt auf die Merkmale von Produkten und deren Auswirkungen auf die Montageplanung. Jedoch werden auch bei ihr die Features nicht für einen automatisierten Vergleich herangezogen, auch wenn dies bei vielen der von ihr genannten Features möglich wäre. Vielmehr verweist sie an dieser Stelle auf den Montageplaner. Dennoch zeigt ihr Modell, dass für jedes Einzelteil bzw. dessen Varianten Features definiert werden müssen und dass die Relevanz von Features stark vom gewählten Prozess abhängt. Weiterhin greift sie den Ansatz von Standard-Ressourcen auf, die nur einmalig modelliert werden müssen.

BRUNNER schließt sich bei der Unterteilung in Partialmodelle an die Reihe der vorherigen Autoren an. Er baut auf dem Ansatz von JONAS auf und erweitert diesen um den Baukastenansatz.

Im Ansatz von BRUNNER werden zunächst generische Arbeitspläne erstellt. Von diesen werden anschließend produktspezifische Ableitungen erstellt. Diese von BRUNNER beschriebenen Ableitungen sind Kopien, die die Beziehung zu den vorherigen Instanzen verlieren. Änderungen am generischen Arbeitsplan werden nicht selbstständig in den produktspezifischen Arbeitsplan übernommen. Die Anforderungen und Fähigkeiten tauchen hier paarweise auf, in Form von ressourcenrelevanten (Produkt-)Merkmalen und produktrelevanten (Ressourcen-)Merkmalen. Mithilfe der Variantenregeln ließe sich die Dynamik im Werkeverbund abbilden, jedoch wäre dies mit einem hohen manuellen Aufwand verbunden, da für jede Änderung des Zustands die Variantenzuweisung manuell angepasst werden muss.

Die Unterstützung der Kommunikation ist in keinem der untersuchten Modelle vorgesehen. Ebenso wird die Dynamik kaum aufgegriffen. Auch die Erstellung automatisierter Auswertungen, die einen direkten Rückschluss auf Entscheidungen in der Montageplanung zulassen, ist nicht Bestandteil der vorhandenen Modelle.

Die Analyse und eine Ableitung von Handlungsempfehlungen für die Montageplanung werden in Ansätzen erwähnt. BRUNNER wird hier am konkretesten als er sein Anwendungsbeispiel vorstellt [BRUN15, S. 89-109]. Eine allgemeingültige Empfehlung, welches Ergebnis des Modells zu welcher Entscheidung im Montageplanungsprozess genutzt werden kann, fehlt jedoch.

Die Analyse der Modelle zeigt, dass keines der Modelle vollständig geeignet ist, um die Anforderungen des praktischen Handlungsbedarfs zu erfüllen. Allerdings zeigt sich auch, dass in vielen Modellen Ansätze vorhanden sind, die für die Beantwortung der Forschungsfrage zielführend sein können. Diese Ansätze sollen aufgegriffen, angepasst und erweitert werden, um die Ziele aus Abschnitt 2.5 zu erfüllen.

Um das Modell, das in dieser Arbeit entwickelt wird, in der Praxis einsetzen zu können, muss es mithilfe von Programmiersprachen in Prototypen implementiert werden.

Hierzu gibt es eine große Zahl an Möglichkeiten. Da in dieser Arbeit die Zusammenhänge innerhalb einer großen Datenmenge gesucht werden, eignen sich Datenbanken besonders, um das Modell abzubilden.

Im Fokus dieser Arbeit soll die Anwendung der Planungsmethodik unter Zuhilfenahme des vernetzten Modells stehen und nicht die Implementierung einer Software. Daher wurde als Datenbanksystem eine möglichst einfache Lösung ausgewählt: SQLite. Da diese Software zudem *public domain* ist, kann die Software frei genutzt werden. Somit lassen sich die Ergebnisse dieser Arbeit vollständig nachvollziehen.

4 Entwicklung des Modells zur Integrationsplanung

Development of the model for integration planning

In Kapitel 4 wird zunächst die Methode zur Integrationsplanung vorgestellt (Abschnitt 4.1). Diese basiert auf bewährten Planungsmethoden aus Abschnitt 2.2.2, wird jedoch für den Zweck des Modells vereinfacht und zusätzlich um die drei grundlegenden Stellhebel erweitert, die einem Montageplaner bei der Integrationsplanung zur Verfügung stehen.

Anschließend wird basierend auf den Modellen aus Abschnitt 3.5 ein Basis-Datenmodell entwickelt, das Linien, Stationen, Produkte und Bauteilvarianten in einen Zusammenhang bringt (Abschnitt 4.2). Anschließend wird dieses Basis-Datenmodell um Restriktionen erweitert, die in einem Montagesystemverbundes auftreten (Abschnitt 4.3).

Zum Abschluss des Kapitels wird aufgezeigt, welche Informationen sich mithilfe des Modells generieren lassen und wie diese im Rahmen einer Integrationsplanung verwendet werden können (Abschnitt 4.4). Dies bildet die Basis für die Implementierung und praktische Anwendung des Modells (vgl. Kapitel 5).

4.1 Methode der Integrationsplanung

Method of integration-planning

Besonders geeignet für die Aufgabenstellung dieser Arbeit ist die Methode der fähigkeitsbasierten Integrationplanung nach MÜLLER. Wie in Abschnitt 2.2.2 beschrieben, ist der Auslöser für den betrachteten Montageplanungsprozess immer eine Veränderung an einem Bauteil oder Produkt ist (vgl. Schritt 1 in Bild 4.1). Am Ende dieses Prozesses steht fest, ob Baubarkeitskonflikte vorhanden sind und somit die Montierbarkeit der Produkte nicht mehr gegeben ist. Ist dies der Fall muss die Montageplanung aktiv werden. Zusätzlich muss bedacht werden, dass im hier betrachteten Montagesystemverbund dieselben Prozesse in mehreren Stationen in verschiedenen Linien vorkommen können.

Integrationsplanung bedeutet, einen (durch Änderungen) mit Konflikten behafteten Montagesystemverbund in einen konfliktfreien Zustand zu überführen.

Dazu stehen drei Stellhebel zur Verfügung:

1. Anpassung der Linienbelegung (Schritt 5 in Bild 4.1)
2. Anpassung der Ressource (Schritt 6 in Bild 4.1)
3. Produktbeeinflussung (Schritt 7 in Bild 4.1)

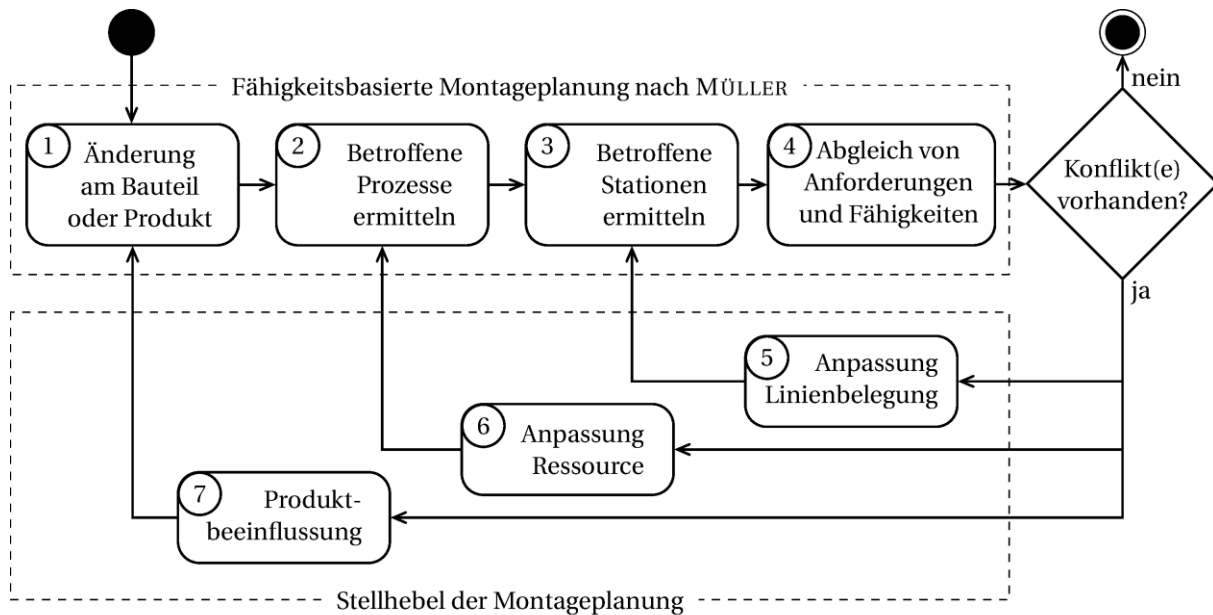


Bild 4.1: Fähigkeitsbasierte Montageplanung erweitert um Stellhebel der Montageplanung [MUEL15, S. 554]

Capability-based assembly planning extended by options of action

Durch die Anpassung der Linienbelegung (vgl. Schritt 5 in Bild 4.1) werden Anforderungen zu anderen Ressourcen und somit anderen Fähigkeiten verschoben. Dies kann dazu führen, dass die Schwere oder Anzahl der Konflikte verringert oder vollständig eliminiert wird. Dabei müssen nicht zwingend die neuen Produkte von der Linienbelegung betroffen sein. Auch durch die Verschiebung bisher hergestellter Produkte können Kapazitäten frei werden, die durch das neue oder geänderte Produkt genutzt werden können. Natürlich kann die Anpassung der Linienbelegung auch zu neuen Konflikten an anderer Stelle führen. Für die Implementierung des Modells (vgl. Kap. 5) ist es daher wichtig, dass stets die Summe aller Anforderungen mit den Fähigkeiten verglichen werden und nicht nur die geänderten Anforderungen.

Für die optimale Belegung der Linie mit Produkten sind nicht nur Belange der Montageplanung relevant. Ebenso spielen Logistik- und Beschaffungskosten sowie Lohnkosten eine wichtige Rolle. Auch strategische Überlegungen und Vorhalte für Volumenflexibilität haben großen Einfluss auf die Belegung der Montagesysteme mit Produkten. Somit sind das Modell und dessen Vorschläge nur ein Aspekt, der zusammen mit den weiteren genannten betrachtet und gegeneinander abgewogen werden muss.

Eine Änderung der Linienbelegung kann schnell zu vielen Konflikten führen – entweder weil die Kapazität überschritten wird oder weil die zusätzlichen Anforderungen nicht durch die Fähigkeiten erfüllt werden können. Es kann jedoch auch vorkommen, dass das Hinzufügen eines neuen Produktes keine Auswirkung auf die Linie hat, da alle Bauteilvarianten dieses neuen Produktes schon in anderen Produkten vorkommen und somit bereits im Montagesystem integriert sind. Außerdem wäre es möglich,

dass die installierte Kapazität ausreicht, um die zusätzlichen Bauteilvarianten aufzunehmen und die neuen Anforderungen durch die Fähigkeiten abgedeckt sind.

Gerade weil die Linienbelegung derart weitreichende Auswirkungen hat, sollte ihr besondere Beachtung geschenkt werden. Um flexibel auf Nachfrageschwankungen reagieren zu können, sollten die Linien für möglichst viele Produkte befähigt werden. Auf der anderen Seite bringt dies erhebliche Mehrkosten mit sich. Diese sind Verschwendung, wenn die vorgehaltene Kapazität nicht benötigt wird.

Die zweite Option ist die Anpassung der Ressource (vgl. Schritt 6 in Bild 4.1). Dadurch lassen sich die Fähigkeiten einer oder mehrerer Stationen anpassen, was wiederum dazu führt, dass Konflikte aufgehoben werden. Dieser Schritt ist in der Regel mit gewissen Aufwänden – bspw. Investitionskosten oder Produktionsunterbrechungen – verbunden.

Als dritte Möglichkeit kann das Produkt beeinflusst werden (vgl. Schritt 7 in Bild 4.1). Im Konzept der Anforderungen und Fähigkeiten bedeutet dies, dass die Anforderungen so angepasst werden, dass sie durch die Fähigkeiten abgedeckt sind. Somit kann die Anzahl der Konflikte ebenfalls reduziert werden.

Die Reihenfolge der Stellhebel der Montageplanung stellt keine Priorisierung dar. Es sollten stets alle Möglichkeiten und ihre Auswirkungen analysiert und in Betracht gezogen werden. Die Entscheidungsfindung basiert einerseits auf den entstehenden Aufwänden, andererseits aber auch auf der (Produktions-)Strategie des Unternehmens.

Im Planungsalltag muss in der Regel nicht eine einzige Änderung bewertet werden. Vielmehr ergeben sich zahlreiche Änderungen aus denen dann zahlreiche Konflikte resultieren können. Zur Lösung dieser Konflikte können die drei genannten Stellhebel kombiniert werden. Die sinnvolle Kombination aus Anpassung der Linienbelegung, Anpassung der Ressourcen und Produktbeeinflussung stellt die beste Möglichkeit zur Lösung aller Konflikte dar. Auf diese Art kann das Montagesystem in einen konfliktfreien Zustand überführt werden und die Montierbarkeit sichergestellt werden. Bei diesem Prozess der Integrationsplanung soll das im Folgenden entwickelte Modell den Montageplaner unterstützen.

4.2 Entwicklung Basis-Datenmodell

Development of basic data model

Um Konflikte finden zu können, müssen zunächst Produkt, Prozess und Ressource in einen Zusammenhang gebracht werden. Dazu werden die in Abschnitt 3.5 analysierten Modelle erweitert, um die in Abschnitt 3.6 beschriebenen Defizite zu beseitigen. Ziel dieses neu entwickelten Basis-Datenmodells ist es, die Objekte durch Verweise

miteinander zu verknüpfen. Analog STEINWASSER und BRUNNER wird das Modell objektorientiert ausgerichtet. Zur Beschreibung des Modells wird – wie von JONAS und BRUNNER – als Modellierungssprache das UML Klassendiagramm verwendet.

Zur besseren Übersicht über ihre Modelle und zur leichteren Erklärung teilen einige Autoren aus Abschnitt 3.5 ihre Modelle in kleinere Einheiten. STEINWASSER spricht von Modulen, KLAUKE von Basisklassen, JONAS und BRUNNER von Hauptklassen. Dieses Vorgehen wird übernommen. Da sich mittels der Unterteilung teilselbstständige, logisch getrennte Modelle ergeben, wird analog NEUHAUSEN und BRUNNER von Partialmodellen gesprochen. Zunächst wird das Partialmodell Produkt (Abschnitt 4.2.1) entwickelt, welches die produktseitigen Zusammenhänge von Bauteilarten und -varianten sowie Produktvarianten darstellt. Darauf folgt die Entwicklung des Partialmodelles Prozess (Abschnitt 4.2.2). Prozesse werden hier als eine umsetzungsneutrale Beschreibung der Operationen, die zur Montage des Produktes notwendig sind, verstanden. Vervollständigt wird das Basis-Datenmodell durch die Modellierung der Ressourcen in einem dritten Partialmodell (Abschnitt 4.2.3). Dieses beschreibt die Zusammenhänge in der Montage selbst und stellt einerseits eine Verknüpfung zu den Montageprozessen und andererseits zu den Produkten her. Folglich sind die Partialmodelle untereinander verknüpft und ergeben in Summe das Basis-Datenmodell.

JONAS baut in seinem Modell auf allgemeine Assoziationen – ohne diese näher zu erläutern – und auf Vererbungen. BRUNNER detailliert die Beziehungen zu Aggregationen. Im Grundsatz wird dieses Vorgehen auch in dieser Arbeit angewendet. Alle Informationen sollen regelbasiert oder als Aggregation verknüpft werden. Die Daten werden nicht kopiert. Vorteile dieses Vorgehens werden in Abschnitt 4.2.4 erläutert. Konstruktionsinformationen werden in der fähigkeitsbasierten Montageplanung vernachlässigt.

Für den in dieser Arbeit benötigten Abgleich von Anforderungen und Fähigkeiten bietet der Ansatz der produktrelevanten und ressourcenrelevanten Merkmale von BRUNNER das größte Potenzial. FRANKES Ansatz, die Features mit dem Prozess zu verknüpfen, soll ebenfalls übernommen werden.

NEUHAUSEN hat als einziger Autor eine Klasse für die Produktionslinie. Da in dieser Arbeit mehrere Montagesysteme betrachtet werden sollen, müssen die Produktionslinien ebenfalls modelliert werden können. Die Abbildung der Dynamik der Zuweisung von Produkten zu Produktionslinien ist bei BRUNNER über die Ableitung auf werksspezifische Prozesse mithilfe der Variantenregeln möglich. Jedoch muss für jeden neuen Zustand eine neue Ableitung erstellt werden.

Der Aspekt der Erleichterung der Kommunikation wird in keinem der beschriebenen Modelle aufgegriffen. Direkte Handlungsempfehlungen, die auf Auswertungen und Ergebnisse des Modells basieren fehlen ebenfalls weiterstehend in allen untersuchten Ansätzen (vgl. Abschnitt 2.5).

4.2.1 Partialmodell Produkt

Partial model of the product

Wie in Abschnitt 2.3 beschrieben stellt die Vielzahl an zu berücksichtigenden Produktvarianten eine Herausforderung für die Montageplanung dar. Daher wird eine Klasse *Produktvariante* angelegt.

Jede real existierende Produktvariante wird später als Instanz der Klasse *Produktvariante* erzeugt. Dies ist wichtig, damit nur die tatsächlichen Produktvarianten im Modell berücksichtigt werden und nicht alle theoretisch möglichen. Beispielsweise gibt es bei vielen Autoherstellern Motoren im Längs- und im Quereinbau. Sechszylindermotoren werden jedoch aus Platzgründen meist nur im Längseinbau angeboten. Bauteile für Quermotoren können bei der Betrachtung von Sechszylindermotoren demnach vernachlässigt werden.

Eine Produktvariante kann eine beliebige Bezeichnung haben. Diese Bezeichnung soll eine für den Menschen gut deutbare Beschreibung der Produktvariante sein. Darüber hinaus verfügt sie über einen Variantencode. Über den Variantencode wird die Produktvariante später mit anderen Klassen verknüpft.

Ein Baukasten-Produkt besteht aus mehreren Baugruppen und Bauteilen (vgl. Abschnitt 5.3.2). Zu diesen Bauteilen gibt es mehrere Varianten. Diese werden fortan Bauteilvarianten genannt. Eine Bauteilvariante kann in mehreren Produktvarianten vorkommen. Die Verwendung einer Bauteilvariante in möglichst vielen Produktvarianten ist ein Ziel des Baukastenansatzes (vgl. Abschnitt 5.3.2). Folglich liegt hier eine m:n-Verknüpfung vor, die mittels der Variantenregel sehr flexibel umgesetzt werden kann (vgl. Bild 4.2). Der Vorteil einer solchen Variantenregel liegt darin, dass sich mittels Wildcard m:n-Verknüpfungen sehr einfach auch für große m und n umsetzen lassen. Eine Wildcard ist dabei ein Platzhalter, der für ein oder mehrere andere Zeichen stehen kann. Weiterhin lassen sich diese leicht erweitern und es müssen vorab weder alle Produktvarianten noch alle Bauteilvarianten bekannt sein. Für ein konkretes Beispiel sei auf Abschnitt 5.1.4 verwiesen.

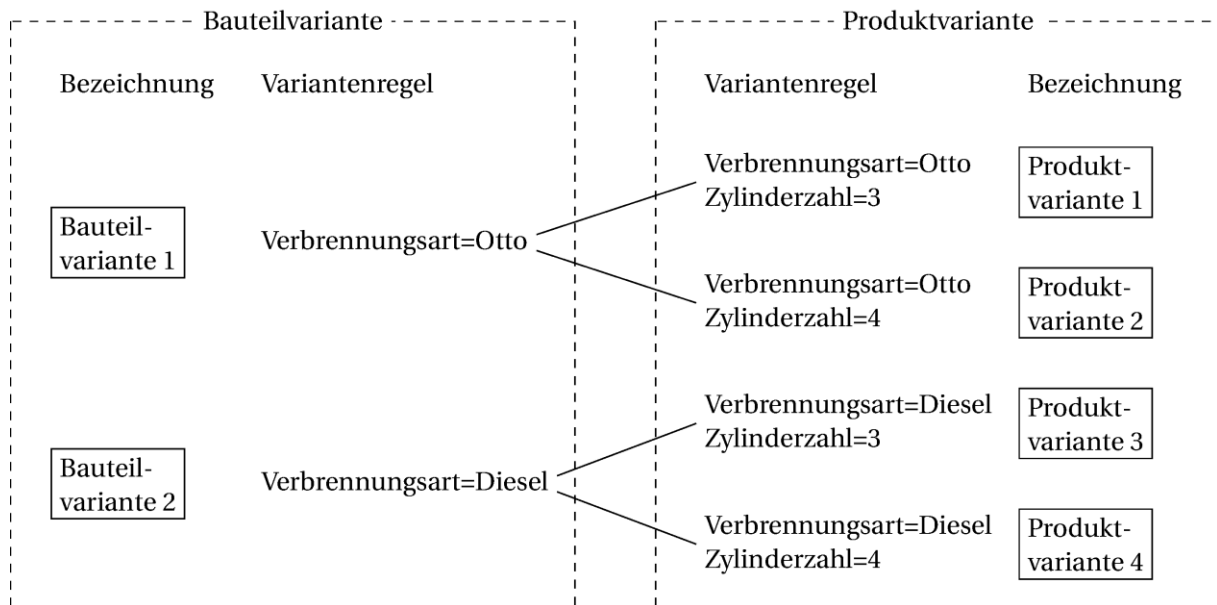


Bild 4.2: Beispiel für die Funktionsweise der Variantenregel

Example for the function of the variant rule

In Bild 4.3 sind die Klasse Produktvariante und die Klasse Bauteilvariante über eine Assoziation verknüpft. Wie oben beschrieben erfolgt die Verknüpfung regelbasiert. Somit lässt sich keine Richtung für diese Verknüpfung festlegen und die Raute entfällt.

Als weitere Klasse kommt im Partialmodell Produkt noch die *Bauteilart* hinzu. Diese beschreibt die Art oder Familie eines Bauteils auf einer Meta-Ebene. So werden sehr ähnliche Bauteile zu einer Gruppe zusammengefasst. Eine solche *Bauteilart* kann *Kurbelwelle* heißen. Diese Bauteilart fasst dann alle Bauteilvarianten von Kurbelwellen zusammen.

Jede *Bauteilart* besteht aus einer oder mehreren *Bauteilvarianten*. Eine *Bauteilvariante* kann jedoch nur zu einer Bauteilart gehören: Es liegt eine 1:n Beziehung vor.

Eine Klasse *Bauteilart* ist auch bei BRUNNER zu finden (vgl. Abschnitt 3.5.6). Jedoch nutzt er diese Klasse zur Verknüpfung der Klassen *Produkt* und *Prozess*. Im hier vorgestellten Datenmodell wird die Klasse *Bauteilart* jedoch nicht mit dem Produkt sondern mit der *Bauteilvariante* verknüpft. Grund hierfür ist die oben beschriebene Clusterung der Bauteilvarianten zu Gruppen mithilfe der Bauteilart. Über die Bauteilvariante ist die Bauteilart aber – ähnlich zu BRUNNER – mit der Produktvariante verknüpft.

Werden die vorgestellten Klassen und Zusammenhänge kombiniert ergibt sich das in Bild 4.3 dargestellte UML-Modell. Dieses beschreibt das Partialmodell Produkt.

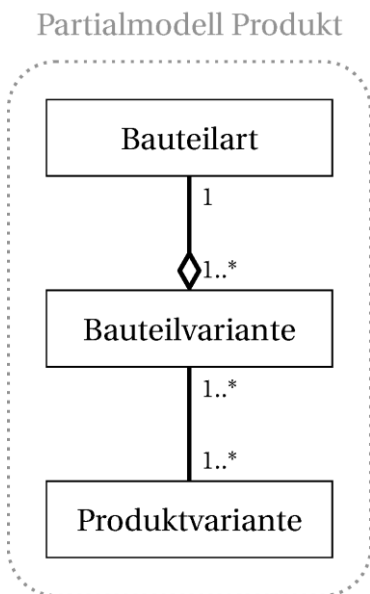


Bild 4.3: UML-Modell des Partialmodell Produkt

UML model of the partial model for the product

4.2.2 Partialmodell Prozess

Partial models of the process

Das Partialmodell Produkt wird – wie auch bei den anderen Autoren – um das Partialmodell Prozess erweitert – welches in diesem Fall nur aus der Klasse Prozess besteht.

In realen Montageprozessen in einer Fabrik, werden Bauteilvarianten verwendet. Demnach müsste die Klasse Prozess mit der Klasse Bauteilvariante verbunden werden. Jedoch lässt sich beobachten, dass in einem Prozess meist die Bauteilvarianten einer Bauteilart verwendet werden. Analog BRUNNER wird daher die Klasse Prozess mit der Klasse Bauteilart assoziiert. Für den Prozess wird also festgelegt, welche Bauteilart er verarbeitet. Eine Bauteilart kann in mehreren Prozessen verarbeitet werden, da sich mehrere Arbeitsschritte auf dieselbe Bauteilart beziehen können. Ebenso können für einen Prozess mehrere Bauteilarten notwendig sein. Die Beziehung hat die Multiplizität m:n (s. Bild 4.4).

Als Besonderheit kommt hinzu, dass eine Bauteilart zwar mehreren Prozessen zugeordnet sein kann, jedoch kann es nur einmal konsumiert werden. Konsumiert bedeutet, dass die Bauteilart zum Produkt hinzugefügt wird. Beispielsweise wird in der Motormontage die Zylinderkopfhaube in einem Prozessschritt auf den Motor aufgelegt. Dies ist der Schritt in dem sie konsumiert wird. Im Anschluss finden noch mehrere Prozesse statt, in denen die Haube zunächst geheftet und dann verschraubt wird. Alle Prozesse sind mit der Zylinderkopfhaube verbunden, jedoch nur einer mit dem Attribut *konsumiert*. Der Ort, an dem eine Bauteilart konsumiert wird, muss mit dem Ort verbunden sein, an dem die Bauteilart an der Montagelinie bereitgestellt wird.

Die Verknüpfung mit der Bauteilart hat für die Pflege und Anpassung des Modells große Vorteile. Die Bauteilarten ändern sich nicht sehr stark, die Bauteilvarianten dagegen schon. Kommen neue Bauteilvarianten hinzu, müssen diese nur der korrekten Bauteilart zugewiesen werden und sind über diese mit allen Prozessen verbunden, ohne dass weiterer Aufwand erforderlich ist. Das Anlegen neuer Bauteilvarianten kann in der Entwicklung erfolgen. Über die Verknüpfung erhält die Montageplanung die Information und kann diese in der Konfliktsuche (vgl. Bild 4.1) berücksichtigen.

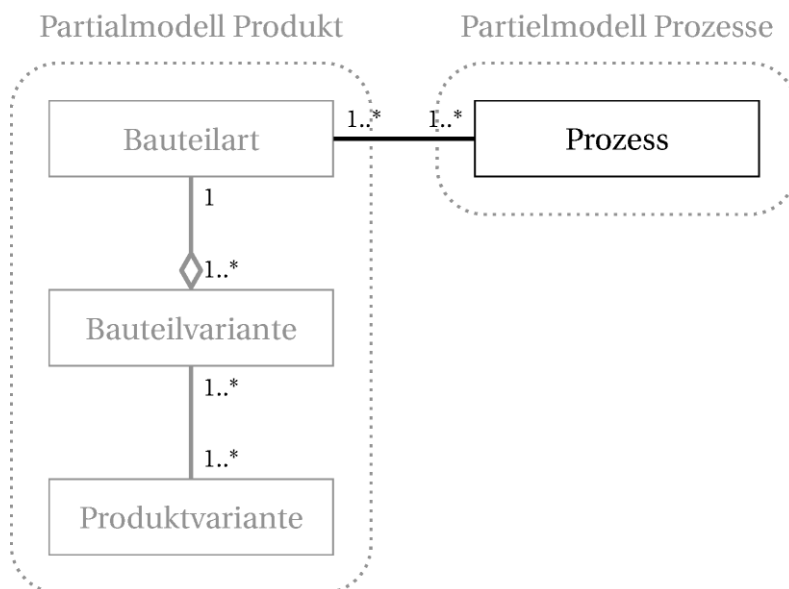


Bild 4.4: UML-Klassendiagramm erweitert um Prozess
UML class diagram extended by process

4.2.3 Partialmodell Ressource

Partial model of the resource

Das Partialmodell Ressource besteht aus den 3 Klassen: Linien, Stationen, Linienbelegung (vgl. Bild 4.5), die teilweise auch in den Modellen aus Abschnitt 3.5 auftauchen. NEUHAUSEN verwendet ebenfalls Klassen für Linien und Stationen. Obwohl JONAS in seinem Stand der Technik auf die Relevanz von Montagestationen hinweist, werden diese nicht explizit modelliert. Nach BRUNNER kann es sich bei Ressourcen auch um Gruppen von Betriebsmitteln – und somit um Stationen – handeln kann.

Im entwickelten Modell sollen Stationen und Linien als eigene Klassen modelliert werden. Im Modell muss mindestens eine Linie abgebildet werden. Diese kann beliebig viele Stationen enthalten.

Je nach Modellierungsansatz kann dieselbe Station mehreren Linien zugeordnet sein oder nur einer. Damit sie mehreren Linien zugeordnet werden kann, müssen ihre Randbedingungen (vgl. Abschnitt 4.3) in allen Linien identisch sein und auch bleiben.

Da dies selten der Fall ist und es zudem irreführend sein kann, dass eine Station mehreren Linien zugeordnet ist, wurde auf diese Möglichkeit verzichtet. Es handelt sich demnach um eine 1:n Beziehung zwischen den Klassen Linie und Station.

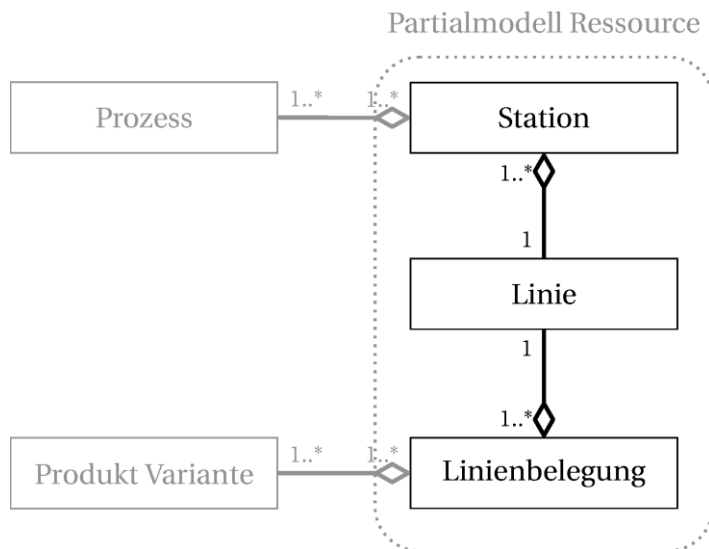


Bild 4.5: Partialmodell Ressource

Partial model for the resource

Die Linienbelegung wird von keinem Autor betrachtet. Sie beschreibt die Zuordnung von Produkten zu Linien. Entsprechend verknüpft sie diese beiden Klassen im Modell miteinander. Für die Verknüpfung zur Klasse Produktvarianten wird wiederum die Variantenlogik aus Abschnitt 4.2.1 verwendet. Ein Beispiel für eine Variantenlogik ist in Bild 4.6 dargestellt.

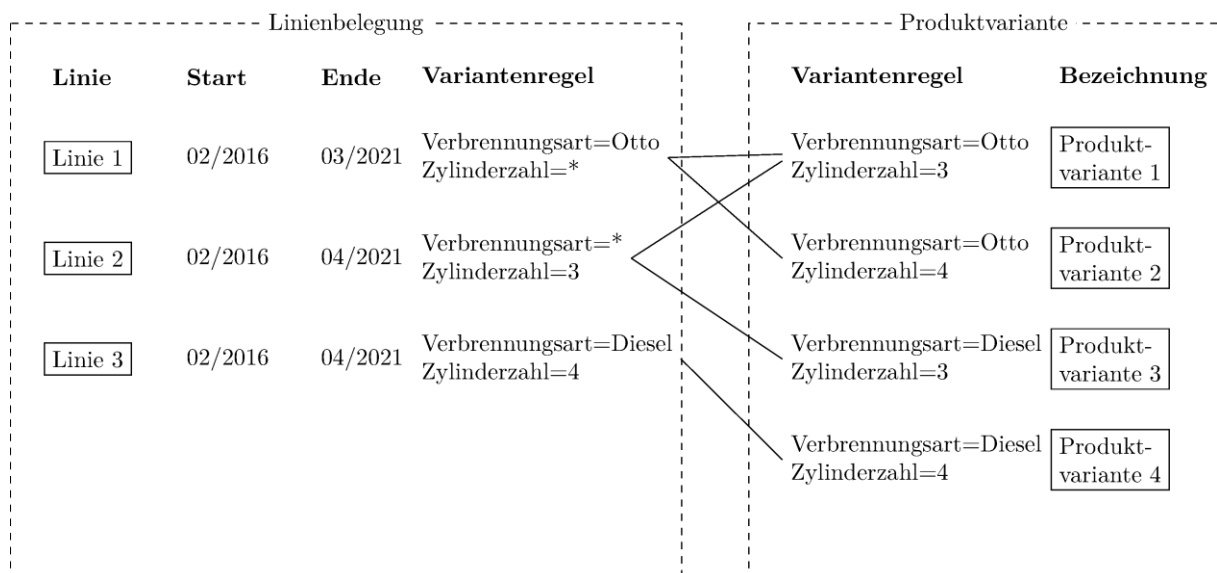


Bild 4.6: Variantenregel am Beispiel der Linienbelegung

Variant code at the example of the assembly line allocation

Die Vorteile sind denen aus Abschnitt 4.2.1 ähnlich: Mittels der bereits beschriebenen Wildcard lassen sich schnell und einfach große Gruppen von Produkten einer Linie zuweisen. Somit lässt sich die m:n Verknüpfung für große m oder n effizient umsetzen. Die Zuweisung kann erfolgen, ohne dass alle Produkte bekannt oder im Modell definiert sein müssen. Weiterhin lässt sich die Zuordnung flexibel anpassen.

Eine solche Klasse oder Logik ist in keinem der Modelle aus Abschnitt 3.5 zu finden.

4.2.4 Erläuterungen zum Basis-Datenmodell

Explanations for the basic data model

Werden die drei Partialmodelle verknüpft, ergibt sich das Basis-Datenmodell (vgl. Bild 4.7). Mithilfe des Basis-Datenmodells lassen sich regelbasiert Produkt, Prozess und Ressource miteinander verknüpfen. Die Linienbelegung schließt den Kreis und ermöglicht es somit, wichtige montagerelevante Erkenntnisse zu gewinnen und gleichzeitig den Pflegeaufwand des Modells überschaubar zu halten.

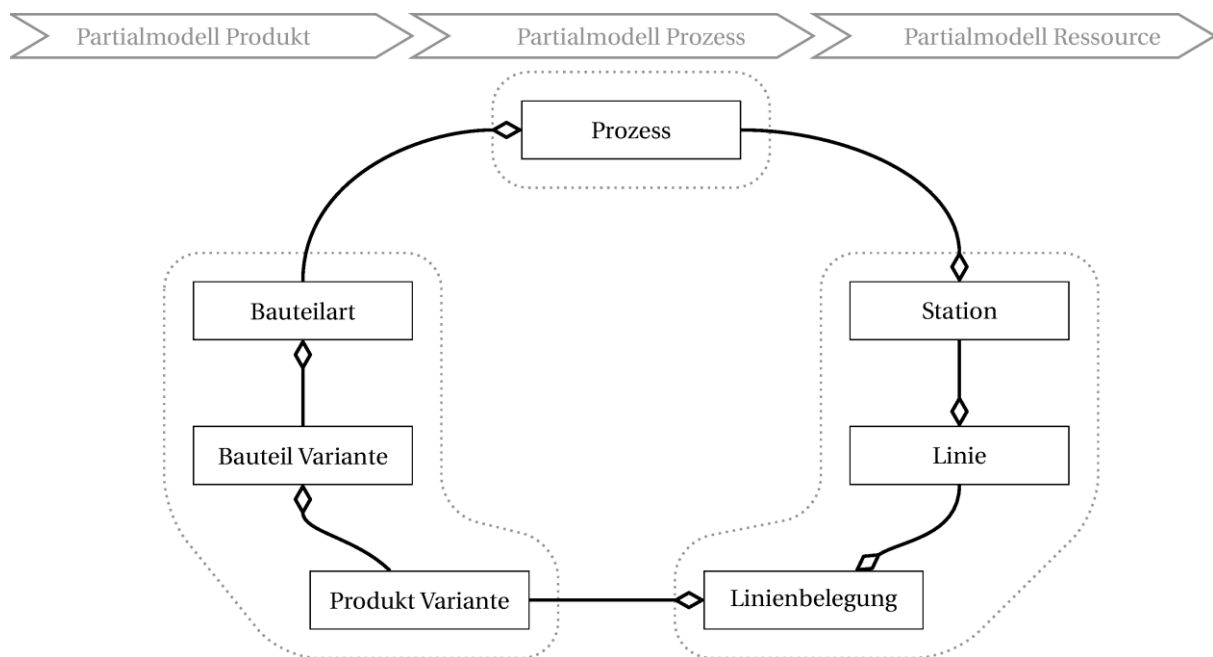


Bild 4.7: Das Basis-Datenmodell zur fähigkeitsbasierten Montageplanung [MUEL15, S. 555]

The basic data model for capability based assembly planning

Somit ist festgelegt, welche Bauteilvarianten an welchen Stationen verwendet werden. Immer wenn sich der Kreis für eine Gruppe von Instanzen der einzelnen Klassen schließen lässt, dann stehen diese in einem Zusammenhang:

Eine Linie ist beispielsweise mit einer Station verknüpft. Zu dieser Station lassen sich die verknüpften Prozesse identifizieren. Zu den Prozessen sind die Bauteilarten zugeordnet und diesen wiederum alle Bauteilvarianten dieses Typs. Die Linie ist jedoch

auch mit der Linienbelegung verknüpft. Diese beschreibt wiederum, welche Produktvarianten in der Linie hergestellt werden sollen. Der Kreis lässt sich nur dann schließen, wenn die Bauteilvariante einerseits zur richtigen Bauteilart gehört und andererseits in einer Produktvariante der Linienbelegung auftaucht. Mit Hilfe der beschriebenen Logik lassen sich folglich die Bauteilvarianten Linien und Stationen zuordnen.

Umgekehrt kann auch untersucht werden, in welchen Stationen oder Linien die Änderung einer bestimmten Bauteilvariante Auswirkungen haben könnte. Zur Bauteilvariante lassen sich die Bauteilart und darüber die betroffenen Prozesse ermitteln. Die Prozesse können wiederum allen Stationen und Linien zugeordnet werden. Um festzustellen, ob die Station und Linie tatsächlich betroffen ist, muss untersucht werden, ob die Bauteilvariante in einer Produktvariante verwendet wird, die über die Linienbelegung der ermittelten Linien zugewiesen ist. Wenn dies der Fall ist, dann kann der Kreis wiederum geschlossen werden und die Elemente stehen in einem Zusammenhang.

Das Datenmodell ermöglicht es eine hohe Dynamik abzubilden: Kommt eine neue Bauteilvariante hinzu, muss diese nur der korrekten Bauteilart zugewiesen und mit einer Variantenregel versehen werden. Falls erforderlich müssen die Variantenregeln bereits vorhandener Bauteilvarianten angepasst werden. Über die Verknüpfungen werden anschließend alle Bauteilvarianten den richtigen Prozessen, Stationen und Linien zugewiesen. Ähnliches gilt für jedes andere Element des Datenmodells. Über die Verknüpfungen ist sichergestellt, dass Anpassungen sich direkt auf das Modell auswirken.

Ändert sich die Zuordnung von Produktvarianten zu Linien, muss nur die Linienbelegung entsprechend angepasst werden. Das manuelle Zuordnen von Produkten zu Linien oder sogar Stationen kann entfallen. Über die Verknüpfungen des Modells ist sichergestellt, dass alle Bauteilvarianten und Prozesse den richtigen Stationen zugeordnet werden. Dieser Schritt fehlt in allen anderen Modellen aus Abschnitt 3.5.

Theoretisch ließe sich diese Verbindung zwischen Linienbelegung und Bauteilvariante auch ohne die Klasse Produktvariante realisieren. Dazu könnten der Variantencode der Bauteilvariante mit dem der Linienbelegung abgeglichen werden. Würden diese zusammenpassen, könnte die Bauteilvariante der Linie zugeordnet werden. Jedoch würden dann Bauteilvarianten einbezogen, zu denen kein reales Produkt existiert (Vgl. Sechszylinder-Beispiel aus Abschnitt 4.2.1).

Bereits dieses einfache Basis-Datenmodell erleichtert die Arbeit des Montageplaners, indem die manuelle Zuordnung von Bauteilvarianten zu Linien und Stationen entfällt. Für eine Analyse müssen die Informationen bisher in verschiedenen Listen und Systemen gesucht und in einem Zusammenhang gebracht werden. Mithilfe des Modells kann dieser Aufwand deutlich reduziert werden.

4.3 Erweiterung des Basis-Datenmodells

Extension of the basic data model

Das Basis-Datenmodell ermöglicht es, Bauteilvarianten und Stationen verknüpfungs- und regelbasiert in einem Zusammenhang zu setzen. Es wurden durchgängig Assoziationen zur Verknüpfung eingesetzt und die Klasse Linienbelegung wurde ergänzt. Somit müssen die Informationen nur an einer Stelle gepflegt werden. Alle abhängigen Informationen sind verknüpft.

Eine Aussage, ob sich in einer Konfiguration des Montagesystemverbundes alle Produktvarianten montieren lassen, ist basierend auf diesen Daten noch nicht möglich. Dazu sind weitere Informationen und somit weitere Klassen im Modell erforderlich. In Abschnitt 4.3.1 wird das Modell daher um Anforderungen und Fähigkeiten erweitert. Kapazitäten stellen eine weitere Restriktion bei der Montierbarkeit eines Produktes dar. Sie werden in Abschnitt 4.3.2 in das Modell integriert.

Um die Montageplanung weiter zu optimieren werden Ansprechpartner und deren Kontaktinformationen in das Modell aufgenommen (vgl. Abschnitt 4.3.3).

4.3.1 Erweiterung um Anforderungen und Fähigkeiten

Extension by requirements and capabilities

Das Modell soll den Planer unterstützen, indem es ihm bei Auffinden von Konflikten hilft. Ein solcher Konflikt kann entstehen, wenn Restriktionen verletzt werden. Diese Restriktionen sollen als Anforderungen und Fähigkeiten beschrieben werden.

Soll beispielsweise eine Schraube mit einem Drehmoment von 20 Nm angezogen werden (Anforderung), die vorhandene Ressource kann aber nur 15 Nm bereitstellen (Fähigkeit), so liegt ein Konflikt vor.

Im Modell von BRUNNER findet sich eine Kombination von Features, die Einfluss auf die Montage haben. Er nennt diese ressourcenrelevante Produktmerkmale und produktrelevante Ressourcenmerkmale. Erstere werden in diesem Modell Anforderungen, letztere Fähigkeiten genannt.

Anforderungen sind demnach Eigenschaften des Produktes bzw. von dessen Bauteilvarianten. Diese Anforderungen müssen durch Eigenschaften des Montagesystems – die Fähigkeiten – erfüllt sein. Insofern ist die Bezeichnung von BRUNNER sehr genau, wenn er von *ressourcenrelevanten Produktmerkmalen* und *produktrelevanten Ressourcenmerkmalen* spricht. In dieser Arbeit werden zur Vereinfachung die Begriffe Anforderungen und Fähigkeiten verwendet. Merkmale des Produktes, die keine Auswirkung auf die Ressourcen haben, sind ebenso wenig relevant für das Modell, wie Merkmale der Ressourcen die aber keine Relevanz für das Produkt haben.

Weiterhin macht diese Erklärung deutlich, dass sich Anforderungen und Fähigkeiten nicht unabhängig voneinander modellieren lassen. Denn die Entscheidung, was relevant ist und was nicht, hängt vom jeweiligen Gegenstück ab.

Das Modell muss derart gestaltet werden, dass sich Anforderungen und Fähigkeiten möglichst gut miteinander vergleichen lassen. Eine Bauteilvariante kann mehrere Anforderungen haben. Es ist aber auch denkbar, dass eine Gruppe von Bauteilvarianten identische Anforderungen besitzt. Zwischen Bauteilvarianten und Anforderungen handelt es sich folglich um eine m:n Beziehung.

Zu den Anforderungen müssen jeweils entsprechende Fähigkeiten vorhanden sein. Eine Station kann mehrere Fähigkeiten haben und eine Fähigkeit in mehreren Stationen enthalten sein. Wiederum handelt es sich um eine m:n Beziehung.

Eine Herausforderung bei der Erstellung des Modells besteht darin, die Anforderungen und Prozesse einander korrekt zuzuordnen. Hat eine Bauteilvariante mehrere Anforderungen und eine Station mehrere Fähigkeiten, ist dies nicht trivial (vgl. Bild 4.8). Trotzdem soll diese Zuordnung automatisiert erfolgen, um manuellen Aufwand zu vermeiden.

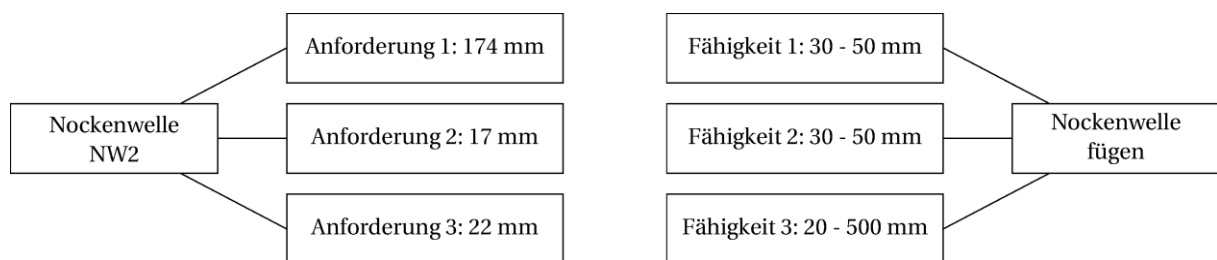


Bild 4.8: Problematik bei der Zuordnung von Anforderungen und Fähigkeiten

Problems in matching requirements and capabilities

Eine Möglichkeit wäre, die Anforderungen und Fähigkeiten entsprechend eine Identifikationsnummer des Gegenstückes zu verknüpfen. Dieses Vorgehen würde bei der zu erwartenden großen Anzahl von Anforderungen und Fähigkeiten jedoch schnell an die Grenzen der Beherrschbarkeit stoßen.

Eine weitere Möglichkeit wäre es, den Anforderungen und Fähigkeiten Bezeichnungen zuzuteilen und anschließend Anforderungen und Fähigkeiten mit identischen Bezeichnungen abzugleichen. Dieses Vorgehen wäre jedoch fehleranfällig. Wären die Zeichenketten nicht identisch, ließen sie sich nicht zuordnen.

Ein weiterer Nachteil besteht darin, dass der Produktentwickler für jede Bauteilvariante erneut überlegen muss, welche Anforderungen diese hat. Wie bereits beschrieben lassen sich Anforderungen nur modellieren, wenn die Gegenseite bekannt ist. Die dafür benötigte Kenntnis des Montagesystems kann beim Produktentwickler nicht vorausgesetzt werden. Somit bestünde ein großes Risiko, dass Anforderungen vergessen würden.

Um diese Schwierigkeiten zu umgehen und dennoch eine automatisierte Zuordnung zu ermöglichen wird eine weitere Klasse eingeführt: der Prozesstyp. Dieser beschreibt, welche Attribute zum Durchführen des Prozesses relevant sind. Diese hängen im Allgemeinen vom Prozess ab. Der Prozesstyp beschreibt dabei nur die Attribute, nicht deren Ausprägung.

Im Beispiel aus Bild 4.8 beschreibt der Prozesstyp einen Abstand, einen Durchmesser und eine Breite. Über den Prozess und die Bauteilart lässt sich eine Verbindung zwischen Prozesstyp und Bauteilvariante herstellen. Somit ist definiert, dass für jede Bauteilvariante der betrachteten Bauteilart als Anforderungen die Attribute Abstand, Durchmesser und Breite definiert werden müssen. Kommt eine neue Bauteilvariante hinzu, müssen lediglich diese Attribute angegeben werden. Somit ist ausgeschlossen, dass einzelne Anforderungen vergessen werden. Analog verhält es sich beim Anlegen oder Anpassen von Stationen. Werden diese angepasst oder neu angelegt, müssen wiederum die Ausprägungen der Attribute des Prozesstyps als Fähigkeiten hinterlegt werden.

Jedem Prozess können Prozesstypen zugeordnet werden. Ein Prozesstyp kann einem oder mehreren Prozessen zugewiesen werden. Die Erweiterung des Basis-Modells um Anforderungen, Fähigkeiten und Prozesstyp ist in Bild 4.9 dargestellt.

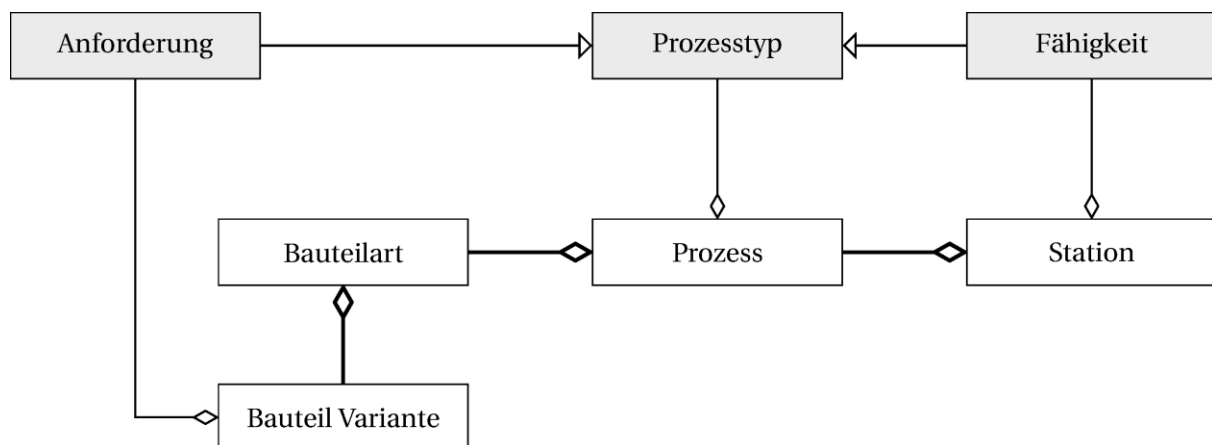


Bild 4.9: Integration von Anforderungen und Fähigkeiten ins Basis-Datenmodell

Integration of requirements and capabilities into the basic data model

Auch bei der Modellierung von Anforderungen, Fähigkeiten und Prozesstyp zeigen sich die Stärken eines verknüpften Modells in einem dynamischen Umfeld. Ändert sich zu einem späteren Zeitpunkt der Prozess oder wird festgestellt, dass eine bisher nicht betrachtete Eigenschaft einen Einfluss auf den Prozess hat, so kann diese im Prozesstyp ergänzt werden.

Über die Verknüpfung wird sie an die betroffenen Stationen und Bauteilvarianten vererbt. Dort müssen nur noch die Ausprägungen der Attribute als Fähigkeit und Anforderung festgelegt werden.

4.3.2 Erweiterung um Kapazitäten an Bauteilvarianten

Extension by capacities of part variants

Die Kapazitäten an Bauteilvarianten beschreiben ebenfalls Fähigkeiten des Montagesystems, allerdings stehen ihnen besondere Anforderungen gegenüber. Die Kapazität beschreibt die Fähigkeit, eine bestimmte Anzahl verschiedener Bauteilvarianten einer Bauteilart an der Linie bzw. einer ihrer Stationen anstellen zu können (vgl. Abschnitt 2.3.2). Ziel ist es, eine Aussage darüber zu generieren, ob sich alle Bauteilvarianten an der Linie anstellen lassen. Ist dies nicht der Fall, müssen Änderungen am Montagesystem gemäß Abschnitt 4.1 und Bild 4.1 vorgenommen werden.

Bei der Anstellung müssen drei Fälle unterschieden werden: Werden die Bauteilvarianten direkt an der Station angestellt, so hängt die Kapazität von der Station ab. So kann beispielsweise die Fläche direkt an der Station begrenzt sein oder der Bereich den der Mitarbeiter in Taktzeit erreichen kann. Wird ein Hilfsmittel wie bspw. ein Kran zum Heben schwerer Lasten benötigt, kann dieser die Fläche vorgeben. Es ist ebenfalls möglich, dass die Station über eine beschränkte Anzahl an Zuführungsbahnen verfügt und die Kapazität hierdurch beschränkt wird. Diese Beispiele zeigen, dass es viele Gründe geben kann, die die Kapazität an einer Station einschränken.

Der zweite zu betrachtende Fall liegt vor, wenn die Bauteilvarianten nicht direkt an der Station angestellt werden und die unmittelbare Umgebung der Station somit keinen Einfluss auf die Kapazität hat. Dies ist dann der Fall, wenn sich die Teile bspw. auf einem Regal oder sogenanntem Set mit dem Produkt durch die Montagelinie bewegen. In der Automobilendmontage werden diese beispielsweise Car-Sets genannt, in der Motormontage Engine-Set. In diesem Fall wird die Kapazität beschränkt durch die Stelle, an der dieses Set befüllt wird. Auch hier kann wieder die Fläche oder die Taktzeit beschränkende Eigenschaft sein.

Der dritte Fall tritt ein, wenn die Bauteile zwar an der Station angestellt werden, die Anlieferung jedoch in der Verbau-Sequenz erfolgt. In diesem Fall kann die Anzahl der direkt anzustellenden Bauteilvarianten reduziert werden. Hier bestimmen sowohl die Station als auch der Ort, an dem die Sequenz gebildet wird, die Kapazität.

Wie aus der Beschreibung zur Kapazität ersichtlich wird, stellt sie eine Schnittstelle zwischen Logistikplanung und Montageplanung dar. In der Realität kommt es zudem häufig vor, dass es sowohl für die einzelne Station als auch für die gesamte Linie eine Maximalkapazität gibt.

Um allen genannten Einflüssen gerecht zu werden, wird die Klasse Kapazität eingeführt. Auf der einen Seite ist diese über eine Assoziation mit einer Bauteilart verknüpft, auf der anderen mit der Station und der Linie (vgl. Bild 4.10). Somit lassen sich beliebig Kapazitäten und Abhängigkeiten abbilden.

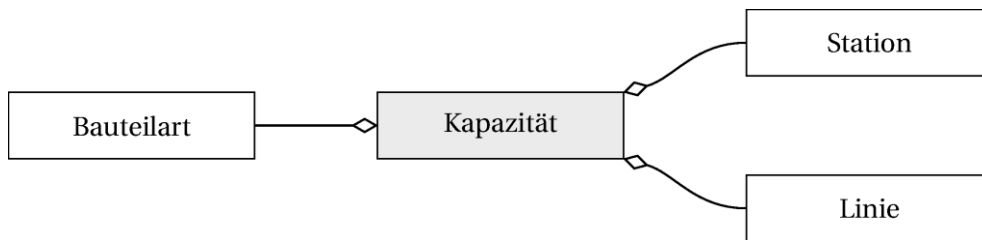


Bild 4.10: Einbindung der Kapazität in das Datenmodell
Integration of capacity into the data model

Im praktischen Anwendungsbeispiel der Motormontage wird die Zylinderkopfhaube beispielsweise direkt an der Montagelinie angestellt. Der Aufbau einer Station ist in Bild 4.11 dargestellt. Die Fläche wird durch die Säulen der Halle und die benachbarten Stationen begrenzt. An der Station lassen sich drei Drehgestelle mit verschiedenen Zylinderkopfhauben anstellen. Die Kapazität, gemäß Definition in Kapitel 2.3.2, beträgt also bedingt durch die Station drei.

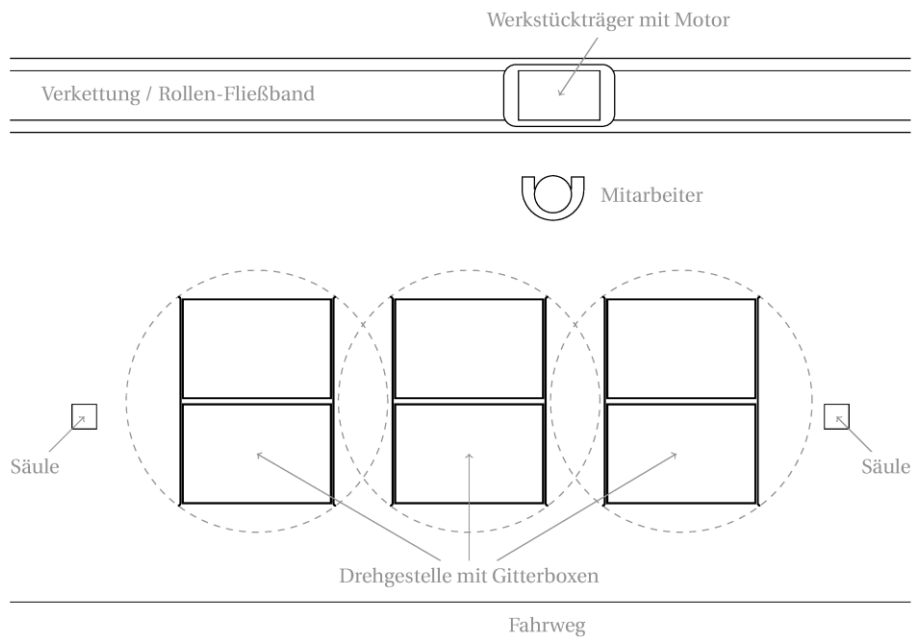


Bild 4.11: Anstellung am Beispiel der Zylinderkopf-Haube
Part supply at the example of the cylinder head cover

Die Kapazität einer Station stellt eine Sonderform der Fähigkeit dar. Denn beim Abgleich wird nicht für jede Bauteilvariante einzeln untersucht, ob die Fähigkeiten zur Erfüllung der Anforderungen ausreichen, sondern es wird die Summe der verschiedenen Bauteilvarianten bzw. deren Anzahl bewertet.

Bei einer Überschreitung der Kapazität lässt sich diese nicht einer einzelnen Bauteilvariante zuordnen, da das Entfernen jeder der zugewiesenen Bauteilvarianten das Problem lösen würde. Unabhängig davon, welche Bauteilvariante entfernt wird, ist die

Fähigkeit eingehalten, wenn die Anzahl der verbleibenden Varianten danach wieder drei beträgt.

Das beschriebene Beispiel entspricht dem ersten der drei oben genannten Fälle. Würde die Kapazität überschritten, dann blieben als Reaktionsmöglichkeiten seitens der Fähigkeiten noch, die Zylinderkopfhaube ins Engine-Set aufzunehmen oder sie in Sequenz anzustellen. Da im betrachteten Beispiel die Abmessungen der Haube deutlich größer sind als die Fächer des Engine-Sets, entfällt die erste Option, sodass nur noch die Anlieferung in Sequenz bleibt. Hierfür müssten Fläche und Personal in der Logistik eingeplant werden.

Alternativ kann auch das Produkt angepasst werden. Es könnten Bauteilvarianten vereinheitlicht und somit deren Anzahl reduziert werden. Oder die Linienbelegung wird so angepasst, dass sich die Anzahl der Hauben für die beschriebene Linie auf maximal drei reduziert.

4.3.3 Erweiterung um Ansprechpartner

Extension by contact persons

Mithilfe der Klassen für Anforderungen, Fähigkeiten, Prozesstypen und Kapazitäten lassen sich wesentliche Aspekte der Montierbarkeit untersuchen. Im Unternehmensalltag betrifft die Dynamik jedoch nicht nur die Bauteilvarianten und Montagelinien, sondern auch das Personal.

Mitarbeiter wechseln ihre Stellen oder bekommen neue Aufgabenbereiche zugewiesen, was dazu führt, dass Ansprechpartner und Verantwortlichkeiten sich ändern. In der Montageplanung wirkt sich dies besonders stark aus, da sie sehr viele Schnittstellen zu Prozesspartnern besitzt. Dies kann dazu führen, dass die Suche nach dem richtigen Ansprechpartner mit großem Aufwand verbunden ist.

Für eine effiziente Montageplanung ist Kommunikation im Netzwerk jedoch ein wichtiger Aspekt. Daher soll das Modell hier ebenfalls unterstützen und die Effizienz verbessern. Zu diesem Zweck wird die Klasse *Ansprechpartner* zum Modell hinzugefügt.

Diese Klasse enthält wichtige Kontaktinformationen wie den Namen, die E-Mailadresse und Telefonnummer eines Ansprechpartners. Somit kann mit diesem Ansprechpartner Kontakt aufgenommen werden. Eine Beschreibung der Position dieses Ansprechpartners ist ebenfalls enthalten. Somit lassen sich verschiedene Rollen innerhalb des Unternehmens ins Modell integrieren. Dies soll das Auffinden des richtigen Ansprechpartners erleichtern. Meist gibt es für eine Bauteilart mehrere Ansprechpartner: Zum Beispiel den Gruppenleiter, den Entwickler und den Konstrukteur.

Die Klasse *Ansprechpartner* ist an zwei Stellen ins Datenmodell eingebunden: Entwickler sind meist je Bauteilart definiert. Ein Entwickler kann für mehrere Bauteilarten zuständig sein. Ebenso können mehrere Entwickler für eine Bauteilart zuständig sein: Es handelt sich also um eine m:n Beziehung.

Die Montageplaner wiederum sind Stationen zuzuordnen. Meist ist ein Abschnitt von mehreren Stationen einem Planer zugeordnet. Es ist aber auch denkbar, dass für eine Station mehrere Planer zuständig sind. Daher gibt es auch hier eine m:n-Beziehung.

Die korrekte Zuordnung ist in hohem Maße von der Organisationsstruktur des Unternehmens abhängig. In sehr großen Unternehmen ist die Gliederung eher feiner, in kleinen Unternehmen eher gröber. In kleinen Unternehmen ist es denkbar, die Ansprechpartner der Montageplanung je Linie und nicht je Station zu definieren.

Für den hier vorliegenden Fall der Großserienmontage wird die Klasse Ansprechpartner mit der Bauteilart und mit der Station verknüpft. Entwickler werden also je Bauteilart definiert und Ansprechpartner der Produktion je Station. Dies ist grafisch in Bild 4.12 dargestellt. Mithilfe der Klasse *Ansprechpartner* lässt sich die personelle Dynamik eines Unternehmens leicht abbilden und Ansprechpartner können auch bei hoher Fluktuation effizient gefunden werden. Bekommt ein Ansprechpartner ein neues Aufgabengebiet, müssen lediglich die Verknüpfungen angepasst werden. Kommt ein neuer Ansprechpartner hinzu, muss einmalig eine neue Instanz erstellt und verknüpft werden.

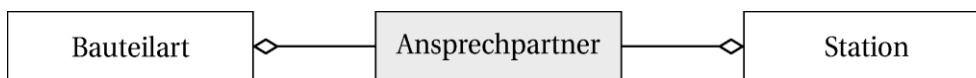


Bild 4.12: Integration von Ansprechpartnern in das Basis-Datenmodell

Integration of contact persons into the basic data model

Das Gesamt-Datenmodell, welches die Zusammenhänge aus den Abschnitten 4.2 und 4.3 zusammenfasst, ist in Bild 4.13 dargestellt.

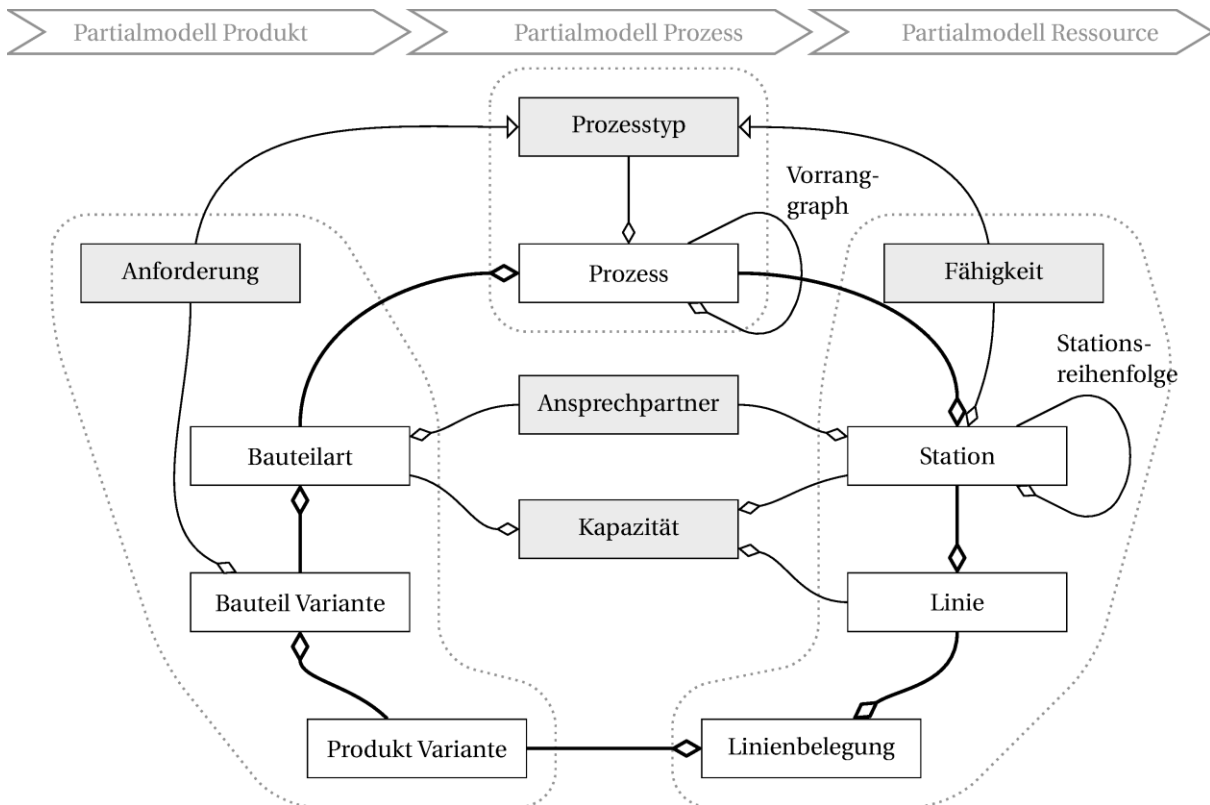


Bild 4.13: Gesamt-Datenmodell
Complete data model

4.4 Verwendung des Modells in der Montageplanung

Using the model in assembly planning

Das Modell und seine Ergebnisse sollen genutzt werden, um Entscheidungen im Rahmen der Montageplanung treffen zu können. Dazu werden in diesem Abschnitt einige Auswertungen und deren Visualisierungen vorgestellt. Sie dienen als Basis für Entscheidungen in der Integrationsplanung. Wie in Abschnitt 4.1 beschrieben stehen hierzu drei Handlungsoptionen zur Verfügung: Anpassung der Linienbelegung, Anpassung der Ressource, Anpassung des Produkts. Die ersten Auswertungen fokussieren auf die Anpassung des Produkts. Die späteren Auswertungen beziehen auch die Ressourcen und die Linienbelegung mit ein.

Bereits in den frühen Phasen des Produktentstehungsprozesses (vgl. Abschnitt 2.2) lassen sich erste Informationen über das Produkt und die zu erwartende Varianz generieren. Auch wenn zu diesem Zeitpunkt noch kein 3D-Modell aller Bauteilvarianten besteht, lässt sich meist schon die Bauteilvariantenvielfalt ermitteln. Für die Integrationsplanung ist insbesondere der Vergleich mit dem bestehenden Produktportfolio interessant.

Für die folgenden Abschnitte 4.4.1 bis 4.4.6 sei angenommen, dass im Montagesystemverbund aktuell die erste Produktgeneration (1. PG) hergestellt wird.

Zu dieser ersten Produktgeneration sind folglich alle Produkt-, Prozess und Ressourceninformationen vorhanden, wenn auch nicht immer in der richtigen Form und im richtigen Format. Weiterhin gelte, dass der Montagesystemverbund alle Produkte dieser Generation konfliktfrei herstellen kann.

Als Planungsprämisse sei angenommen, dass eine zweite Produktgeneration (2. PG) integriert werden soll. Diese besteht aus neuen Produktvarianten, die teilweise neue Bauteilvarianten benötigen und teilweise Bauteilvarianten aus der 1. Produktgeneration übernehmen. Die Anlaufszzenarien dieser zweiten Produktgeneration seien so gestaltet, dass es eine gewisse zeitliche Überlappung der Herstellung der Produktvarianten der ersten Produktgeneration und der zweiten Produktgeneration gebe.

4.4.1 Bauteilvarianten in Produktgenerationen

Mithilfe des Diagramms in Bild 4.14 lässt sich die Bauteilvielfalt über alle Produktgenerationen ermitteln. Es werden zum einen die Bauteilvariantenanzahl der ersten Produktgeneration für jede Bauteilart gezeigt. Im Bild sind es beispielsweise fünf Bauteilvarianten je Bauteilart. Zum anderen wird die Anzahl der Bauteilvarianten der zweiten Produktgeneration dargestellt.

Bauteilvarianten aus der ersten Produktgeneration können in die zweite Generation übernommen werden. Ebenso können Bauteile aus der zweiten Generation in die erste Generation übernommen werden. Diese Bauteilvarianten nennt man (rückwärts)kompatibel. Beide Phänomene sind mittels der Überlappung von hellgrauen und dunkelgrauen Balken dargestellt. Es werden mehrere Fälle unterschieden, die in Bild 4.14 dargestellt sind.

Bei Bauteilart 1 gibt es in der ersten Produktgeneration fünf Bauteilvarianten. Mit der zweiten Produktgeneration kommen fünf weitere, neue Bauteilvarianten hinzu. In Summe ergeben sich somit zehn Bauteilvarianten für Bauteilart 1.

Für Bauteilart 2 ist ein anderer Extremfall beschrieben. In diesem gibt es in der ersten Produktgeneration ebenfalls fünf Bauteilvarianten. Die zweite Produktgeneration verwendet die identischen fünf Bauteilvarianten. Als Summe über beide Produktgenerationen ergeben sich fünf Bauteilvarianten – im Diagramm überlappen die Balken vollständig.

Es sind auch Fälle möglich, die zwischen denen von Bauteilart 1 und Bauteilart 2 liegen. Ein solchen Fall ist für Bauteilart 3 beschrieben. Auch hier haben die beiden Produktgenerationen je fünf Bauteilvarianten. Eine Bauteilvariante der neuen Produktgeneration ist jedoch ein Übernahmeteil aus der alten Produktgeneration. Daher überlappen sich die beiden Balken in Bild 4.14 um ein Feld und es ergibt sich eine Summe von neun Bauteilvarianten.

Die Information, welche Teile neu sind und welche aus einer vorherigen Produktgeneration übernommen werden, ist essentiell um erste Abschätzungen über

die Montierbarkeit treffen zu können. Diese Auswertung leistet einen Wichtigen Beitrag bei der Produktbeeinflussung (vgl. Schritt 7: Produktbeeinflussung, Bild 4.1).

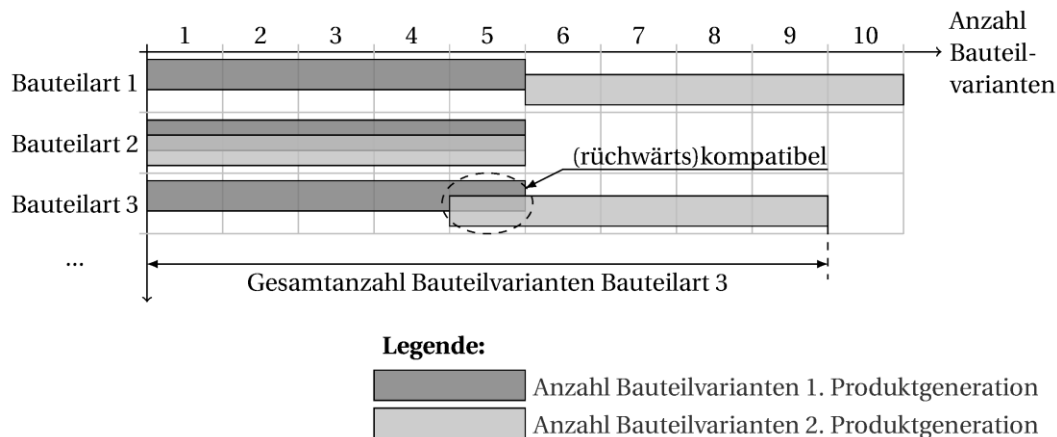


Bild 4.14: Absolute Bauteilvielfalt zweier Produktgenerationen

Absolute part variety of two product generations

Für eine große Anzahl von Bauteilarten, wie sie in der Praxis vorkommen, kann das Diagramm aus Bild 4.14 zwar verwendet werden. Um eine schnelle und einfache Aussage über den Trend der Bauteilvariantenvielfalt von Produktgenerationen zu treffen, ist das Diagramm allerdings bedingt geeignet. Insbesondere bei Baukastenprodukten ist diese Trendaussage jedoch wichtig. Die Bauteilvarianten sollen möglichst übergreifend verwendet werden. Wird der Ansatz erfolgreich verfolgt sollte die Bauteilvarianz im Laufe der Zeit bei gleichbleibender Produktvarianz sinken. Oder die Produktvarianz steigt ohne dass die Bauteilvarianz gleichermaßen mit ansteigt.

4.4.2 Bauteilvariantenentwicklung

Um diese Trendaussage kompakt darzustellen, wurde das in Bild 4.15 dargestellte Ringdiagramm entwickelt. Hierin werden vier verschiedene Kategorien gebildet: [MUEL17, S. 255-256]

1. Übernahme
2. Optimierung
3. Neuentwicklung
4. Mehrung

Die Bauteilvariantenentwicklung jeder Bauteilart wird mit Hilfe des Modells bewertet und die Bauteilart in eine der vier Kategorien eingeordnet. Für die Einordnung sind drei Zeiträume zu betrachten. Zum einen die Phase in der nur die erste Produktgeneration hergestellt wird. Weiterhin die Phase in der nur die zweite Produktgeneration hergestellt wird und der bereits erwähnte Überlappungszeitraum, in dem beide Produktgenerationen gleichzeitig hergestellt werden.

Im Fall *Übernahme* werden keine neuen Teile für die neue Produktgeneration ermittelt. Die Bauteile aus der ersten Produktgeneration werden vollständig übernommen. Dieser Fall wurde bereits für Bauteilart 2 in Bild 4.14 beschrieben. Es ist ebenfalls denkbar, dass die Teile rückwärtskompatibel entwickelt werden. Das bedeutet, dass die neu entwickelten Teile der zweiten Produktgeneration auch (rückwärts) in der ersten Produktgeneration eingesetzt werden können. In Bild 4.15 ist die Rückwärtskompatibilität durch einen Pfeil dargestellt. [MUEL17, S. 255-256]

Bei einer *Optimierung* konnte die Anzahl der Bauteilvarianten von der ersten Generation zur zweiten Generation verringert werden. Dazu muss in der Regel eine neue Bauteilvariante entwickelt werden, was dazu führt, dass im Überlappungszeitraum mehr Bauteilvarianten verarbeitet werden müssen als vorher. [ebd.]

Bei einer *Neuentwicklung* wird die gleiche Anzahl an Bauteilvarianten wie in der alten Produktgeneration für die neue Produktgeneration entwickelt. Dies entspricht der Bauteilart 1 aus Bild 4.14. Im Überlappungszeitraum verdoppelt sich die Anzahl der Bauteilvarianten. Nach dem Überlappungszeitraum ist sie wieder auf dem Niveau der ersten Produktgeneration. [ebd.]

Der letzte Fall ist die *(Varianten-)Mehrung*. Hierbei müssen für die zweite Produktgeneration mehr Bauteilvarianten entwickelt werden, als in der ersten Produktgeneration vorhanden sind. Somit sind in der Überlappungsphase mehr als die doppelte Anzahl an Bauteilvarianten vorhanden. Selbst nach Auslaufen der ersten Generation müssen noch mehr Varianten gehandhabt werden. In Tabelle 4.1 sind diese Zusammenhänge kompakt dargestellt. [ebd.]

Tabelle 4.1: Übersicht der vier Kategorien zur Trendbeurteilung von Baukastenprodukten
[MUEL17, S. 256]

Overview of the four categories for evaluating trends for modular products

	Bauteilvariantenvielfalt verglichen mit 1. PG	
	Überlappung 1. PG & 2. PG	Nach Umstellung auf 2. PG
Übernahme	Identisch	Identisch
Optimierung	Mehrung, weniger als verdoppelt	Reduzierung
Neuentwicklung	Mehrung, Verdopplung	Identisch
Mehrung	Mehrung, mehr als verdoppelt	Mehrung

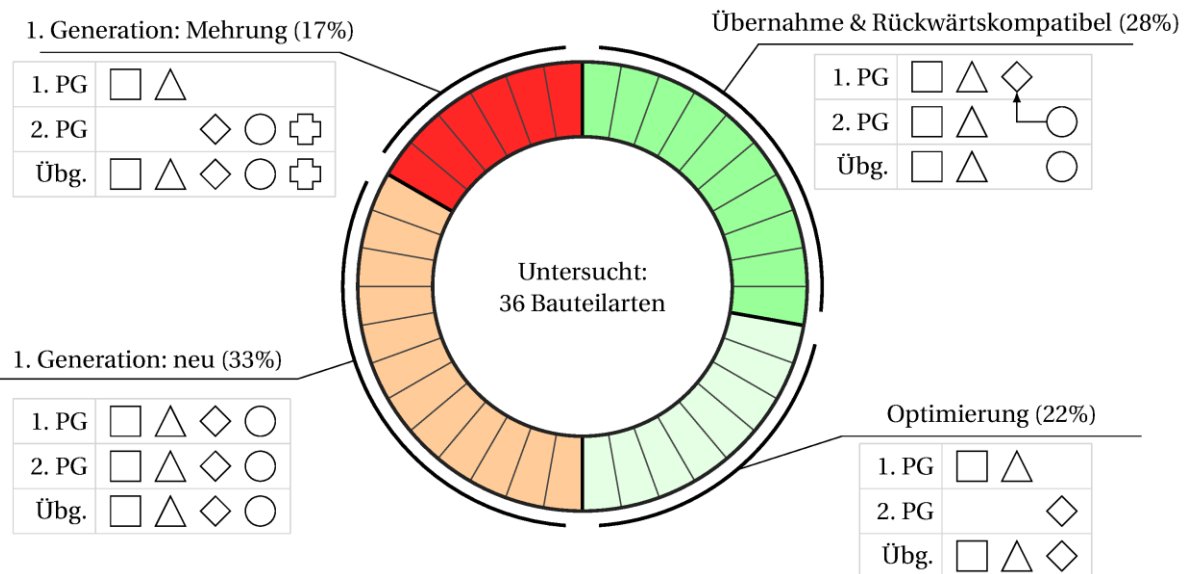


Bild 4.15: Darstellung der Bauteilvarianten-Entwicklung von der ersten Produktgeneration zur zweiten Produktgeneration (in Anlehnung an [MUEL17, S. 256])

Diagram for the evolution of part variants from the first product generation to the second product generation

4.4.3 Auswertung Produktkommunalität

Basierend auf der Kenntnis der Produktvarianten, Bauteilarten und Bauteilvarianten lässt sich ein weiteres Diagramm erzeugen. Dazu werden die Produktvarianten paarweise miteinander verglichen.

Es wird ermittelt, wie viele identische Bauteilvarianten sich diese beiden Produktvarianten teilen. Als kommunal werden in dieser Auswertung nur exakt identische Bauteilvarianten gezählt. Für Baukastenprodukte sollte wiederum ein möglichst hoher Wert angestrebt werden. In Tabelle 4.2 ist ein solcher paarweise Vergleich für die Beispielprodukte A, B, C und D dargestellt. Auf der Diagonalen würde das Produkt mit sich selber verglichen und eine Kommunalität von 100% ermittelt, daher wurde diese nicht aufgeführt. Weiterhin gilt für den Vergleich das Kommutativgesetz der Mengenlehre. Das bedeutet, in der oberen rechten Hälfte und der unteren linken Hälfte des Diagramms stehen identische Werte, die Werte sind entlang der Diagonalen gespiegelt. Im Diagramm wird daher nur eine Hälfte dargestellt.

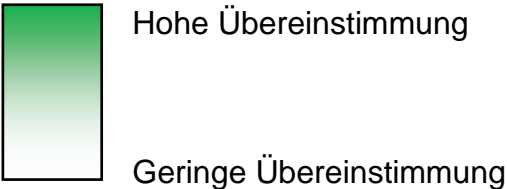
Produkte, die sich sehr ähnlich sind, sollten möglichst im selben Montagesystem hergestellt werden, da sie sich vermutlich mit einem geringen Aufwand integrieren lassen. Für Produkte mit geringer Ähnlichkeit gilt das Gegenteil. Falls möglich sollten diese in verschiedenen Montagesystemen hergestellt werden. [MUEL17, S. 255] Im Beispiel aus Tabelle 4.2 sind sich die Produkte A und B sehr ähnlich und sollten folglich im selben Montagesystem hergestellt werden. Analoges gilt für die Produkte C und D. Alle anderen Kombinationen (A+C, A+D, B+C, B+D) weisen große Unterschiede auf und sollten daher nicht in einem Montagesystem hergestellt werden.

Tabelle 4.2: Heatmap für die Bauteilvariantenkommunalität verschiedener Produkte (in Anlehnung an [MUEL17, S. 256])

Heatmap of overlapping part variety for different products

	A	B	C	D
A		91%	43%	12%
B			36%	21%
C				64%
D				

Legende:



Zu einem frühen Zeitpunkt im PEP ist dies eine grobe Abschätzung, da die spezifischen Anforderungen und Fähigkeiten der Montagesysteme noch nicht in die Überlegung einbezogen wurde. Dennoch lassen sich erste Szenarien untersuchen.

4.4.4 Auswertung Stationskapazität

Das Produkt wird aus mehreren Perspektiven analysiert. Im nächsten Schritt wird die Linienbelegung hinzugezogen. Die Auswertung der Stationskapazitäten kombiniert die Balkendiagramme aus Bild 4.14 mit der Linienbelegung und der Kapazität als Fähigkeit der Montagestation (vgl. Abschnitt 4.3.2).

Es wird untersucht, welche Bauteilvarianten gemäß Linienbelegung in den einzelnen Linien vorkommen. Wie im Balkendiagramm erfolgt dies für jede Produktgeneration einzeln und für den Überlappungszeitraum. Dies ist beispielhaft in Tabelle 4.3 dargestellt.

In den Spalten sind alle Bauteilarten aufgetragen. In den Zeilen sind je Linie die Auslastung in der 1. Produktgeneration, der 2. Produktgeneration, dem Überlappungszeitraum und die Kapazität der Linie bzw. der betroffenen Stationen dargestellt. In die sich ergebende Tabelle werden die jeweiligen Werte eingetragen. Anschließend werden die Zeilen mit 1. Produktgeneration, 2. Produktgeneration und Überlappungszeitraum eingefärbt. Wenn die Auslastung unterhalb der Kapazität liegt, wird die Zelle grün eingefärbt. Hier besteht ein Puffer, falls neue Bauteilvarianten hinzukommen. Sind Kapazität und Auslastung gleich, so wird die Zelle gelb gefärbt. Es können zwar alle Bauteilvarianten verarbeitet werden, jedoch besteht kein Spielraum für die Integration weiterer Bauteilvarianten mehr. Kommen weitere Bauteilvarianten hinzu, ist die Herstellbarkeit nicht mehr gegeben. Überschreitet die Auslastung die Kapazität, sind also zu viele Bauteilvarianten der Station zugewiesen, so muss die Montageplanung aktiv werden. Die Zelle wird rot eingefärbt. [MUEL17, S. 257]

Eine erste Lösungsoption zur Auflösung der rot markierten Nicht-Baubarkeiten ist die Anpassung der Ressource, hier zur Erweiterung der Kapazität. Um weitere Lö-

sungsoptionen zu finden, sollte zunächst die Station mit überschrittener Kapazität näher untersucht werden, um das Problem besser zu verstehen. Tabelle 4.3 zeigt die Daten in aggregierter Form, Bild 4.16 zeigt die Hintergrunddaten zu einem bestimmten Feld. Auf der Abszisse ist die Zeit aufgetragen, hier monatsweise für 1 Jahr. Die Ordinate ist zweigeteilt. Die obere Hälfte des Diagramms enthält ein Liniendiagramm über die Anzahl aller zu verbauender Bauteilvarianten. Welche dies genau sind und wie sich diese überlagern ist in der unteren Hälfte des Diagramms dargestellt.




Tabelle 4.3: Stationskapazitäten (in Anlehnung an [MUEL17, S. 257])

Station capacity

		Bauteilart 1	Bauteilart 2	Bauteilart 3	...
Linie A		Station 1	Station 2	Station 2	...
	1. PG	3	3	5	
	Überlapp.	6	3	9	
	2. PG	3	3	5	
	Kapa.	4	3	7	
Linie B		Station 1	Station 2	Station 3	...
	1. PG	2	3	3	
	Überlapp.	4	3	6	
	2. PG	2	3	4	
	Kapa.	4	3	5	

Legende:

1.PG: 1. Produktgeneration
 2.PG: 2. Produktgeneration
 Überlapp.: Überlappungszeitraum
 Kapa.: Kapazität der Station bzgl. der Bauteilart

 Auslastung < Kapazität
 Auslastung = Kapazität
 Auslastung > Kapazität

Im Beispiel aus Bild 4.16 ist zu erkennen, dass der kritische Punkt mit 6 Bauteilvarianten nur für einen Monat existiert. Hier könnten Investitionen reduziert oder sogar vermieden werden, indem eine der Bauteilvarianten des Monats Juli eingespart wird. Hierzu könnten der EOP (engl.: end of production, Ende der Serienproduktion) für Produkte mit der Bauteilvariante V2 vorgezogen oder der SOP (engl.: start of production, Start der Serienproduktion) für Produkte mit den Bauteilvarianten V6 oder V7 verschoben werden. Genau diese Transparenz für eine Entscheidung soll mithilfe des Diagramms geschaffen werden.

Aufgrund der Mehrfachverwendung von Bauteilvarianten in Produktvarianten lässt sich auf Basis der erläuterten Auswertungen nicht trivial entscheiden, welche Produktvarianten aus der Linie gestrichen werden müssen. Es kann sein, dass die Streichung einer einzelnen Produktvariante keinen Effekt an der betroffenen Stelle hat.

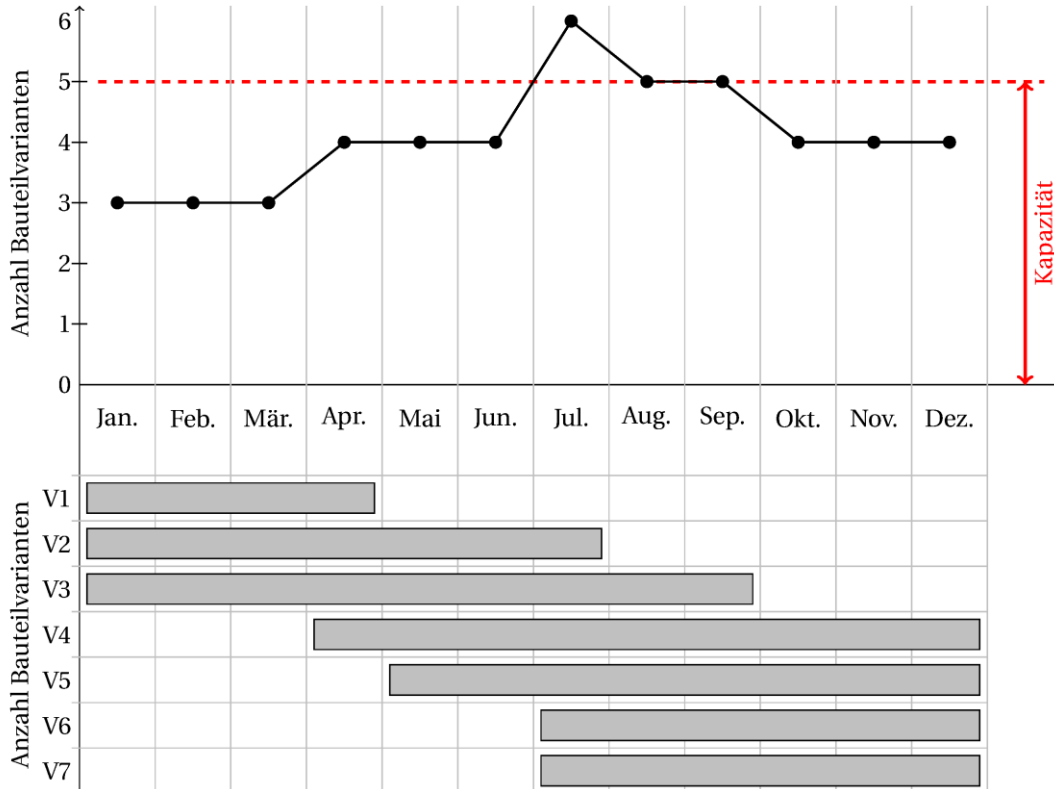


Bild 4.16: Beispielter zeitlicher Verlauf der Bauteilvariantenauslastung einer Station
Exemplary chronological sequence of part occupancy for a station

4.4.5 Analyse von Exklusivbauteilvarianten

Um dieses Dilemma zu lösen, wurde die Auswertung der Exklusivteile geschaffen. Hierin wird untersucht, welche Produktvariante oder welche Gruppe von Produktvarianten welche Bauteilvarianten exklusiv verwendet. Eine Bauteilvariante einer Ölwanne, die nur für Fahrzeuge mit Allradantrieb benötigt wird, ist ein Beispiel für ein solches Exklusivteil.

Um die Exklusivteile zu ermitteln werden die in einer Linie hergestellten Produkte in zwei Gruppen unterteilt. Eine Gruppe enthält die Produktvarianten, deren Exklusivteile ermittelt werden sollen. Die andere Gruppe sind alle weiteren Produktvarianten, die nicht schon in der ersten Gruppe enthalten sind. Es ergeben sich zwei Mengen von Produktvarianten.

Für jede der beiden Mengen von Produktvarianten werden anschließend alle zugehörigen Bauteilvarianten ermittelt. Es ergeben sich zwei Mengen an Bauteilvarianten. Schematisch sind die Mengen der Bauteilvarianten als zwei Kreisflächen in Bild 4.17

dargestellt. Anders als die Mengen der Produktvarianten können sich diese Mengen überschneiden (vgl. Schnittfläche der Kreisflächen in Bild 4.17). Dies ist dann der Fall, wenn eine Bauteilvariante sowohl für Menge 1 als auch für die Menge 2 benötigt wird. Die Konsequenz hiervon ist: Werden die Produktvarianten aus der Menge 1 aus der Linienbelegung entfernt, entfallen diese gemeinsam genutzten Bauteilvarianten nicht, eben weil sie noch für die Produkte in Menge 2 benötigt werden.

Die Menge der Bauteilvarianten, die in der Menge 1, nicht aber in der Menge 2 enthalten sind, sind die Exklusivteile (vgl. rechter Teil der rechten Kreisfläche in Bild 4.17). Gelingt es also die Produktvarianten der Gruppe 1 aus der Linienbelegung zu entfernen, so entfallen alle darin enthaltenen Bauteilvarianten auf der betrachteten Linie.

Mittels einer Untersuchung der Exklusivteile lässt sich systematisch ermitteln, wie groß die Reduzierung von Bauteilvarianten ausfällt, wenn bestimmte Produktvarianten oder Gruppen von Produktvarianten aus der Linienbelegung gestrichen werden. Außerdem lässt sich bestimmen, ob auch an den kritischen Stationen Bauteilvarianten entfallen oder nicht. Ebenso kann diese Auswertung genutzt werden, um zu analysieren, welche neuen Bauteile hinzukommen, wenn die Linienbelegung um ein oder mehrere Produktvarianten erweitert wird.

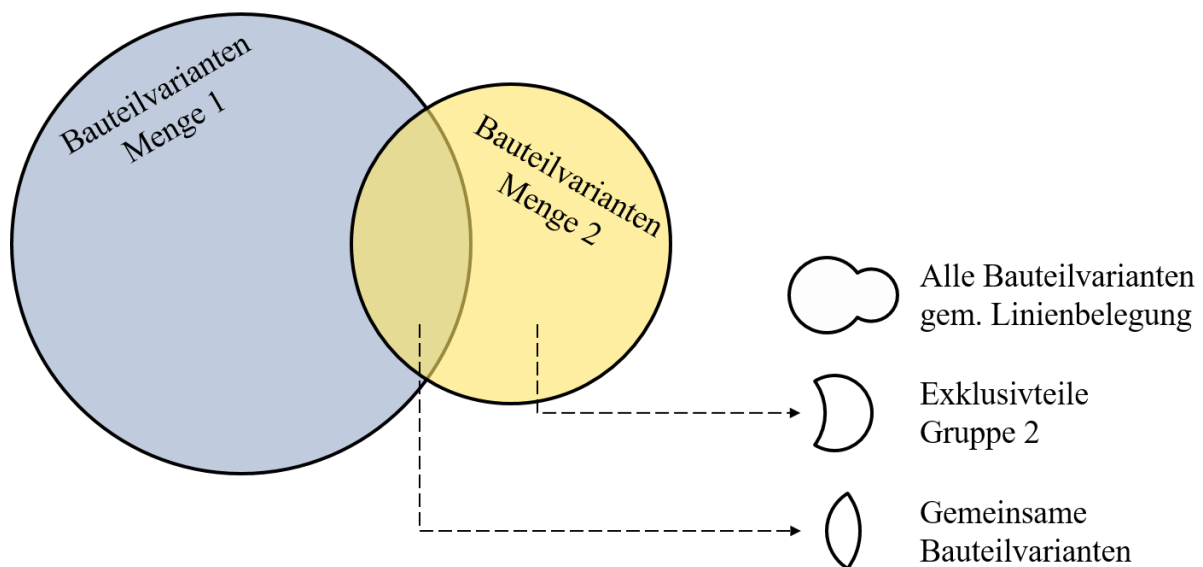


Bild 4.17: Mengen der Bauteilvarianten für zwei Mengen von Produktvarianten (Bild in Arbeit)

Amounts of part variety for two amounts of product variants

4.4.6 Auswertung und Abgleich von Anforderungen und Fähigkeiten

Mithilfe des Modells lassen sich ebenfalls die Anforderungen und Fähigkeiten vergleichen und auswerten. Aufgrund der Vielfalt von Anforderungen und Fähigkeiten ist eine grafische Darstellung jedoch wenig hilfreich. Vielmehr hat sich hierzu eine tabellarische Darstellung bewährt.

In dieser Tabelle, ein Beispiel ist in Tabelle 4.4 dargestellt, werden alle Stationen und Linie aufgeführt. Für jede Station sind die Bauteilarten und Bauteilvarianten angegeben, die gemäß der Linienbelegung in der Linie zu verbauen sind. Außerdem sind die betroffenen Produktvarianten angeführt. Im Anschluss folgen die bereits bekannten Informationen über die Anzahl der Bauteilvarianten und die Kapazität der Station.

Der Prozesstyp (vgl. Abschnitt 4.2.2) hilft bei der Zuordnung von Anforderungen und Fähigkeiten. Sind diese zugeordnet wird abgeglichen, ob ein Konflikt vorliegt oder nicht und das Ergebnis wird in der letzten Spalte ausgegeben.

Tabelle 4.4 fasst alle wichtigen Informationen zusammen. Für alle Objekte des Modells kann sie sehr lang werden, jedoch kann sie mithilfe von Filtern und der Sortierfunktion als effektives Werkzeug genutzt werden.

4.5 Gesamtbetrachtung zum Modell zur Integrationsplanung

Summary of the model for integration planning

Für die Befüllung des Modells ist es erforderlich, dass bereits Daten zum Montagesystem vorhanden sind, weshalb es vorwiegend für die Integrationsplanung geeignet ist. Welche der Eingangsdaten aus Bild 2.16 an welcher Stelle ins Modell eingepflegt werden müssen ist in Bild 4.18 darstellt.

Mithilfe der Stückliste, der Variantenmatrix und dem Variantenbaum (Nr. 1 bis 3 in Bild 4.18) wird das Partialmodell Produkt befüllt. Die Linienbelegung (Nr. 4 in Bild 4.18) und die Prozess und Ressourcenlisten (Nr. 5 in Bild 4.18) bilden die Partialmodelle Prozess und Ressource. Somit lässt sich das Basis-Datenmodell befüllen und verknüpfen. Die Anforderungen und Fähigkeiten sowie Kapazität (Nr. 6 und 7 in Bild 4.18) sind implizites Planerwissen und müssen somit vom Planer befüllt werden.

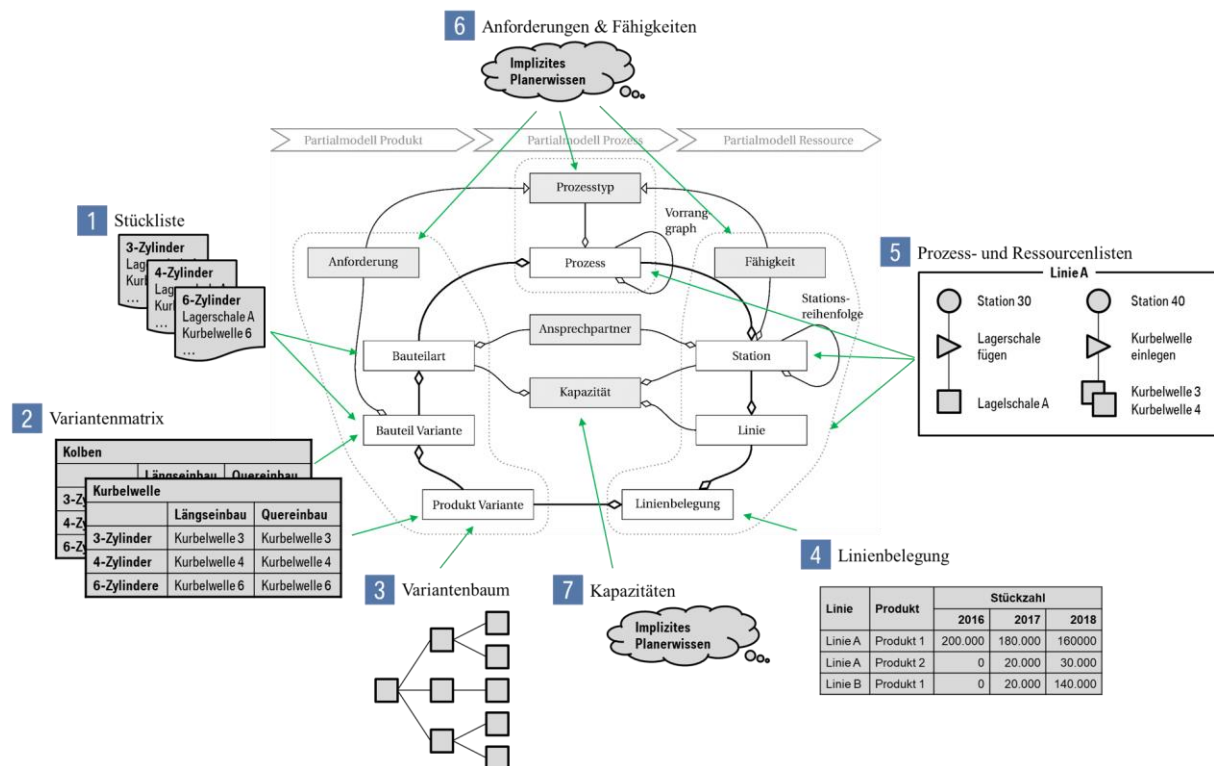


Bild 4.18: Zuordnung der Eingangsdaten zu den Klassen des Datenmodells

Allocation of input data to the classes of the data model

Nach der Befüllung des Datenmodells soll diesem im Rahmen der Integrationsplanung eingesetzt werden. Hierzu wurden in Abschnitt 4.4 verschiedene Möglichkeiten der Auswertung entwickelt und erläutert. In Bild 4.19 sind diese Auswertungen den verschiedenen Stellhebeln des fähigkeitsbasierten Planungsprozesses zugeordnet.

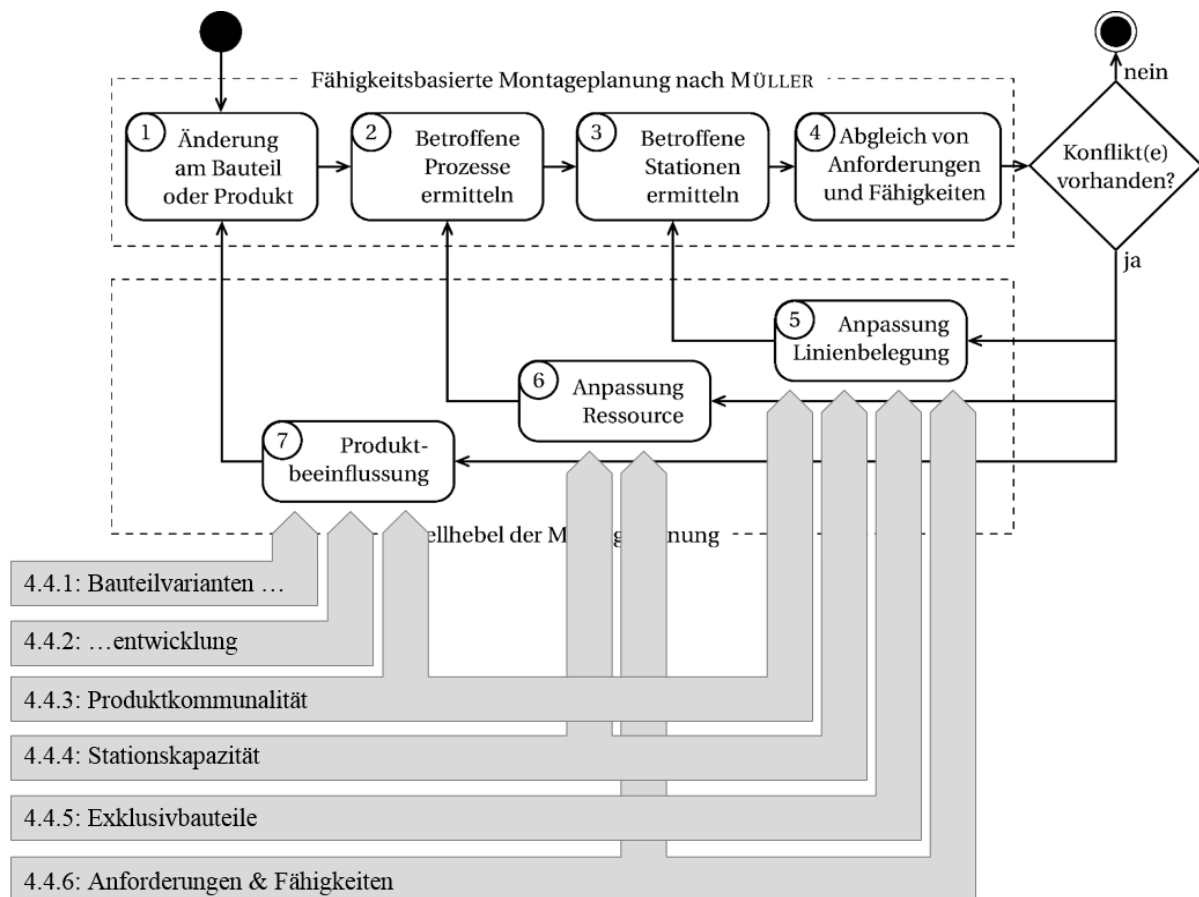


Bild 4.19: Zuordnung der Auswertungen zu den Stellhebeln des Planungsprozesses

Allocation of the evaluations to the steps of the options of action

Im Folgenden soll untersucht werden, wie das Modell die Anforderungen aus Abschnitt 2.5 erfüllt.

1. Verknüpfung Produkt, Prozess und Ressource

Die Verknüpfung von Produkt, Prozess und Ressource ist bei allen untersuchten Modellen vorhanden und kann weitestgehend übernommen werden. Alle Autoren verwenden objektorientierte Ansätze, so auch der Ansatz dieser Arbeit. Anforderung 1 aus Abschnitt 2.5, die Verknüpfung von Produkt, Prozess und Ressource, ist somit erfüllt.

2. Logik über Verknüpfungen / Redundanzfreiheit

Für die Verbindung der Elemente wurden entweder Regeln oder Aggregationen verwendet. Anstatt die Instanzen zu kopieren werden sie miteinander verknüpft. Ein großer Vorteil hiervon ist, dass Änderungen an einer Stelle des Modells sofort im ganzen Modell verfügbar sind. Zu jeder Zeit wird im gesamten Modell mit den aktuellsten Daten gearbeitet. Zudem wird die Datenpflege erheblich erleichtert, da in jedem Partialmodell unabhängig und somit parallel gearbeitet werden kann. Insbesondere in einem

dynamischen Umfeld wie der Montageplanung und bei sich häufig ändernden Produkten ist dies sehr hilfreich. Es lassen sich leicht neue Bauteil- oder Produktvarianten, Prozesse oder Stationen hinzufügen und verknüpfen. Die Auswirkungen davon auf das Gesamtsystem werden sofort ersichtlich. Dies greift Anforderung 2 aus Abschnitt 2.5 auf.

3. Integration des Baukastenansatzes

Der Baukastenansatz wurde von BRUNNER bereits durch die Integration der Klasse Bauteilart umgesetzt. Dieser Ansatz wird für diese Arbeit übernommen, um Anforderung 3 zu erfüllen.

4. Regelbasierte Verknüpfung / Variantenvielfalt

Ebenfalls auf dem Grundgedanken des Produktbaukastens basieren die aufgestellten Variantenregeln, die zur Herstellung der zahlreichen Verknüpfungen von Bauteilvariante, Produktvariante und Linienbelegung dienen. JONAS führt hierzu ein zahlenbasiertes System ein. Dieses wurde übernommen und in eine Kombination aus Buchstaben und Zahlen überführt, um somit einen sprechenden Code zu erhalten. Ein solcher Code lässt sich auch durch den Menschen lesen und übersetzen. Anforderung 4 ist somit eingehalten.

5. Anforderungen/Fähigkeiten

Im erweiterten Datenmodell werden Klassen für die Anforderungen und Fähigkeiten eingeführt. Diese waren bei allen Autoren im Ansatz vorhanden. Der Ansatz von BRUNNER war am weitesten fortgeschritten und konnte daher mit der Erweiterung des Prozesstyps übernommen werden, um Anforderung 5 zu erfüllen. In dieser Arbeit werden lediglich die Begriffe Anforderungen und Fähigkeiten verwendet.

6. Montagesystemverbund

Die Klasse *Linie* dient dazu, dass auch mehrere Linien modelliert werden können. Für Anforderung 6 konnte der Ansatz von NEUHAUSEN in das Modell integriert werden.

7. Dynamik

Die Linien unterliegen einer Dynamik im Hinblick auf die Zuordnung von Produkten. Daher wurde in dieser Arbeit die Klasse Linienbelegung eingeführt, in der für jedes Produkt ein Start- und ein Endzeitpunkt definiert und somit eine Dynamik dargestellt werden kann. Anforderung 7 ist somit ebenfalls erfüllt.

8. Kommunikation.

Der Aspekt der Erleichterung der Kommunikation aus Anforderung 8 konnte durch die Integration der Klasse *Ansprechpartner* in das erweiterte Datenmodell umgesetzt werden.

9. Analysen

Analysen wurden von einigen Autoren bezogen auf ihren jeweiligen Schwerpunkt thematisiert. Umfassende Auswertungen wurden jedoch nicht vorgestellt. In Abschnitt 4.4 wurde daher gezeigt, wie das Modell genutzt werden kann, um für die Montageplanung relevante Auswertungen zu generieren. Dies führt zur Erfüllung von Anforderung 9 aus Abschnitt 2.5.

Der Pflegeaufwand des Systems wird mittels der durchgängigen Verwendung von Verknüpfungen reduziert. Hinzu kommt, dass sich die Partialmodelle zunächst auch unabhängig voneinander und ohne Kenntnis über die anderen Partialmodelle erstellen lassen. Das Partialmodell Produkt lässt sich komplett befüllen, ohne Kenntnis über die Prozesse oder die Montagestationen. Ebenso können Stationen und Linien angelegt werden, ohne dass bereits alle Informationen über alle möglichen Produkte oder eine konkrete Linienbelegung vorliegen muss.

10. Fehleridentifikation

Zusätzlich ist es mittels der Verknüpfungen und der Wahl eines ringförmigen Modells möglich, Fehler oder Inkonsistenzen bei der Befüllung festzustellen. Wird beispielsweise eine Produktvariante in keiner Linienbelegung gefunden, so sollte geprüft werden, ob dies tatsächlich korrekt ist. Es wurde ein Produkt spezifiziert, dass niemals gebaut wird. Da die Linienbelegung verschiedene Zeitpunkte darstellen kann, werden bei dieser Betrachtung auch bereits ausgelaufene oder zukünftig geplante Motoren korrekt berücksichtigt.

Bauteilvarianten, die zu keinem Produkt gehören, lassen sich ebenfalls identifizieren. Mit einer Grundkenntnis über den Aufbau des Produktes lassen sich auch semantische Fehler in der Stückliste finden. So muss beispielsweise jeder Motor über genau eine Kurbelgehäuse-Bauteilvariante verfügen. Mittels des Partialmodells Produkt lassen sich alle Produktvarianten finden, die entweder keine oder mehr als eine Bauteilvariante des Kurbelgehäuses enthalten.

Wenn eine Bauteilart keinem Prozess zugewiesen ist oder gewisse Prozesse keiner Station zugewiesen sind, liegt häufig ebenfalls ein Fehler vor. Dies kann passieren, wenn neue Bauteilarten zum Produkt hinzukommen, aber an den betroffenen Standorten nicht die notwendigen Prozesse und Ressourcen eingerichtet wurden.

Fähigkeiten, zu denen keine Anforderungen existieren oder Anforderungen, zu denen keine Fähigkeiten existieren, lassen sich ebenfalls leicht finden. Auch dies passiert in der Regel durch die Dynamik des Systems. Zu irgendeinem Zeitpunkt wird eine Anforderung oder eine Fähigkeit hinzugefügt, ohne dass die entsprechende Gegenseite auch erweitert wird.

Die genannten Beispiele von Fehlern können mit den betroffenen Ansprechpartnern thematisiert werden, bevor der Planer mit der Ausgestaltung von Alternativen beginnt.

Der Planungsprozess wird somit erleichtert. Anforderung 10 aus Abschnitt 2.5 ist erfüllt.

Es werden nur die Klassen ohne Attribute und Methoden definiert, da letztere stark vom Anwendungsfall abhängen. Im Rahmen der Implementierung (vgl. Abschnitt 5.1) werden die Klassen um Attribute erweitert. Ein entscheidendes Merkmal des Modells sind die Beziehungen der Klassen zueinander.

Das Datenmodell orientiert sich am Stand der Technik der Modellierung, es kombiniert, modifiziert und erweitert dessen Modelle jedoch an den notwendigen Stellen, um die in Abschnitt 2.5 definierten Anforderungen zu erfüllen. Für die Zuordnung von Bauteilvarianten und Produkten zu Linien und Stationen wird das Basis-Datenmodell geschaffen. Mit diesem Basis-Datenmodell lässt sich noch keine Aussagen bezüglich der Montierbarkeit von Produkten treffen. Um diese ermitteln zu können wurde es um einige Klassen erweitert, die insbesondere die Restriktionen des Montagesystems abbilden. Das sich ergebende Gesamt-Datenmodell ist in Bild 4.13 dargestellt.

5 Implementierung und Auswertungen anhand der Anwendungsbeispiele

Implementation and evaluation using the example of ballpoint pen and engine assembly

In diesem Kapitel werden zunächst verschiedene Optionen zur Implementierung des Modells betrachtet und eine ausgewählt. Anschließend wird das UML-Gesamtmodell aus Bild 4.13 um Attribute erweitert und somit vervollständigt und für die beiden Anwendungsfälle adaptiert. Im Weiteren wird das Modell als Softwareprototyp implementiert (vgl. Abschnitt 5.1).

Mit dem implementierten Modell wird eine zweistufige Validierung durchgeführt. Im ersten Teil werden die verschiedenen Funktionsweisen des Modells anhand des einfachen, fiktiven Beispiels einer Kugelschreibermontage erläutert (vgl. Abschnitt 5.2). Im Anschluss wird das deutlich komplexere, aber reale Anwendungsbeispiel einer Großserien-Motormontage ins Modell eingefügt und diverse – ebenfalls aus der Realität stammende – Anwendungsszenarien untersucht (vgl. Abschnitt 5.3).

Diese Schritte dienen zur Validierung des Modells und zur Überprüfung auf die Anwendbarkeit in der Praxis. Die Ergebnisse der Validierung werden in Abschnitt 5.5 zusammengefasst.

5.1 Implementierung Datenmodell

Implementation of the data model

Das in Kapitel 4 geschaffene Modell und die beschriebenen Zusammenhänge sind gültig, unabhängig von einer gewählten Umsetzungsalternative wie bspw. einer Programmiersprache. Um das Modell in einem für die Praxis geeigneten Prototypen umzusetzen, muss es erweitert und ausdetailliert werden. Dazu müssen vor allem relevante Attribute in den Klassen ergänzt werden. Je nach gewählter Form und Programmiersprache der Umsetzung müssen weitere Klassen ergänzt werden.

5.1.1 Umsetzungsalternativen

Options for implementation

Das Modell ist umsetzungsneutral in UML beschrieben. Dies ermöglicht es, für die Implementierung verschiedene Alternativen zu betrachten. Die Integration in bestehende Systeme ist ebenso möglich wie eine eigenständige Implementierung.

Durch die Verwendung einer Datenbank und einer Datenbanksprache kann der Fokus somit auf das Wesentliche gelegt werden. Zudem ließe sich das Modell als Datenbank leicht implementieren und somit auch leicht auf den Bedarf des Anwenders anpassen. Für weitere Vorteile von Datenbanken sei auf Abschnitt 3.3 verwiesen.

Die Verwendung von Datenbanken und die Möglichkeit zur Verknüpfung der Daten kann eine redundante Datenhaltung vermeiden. Datenbanken lassen sich mittels Abfragen leicht und schnell durchsuchen. Durch verschiedene Bedingungen lassen sich die Suchergebnisse filtern und gruppieren. [SCHU04, S. 87-88; MOOS04, S. 228-230, S. 255-261; KREI19, S. 68-70]

Auswertungen können einmalig erstellt und jederzeit wieder abgerufen werden. Somit basieren die Ergebnisse der Auswertungen immer auf dem neuesten Datenstand. Wenn die Auswertungen mithilfe derselben Abfrage erstellt werden, lassen sich die Ergebnisse miteinander vergleichen und Trends und Entwicklungen ableiten.

Die im Folgenden vorgestellte Lösung lässt sich im Prinzip mit jeder beliebigen Datenbank umsetzen. Für diese Arbeit soll ein Datenbankmanagement-System ausgewählt werden, das ohne Server als Standalone-Variante betrieben werden kann. Dies bietet den Vorteil, dass sich die Ergebnisse der Arbeit ohne Server-Struktur nachbauen und nachvollziehen lassen. Ein solches System ist das Datenbankmanagement-System SQLite, das in Abschnitt 3.3 vorgestellt wurde. Alle benötigten Quellcodes sind in Listings in Kapitel 5 und im Anhang enthalten.

5.1.2 Detaillierung des UML-Datenmodells

Extended UML-data model

In diesem Abschnitt soll die Umsetzung des Modells in Form einer SQLite-basierten Datenbank vorgestellt werden. Jede Klasse des Klassendiagramms entspricht einer Tabelle dieser Datenbank.

Das UML-Modell aus Bild 4.13 stellt die verkürzte Form dar. Attribute und Methoden fehlen. Grund hierfür ist, dass Attribute sich je nach Anwendungsfall stark verändern können. Methoden innerhalb der Klassen werden bei der Umsetzung in Form einer Datenbank nicht benötigt.

Um das Modell basierend auf dem Klassendiagramm implementieren zu können, muss dieses zunächst vervollständigt werden. Der Name der Klassen wurde um ein Präfix ergänzt. Jede Klasse beginnt mit *tab_* oder *lnk_*. *tab_* deutet an, dass es sich hierbei um eine Tabelle handelt. Die Tabellen mit dem Präfix *lnk_* dienen zur Umsetzung der m:n-Verknüpfungen. Sie sollen für die weiteren Erläuterungen zunächst ignoriert werden. Später wird näher auf diese eingegangen.

Weiterhin wurden die Namen in den Plural gesetzt, da sich in jeder Tabelle mehrere Elemente dieser Klasse befinden. Aus der Klasse *Prozess* wurde somit *tab_prozesse*. Die Groß- und Kleinschreibung hat in SQL keine Auswirkungen.

Jede Tabelle erhält als Primärschlüssel das Attribut ID. Mit ihr lassen sich die enthaltenen Daten eindeutig beschreiben und leicht wiederfinden. Aus diesem Grund werden die meisten Verknüpfungen über die ID-hergestellt. Ausnahme bilden hier die Variantenregeln, die in Abschnitt 5.1.4 vorgestellt werden.

tab_bauteilvarianten: Die Klasse *tab_bauteilvarianten* enthält einen Fremdschlüssel *ID_Bauteilart*. In diesem Attribut wird für jedes Objekt gespeichert, zu welcher Bauteilart diese Bauteilvariante gehört. Über das Attribut *Bezeichnung* kann der Bauteilvariante ein Name zugewiesen werden. Wichtig ist hierbei, dass eine identische Bezeichnung auch eine identische Bauteilvariante bedeutet.

tab_bauteilarten: In der Klasse *Bauteilart* wird mittels des Attributes *Bezeichnung* ein Name vergeben. Das Attribut *Anstellungsart* spezifiziert, in welcher Form die Bauteilvarianten dieser Bauteilart an den Linien angestellt werden. Die Klasse weiterhin über eine ID mit einem *Ansprechpartner* verknüpft.

tab_produkte: Die Klasse *Produkte* besteht neben der *ID* nur aus einem *Variantencode* und einer *Bezeichnung*, mittels der dem Produkt ein Name gegeben werden kann (vgl. Abschnitte 5.3.2 und 5.1.4).

tab_prozesse: Prozesse beschreiben umsetzungsneutral, welche Vorgänge zur Montage des Produktes durchgeführt werden müssen. Somit enthält diese Klasse nur eine *Bezeichnung*. Jedoch ist der Prozess über zwei Verknüpfungs-Klassen mit den *Bauteilarten* und den *Stationen* verknüpft.

tab_stationen: Jeder Station kann eine *Nummer* zugewiesen werden, hierzu dient das entsprechende Attribut. Neben einer *Bezeichnung* – also einem Namen für die Station – kann diese mit einer zusätzlichen *Beschreibung* näher erläutert werden. Im Attribut *Reihenfolge* kann ebendiese für die Stationen festgelegt werden. Wie schon den Bauteilarten ist auch den Stationen ein *Ansprechpartner* zugewiesen. Da eine Montagelinie aus sehr vielen Stationen bestehen kann, können diese mithilfe der Eigenschaft *Gewerk* einer Gruppe von Montagestationen zugewiesen und somit gegliedert werden. *Typ* beschreibt, um welche Art einer Station es sich handelt. Hier kann beschrieben werden, ob es sich um eine automatische oder eine manuelle Station handelt.

tab_linien: Die Tabelle für die *Linien* hat zwei Eigenschaften: Die *Bezeichnung* gibt der Linie einen Namen. Mittels des Feldes *Standort* kann zudem beschrieben werden, wo sich diese Linie befindet. Insbesondere wenn sich das Produktionsnetzwerk auf mehrere Standorte verteilt ist dies eine Hilfe.

tab_linienbelegung: Die Tabelle für die *Linienbelegung* ist ein Hybrid aus Verknüpfungstabellen mit dem Präfix *Ink_* und Datentabellen mit dem Präfix *tab_*. Sie enthält neben den zu verknüpfenden Elementen noch einige weitere Informationen, weswegen sie im Datenmodell als Datentabelle geführt wird. Über die *ID_Linie* wird bestimmt, auf welche Linie sich diese Linienbelegung bezieht. Im Feld *Variantencode* wird eine Variantenregel (vgl. Abschnitte 5.3.2 und 5.1.4) spezifiziert. Alle Produkte, für die diese Variantenregel gilt, werden der Linie zugeordnet. Die drei Attribute *PVL* (Produktionsvorläufer), *SOP* und *EOP* (vgl. Abschnitt 2.1) enthalten zeitliche Angaben. Der *PVL* beschreibt den Zeitpunkt, ab dem die Linie in der Lage ist, diese Produktvarianten herzustellen (vgl. Abschnitt 2.1). Die Produkte werden von *SOP* bis *EOP* in Großserie

hergestellt. Diese Informationen sind wichtig, damit vor PVL und nach EOP die Bau-
teilvarianten der Produkte nicht der Linie zugewiesen werden.

tab_anforderungen: Wie in Abschnitt 4.3.1 beschrieben hängen die Klassen für die
Anforderungen, Prozesstypen und Fähigkeiten eng zusammen. Die Klasse *tab_An-*
forderungen enthält die Anforderung als Text. Grund hierfür ist, dass „blau“ eine An-
forderung sein kann, wenn bspw. die Farbe beschrieben werden soll. Ebenso kann
M8 als Anforderung zur Beschreibung eines metrischen ISO-Schraubengewindes die-
nen. Beide Beispiele zeigen die Notwendigkeit, hier eine Textvariable zu verwenden.
Die Anforderungen können auch ein Zahlenwert sein, dieser wird jedoch als Text ge-
speichert. Für die Auswertung wird er wieder wie ein Zahlenwert behandelt.

tab_prozesstypen: Wie in der Erläuterung für die Anforderungen beschrieben, kann
der Vergleich entweder auf Texten oder auf Zahlen basieren. Über Texte werden dis-
krete Werte miteinander vergleichen, über Zahlen entweder diskrete Werte oder Wer-
tebereiche. Eine ausführliche Beschreibung über diese beiden Arten von Abgleichen
hierzu folgt in Abschnitt 5.2.5. Um festzulegen, welche Art von Abgleich durchgeführt
werden soll, erhält der die Klassen *tab_prozesstypen* das Attribut *Abgleichart*. Zusätz-
lich kann noch eine Bezeichnung für den Prozesstyp vergeben werden.

tab_faehigkeiten: Das Gegenstück zur den (produktinduzierten) Anforderungen sind
die Fähigkeiten, die die Betriebsmittel in den Stationen besitzen. Als Attribute der *Fä-*
higkeit sind besonders *Fähigkeit_min* und *Fähigkeit_max* relevant. Diese legen bei
einem Wertebereich die Grenzen für diesen Bereich fest. Ein Beispiel hierfür findet
sich in Bild 5.7 in Abschnitt 5.2.5. Im Falle diskreter Werte können ein oder beide Fel-
der verwendet werden. Jeder *Station* können auch mehrere *Fähigkeiten* des gleichen
Prozesstyps zugeordnet werden. Wenn beim Vergleich diskreter Werte die Anlage
vier Fähigkeiten besitzt, bspw. metrische ISO-Gewinde M6, M8, M10 und M12, dann
werden zwei Objekte von Fähigkeiten erstellt, die jeweils zwei dieser Fähigkeiten be-
schreiben. Beim Abgleich diskreter Werte spielt es keine Rolle, welcher Wert in *Fä-*
higkeit_min und welcher in *Fähigkeit_max* geführt wird.

tab_personen: Die Klasse *tab_personen* enthält die Kontaktinformationen verschie-
dener Menschen, die entweder als Entwickler für bestimmte Bauteilarten verantwort-
lich sind, oder als Montageplaner die Gestaltung von Stationen verantworten. Für nä-
here Erläuterungen sei auf Abschnitt 4.3.3 verwiesen.

tab_kapazitaeten_stationen, tab_kapazitaeten_linien: Zur Erfassung der Kapazi-
täten wurden zwei Klassen geschaffen. Grund hierfür ist, dass eine Kapazität entwe-
der von einer *Station* oder von einer *Linie* abhängen kann (vgl. Abschnitt 4.3.2). Somit
entstehen zwei Beziehungen. Diese ließen sich auch in einer Klasse realisieren, je-
doch ist die Aufteilung auf zwei Klassen übersichtlicher und es lässt sich eindeutiger
interpretieren. Als Attribut enthalten beide Klassen neben den *IDs* der Beziehungs-
partner eine Integer-Variable, in der der Wert für die *Kapazität* abgelegt werden kann.

tab_zeitraume: In dieser Tabelle werden alle Zeitpunkte hinterlegt, für die eine Auswertung durchgeführt werden soll. Jeder Zeitpunkt muss als einzelner Eintrag in dieser Tabelle angelegt werden. Die Zeitpunkte sind frei wählbar und müssen keiner Regel oder Logik folgen. In der Regel sollen jedoch Verläufe über die Zeit gezeigt werden. Dafür bietet es sich an, äquidistante Abstände zwischen den Zeitpunkten zu wählen. Für die hier beschriebenen Fälle sind das Wochen oder Monate. Eine detailliertere Beschreibung, wie die Tabelle *Zeiträume* in die Abfragen eingebunden ist, findet sich in Abschnitt 5.2.4.

In Bild 5.1 sind alle Klassen und ihre Zusammenhänge dargestellt.

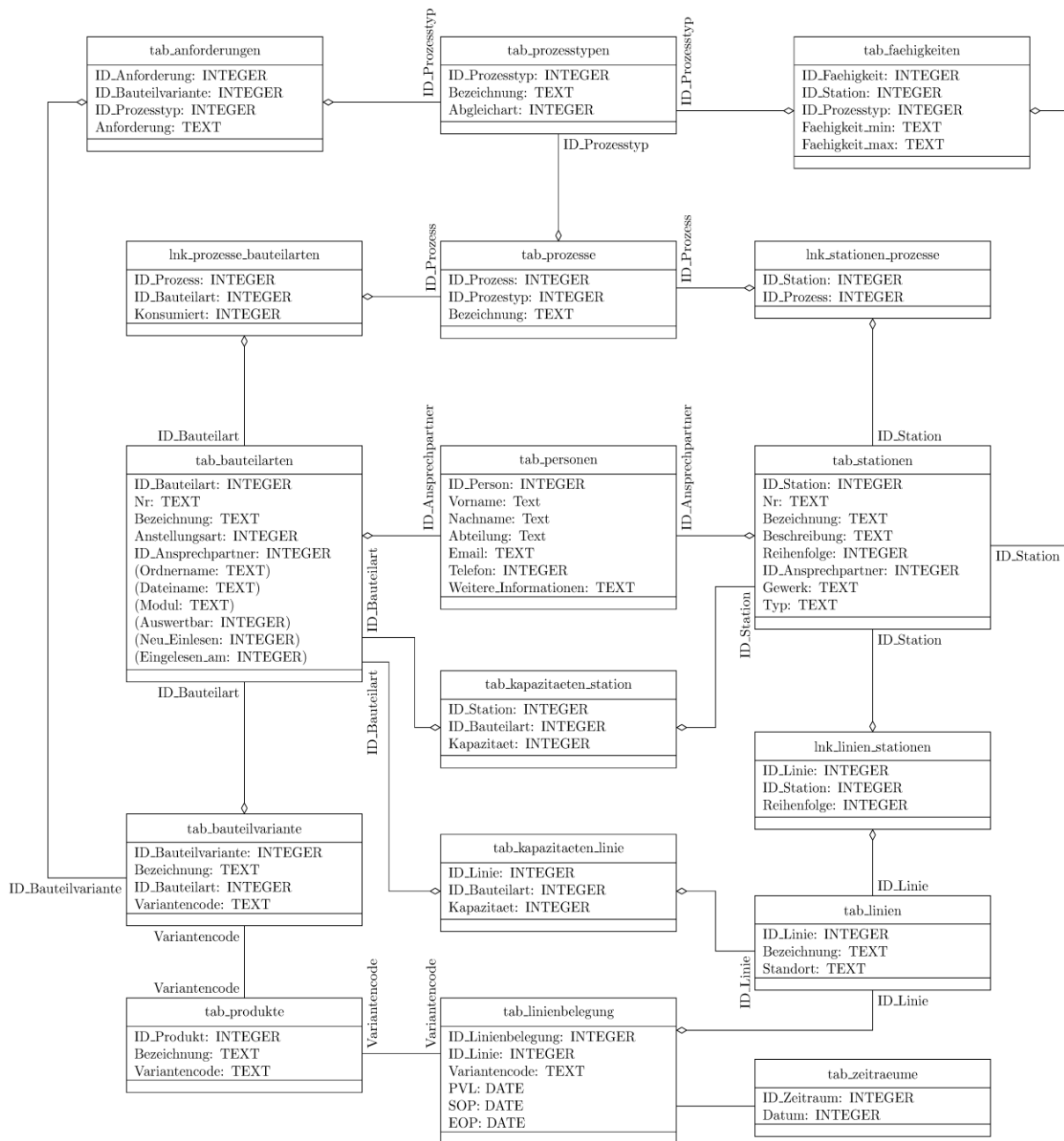


Bild 5.1: Ausführliches UML Klassendiagramm zur Implementierung
Extended UML class diagram for the implementation

5.1.3 Erstellung der Datenbank-Tabellen

Creation of the tables of the database

Die Attribute und somit die Spalten der Tabellen der Datenbank werden in Abschnitt 5.1.2 hergeleitet und in Bild 5.1 zusammengefasst. Aus diesen Informationen lässt sich die Struktur der Datenbank anlegen. Die Vorgehensweise hierzu wird in diesem Kapitel an einigen Beispielen erläutert.

Mithilfe des `CREATE TABLE` Befehls aus Abschnitt 3.3 können die benötigten Tabellen angelegt werden. In Listing 5.1 ist der Code für die Erstellung der Tabelle `tab_linien` als Beispiel aufgeführt. Nach diesem Schema werden alle Tabellen des Datenmodells erstellt. Der Code zur Erstellung der weiteren Tabellen ist im Anhang (Kapitel 8) aufgeführt.

```
1 CREATE TABLE IF NOT EXISTS tab_linien (  
2 ID_Linie INTEGER PRIMARY KEY,  
3 Bezeichnung TEXT,  
4 Standort TEXT  
5 );
```

Listing 5.1: Erstellung einer Tabelle am Beispiel von `tab_linien`

Creation of a table at the example of tab_linien

Eine besondere Art von Tabellen stellen die Verknüpfungstabellen dar. Diese sind im erweiterten Datenmodell aus Bild 4.13 nicht enthalten. Sie dienen dazu, in der Implementierung die m:n Verknüpfungen herzustellen. Zur besseren Übersicht haben diese Tabellen als Präfix `Ink_`. Danach folgen die Namen der beiden zu verknüpfenden Tabellen, getrennt durch einen Unterstrich. Die Tabelle zur Verknüpfung der Prozesse und der Bauteilarten heißt demnach `Ink_prozesse_bauteilarten`. Weitere Verknüpfungstabellen sind die `Ink_linien_stationen` und die `Ink_stationen_prozesse`.

```
1 CREATE TABLE IF NOT EXISTS lnk_prozesse_bauteilarten (  
2 ID_Prozess INTEGER,  
3 ID_Bauteilart INTEGER,  
4 Konsumiert INTEGER,  
5 PRIMARY KEY(ID_Prozess, ID_Bauteilart)  
6 );
```

Listing 5.2: Erstellung einer Verknüpfungstabelle am Beispiel von `Ink_prozesse_bauteilarten`

Creation of a link-table at the example of Ink_prozesse_bauteilarten

Als Parameter enthalten diese Tabellen die IDs der beiden zu verknüpfenden Tabellen und falls erforderlich weitere Parameter. Besonders ist hier, dass die Instanzen dieser Tabellen keine eigenen IDs haben. Als Primärschlüssel wird die Kombination der IDs der beiden zu verknüpfenden Klassen verwendet. Listing 5.2 zeigt am Beispiel von

Ink_prozesse_bauteilarten den Befehl zur Erstellung einer solchen Tabelle. Die vollständigen Befehle für die anderen Verknüpfungstabellen sind im Anhang (Kapitel 8) zu finden.

5.1.4 Umsetzung Variantenregeln

Implementation of variant rules

Als zweite Option für die m:n-Verknüpfung – insbesondere für große m und n – wird die Variantenregel genutzt. Hier werden nicht alle Verknüpfungen einzeln hergestellt, sondern Regeln definiert, anhand derer zwei Gruppen von Objekten miteinander verknüpft werden.

Im Abschnitt 4.2.1 wird diese Regel generisch beschrieben. In diesem Abschnitt wird ein konkretes, implementierbares Beispiel für eine solche regelbasierte Verknüpfung erstellt.

Die Verknüpfung mittels Variantenregel basiert auf dem Vergleich zweier Zeichenketten. Die Multiplizität der Verknüpfung wird über Wildcards hergestellt. Eine Wildcard ist ein Platzhalter, der für ein oder mehrere beliebige Zeichen stehen kann.

In SQL stehen zwei Wildcards zur Verfügung: Das Prozentzeichen (%) steht für beliebig viele – auch null – Zeichen und der Unterstrich (_) für genau ein beliebiges Zeichen. Zum Abgleich zweier Zeichenketten kann in SQL der `LIKE` Befehl verwendet werden. Dessen Verwendung ist in Listing 5.3 dargestellt. Der gezeigte Code liefert als Ergebnis 1, der Vergleich ist also positiv. Tabelle 5.1 zeigt zum besseren Verständnis das Ergebnis einiger Abfragen mittels `LIKE` Befehl. Die Erläuterung des im Beispiel verwendeten Variantencodes des Baukastenmotors folgt in Abschnitt 5.3.2.

```
SELECT „B38A15U0“ LIKE „B38%“;
```

Listing 5.3: Beispiel für den Abgleich zweier Zeichenketten mittels `LIKE` Befehl

Example of the string comparison using the `LIKE` command

Die Variantenregel muss sich am Anwendungsfall orientieren, da dieser vorgibt, nach welchen Kriterien Produkte zu unterscheiden sind. Für den fiktiven (vgl. Abschnitt 5.2) und den realen (vgl. Abschnitt 5.3) Anwendungsfall werden daher jeweils eigene Variantenregeln aufgestellt.

Der Nachteil bei der Verwendung einer solchen Zeichenkette als Variantencode ist, dass mit steigender Varianz der Variantencode zunehmend länger wird. Produktmerkmale, die kombinierbar sind, werden als eigene Zeichen dargestellt und verlängern somit die Zeichenkette.

Tabelle 5.1: Beispiele für die Verwendung des `LIKE` Befehls mit Wildcards*Example of the usage of the `LIKE` statement with wildcards*

Abfrage	Ergebnis	Beschreibung
<code>SELECT "B38A15U0" LIKE "B38%"</code>	1 (wahr)	Zur Auswahl aller Motoren, deren Variantencode mit <i>B38</i> anfängt, d.h. alle Baukasten 3-Zylinder Ottomotoren
<code>SELECT "B38A15U0" LIKE "B_8%"</code>	1 (wahr)	Zur Auswahl aller Motoren mit <i>B</i> an erster und <i>8</i> an dritter Stelle: Alle Baukasten Ottomotoren
<code>SELECT "B38A15U0" LIKE "B48%"</code>	0 (falsch)	Die Abfrage ist nicht wahr, da <i>B48%</i> alle Baukasten Vierzylinder Ottomotoren beschreibt, der <i>B38A15U0</i> aber nicht in diese Menge fällt

Im praktischen Anwendungsfall werden die Motorcodes als Variantenregeln verwendet. Eine einfache Abfrage mithilfe der Variantenregel ist in Listing 5.4 dargestellt. In dieser Abfrage werden die Bezeichnungen aller Bauteilvarianten ausgegeben, bei denen der Variantencode mit der Zeichenkette *B48A* beginnt. Für nähere Details zum Motorcode sei auf Abschnitt 5.3.2 verwiesen.

```

1  SELECT
2  Bezeichnung AS Bauteilvariante
3  FROM
4  tab_bauteilvarianten
5  WHERE
6  Variantencode LIKE "B48A%";

```

Listing 5.4: Einfache Abfrage mithilfe der Variantenregel*Simple query using the variant ode*

Die Variantenregel wird im Modell verwendet, um die Beziehung zwischen den Klassen Bauteilvariante, Produkt und Linienbelegung herzustellen. Dazu wird allen Bauteilvarianten ein Variantencode, meist unter Verwendung von Wildcards zugewiesen. Die Produktvarianten dürfen keine Wildcards enthalten, da diese ein Produkt eindeutig anhand aller seiner Eigenschaften beschreiben müssen. In der Linienbelegung dürfen Wildcards verwendet werden, um viele Produkte einer Linie zuzuordnen. Die Zuordnung von Produkten zu Linien mittels der Linienbelegung ist in Listing 5.5 dargestellt. Tabelle 5.2 zeigt ein beispielhaftes Ergebnis dieser Abfrage.

```

1  SELECT
2  tab_linien.Bezeichnung AS Linie,
3  tab_produkte.Bezeichnung AS Produkt,
4  tab_linienbelegung.Variantencode AS Variantencode_Linienbelegung,
5  tab_produkte.Variantencode AS Variantencode_Produkt
6  FROM tab_linien
7  LEFT JOIN tab_linienbelegung USING (ID_Linie)
8  LEFT JOIN tab_produkte ON (
9  tab_produkte.Variantencode LIKE tab_linienbelegung.Variantencode);

```

Listing 5.5: Umsetzung der Variantenregel am Beispiel der Linienbelegung

Implementation of the variant code at the example of the line allocation

Durch die beschriebene Implementierung konnte der gewünschte Effekt erzielt werden: Es lassen sich leicht neue Produkte hinzufügen und überflüssige Produkt entfernen. Zu Beginn der Befüllung des Modells müssen nicht alle Produktvarianten bekannt sein. Mit nur einer einzigen Produktvariante lässt sich das Modell bereits nutzen.

Somit kann mit verschiedenen Linienbelegungsszenarien experimentiert werden, um den optimalen Betriebspunkt für einen Montagesystemverbund zu finden.

Tabelle 5.2: Beispielhaftes Ergebnis der Abfrage aus Listing 5.5

Example result from Listing 5.5

Linie	Produkt	Variantencode_ Linienbelegung	Variantencode_ Produkt
Linie A	B38A12U0 – EU – LL	B38A12U0:___	B38A12U0:1L
Linie A	B38A12U0 – EU – RL	B38A12U0:___	B38A12U0:1R
Linie A	B38A15M0 – EU - LL	B38A12M0:___	B38A12U0:1L
Linie A	B38A15M0 – EU - RL	B38A12M0:___	B38A12U0:1R
Linie A	B38A15M0 – US - LL	B38A12M0:___	B38A12U0:2L
...

5.2 Funktionale Validierung anhand einer fiktiven Kugelschreiber-Fabrik

Functional validation on the basis of the fictional ballpoint pen factory

Die funktionale Validierung soll zeigen, dass das Modell und der darauf basierende Prototyp die gestellten Anforderungen erfüllen. Weiterhin soll das bisher abstrakte Modell anhand konkreter Praxisbeispiele tiefer erläutert werden.

Die Zusammenhänge in der Montageplanung für Baukastenmotoren sind vielfältig und komplex. Um zunächst tieferes Verständnis für das Modell und dessen Funktionsweise zu erlangen, soll daher ein einfacheres, jedoch fiktives Beispiel herangezogen werden: Eine Kugelschreiber-Fabrik. Im Beispiel treten jedoch dieselben Herausforderungen auf, wie im realen Praxis-Beispiel.

5.2.1 Einführung in das fiktive Beispiel der Kugelschreiber-Fabrik

Introduction to the fictional example of the ballpoint pen factory

Das Produkt Kugelschreiber besteht aus vier Bauteilarten: Das Oberteil und das Unterteil bilden das Gehäuse. Feder und Mine bilden das Innenleben (vgl. Bild 5.2).

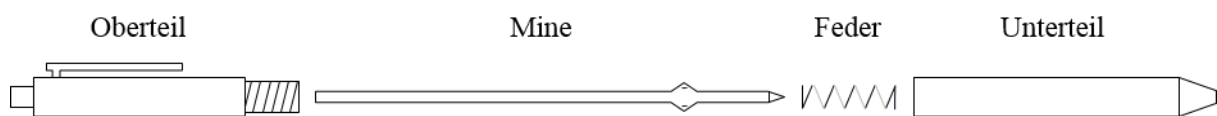


Bild 5.2: Aufbau des fiktiven Produktes Kugelschreiber

Design of the fictional product ballpoint pen

Zu jeder Bauteilart gibt es mindestens eine Bauteilvariante. Die Gehäuse, bestehend aus Unter- und Oberteil, sind in zwei verschiedenen Farben verfügbar. Mischungen aus verschiedenfarbigen Unter- und Oberteilen werden nicht angeboten. Die Mine ist blau, die Feder normal. Somit ergeben sich insgesamt zwei Produkte. In Bild 5.3 sind die Produktvarianten als Variantenbaum (vgl. Abschnitt 2.3) dargestellt. Jede Bauteilart ist als eigene Spalte aufgeführt. Hinter jeder Kombination von Bauteilvarianten ist ein Variantencode (vgl. Bild 5.5) aufgeführt. Weiterhin sind die Produktvarianten fortlaufend nummeriert. (in Anlehnung an [SCHU97, SCHU05, ZAGE06])

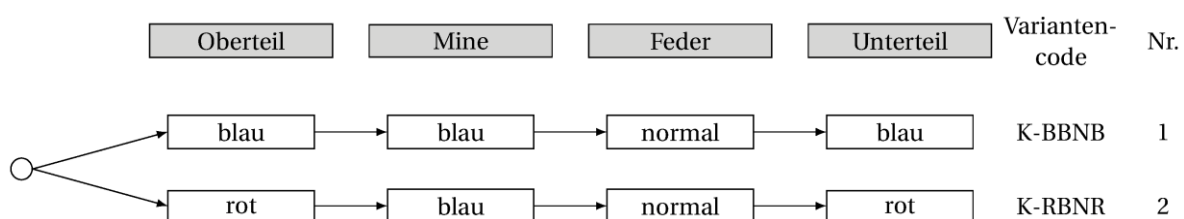


Bild 5.3: Variantenbaum der Produktvielfalt des Kugelschreiber-Beispiels

Variant tree for the product range of the ballpoint pen example

Ein Kugelschreiber wird im Szenario in fünf Arbeitsschritten montiert: Zunächst wird das Oberteil aufgelegt und anschließend die Mine in dieses eingefügt. Im nächsten Schritt wird die Feder auf die Mine gefädelt und das Unterteil gefügt. Zum Schluss wird das Unterteil verschraubt. Im Modell wird jeder Arbeitsschritt als Instanz der Klasse *Prozess* angelegt.

Die Montage findet in zwei verschiedenen Linien statt. Diese werden gemäß ihren Standorten *Linie München* und *Linie Saarbrücken* genannt. Jede dieser Linien hat drei Stationen. Diese haben eine vierstellige Bezeichnung: Der erste Buchstabe gibt an, welche Linie gemeint ist. M steht hierbei für München, S für Saarbrücken. Danach folgt eine dreistellige Zahl. Damit später Stationen ergänzt werden können, werden die Stationen in Zehnerschritten durchnummeriert.

Die erste Station heißt demnach M010 bzw. S010: Hier wird das Oberteil aufgelegt. Eine Kamera mit Farbsensor prüft, ob tatsächlich das richtige Oberteil aufgelegt wurde. Anschließend wird das Oberteil zur nächsten Station transportiert.

In Station M020 / S020 wird die Mine eingesetzt und die Feder auf die Mine gefädelt. In der nächsten Station, M030 / S030, wird das Unterteil gefügt und es wird optisch kontrolliert, ob das richtige Unterteil verwendet wurde. Zum Schluss wird das Unterteil an das Oberteil geschraubt. Bild 5.4 zeigt diese Zusammenhänge in einem vereinfachten Objektdiagramm. Die Verknüpfung zwischen Linienbelegung, Produkt und Bauteilvariante erfolgt wie in Abschnitt 5.1.4 beschrieben über den Variantencode. Diese Zusammenhänge sind in Bild 5.3 dargestellt und werden im Objektdiagramm nicht wiederholt.

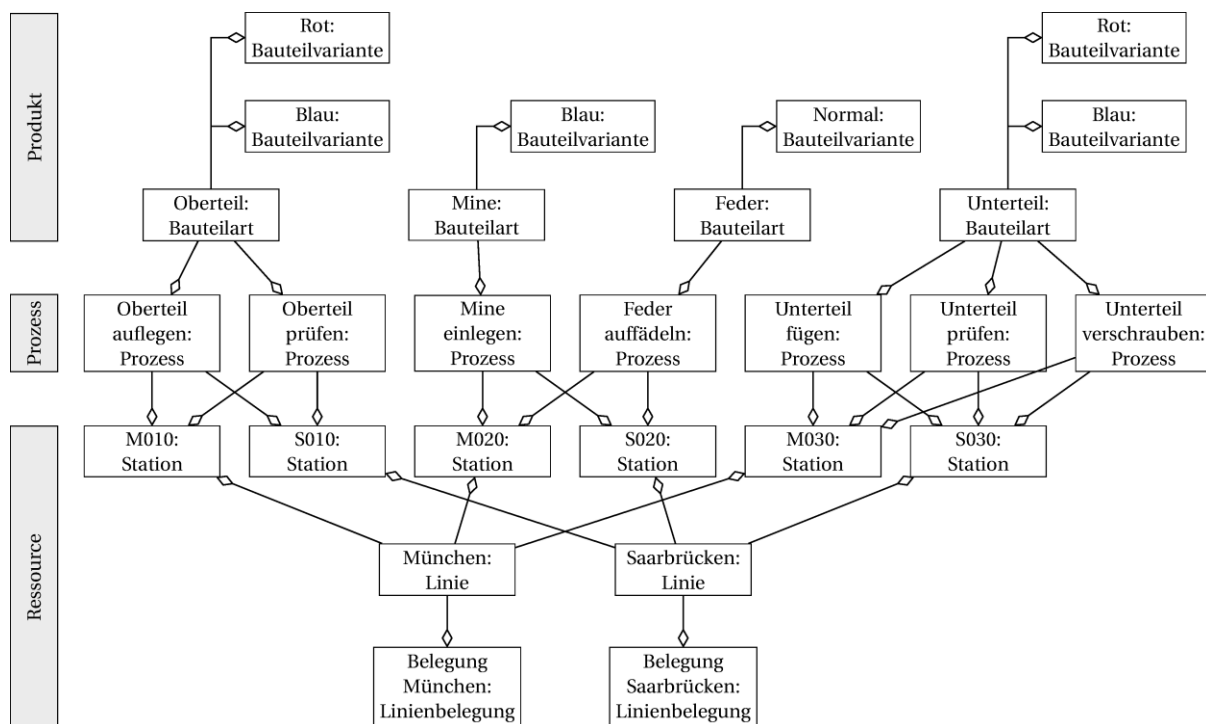


Bild 5.4: Ausgangssituation im Kugelschreiber-Szenario
Initial situation at the ballpoint pen scenario

Idealerweise ist die Variantenregel nach einer für den Menschen nachvollziehbaren Logik aufgebaut. Eine gute Möglichkeit hierzu ist, jeder Stelle des Variantencodes eine

Eigenschaft des Produktes zuzuweisen – analog der Variantenlogik des Baukastenmotors aus Abschnitt 5.3.2. Im Falle des fiktiven Kugelschreiber-Beispiels könnte der Variantencode wie folgt gestaltet werden.

Die verschiedenen Produktvarianten differenzieren sich über 4 Eigenschaften: Die Farbe des Ober- und Unterteils des Gehäuses, die Farbe der Tinte in der Mine und die Härte der Feder. Da es möglich ist, dass neben Kugelschreibern noch weitere Produkte hergestellt werden, sollte die Information über das Grundprodukt Kugelschreiber ebenfalls enthalten sein. Folglich empfiehlt sich der in Bild 5.5 dargestellte Aufbau für den Variantencode.

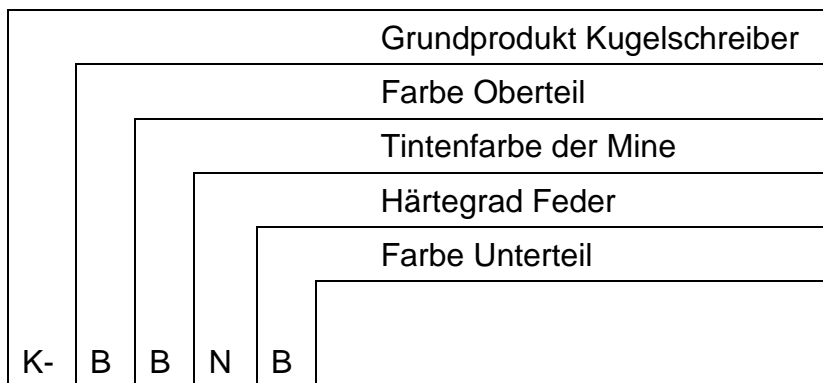


Bild 5.5: Variantenregel für das fiktive Kugelschreiber-Beispiel
Variant code for the fictional ballpoint pen example

5.2.2 Befüllung des Datenmodells

Filling the data model

Nachdem die Tabellen gemäß der Kapitel 5.1.2 und 5.1.3 erstellt wurden, müssen diese mit den in Abschnitt 4.5 beschriebenen Daten gefüllt werden (vgl. Bild 4.18). Hierzu stehen folgende vier Möglichkeiten zur Verfügung:

1. Einfügen als SQL-Query
2. Einfügen über ein Software-Tool
3. Einfügen aus anderer Datenbank
4. Einfügen aus einer Text- / Tabellenverarbeitungsdatei

Diese Möglichkeiten werden im Folgenden kurz vorgestellt.

Die Daten können über einen Befehl eingefügt werden. Dieser Befehl kann direkt in die Konsole eingegeben werden, aus einer Textdatei stammen, die ausgeführt wird, oder über ein anderes Programm an die Datenbank übermittelt werden. Das Ergebnis ist dasselbe, die beschriebene Zeile wird zur Tabelle hinzugefügt. Listing 5.6 zeigt ein Beispiel hierfür. Die Syntax des Befehls `INSERT INTO` wird in Abschnitt 3.3 erläutert. Daher wird hier auf eine nähere Beschreibung verzichtet.

```
1 INSERT INTO tab_linien(Bezeichnung, Standort)
2 VALUES ("Linie München", "München");
```

Listing 5.6: Einfügen von Daten in die Datenbank mittels SQL Query

Adding Data to the Database with a SQL query

Das Einfügen und Manipulieren von Daten über die Konsole kann mühsam und aufwendig sein. Abhilfe schaffen hier speziell entwickelte Softwaretools, sogenannte Datenbank Management Software. Für SQLite Datenbanken gibt es einige Anbieter solcher Software, wie zum Beispiel SQLite Manager von SQLabs LLC oder den SQLite Database Browser [SQLA17, SQLI17].

Die dritte Möglichkeit ist das Einfügen der Daten aus einer anderen Datenbank. Viele Daten liegen bereits in Datenbanken vor. Wie sich die Daten von einer SQLite Datenbank in eine andere übertragen lassen ist in Listing 5.7 dargestellt.

```
1 ATTACH DATABASE "Quelldatenbank.db" AS "Quelldatenbank";
2 INSERT INTO main.Zieltabelle SELECT * FROM
  Quelldatenbank.Quelltable;
```

Listing 5.7: Befüllung einer Datenbank mithilfe einer anderen Datenbank

Filling the database using another database

Handelt es sich um unterschiedliche Datenbanksysteme, können Daten in eine Tabelle exportiert und diese Tabelle anschließend in die Zieldatenbank eingelesen werden. Diese Möglichkeit kann auch verwendet werden, wenn die Eingangsdaten nur als Tabelle gepflegt werden. Ein Vorteil bei diesem Vorgehen ist, dass sich die Daten in einem Tabellenverarbeitungsprogramm besser lesen lassen, hierin manipuliert und dann wieder in die Datenbank eingelesen werden können.

In die SQLite Datenbank können Tabellen eingelesen werden. Am einfachsten lässt sich dies umsetzen, wenn die Tabelle als CSV-Datei (Comma-separated values / Komma-getrennte Werte) angelegt ist.

Zum Einlesen einer Tabelle mittels SQLite muss zunächst der Modus auf CSV gewechselt werden (Zeile 1 aus Listing 5.8). Ist das Trennzeichen der einzulesenden CSV-Datei nicht das Komma, so muss das Trennzeichen angegeben werden (Zeile 2). Mithilfe des Befehls `header on` kann SQLite mitgeteilt werden, dass die erste Zeile Spaltenüberschriften enthält und nicht den ersten Datensatz darstellt (Zeile 3). Anschließend kann der Import erfolgen.

```
1 .mode csv
2 .separator ";"
3 .header on

4 .import tab_anforderungen.csv tab_anforderungen
5 --- weitere Tabellen einlesen
```

Listing 5.8: Einlesen einer CSV-Datei

Importing a CSV-file

Die in Abschnitt 5.2.1 beschriebenen Daten und Verknüpfungen werden als CSV-Tabellen angelegt und anschließend in die Datenbank eingelesen. Zum einen lassen sich diese aus anderen Systemen generieren. Zum anderen können die CSV-Dateien mit gängigen Tabellenverarbeitungsprogrammen anwenderfreundlich angepasst und überarbeitet werden. Im Weiteren wird davon ausgegangen, dass alle Tabellen in der Datenbank gemäß Anhang (Kapitel 8) vorhanden sind und befüllt wurden.

5.2.3 Abfrage der Zuordnungen

Querying the interrelations

Zur Erstellung der Auswertungen aus Abschnitt 4.4 müssen die Zuordnungen innerhalb des Datenmodells abgefragt werden. Bei der Abfrage von Daten zeigt sich die Stärke der ausgewählten Umsetzung in SQL. Abfragen lassen sich sehr leicht erstellen und anpassen.

Will der Planer wissen, welche Stationen mit welcher Stationsnummer in welcher Linie installiert sind, so kann dies mit Listing 5.9 ermittelt werden. Der `SELECT`-Befehl bestimmt, welche Spalten der Datenbank ausgegeben werden sollen. Mittels des `AS`-Befehls kann die Bezeichnung der Spalte in der Ausgabe verändert werden. Diese wurden hier genutzt, um die Ausgabe kompakter und übersichtlicher zu gestalten. Somit steht dort anstatt *tab_linien.Bezeichnung* nur *Linie*.

Per `LEFT JOIN` werden die drei für die Abfrage notwendigen Tabellen miteinander verknüpft. Praktischerweise hat die Spalte, über die die Verknüpfung hergestellt werden soll, in den Tabellen jeweils die gleiche Bezeichnung, sodass der verkürzte `USING`-Befehl anstatt des typischen `ON`-Befehls verwendet werden kann.


```

1  SELECT
2  tab_linien.Bezeichnung AS Linie,
3  tab_stationen.Nr AS Stations_Nr,
4  tab_stationen.Bezeichnung AS Station
5  FROM tab_linien
6  LEFT JOIN lnk_linien_stationen USING(ID_Linie)
7  LEFT JOIN tab_stationen USING(ID_Station)
8  ORDER BY
9  tab_linien.Bezeichnung,
10 tab_stationen.Reihenfolge;

```

Listing 5.9: Abfrage der Stationen je Linie*Query for all stations per line***Tabelle 5.3:** Beispielhaftes Ergebnis der Abfrage aus Listing 5.9*Exemplary result for query from Listing 5.9*

Linie	Stations_Nr	Station
München	M010	Oberteil auflegen und prüfen
München	M020	Mine und Feder einlegen
München	M030	Unterteil fügen, prüfen und verschrauben
Saarbrücken	S010	Oberteil auflegen und prüfen
Saarbrücken	S020	Mine und Feder einlegen
Saarbrücken	S030	Unterteil fügen, prüfen und verschrauben

Wird Listing 5.9 ausgeführt, ergibt sich das Ergebnis, das in Tabelle 5.3 dargestellt ist. Diese Abfrage kann unter Anderem genutzt werden, um zu prüfen, ob alle Zuweisungen zwischen Linien und Stationen korrekt erfolgt sind.

Analog zu Listing 5.9 lassen sich diverse Abfragen generieren. Zum Beispiel können alle Prozesse an einer Station oder alle Bauteilvarianten einer Bauteilart ausgegeben werden.

In die Abfragen können Gruppierungsfunktionen einbezogen werden. Dies ist hilfreich, wenn beispielsweise je Bauteilart nicht alle Bauteilvarianten ausgegeben werden sollen, sondern nur die Anzahl der Bauteilvarianten, die mit einer Bauteilart verbunden sind.

Dazu wird die zu zählende Spalte an den COUNT-Befehl übergeben. Dieser zählt dann alle Elemente. Im Fall der Bauteilvariantenanzahl sollen doppelte Elemente nicht mitgezählt werden. Dies wird durch den DISTINCT-Befehl sichergestellt. Dieser zählt mehrfach auftretende Werte nur einmal. Weiterhin sollen die Bezeichnungen der Bauteilvarianten nicht nach Groß- und Kleinschreibung unterschieden werden. Daher

wird mittels der `UPPER`-Funktion die Bauteilbezeichnung immer in Großbuchstaben umgewandelt.

Damit die SQL-Abfrage die Ergebnisse korrekt zusammenfasst, muss an ihr Ende noch ein `GROUP BY`-Befehl angefügt werden. Dieser legt fest, nach welchen Kriterien bei der Ausgabe unterschieden werden soll. In Listing 5.10 sind dies zum Beispiel die Bauteilarten. Das Ergebnis von Listing 5.10 ist in Tabelle 5.4 dargestellt.

```

1  SELECT
2  tab_bauteilarten.Bezeichnung AS Bauteilart,
3  COUNT(DISTINCT UPPER(tab_bauteile.Bezeichnung)) AS Anzahl_Bauteile
4  FROM tab_bauteilarten
5  LEFT JOIN tab_bauteile USING(ID_Bauteilart)
6  GROUP BY tab_bauteilarten.ID_Bauteilart;

```

Listing 5.10: Abfrage der Anzahl von Bauteilvariante je Bauteilart

Query of the amount of part variants for each part type

Tabelle 5.4: Beispielhaftes Ergebnis der Abfrage aus Listing 5.10

Exemplary results for listing Listing 5.10

Bauteilart	Anzahl_Bauteilvarianten
Oberteil	2
Mine	1
Feder	1
Unterteil	2

In Abschnitt 4.2.4 wurde erläutert, dass Elemente im entwickelten Datenmodell einen geschlossenen Kreis bilden, wenn diese zusammengehören. Diesen Effekt macht sich die Abfrage aus Listing 5.11 zu Nutzen. In ihr werden die Elemente des Datenmodells mittels `LEFT JOIN` und den IDs verknüpft.

In den Zeilen 1-7 werden die Spalten der Tabellen ausgewählt, die später in der Ausgabe oder Ausleitungsdatei erscheinen sollen. Zur besseren Lesbarkeit werden den Bezeichnungsfeldern neue Namen gegeben. Die Spalte `tab_linien.Bezeichnung` wird in *Linien* umbenannt. Gleichermaßen wird mit den anderen Spalten verfahren.

Die Zeilen 8-19 verknüpfen alle Klassen bzw. Tabellen des Basis-Datenmodells (vgl. Bild 4.7) miteinander. Für alle Tabellen wird dazu die jeweilige ID verwendet, mit Ausnahme der Tabellen, die mit der Produkt-Tabelle verknüpft sind. Die Verknüpfung sowohl von der Bauteilvariante als auch von der Linienbelegung zum Produkt erfolgt über den Variantencode (vgl. Abschnitt 5.2.1). Um diesen Vergleich zu realisieren wurde das `LIKE`-Statement (vgl. Abschnitt 5.1.4) verwendet.

Das Ergebnis dieser Zeilen sind alle möglichen Kombinationen im Datenmodell. Jedoch sollen Kombinationen, die sich keinem Produkt zuordnen lassen und daher nicht real existieren, auch nicht Bestandteil des Ergebnisses sein. Daher werden in Zeile 20 alle Ergebnis-Zeilen in denen kein Produkt enthalten ist eliminiert.

```
1  SELECT
2  tab_linien.Bezeichnung AS Linie,
3  tab_stationen.Bezeichnung AS Station,
4  tab_prozesse.Bezeichnung AS Prozess,
5  tab_bauteilarten.Bezeichnung AS Bauteilart,
6  tab_bauteile.Bezeichnung AS Bauteilvariante,
7  tab_produkte.Variantencode AS Produkt
8  FROM tab_linien
9  LEFT JOIN tab_linienbelegung USING(ID_Linie)
10 LEFT JOIN lnk_linien_stationen USING(ID_Linie)
11 LEFT JOIN tab_stationen USING(ID_Station)
12 LEFT JOIN lnk_stationen_prozesse USING(ID_Station)
13 LEFT JOIN tab_prozesse USING(ID_Prozess)
14 LEFT JOIN lnk_prozesse_bauteilarten USING(ID_Prozess)
15 LEFT JOIN tab_bauteilarten USING(ID_Bauteilart)
16 LEFT JOIN tab_bauteile USING(ID_Bauteilart)
17 LEFT JOIN tab_produkte ON (
18 tab_produkte.Variantencode LIKE tab_bauteile.Variantencode AND
19 tab_produkte.Variantencode LIKE tab_linienbelegung.Variantencode)
20 WHERE tab_produkte.ID_Produkt NOT NULL
21 ORDER BY tab_linien.ID_Linie,
22 tab_stationen.ID_Station,
23 tab_prozesse.ID_Prozess;
```

Listing 5.11: Abfrage aller Zusammenhänge

Query for all interrelations

Die Zeilen 21-23 sortieren das Ergebnis aus Sicht der Montageplanung, also nach Linie, Station und Prozess. Jede andere Sortierung ist möglich und gleichermaßen richtig. Diese Zeilen sind als optional zu betrachten.

Die Ausführung von Listing 5.11 ergibt das in Tabelle 5.5 ausschnittsweise dargestellte Ergebnis. Dieses Ergebnis zeigt, dass nur tatsächlich miteinander verknüpfte Elemente enthalten sind. Nicht existierende Bauteilvariantenkombinationen werden mithilfe der Klasse Produkt herausgefiltert (vgl. Abschnitte 4.2.1 und 4.2.4). Den Stationen sind die korrekten Standorte zugewiesen. Die verschiedenen Stationen verwenden dieselben Prozesse, trotzdem erscheinen hinter den Prozessen nur die Bauteilvarianten, die gemäß Linienbelegung auch dieser Linie zugewiesen sind. Als Produkt wird der Variantencode aus Bild 5.3 und Bild 5.5 ausgegeben. Über diesen ist das Produkt eindeutig identifiziert.

Für die Abfragen aus den folgenden Abschnitten 5.2.4 und 5.2.5 ist die Abfrage aller Zuordnungen ebenfalls relevant, da nur Kombinationen von Anforderungen und Fähigkeiten bzw. Kapazitäten abgeglichen werden sollen, die sich gemäß Linienbelegung tatsächlich ergeben. Zur besseren Lesbarkeit wird in den Listings dieser Abschnitte auf die Darstellung des ausführlichen SQL-Quellcodes verzichtet. In Abschnitt 5.2.7 wird eine Abfrage erstellt, die alle relevanten Informationen inklusive Anforderungen, Fähigkeiten und Kapazitäten generiert. Der SQL-Quellcode hierzu ist vollständig angegeben.

Tabelle 5.5: Ausschnitt des Ergebnisses aus Listing 5.11

Extract of the results from Listing 5.11

Linie	Station	Prozess	Bauteilart	Bauteilvariante	Produkt
München	Oberteil auflegen und prüfen	Oberteil auflegen	Oberteil	Oberteil Rot	K-RSSR
München	Oberteil auflegen und prüfen	Oberteil auflegen	Oberteil	Oberteil Rot	K-RSMR
...

5.2.4 Abfrage von Zeiträumen

Query of periods of time

Viele der Abfragen, die in Abschnitt 4.4 beschrieben werden, stellen einen statischen Zustand dar. Andere Abfragen müssen den zeitlichen Verlauf in ihre Analyse integrieren. Dieser zeitliche Verlauf wird durch die Angaben von PVL, SOP und EOP in der Linienbelegung bestimmt (vgl. Abschnitt 5.1.2).

Zu mehreren Zeitpunkten müssen die statischen Zustände des Systems bestimmt werden. Daraus entsteht die Dynamik. In vielen Programmiersprachen wird eine solche Dynamik über eine Schleife ähnlich Listing 5.12 realisiert.

```

1 void AbfrageZeitraum(int startjahr, int endjahr) {
2 // Alle Jahre zwischen startjahr und endjahr durchlaufen
3     for(int jahr = startjahr; jahr <= endjahr; ++i) {
4         Auswertung(jahr); // Auswertung (=statisch) für ein Jahr
5     }
6 }

```

Listing 5.12: For-Schleife zur Erstellung mehrere statischer Auswertungen in C++

For loop to create multiple static results in C++

In SQLite sind solche Schleifen nicht möglich. Entweder muss die Umsetzung der Dynamik außerhalb von SQLite erfolgen oder eine andere Lösung gefunden werden. Die für diese Arbeit verwendete Lösung wird im Folgenden beschrieben.

Um eine Abfrage für jeden Monat zu erzeugen wird die Tabelle *tab_zeitraeume* verwendet. In ihr sind die zu untersuchenden Zeiträume gespeichert. Jeder Zeitpunkt, für den eine Auswertung erstellt werden soll, ist eine eigene Zeile in der Datenbank. Das Datum wird dabei nach dem Format YYYY aufgebaut. Mit diesem Format lassen sich für einzelne Monate Abfragen erstellen. Bei Bedarf kann das Datumsformat auf YYYYMMDD erweitert werden um Tagegenaue Abfragen zu erstellen. Theoretisch kann das Format beliebig verfeinert oder gröber gestaltet werden, um die gewünschte Genauigkeit zu erhalten.

Ist die Tabelle Zeiträume mit den gewünschten Zeitpunkten gefüllt, kann die Abfrage aus Listing 5.13 ausgeführt werden.

```
1  SELECT
2  tab_zeitraeume.Datum AS Datum,
3  tab_linien.Bezeichnung AS Linie,
4  COUNT(DISTINCT tab_linienbelegung.Variantencode) AS
   Anzahl_Produktvarianten
5  FROM tab_linien
6  LEFT JOIN tab_linienbelegung USING(ID_Linie)
7  LEFT JOIN tab_zeitraeume ON (
8  tab_zeitraeume.Datum >= tab_linienbelegung.SOP AND
9  tab_zeitraeume.Datum <= tab_linienbelegung.EOP)
10 GROUP BY
11 tab_linien.Bezeichnung,
12 tab_zeitraeume.Datum
13 ORDER BY
14 tab_linien.Bezeichnung,
15 tab_zeitraeume.Datum;
```

Listing 5.13: Abfrage zur Erstellung zeitdynamischer Abfragen in SQL

Query for the creation of a chronological sequence in SQL

Ein beispielhaftes Ergebnis von Listing 5.13 ist in Tabelle 5.6 dargestellt. Für jeden Monat wird eine Abfrage erstellt. In diesem Fall werden je Datum die Linienbezeichnung und die Anzahl zugewiesener Produktvarianten ausgegeben.

Tabelle 5.6: Ergebnis aus Listing 5.13*Result from Listing 5.13*

Datum	Linie	Anzahl_Produktvarianten
1701	München	16
1702	München	18
...
1701	Saarbrücken	10
1702	Saarbrücken	9
...

5.2.5 Abgleich Anforderungen und Fähigkeiten

Comparison of requirements and capabilities

Zur Beurteilung der Montierbarkeit müssen (Produkt-)Anforderungen und (Betriebsmittel-)Fähigkeiten miteinander abgeglichen werden (vgl. Abschnitt 4.3.1). Dabei lassen sich die Anforderungen und Fähigkeiten, die im Rahmen dieser Arbeit betrachtet werden, in zwei Gruppen einteilen:

1. Werte und Wertebereich
2. Werte und Werteliste

Im ersten Fall ist die Anforderung ein numerischer Wert. Die Fähigkeit wird durch einen Wertebereich beschrieben, der durch die Angabe eines Minimums und eines Maximums spezifiziert wird. Liegt der Wert der Anforderung zwischen diesen beiden Werten, kann die Anforderung erfüllt werden. Andernfalls wird die Anforderung nicht erfüllt.

Bild 5.6 verdeutlicht dies anhand des Beispiels eines Parallelgreifers. Dieser kann Objekte mit einer Breite von $F_{\min}=10\text{mm}$ bis $F_{\max}=60\text{mm}$ greifen. Die Anforderungen $A_2=20\text{mm}$, $A_3=36\text{mm}$ $A_4=50\text{mm}$ sind in diesem Wertebereich enthalten und lassen sich somit umsetzen. Die Anforderungen $A_1=6\text{mm}$ und $A_5=66\text{mm}$ liegen außerhalb und sind somit nicht umsetzbar. Grafisch sind die Anforderungen A_1 bis A_6 und Fähigkeiten F_{\min} und F_{\max} in Bild 5.7 zusammengefasst.

Diese Art des Abgleichs wird in der Datenbank mit einer 1 gekennzeichnet. Listing 5.14 zeigt, wie in SQL abgefragt werden kann, ob die Anforderungen umsetzbar sind.

Der Grundgedanke der Abfrage ist, dass alle Fähigkeiten durchsucht werden, sofern diese über die Station und Bauteilart mit der Anforderung verknüpft sind. Wird mindestens eine passende Fähigkeit gefunden, so ist der Abgleich positiv. Wird keine passende Fähigkeit gefunden, so fällt der Abgleich negativ aus.

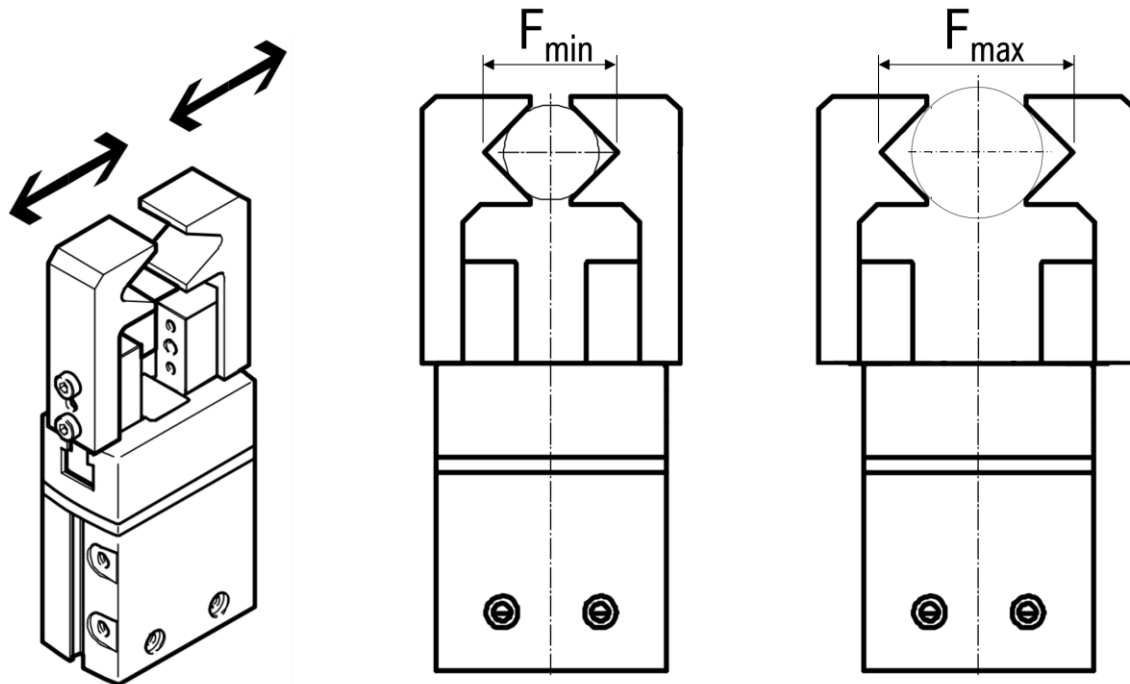


Bild 5.6: Parallelgreifer Festo DHPS-20-A als Beispiel für eine Fähigkeit mit Wertebereich
Festo DHPS parallel gripper as an example for value range capabilities

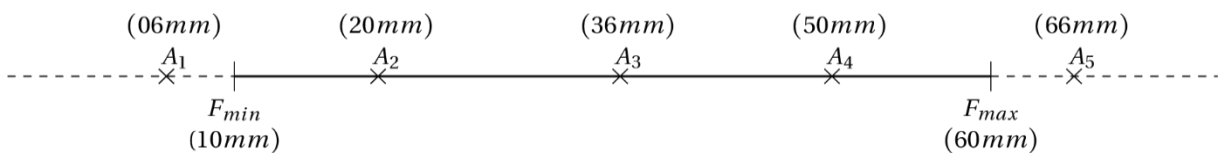


Bild 5.7: Grafische Darstellung von Anforderungen und Fähigkeiten als Wert und Wertebereich (Abgleichart 1)
Image of requirements and capabilities as a value range (Comparison type 1)

Die Umsetzung dieses Grundgedanken in SQL erfolgt mittels `LEFT JOIN`. Für jedes Element einer Anforderung wird eine Fähigkeit gesucht, die alle der folgenden Bedingungen erfüllt:

1. Der Prozesstyp ist identisch.
2. Der Wert der Anforderung ist kleiner oder gleich dem Maximum der Fähigkeit.
3. Der Wert der Anforderung ist größer oder gleich dem Minimum der Fähigkeit.
4. Die Abgleichart ist gleich 1.

Sind alle Bedingungen erfüllt, ist eine passende Fähigkeit gefunden. Die Bedingungen 1-4 sind in Listing 5.14 als Bedingungen im `LEFT JOIN`-Befehl enthalten.

```

1  SELECT
2  Anforderung,
3  Faehigkeit_min,
4  Faehigkeit_max,
5  (tab_faehigkeiten.ID_Faehigkeit IS NULL) AS AF_Konflikt
6  FROM tab_Anforderungen
7  LEFT JOIN tab_Faehigkeiten
8  ON ( tab_Anforderungen.Typ = tab_Faehigkeiten.Typ
9  AND Anforderung <= Faehigkeit_max
10 AND Anforderung >= Faehigkeit_min
11 AND Abgleichart = 1);

```

Listing 5.14: Abfrage für Anforderungen und Fähigkeiten für Werte und Wertebereich (Abgleichart 1)

Query of requirements and capabilities for values and a range

Ein Beispiel für eine solche Abfrage ist in Tabelle 5.7 dargestellt. Zunächst werden die Anforderungen und Fähigkeiten in Spalten ausgegeben. Wenn sich zu einer Anforderung keine passende Fähigkeit finden lässt, dann bleiben die Spalten für die Fähigkeit leer und die Spalte AF_Konflikt wird mit 1 belegt. Auf diese Art lässt sich die Ausgabetable einfach nach Konflikten filtern.

Tabelle 5.7: Ergebnis der Abfrage aus Listing 5.14

Result of the query in Listing 5.14

Anforderung	Faehigkeit_min	Faehigkeit_max	AF_Konflikt
5	0	10	0
10	0	10	0
15			1

Diese Abfrage ist so in die Gesamtanfragen integriert, dass nur Anforderungen von Bauteilvarianten mit Fähigkeiten von Stationen verglichen werden, die auch im Basis-Datenmodell miteinander verknüpft sind.

Bei der zweiten Abgleichart ist die Anforderung ein numerischer Wert oder eine alphanumerische Zeichenkette. Die Fähigkeit ist eine Liste solcher Werte oder Zeichenketten. Ist die Anforderung in der Liste der Fähigkeiten enthalten, so fällt der Vergleich positiv aus. Ist die Anforderung nicht enthalten, so fällt der Vergleich negativ aus. (vgl. Bild 5.8)

Werte-Liste (Fähigkeiten):	Prüfung (Anforderungen):	Legende:
<div style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px; display: inline-block;"> M5 M6 M8 </div>	M6 <input checked="" type="checkbox"/> M4 <input type="checkbox"/>	<input checked="" type="checkbox"/> Anforderung in der Liste der Fähigkeiten enthalten <input type="checkbox"/> Anforderung nicht in der Liste der Fähigkeiten enthalten

Bild 5.8: Grafische Darstellung von Anforderungen und Fähigkeiten als Wert und Werteliste (Abgleichart 2)

Representation of requirements and capabilities as value and list of values (Comparison type 2)

Für den Abgleich von Werten und Werte-Liste (vgl. Listing 5.15) wird das Grundprinzip von Listing 5.14 übernommen, jedoch mit folgenden Bedingungen:

1. Der Prozesstyp ist identisch.
2. Der Wert der Anforderung ist entweder im Minimum oder im Maximum der Fähigkeit enthalten.
3. Die Abgleichart ist gleich 2.

Da die Klasse *Fähigkeit* für Abgleichart 1 und 2 identisch ist, enthält sie immer ein Minimum und ein Maximum Feld. Um diesen Platz effizient zu nutzen werden diese auch für Abgleichart 2 genutzt. Somit lassen sich Wertepaare angeben. Es ist aber ebenso möglich, nur einen der beiden Werte zu belegen.

```

1  SELECT
2  Anforderung,
3  Faehigkeit,
4  (tab_faehigkeiten.ID_Faehigkeit IS NULL) AS AF_Konflikt
5  FROM tab_Anforderungen
6  LEFT JOIN tab_Faehigkeiten
7  ON ( tab_Anforderungen.Typ = tab_Faehigkeiten.Typ
8  AND Anforderung = Faehigkeit
9  AND Abgleichart = 2) ;

```

Listing 5.15: Abfrage für Anforderungen und Fähigkeiten für Werte und Werte-Listen (Abgleichart 2)

Query of requirements and capabilities for values and a value list

Durch die Verbindung der beiden Abfragen aus Listing 5.14 und Listing 5.15 lassen sich alle Konflikte, die auf Anforderungen und Fähigkeiten basieren, ermitteln.

Wird eine Bauteilart an mehreren Stationen bearbeitet, dann werden dort mittels der Abfrage auch alle Anforderungen aufgeführt. In der Auswertung muss also noch manuell geprüft werden, ob jede Anforderung mindestens einmal erfüllt ist. Hierzu kann in der Spalte für die Fähigkeiten nach leeren Zellen gefiltert werden. In diesen Fällen konnte keine der Fähigkeiten die Anforderung erfüllen.

5.2.6 Abgleich von Kapazität

Comparison of capacity

Wie in Abschnitt 4.3.2 beschrieben stellen die Auslastungen und Kapazitäten eine besondere Art von Anforderungen und Fähigkeiten dar. Mittels der Abfrage aus Listing 5.16 lässt sich überprüfen, ob die Kapazität der Stationen ausreicht, um alle benötigten Bauteilvarianten anzustellen und zu verarbeiten. Ein Ergebnis der Abfrage ist in Tabelle 5.8 zu sehen.

```

1  SELECT
2  (...)
3  COUNT(DISTINCT tab_bauteile.Bezeichnung) AS Anzahl_Bauteile,
4  tab_kapazitaeten_stationen.Kapazitaet AS Kapazitaet_Stationen,
5  (Anzahl_Bauteile > tab_kapazitaeten_stationen.Kapazitaet AND lnk_pro-
   zesse_bauteilarten.Konsumiert > 0) AS
   Kapa_Konflikt_Station
6  (...)
7  LEFT JOIN tab_kapazitaeten_stationen USING(ID_Station, ID_Bauteilart)
8  (...)

```

Listing 5.16: Abfrage für Kapazitäten

Query for capacities

Listing 5.16 zeigt in verkürzter Form die zusätzlichen Zeilen für eine Abfrage der Kapazitäten. Die ausführliche Abfrage ist in Listing 5.17 dargestellt. Mittels der bereits aus Listing 5.10 bekannten Abfrage wird die Anzahl der Bauteilvarianten ermittelt (Zeile 3). Außerdem wird die Kapazität ermittelt (Zeile 4). In Zeile 5 werden die Anzahl der Bauteilvarianten und die Kapazität miteinander verglichen, allerdings nur, wenn die Bauteilart an dieser Station konsumiert wird.

Tabelle 5.8: Abfrage der Stationskapazitäten

Query for station capacities

Linie	Station	Bauteilart	Anzahl Bauteilvarianten	Kapazität	Kapa_Konflikt
München	Oberteil auflegen	Oberteil	6	5	1
München	Mine und Feder einlegen	Mine	3	2	1
München	Mine und Feder einlegen	Feder	1	2	0
...

In Listing 5.16 ist nur die Abfrage für die Kapazität der Station und nicht die der Linie dargestellt. Deren Aufbau ist ähnlich, lediglich muss in Zeile 7 *ID_Station* durch *ID_Linie* ersetzt werden, da die die Kapazität mit der Linie verknüpft ist. (vgl. Bild 5.1)

Wie schon beim Abgleich von Anforderungen und Fähigkeiten wird mit *Kapa-Konflikt* eine zusätzliche Ausgabe-Spalte generiert, in der nach Konflikten gefiltert werden kann.

5.2.7 Kombination aller Abfragen

Combination of all queries

In den Abschnitten 5.2.3 bis 5.2.6 werden ausgewählte Einzelauswertungen vorgestellt. Darüber hinaus gibt es die Möglichkeit, alle Abfragen zu kombinieren und somit eine Datei zu erstellen, die die Ergebnisse aller Abfragen enthält (vgl. Listing 5.17). Analog dem gewünschten Ergebnis aus Abschnitt 4.4 erreicht diese Tabelle schnell eine große Zeilenanzahl. Für die effektive Arbeit mit dieser Tabelle müssen die Daten mithilfe von Filter- und Sortierfunktionen analysiert werden.

In den Zeilen 1-21 werden die Spalten der Ausgabedatei festgelegt. Dies sind einerseits Werte aus der Datenbank (Zeilen 1-8, 10, 11 und 16-19). Darüber hinaus werden einige Berechnungen angestellt. In den Zeilen 12 und 13 werden Konflikte mit der Kapazität gemäß Abschnitt 5.2.6 ermittelt. Mit der Formel in Zeile 14 wird ermittelt, ob die Anforderungen für jede Bauteilvariante vollständig beschrieben sind. In Zeile 15 wird ermittelt, ob ein Konflikt zwischen Anforderungen und Fähigkeiten vorliegt – wie in Abschnitt 5.2.5 beschrieben.

Die Zeilen 20 und 21 fassen die Kontaktdaten von Entwicklern und Montageplanern in einer Zeile zusammen. Alternativ könnten auch alle Kontaktdaten in einzelnen Spalten ausgegeben werden.

Eine Besonderheit stellt Zeile 9 dar. In der Gesamt-Abfragen sollen die Bauteilvarianten explizit ausgegeben werden. Außerdem soll die Anzahl der Bauteilvarianten mit angegeben werden, um Konflikte in der Kapazität transparent zu machen. Um beides zu ermöglichen, sind zwei SQL Abfragen erforderlich – eine, die die Bauteilvarianten einzeln ausgibt, und eine, die diese kombiniert und zählt.

Für den zweiten Fall wird die Abfrage Q (Zeile 39-58) implementiert. Diese enthält die bereits bekannten Elemente zur Kombination der Daten und Ermittlung der Anzahl der Bauteilvarianten je Linie, Station und Bauteilart unter Berücksichtigung der Linienbelegung. Diese Abfrage Q wird dann mit der Gesamtanfrage verknüpft (Zeile 58).

In Zeile 9 wird auf das Ergebnis dieser verschachtelten Abfrage zugegriffen. Die Zeilen 12 und 13 greifen für die Berechnung ebenfalls auf diese Abfrage zu. In den Zeilen 22-38 werden die Tabellen miteinander verknüpft. In Zeile 59 werden alle Zeilen ohne existierendes Produkt wieder herausgefiltert und in Zeile 60 wird das Ergebnis sortiert.

Die sich ergebende Tabelle ist zwar lang, aber eine gute Basis für den Planer. Sie enthält alle benötigten Informationen in einer Datei, zeigt auf, wo Konflikte vorliegen, welche Bauteilvarianten darin involviert sind und wer die zuständigen Ansprechpartner sind.

```

1  SELECT
2  tab_linien.Bezeichnung AS Linie,
3  tab_stationen.Bezeichnung AS Station,
4  tab_prozesse.Bezeichnung AS Prozess,
5  tab_bauteilarten.Bezeichnung AS Bauteilart,
6  tab_bauteile.Bezeichnung AS Bauteil,
7  tab_produkte.Variantencode AS Produkt,
8  lnk_prozesse_bauteilarten.Konsumiert AS Konsumiert,
9  Q.Anzahl_Bauteile AS Anzahl_Bauteile,
10 tab_kapazitaeten_logistik.Kapazitaet AS Kapazitaet_Logistik,
11 tab_kapazitaeten_stationen.Kapazitaet AS Kapazitaet_Stationen,
12 (Q.Anzahl_Bauteile > tab_kapazitaeten_stationen.Kapazitaet AND
13 lnk_prozesse_bauteilarten.Konsumiert > 0) AS Kapa_Konflikt_Stationen,
14 (Q.Anzahl_Bauteile > tab_kapazitaeten_logistik.Kapazitaet AND lnk_pro-
15 zesse_bauteilarten.Konsumiert > 0) AS Kapa_Konflikt_Logistik,
16 (tab_prozesstypen.ID_Prozesstyp NOT NULL AND tab_anforderungen.ID_An-
17 forderung IS NULL) AS Anforderungen_Unvollstaendig,
18 (tab_faehigkeiten.ID_Faehigkeit IS NULL AND tab_prozesstypen.ID_Pro-
19 zesstyp IS NOT NULL) AS AF_Konflikt,
20 tab_prozesstypen.Bezeichnung AS Prozesstyp,
21 tab_anforderungen.Anforderung AS Anforderung,
22 tab_faehigkeiten.Faehigkeit_min AS Faehigkeit_min,
23 tab_faehigkeiten.Faehigkeit_max AS Faehigkeit_max,
24 (Ansprechpartner_Produktion.Vorname || " " || Ansprechpartner_Produk-
25 tion.Nachname || "(" || Ansprechpartner_Produktion.Abteilung || ", "
26 || Ansprechpartner_Produktion.Email || ", " || Ansprechpartner_Produk-
27 tion.Telefon || ")") AS Ansprechpartner_Produktion,
28 (Ansprechpartner_Entwicklung.Vorname || " " || Ansprechpartner_Ent-
29 wicklung.Nachname || "(" || Ansprechpartner_Entwicklung.Abteilung ||
30 ", " || Ansprechpartner_Entwicklung.Email || ", " || Ansprechpart-
31 ner_Entwicklung.Telefon || ")") AS Ansprechpartner_Entwicklung
32 FROM tab_linien
33 LEFT JOIN tab_linienbelegung USING(ID_Linie)
34 LEFT JOIN lnk_linien_stationen USING(ID_Linie)
35 LEFT JOIN tab_stationen USING(ID_Station)
36 LEFT JOIN lnk_stationen_prozesse USING(ID_Station)
37 LEFT JOIN tab_prozesse USING(ID_Prozess)
38 LEFT JOIN lnk_prozesse_bauteilarten USING(ID_Prozess)
39 LEFT JOIN tab_bauteilarten USING(ID_Bauteilart)
40 LEFT JOIN tab_bauteile USING(ID_Bauteilart)
41 LEFT JOIN tab_produkte ON (tab_produkte.Variantencode LIKE
42 tab_bauteile.Variantencode AND tab_produkte.Variantencode LIKE
43 tab_linienbelegung.Variantencode)
44 LEFT JOIN tab_kapazitaeten_stationen USING(ID_Station, ID_Bauteilart)
45 LEFT JOIN tab_kapazitaeten_logistik USING(ID_Linie, ID_Bauteilart)
46 LEFT JOIN tab_prozesstypen USING(ID_Prozess)
47 LEFT JOIN tab_anforderungen ON (tab_bauteile.ID_Bauteil = tab_anforde-
48 rungen.ID_Bauteil AND tab_anforderungen.ID_Prozesstyp = tab_prozessty-
49 pen.ID_Prozesstyp)

```

```

36 LEFT JOIN tab_faehigkeiten ON(tab_stationen.ID_Station = tab_faehig-
keiten.ID_Station AND tab_faehigkeiten.ID_Prozesstyp = tab_anforderun-
gen.ID_Prozesstyp AND ((Abgleichart = 1 AND Anforderung <= Faehig-
keit_max AND Anforderung >= Faehigkeit_min) OR (Abgleichart = 2 AND
37 LEFT JOIN tab_personen AS Ansprechpartner_Entwicklung ON(tab_bauteil-
arten.ID_Ansprechpartner = Ansprechpartner_Entwicklung.ID_Person)
38 LEFT JOIN (
39 SELECT
40 tab_linien.ID_Linie AS ID_Linie,
41 tab_stationen.ID_Station AS ID_Station,
42 tab_prozesse.ID_Prozess AS ID_Prozess,
43 tab_bauteilarten.ID_Bauteilart AS ID_Bauteilart,
44 COUNT(DISTINCT tab_bauteile.Bezeichnung) AS Anzahl_Bauteile,
45 tab_produkte.ID_Produkt AS ID_Produkt
46 FROM tab_linien
47 LEFT JOIN tab_linienbelegung USING(ID_Linie)
48 LEFT JOIN lnk_linien_stationen USING(ID_Linie)
49 LEFT JOIN tab_stationen USING(ID_Station)
50 LEFT JOIN lnk_stationen_prozesse USING(ID_Station)
51 LEFT JOIN tab_prozesse USING(ID_Prozess)
52 LEFT JOIN lnk_prozesse_bauteilarten USING(ID_Prozess)
53 LEFT JOIN tab_bauteilarten USING(ID_Bauteilart)
54 LEFT JOIN tab_bauteile USING(ID_Bauteilart)
55 LEFT JOIN tab_produkte ON (tab_produkte.Variantencode LIKE
tab_bauteile.Variantencode AND tab_produkte.Variantencode LIKE
tab_linienbelegung.Variantencode)
56 WHERE tab_produkte.ID_produkte NOT NULL
57 GROUP BY tab_linien.ID_Linie, tab_stationen.ID_Station, tab_pro-
zesse.ID_Prozess, tab_bauteilarten.ID_Bauteilart
58 ) AS Q USING(ID_Linie, ID_Station, ID_Prozess, ID_Bauteilart)
59 WHERE tab_produkte.ID_Produkt NOT NULL
60 ORDER BY tab_linien.ID_Linie, tab_stationen.ID_Station,
tab_prozesse.ID_Prozess;

```

Listing 5.17: Abfrage aller Zusammenhänge des Modells

Query for all interrelations of the model

5.2.8 Analyse von Szenarien durch Änderungen im Datenmodell

Analyzing scenarios by changing the data model

Wie in den Abschnitten 2.2.2 und 4.1 erläutert besteht die Aufgabe der Montageplanung bei der Integrationsplanung darin, Konflikte zwischen Anforderungen und Fähigkeiten zu identifizieren und aufzulösen. Hierbei unterstützt das Modell. Die notwendigen Abfragen sind in den Abschnitten 5.2.3 bis 5.2.6 vorgestellt worden.

Um einen realen Anwendungsfall des Modells zu demonstrieren wird das fiktive Szenario erweitert: Es wird eine neue Produktgeneration eingeführt. Diese enthält neue Produkte und mit diesen neue Bauteilvarianten. Außerdem kommt ein neuer Standort im Ausland hinzu. Bauteilarten und Prozesse kommen nicht hinzu.

Der neue Standort ist zunächst als Kleinserienstandort ausgelegt. Hier wird die gesamte Montage in nur zwei Stationen durchgeführt. Bild 5.4 wird um diesen Standort

und dessen Stationen erweitert. Bild 5.9 gibt einen Überblick über die zusätzlichen Elemente im Objektdiagramm. Um die neuen Elemente besser hervor zu heben, sind diese in schwarz dargestellt, die Elemente, die bereits in Bild 5.4 vorhanden sind, sind grau markiert.

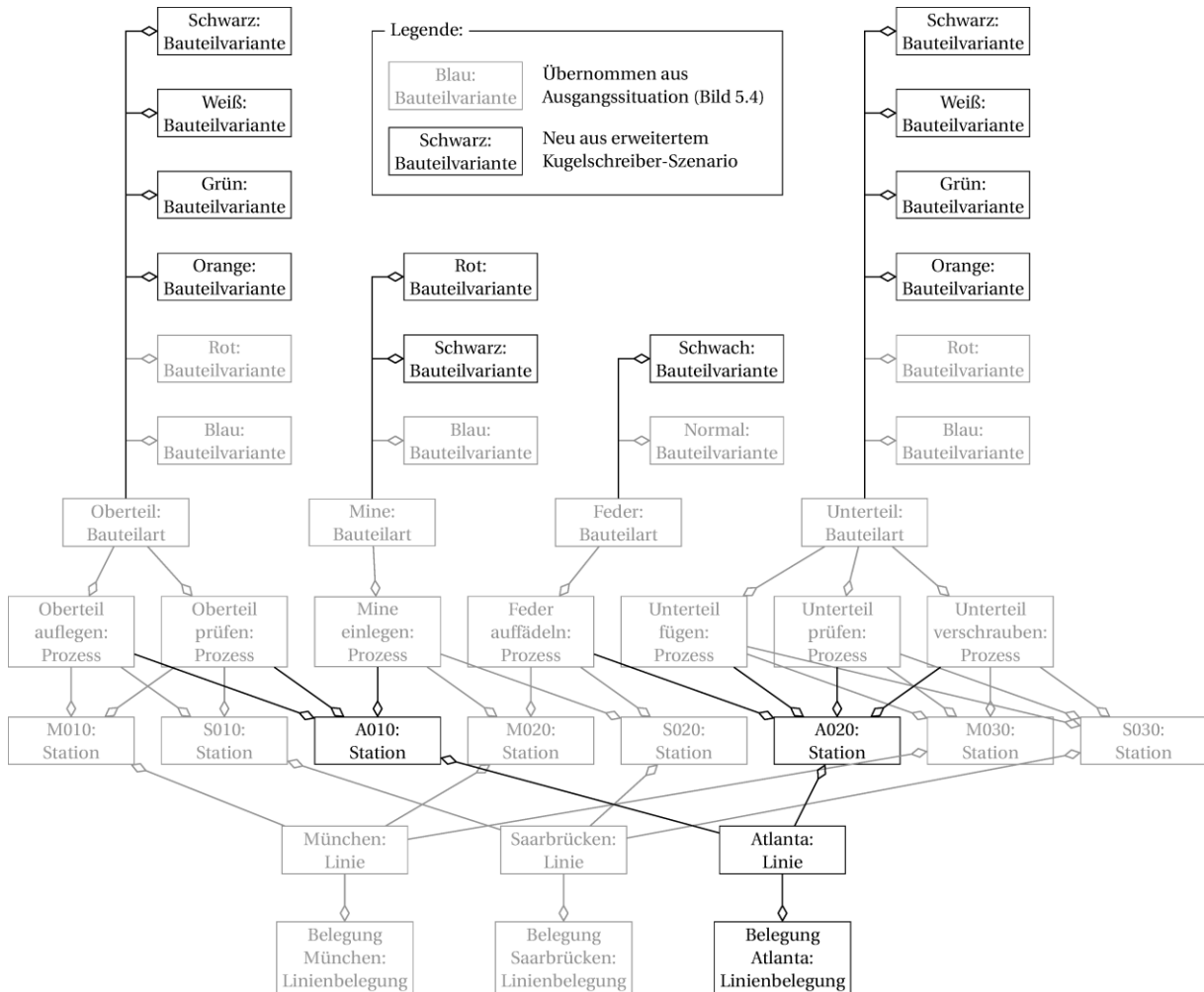


Bild 5.9: Bild 5.4 erweitert um die neue Linie mit ihren Stationen und die neuen Bauteilvarianten

Bild 5.4 extended by the new assembly lines with their stations and new part variants

Neben den bisher hergestellten Kugelschreibern mit rotem und blauem Ober- und Unterteil sollen in Zukunft zusätzlich orange, grüne, weiße und schwarze Kugelschreiber hergestellt werden. Kombinationen verschiedener Farben von Ober- und Unterteil werden weiterhin nicht angeboten. Ein schwarzes Oberteil lässt sich demnach nur mit einem schwarzen Unterteil kombinieren. Jeder Kugelschreiber soll mit drei verschiedenen Minenfarben angeboten werden können: blau, schwarz und rot. Neben der bisher verwendeten normalen Feder kann auch eine starke Feder bestellt werden. Die sich ergebenden 36 Produktvarianten sind in Bild 5.10 dargestellt.

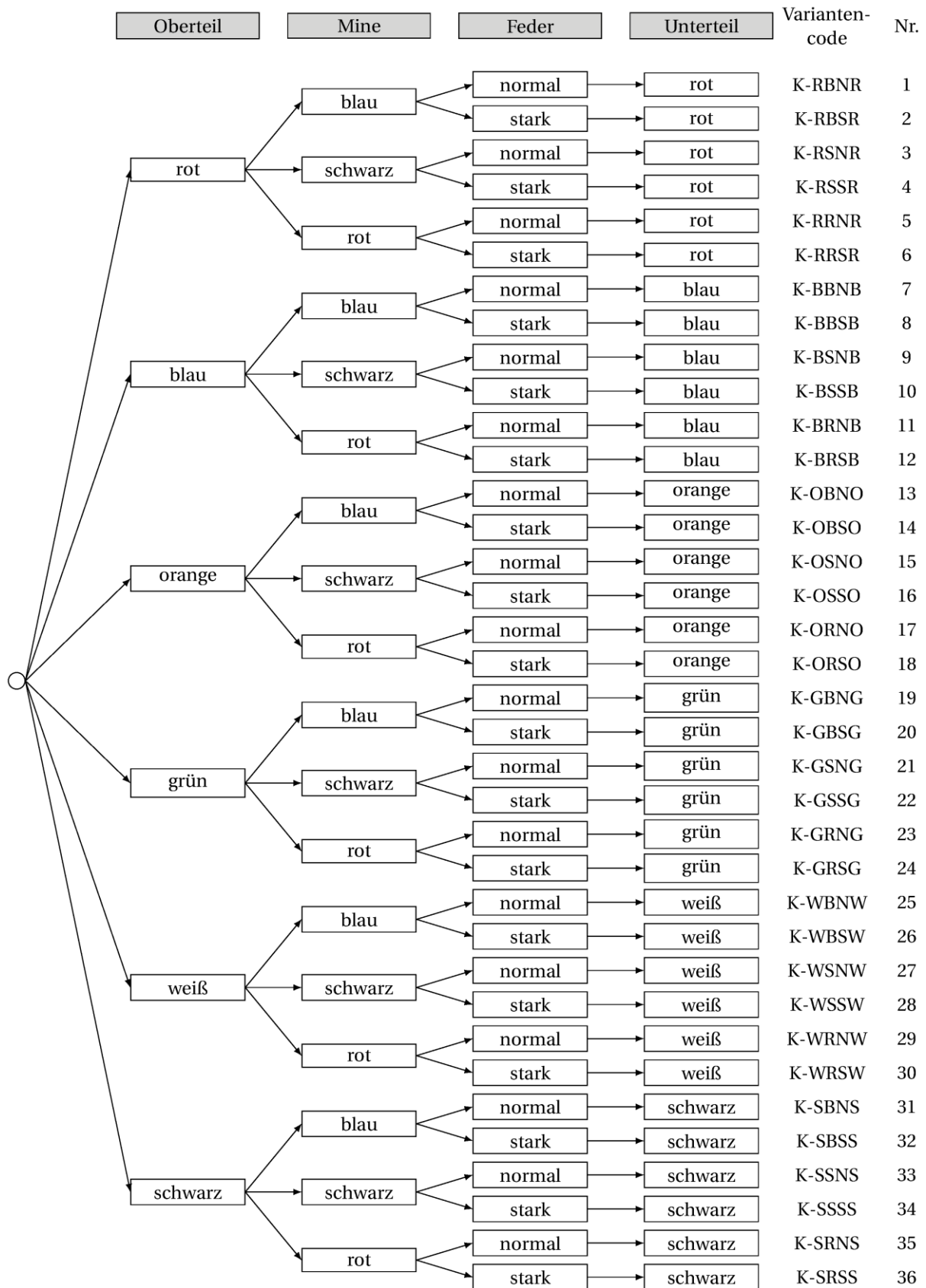


Bild 5.10: Produktvariantenvielfalt der erweiterten Kugelschreiber-Szenarios
Product variant range of the extended ballpoint pen scenario

Tabelle 5.9: Stationskapazitäten ohne (1. PG) und mit den neuen Bauteilvarianten (2.PG)
Station capacity without (1. PG) and with the new part variants (2 PG)




		Oberteile	Mine	Feder	Unterteil
Linie München		M010	M020	M020	M030
	1. PG	2	1	1	2
	2. PG	6	3	2	6
	Kapa.	5	2	2	3
Linie Saarbrücken		S010	S020	S020	S030
	1. PG	2	1	1	2
	2. PG	6	3	2	6
	Kapa.	4	3	1	4
Linie Atlanta		A010	A010	A020	A020
	1. PG				
	2. PG	6	3	2	6
	Kapa.	1	2	1	1

Legende:

1. PG: 1. Produktgeneration

2. PG: 2. Produktgeneration

Kapa.: Kapazität der Station

 Auslastung < Kapazität
 Auslastung = Kapazität
 Auslastung > Kapazität

Ausgangspunkt für die Montageplanung ist, dass jedes Produkt auf jeder Linie hergestellt wird. Hieraus ergeben sich einige Konflikte. In Tabelle 5.9 werden die Stationsauslastungen und -kapazitäten miteinander verglichen. Die Stationskapazität wird häufig überschritten. Im Folgenden soll mithilfe der von Tabelle 5.10, Tabelle 5.11 und Tabelle 5.12 manuell überprüft werden, ob das Modell korrekte Ergebnisse liefert. Die Tabellen sind dabei keine neue Auswertung, sondern dienen nur der Visualisierung der Auswertungen aus den Abschnitten 4.4.4 und 4.4.6.

Neben den Kapazitäten sind auch einige Anforderungen nicht durch Fähigkeiten abgedeckt. Im Beispiel kann das orangene Unter- und Oberteil nicht richtig erkannt werden. Die Fähigkeit orange Teile zu erkennen ist nicht gegeben. Die rote Mine ist dünner als die schwarze und blaue und kann daher nicht durch die vorhandenen Betriebsmittel gegriffen werden. Für die starke Feder sind hohe Verschraubkräfte erforderlich. Diese können am Standort in Atlanta nicht durchgeführt werden.

Tabelle 5.10 stellt diese Ausgangssituation inkl. der 17 vorhandenen Konflikte dar. Je Bauteilart und Station ist eine Spalte und je Standort drei Zeilen vorhanden. In der ersten Zeile jedes Standortes ist die Belegung dargestellt, also die Bauteilvariante, die nach der aktuellen Linienbelegung an diesem Standort an der Station benötigt werden. Hinter jeder Bauteilvariante ist dargestellt, ob die Anforderungen dieser Bauteilvariante durch die Fähigkeiten der Station abgedeckt sind. Diese Grafik dient nur zur Durchführung der Validierung, da sich in ihr die Schritte des Modells nachvollziehen und überprüfen lassen. Für reale Anwendungsfälle wird diese Grafik unübersichtlich und ist daher nicht Teil der Methode.

Tabelle 5.10: Ausgangssituation: Jede Linie stellt jedes Produkt her

Initial situation: Every assembly line produces every product

		Oberteil	Mine	Feder	Unterteil
		M010	M020	M020	M030
München	Belegung	Rot <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>	Normal <input checked="" type="checkbox"/>	Rot <input checked="" type="checkbox"/>
		Blau <input checked="" type="checkbox"/>	Schwarz <input checked="" type="checkbox"/>	Stark <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>
		Orange <input type="checkbox"/>	Rot <input type="checkbox"/>		Orange <input type="checkbox"/>
		Grün <input checked="" type="checkbox"/>			Grün <input checked="" type="checkbox"/>
		Weiß <input checked="" type="checkbox"/>			Weiß <input checked="" type="checkbox"/>
		Schwarz <input checked="" type="checkbox"/>			Schwarz <input checked="" type="checkbox"/>
	Anzahl	6 <input type="checkbox"/>	3 <input type="checkbox"/>	2 <input checked="" type="checkbox"/>	6 <input type="checkbox"/>
	Kapazität	5 <input type="checkbox"/>	2 <input type="checkbox"/>	2 <input checked="" type="checkbox"/>	3 <input type="checkbox"/>
Saar-brücken	Belegung	Rot <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>	Normal <input checked="" type="checkbox"/>	Rot <input checked="" type="checkbox"/>
		Blau <input checked="" type="checkbox"/>	Schwarz <input checked="" type="checkbox"/>	Stark <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>
		Orange <input type="checkbox"/>	Rot <input type="checkbox"/>		Orange <input type="checkbox"/>
		Grün <input checked="" type="checkbox"/>			Grün <input checked="" type="checkbox"/>
		Weiß <input checked="" type="checkbox"/>			Weiß <input checked="" type="checkbox"/>
		Schwarz <input checked="" type="checkbox"/>			Schwarz <input checked="" type="checkbox"/>
	Anzahl	6 <input type="checkbox"/>	3 <input checked="" type="checkbox"/>	2 <input type="checkbox"/>	6 <input type="checkbox"/>
	Kapazität	4 <input type="checkbox"/>	3 <input checked="" type="checkbox"/>	1 <input type="checkbox"/>	4 <input type="checkbox"/>
Atlanta	Belegung	Rot <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>	Normal <input checked="" type="checkbox"/>	Rot <input checked="" type="checkbox"/>
		Blau <input checked="" type="checkbox"/>	Schwarz <input checked="" type="checkbox"/>	Stark <input type="checkbox"/>	Blau <input checked="" type="checkbox"/>
		Orange <input checked="" type="checkbox"/>	Rot <input type="checkbox"/>		Orange <input checked="" type="checkbox"/>
		Grün <input checked="" type="checkbox"/>			Grün <input checked="" type="checkbox"/>
		Weiß <input checked="" type="checkbox"/>			Weiß <input checked="" type="checkbox"/>
		Schwarz <input checked="" type="checkbox"/>			Schwarz <input checked="" type="checkbox"/>
	Anzahl	6 <input type="checkbox"/>	3 <input checked="" type="checkbox"/>	2 <input type="checkbox"/>	6 <input type="checkbox"/>
	Kapazität	1 <input type="checkbox"/>	3 <input checked="" type="checkbox"/>	1 <input type="checkbox"/>	1 <input type="checkbox"/>

Legende:	<input type="checkbox"/>	Anforderung nicht durch Fähigkeit abgedeckt	<input checked="" type="checkbox"/>	Anforderung durch Fähigkeit abgedeckt
-----------------	--------------------------	---	-------------------------------------	---------------------------------------

In der zweiten Zeile zu jedem Standort ist die Anzahl der Bauteilvarianten aufgeführt. Darunter ist die Kapazität der Station dargestellt. Hier gibt der Haken am Zeilenende an, ob die Kapazität ausreicht, um die Anzahl an Bauteilvarianten anzustellen. Diese Tabelle stellt also eine Erweiterung zu Tabelle 5.9 dar. Im Folgenden soll diese Darstellung verwendet werden, um den Fortschritt bei der Eliminierung von Konflikten darzustellen, da hier sowohl Anforderungen und Fähigkeiten als auch Bauteilvariantenanzahl und Kapazitäten dargestellt sind.

Tabelle 5.11: Erste Option: Anpassung der Linienbelegung

First option: Changing the assembly line allocation

		Oberteil	Mine	Feder	Unterteil
		M010	M020	M020	M030
München	Belegung	Rot <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>	Normal <input checked="" type="checkbox"/>	Rot <input checked="" type="checkbox"/>
		Blau <input checked="" type="checkbox"/>	Schwarz <input checked="" type="checkbox"/>	Stark <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>
	Anzahl	5 <input type="checkbox"/>	2 <input checked="" type="checkbox"/>	2 <input checked="" type="checkbox"/>	5 <input type="checkbox"/>
		Orange	Rot		Orange
		Grün <input checked="" type="checkbox"/>			Grün <input checked="" type="checkbox"/>
		Weiß <input checked="" type="checkbox"/>			Weiß <input checked="" type="checkbox"/>
		Schwarz <input checked="" type="checkbox"/>			Schwarz <input checked="" type="checkbox"/>
Kapazität		5	2	2	3
Saar-brücken	Belegung	Rot <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>	Normal <input checked="" type="checkbox"/>	Rot <input checked="" type="checkbox"/>
		Blau <input checked="" type="checkbox"/>	Schwarz <input checked="" type="checkbox"/>	Stark <input checked="" type="checkbox"/>	Blau <input checked="" type="checkbox"/>
	Anzahl	4 <input checked="" type="checkbox"/>	3 <input checked="" type="checkbox"/>	2 <input type="checkbox"/>	4 <input checked="" type="checkbox"/>
		Orange	Rot <input type="checkbox"/>		Orange
		Grün <input type="checkbox"/>			Grün <input type="checkbox"/>
		Weiß <input checked="" type="checkbox"/>			Weiß <input checked="" type="checkbox"/>
		Schwarz <input checked="" type="checkbox"/>			Schwarz <input checked="" type="checkbox"/>
Kapazität		4	3	1 <input type="checkbox"/>	4 <input checked="" type="checkbox"/>
Atlanta	Belegung	Rot <input type="checkbox"/>	Blau <input checked="" type="checkbox"/>	Normal <input checked="" type="checkbox"/>	Rot <input type="checkbox"/>
		Blau <input type="checkbox"/>	Schwarz <input checked="" type="checkbox"/>	Stark <input type="checkbox"/>	Blau <input type="checkbox"/>
	Anzahl	1 <input checked="" type="checkbox"/>	3 <input checked="" type="checkbox"/>	2 <input type="checkbox"/>	1 <input checked="" type="checkbox"/>
		Orange <input checked="" type="checkbox"/>	Rot <input type="checkbox"/>		Orange <input checked="" type="checkbox"/>
		Grün <input type="checkbox"/>			Grün <input type="checkbox"/>
		Weiß <input type="checkbox"/>			Weiß <input type="checkbox"/>
		Schwarz <input type="checkbox"/>			Schwarz <input type="checkbox"/>
Kapazität		1	3	1 <input type="checkbox"/>	1 <input checked="" type="checkbox"/>

Legende	▲	Anzahl / Kapazität gesteigert
	▼	Anzahl / Kapazität verringert

Aus der Produktkombinatorik ist bekannt, dass grüne Gehäuse nicht zusammen mit roten Minen angeboten werden sollen. Die rote Mine führt in der Linie München zu einer Überschreitung der Kapazität, in Saarbrücken nicht. Folglich wird die rote Mine aus München gestrichen und das grüne Gehäuse aus Saarbrücken. Kugelschreiber mit orangem Gehäuse werden wenig nachgefragt und daher in der Linienbelegung in die Kleinserienfertigung in Atlanta verlagert. Dafür werden alle anderen Produktvarianten aus Atlanta entfernt. Das Resultat dieser Anpassungen ist in Tabelle 5.11 dargestellt. Durch die Schritte konnten 11 der 17 Konflikte gelöst werden. Die Stellen, an denen sich die Kapazität oder die Anzahl der Bauteilvarianten im Vergleich zum vorherigen Schritt verändert haben, sind durch nach unten oder oben zeigende Dreiecke markiert.

Die Anforderungen der roten Mine stellen sowohl in Atlanta als auch in Saarbrücken eine Herausforderung dar. Daher wird an dieser Stelle die Anforderung so angepasst, dass sie zu denen der blauen und schwarzen Mine passt.

Die Kapazitäten an den Station M040 und S030 sind noch überschritten. Hier soll den Kunden die volle Produktvielfalt geboten werden, daher werden Investitionen getätigt und die Kapazitäten an den Stationen erweitert, die Ressourcen werden also angepasst.

Folglich verbleiben noch zwei Konflikte in Station A020: Einerseits können die Anforderungen der starken Feder nicht durch Fähigkeiten abgedeckt werden, andererseits ist die Kapazität überschritten. Da die orangen Kugelschreiber ohnehin wenig nachgefragt werden, entscheidet sich die Firma gegen die erforderlichen Investitionen und entfernt die Produktvariante Oranger Kugelschreiber mit starker Feder aus dem Produktprogramm. Im Ergebnis sind alle Konflikte aufgelöst (vgl. Tabelle 5.12) und der Montagesystemverbund kann in der aktuellen Konfiguration betrieben werden. Alle Stellhebel aus Abschnitt 4.1 sind zum Einsatz gekommen, um die Konflikte zu beseitigen.

Tabelle 5.12: Situation nach allen beschriebenen Anpassungen*Situation after applying all described changes*

		Oberteil	Mine	Feder	Unterteil				
		M010	M020	M020	M030				
München	Belegung	Rot	<input checked="" type="checkbox"/>	Blau	<input checked="" type="checkbox"/>	Normal	<input checked="" type="checkbox"/>	Rot	<input checked="" type="checkbox"/>
		Blau	<input checked="" type="checkbox"/>	Schwarz	<input checked="" type="checkbox"/>	Stark	<input checked="" type="checkbox"/>	Blau	<input checked="" type="checkbox"/>
		Orange		Rot				Orange	
Grün		<input checked="" type="checkbox"/>					Grün	<input checked="" type="checkbox"/>	
Weiß		<input checked="" type="checkbox"/>					Weiß	<input checked="" type="checkbox"/>	
Schwarz	<input checked="" type="checkbox"/>					Schwarz	<input checked="" type="checkbox"/>		
Anzahl	5	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	5	<input checked="" type="checkbox"/>	
Kapazität	5	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	5▲	<input checked="" type="checkbox"/>	
		S010	S020	S030	S030				
Saar-brücken	Belegung	Rot	<input checked="" type="checkbox"/>	Blau	<input checked="" type="checkbox"/>	Normal	<input checked="" type="checkbox"/>	Rot	<input checked="" type="checkbox"/>
		Blau	<input checked="" type="checkbox"/>	Schwarz	<input checked="" type="checkbox"/>	Stark	<input checked="" type="checkbox"/>	Blau	<input checked="" type="checkbox"/>
		Orange		Rot	<input checked="" type="checkbox"/>			Orange	
Grün							Grün		
Weiß		<input checked="" type="checkbox"/>					Weiß	<input checked="" type="checkbox"/>	
Schwarz	<input checked="" type="checkbox"/>					Schwarz	<input checked="" type="checkbox"/>		
Anzahl	4	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	2	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	
Kapazität	4	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	2▲	<input checked="" type="checkbox"/>	4	<input checked="" type="checkbox"/>	
		A010	A010	A020	A020				
Atlanta	Belegung	Rot		Blau	<input checked="" type="checkbox"/>	Normal	<input checked="" type="checkbox"/>	Rot	
		Blau		Schwarz	<input checked="" type="checkbox"/>	Stark		Blau	
		Orange	<input checked="" type="checkbox"/>	Rot	<input checked="" type="checkbox"/>			Orange	<input checked="" type="checkbox"/>
Grün							Grün		
Weiß							Weiß		
Schwarz						Schwarz			
Anzahl	1	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	1▼	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	
Kapazität	1	<input checked="" type="checkbox"/>	3	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>	

5.3 Einsatz des Modells in der Montageplanung für Baukastenmotoren

Use of the model in assembly planning for modular engines

Das fiktive Beispiel hat gezeigt, dass das Modell wie in Kapitel 4 beschrieben umgesetzt werden kann und Montageplaner beim Treffen von Entscheidungen unterstützt. Dieser Teil der Validierung ist somit abgeschlossen. Im Folgenden soll das Modell und die implementierte Datenbank im praktischen Einsatz getestet werden.

Dazu wird zunächst auf die Rahmenbedingungen in der Motormontage eingegangen (vgl. Abschnitt 5.3.1). Beim praktischen Einsatz des Modells in bestehenden Produktionsstrukturen ergaben sich weitere Herausforderungen, die in Abschnitt 5.3.3 näher beschrieben werden.

In den darauffolgenden Abschnitten wird der praktische Einsatz des Modells anhand von real auftretenden Fragenstellungen demonstriert (vgl. Abschnitte 5.3.4-5.3.6)

5.3.1 Rahmenbedingungen in der Motorenmontage

Overall conditions in engine assembly

Als Anwendungsfall für diese Arbeit dient die Montage von PKW-Motoren der BMW AG. Die konkreten Werte wurden dabei aus Datenschutzgründen modifiziert. Die betrachteten Montagesysteme produzieren die sogenannten Baukastenmotoren, die im folgenden Abschnitt 5.3.2 vorgestellt werden. Ein Montagesystem besteht aus mehreren (Montage-)Linien, die sich teilweise in mehrere Montageabschnitte, sogenannte Gewerke, gliedern. Die BMW-Motormontage gliedert sich in die Gewerke Zylinderkopf, Short-Block, Rumpfmotor, Fertigmotor und Prüffeld. In den Linien sind automatische, manuelle und hybride Montagekonzepte im Einsatz (vgl. Bild 2.8). Die Motoren bleiben vor dem Werker stehen, bis dieser seine Arbeit verrichtet hat. Nach der Freigabe durch den Werker werden die Motoren durch die automatisierte Verkettung zum nächsten Arbeitsplatz bewegt.

Die Montagesysteme sind so ausgelegt, dass mehrere Motorderivate hergestellt werden können. So ist eine Linie bspw. in der Lage, alle Varianten von Otto Drei- und Vierzylindermotoren herzustellen. Die Linien sind so gestaltet, dass sie in Losgröße 1 betrieben werden können. Die herstellbaren Motoren können in beliebiger Reihenfolge eingesteuert werden. Durch diese Flexibilität kann BMW auf die in der Einleitung beschriebenen Turbulenzen und Nachfrageschwankungenmatrix reagieren.

Geplant werden die Montagesysteme von der Planung Montage Reihenmotoren. Diese bewertet alle Produktänderungen und beeinflusst das Produkt und die Linienbelegung. Sie plant zudem die erforderlichen Anpassungen der Montagesysteme und steuert deren Umsetzung. Folglich ist sie für die Auswahl und Kombination der Stellhebel (vgl. Abschnitt 4.1) zuständig.

5.3.2 Baukastenmotoren

Modular engines

Mit dem Baukastenmotor wird dem Kunden eine große Vielfalt an Produktvarianten angeboten, ohne dass die Bauteilvariantenvielfalt in der Produktion gleichermaßen ansteigt. Grundgedanke des Baukastenansatzes ist es, austauschbare Module zu erstellen und die Kundenvarianz durch die Kombination dieser Module zu erzeugen. Dazu müssen einheitliche Schnittstellen zwischen den Modulen definiert werden. Die

Module werden dabei standardisiert und in allen Produkten zu verwenden. [GRAE04, S. 6; STE114, S.31]

Ein sehr einfaches Beispiel kann der Verbund aus Kolben und Pleuel eines Verbrennungsmotors sein. Wenn die Zylinderbohrungen im Kurbelgehäuse identische Durchmesser haben und der Hub in allen Motoren identisch ist, dann kann ein Verbund von Kolben und Pleuel für alle Ottomotoren verwendet. Ebenso ist es denkbar, den Ventilttrieb über alle Motoren zu vereinheitlichen und nur die Anzahl gewisser Bauteile mit der Zylinderzahl zu variieren. [STE114, LAND15]

Dem Kunden können somit verschiedene Zylinderzahlen und Leistungsstufen angeboten werden, ohne dass die Bauteilvielfalt in der Produktion gleichermaßen ansteigt. Dabei werden alle Bauteile in die Überlegung einbezogen. [STE114, LAND15]

Zur Benennung der verschiedenen BMW Baukastenmotoren gibt es einen Varianten-code. Dieser ist so ausgeführt, dass jede Stelle eine Eigenschaft des Motors repräsentiert. Die Bedeutung jeder Stelle ist in Tabelle 5.13 dargestellt.

Aus dem Baukastenansatz ergeben sich einige Vorteile. Für die Entwicklung ergibt sich ein reduzierter Entwicklungsaufwand, da ein Modul nur einmal entwickelt werden muss. Für den Einkauf ergeben sich durch die Mehrfachverwendung der Module Skaleneffekte, was sich wiederum positiv auf den Einkaufspreis auswirkt. [VIET13, S. 864]

Tabelle 5.13: Variantencode für den Baukastenmotor

Variant code for the modular engine

	Motortyp	B: Baukastenmotor
	Zylinderanzahl	3: 3 Zylinder 4: 4 Zylinder
	Verbrennungsart	7: Diesel 8: Otto
	Einbaurichtung	A/C: Quereinbau B/D: Längseinbau
	Hubraum	15: 1,5 L 20: 2,0 L
	Leistungsstufe	U: Untere Leistungsstufe M: Mittlere Leistungsstufe O: Obere Leistungsstufe
	Produktgeneration	
B	3	8
A	15	U
		0

Für die Logistik ergeben sich geringere Lagerhaltungskosten, da weniger verschiedene Bauteilvarianten vorgehalten werden müssen. In der Montage können Mitarbeiter schneller angelernt werden, da sie weniger verschiedene Varianten erlernen müssen. Zusätzlich sinkt die Fehlerwahrscheinlichkeit und die Qualität steigt. Da weniger Varianten zu berücksichtigen sind, wird die Montageplanung ebenfalls einfacher. Die Automatikstationen in der Montage können bei weniger Varianten einfacher gestaltet werden, sodass die Komplexität sinkt.

Eine Herausforderung bei der Entwicklung und Produktion der Baukastenprodukte ist die deutlich höhere Vernetzung. Wenn eine Bauteilvariante in vielen Produkten und somit vielen Montagesystemen eingesetzt wird, ist ihre Änderungen mit deutlich erhöhtem Prüfaufwand verbunden.

5.3.3 Herausforderungen beim praktischen Einsatz des Modells

Challenges with the practical use of the model

Viele Informationen, die für die Durchführung der Methode und somit zur Befüllung des Modells erforderlich sind, liegen bereits vor – beispielsweise Stuecklisten, Prozesslisten und Variantenmatrizen (vgl. Bild 4.18). Jedoch liegen sie in Formen, die nicht maschinenlesbar sind. Weiterhin sind die gewählten Darstellungen häufig unvollständig oder nicht eindeutig und lassen sich folglich nur mit großem Hintergrundwissen korrekt interpretieren.

Für die Befüllung des Partialmodells Produkt muss bekannt sein, welche Bauteilarten und -varianten es gibt. Hierzu verwendet die BMW Montageplanung sogenannte Bauteilvariantenmatrizen. In diese werden alle Bauteilvarianten einer Bauteilart eingetragen. In ihrer ursprünglichen Form enthielten die Bauteilvariantenmatrizen nur einen Teil der Bauteilvarianten. Weiterhin sind die Bauteilvariantenmatrizen nicht nach einer einheitlichen und eindeutigen Logik aufgebaut, sodass sie sich nicht maschinell auslesen lassen. Im ersten Schritt wird eine Vorlage für diese Bauteilvariantenmatrizen erstellt. Im Rahmen dieser Arbeit wurden die Bauteilvariantenmatrizen so weiterentwickelt, dass sie alle Bauteilvarianten enthält und gleichzeitig sicherstellt, dass die Bauteilvarianten eindeutig den Produktvarianten und Bauteilarten zugeordnet werden können. Ein stark vereinfachtes Beispiel, in dem lediglich das Prinzip gezeigt werden soll, ist in Tabelle 5.14 dargestellt.

Tabelle 5.14: Prinzipdarstellung einer Bauteilvariantenmatrix*Schematic part variant matrix*

Motordefinition				Weitere Variantenmerkmale		
VBR	ZYL	EBR	V-Code	HS	A	...
Otto	3-Zylinder	Quer	B38A	SR1	X	...
		Längs	B38B	SR2	SR3	...
	4-Zylinder	Quer	B48A	SR1	SR3	...
		Längs	B48B	SR2	SR3	...
...

Legende:
VBR = Verbrennungsart
ZYL = Zylinderanzahl
EBR = Einbaurichtung
V-Code = Variantencode
HS = Handschalter
A = Automatik

Die grauen Zellen beinhalten dabei die Informationen, die zur Beschreibung der betroffenen Produktvariante erforderlich sind. Im Gegensatz zum Beispiel aus Tabelle 2.2 wird hier darauf geachtet, dass einheitliche Bezeichnungen verwendet werden und alle Produkt- und Bauteilvarianten aufgeführt werden. Die weißen Zellen beinhalten die zugehörige Bauteilvariante. Das erste Feld links oben im weißen Bereich beschreibt demnach die Bauteilvariante für das Produkt Otto 3-Zylindermotor im Quereinbau als Handschalter. Die für diese Produktvariante verwendete Bauteilvariante hat die Bezeichnung SR1. Diese Bauteilvariante wird ebenfalls in der Otto 4-Zylinder-Quereinbau Handschalter Produktvariante verwendet. Ein leeres Feld ist in dieser Matrix ein Hinweis auf einen Fehler oder auf eine Unvollständigkeit. Existiert eine Produktvariante nicht und gibt es daher keine Bauteilvariante, wird das Feld mit einem X gekennzeichnet und geschwärzt. Somit ist sichergestellt, dass kein Feld vergessen wird.

Eine Bauteilvariantenmatrix enthält jedoch keine Informationen darüber, welche Produkte tatsächlich hergestellt werden sollen. Diese Information ist in einem Variantenbaum (vgl. Bild 2.15) zu finden.

Variantenbäume sind im betrachteten Anwendungsfall zwar vorhanden, jedoch passt deren Systematik und Beschreibung nicht zu der der Bauteilvariantenmatrizen, was eine automatisierte Auswertung verhindert. In der Folge werden die Produktbezeichnungen in den Variantenbäumen und Bauteilvariantenmatrizen harmonisiert.

Eine weitere Schnittstelle zu den Produktbezeichnungen stellt die Linienbelegung dar. Auch diese verwendet eine andere Detailtiefe als der Variantenbaum und die Bauteilvariantenmatrizen. Der Detaillierungsgrad der Linienbelegung konnte so erhöht werden, dass er in allen drei Informationsquellen einheitlich ist.

Die Daten zu den Produktionsprozessen sind ebenfalls nicht auf eine automatisierte Verarbeitung ausgelegt. Zur Beschreibung der Prozesse ist ein Vorranggraph vorhanden, jedoch sind viele relevante Informationen in graphischer Form aufbereitet. Informationen über die Fähigkeiten waren in mehreren Systemen verteilt, nicht einheitlich und auch nicht vollständig.

In Abstimmung mit den betroffenen Personen wird eine Matrix geschaffen, in der die benötigten Informationen vereint werden. Durch die einheitliche Darstellung und Überführung in ein System lässt sich schnell prüfen, ob alle benötigten Informationen vorhanden sind. Weiterhin wird die Matrix so gestaltet, dass sich die Verknüpfung zwischen den zusammenhängenden Daten leicht herstellen lässt.

Nachdem alle Daten in eine maschinenlesbare Form gebracht worden sind, können sie in die Datenbank eingelesen werden. Einen Sonderfall stellen die Bauteilvariantenmatrizen dar. Durch ihren Aufbau in Form einer Matrix müssen diese interpretiert werden, bevor sie in die Datenbank geschrieben werden können. Alle Herausforderungen sind zusammengefasst in Bild 5.11 dargestellt.

Die initiale Befüllung der Datenbank ist damit möglich. In Abschnitt 2.4 werden einige Szenarien definiert, die aktuell Herausforderungen für die Montageplanung darstellen. Anhand der definierten Szenarien werden mithilfe der befüllten Datenbank verschiedene Ergebnisse ermittelt.

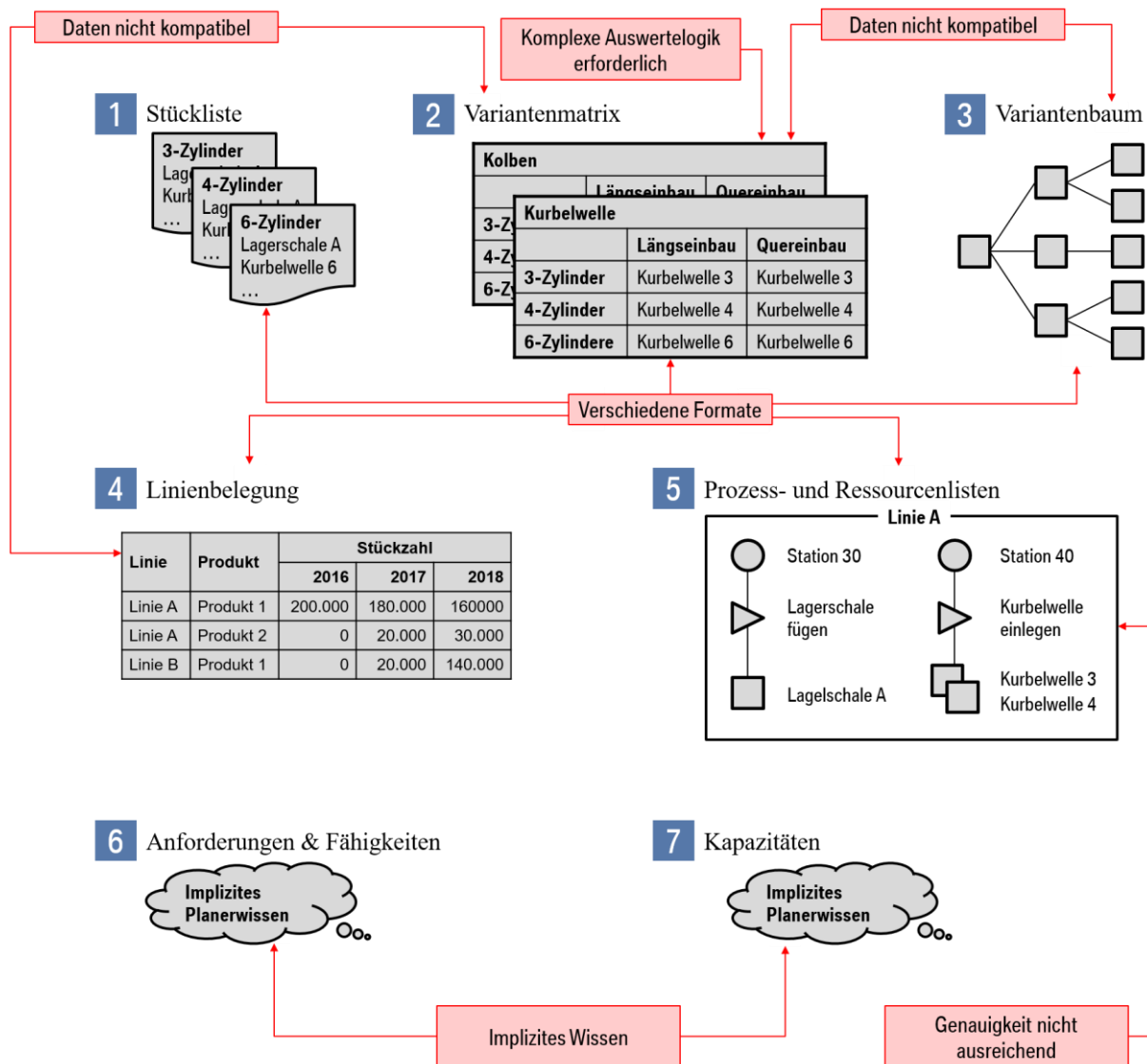


Bild 5.11: Herausforderungen beim Einlesen der Daten
Challenges with reading the input files

5.3.4 Variantenmanagement und Stationskapazität

Variant management and station capacity

Die in Abschnitt 4.4 vorgestellten Auswertungen und Diagramme lassen sich für mehrere Zwecke nutzen. Hier soll nur ein Auszug aus den vielen Möglichkeiten gegeben werden.

Zur Einführung der neuen Motorgeneration werden zunächst Auswertungen basierend auf der Produktvielfalt erstellt. Eine erste Übersicht bietet die Auswertung „Bauteilvarianten in Produktgenerationen“ (vgl. Abschnitt 4.4.1)

Bild 5.12 zeigt das Ergebnis der Auswertung für ausgewählte Bauteilarten. Es ist sowohl die absolute Bauteilvariantenvielfalt gut ablesbar als auch die Überlappung. Beim

Abgasturbolader (ATL) und Kurbelgehäuse ist die absolute Vielfalt mit je 25 Bauteilvarianten über beide Generationen relativ groß und es können keine Teile in die neue Generation übernommen werden. Dennoch wird bei beiden Bauteilen eine Optimierung erreicht: Beim ATL wird von 13 auf 12 Varianten reduziert und beim Kurbelgehäuse von 14 auf 11. Nach Auslaufen der ersten Generation sind somit weniger Varianten vorhanden als vor Anlauf der zweiten Generation.

Eine teilweise Übernahme von Bauteilen ist bei der Einlass-Nockenwelle gelungen. Hier können zwei Teile in die neue Generation übernommen werden. Somit ergeben 7 Bauteile in der ersten Generation und 5 Bauteile in der zweiten Generation im Überlappungszeitraum 10 anstatt 12 Bauteilvarianten.

Beim Kurbelwellensensor und der Rückstellfeder werden die Bauteilvarianten vollständig aus der ersten Generation übernommen.

Ein aus Sicht der Montage guter Fall liegt beim Auslassventil vor. Hier können alle Varianten für die neue Generation aus der alten übernommen werden. Darüber hinaus sind von den 6 Varianten, die in der ersten Generation benötigt werden in der zweiten Generation nur noch 3 erforderlich.

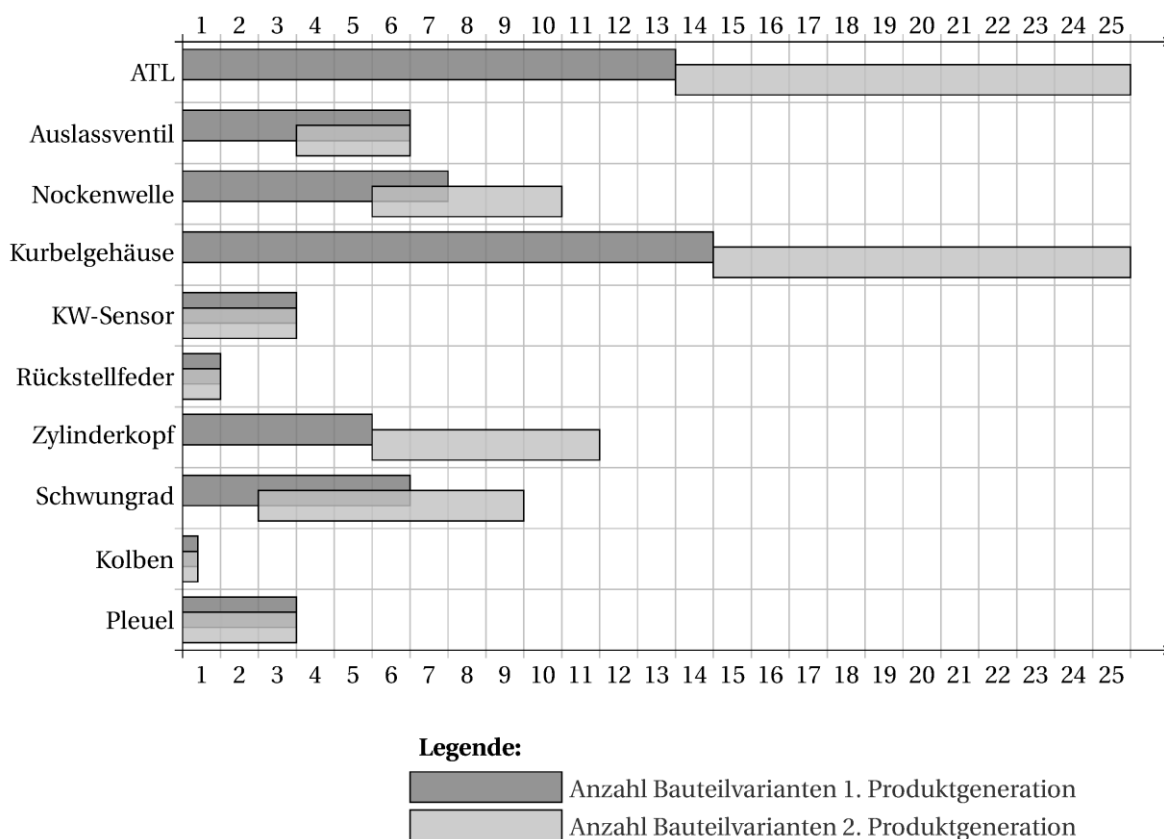


Bild 5.12: Anzahl Bauteilvarianten zweier Produktgenerationen je Bauteilart

Amount of part variants for two product generations for each part type

Eine Übersicht zur Bauteilvarianten-Entwicklung bietet das Ringdiagramm aus Abschnitt 4.4.2. Untersucht werden die wichtigsten 90 Bauteilarten des Baukastenmotors. Dabei wird je Bauteilart ermittelt, wie sich die Anzahl der Bauteilvarianten dieser Bauteilart von der 0. Produktgeneration zur 1. Produktgeneration ändert.

Bei 14 Bauteilarten werden die Bauteilvarianten aus der ersten Produktgeneration übernommen. An Prozesse und Stationen mit diesen Bauteilarten sind demnach keine Änderungen erforderlich – vorausgesetzt die Linienbelegung der zweiten Produktgeneration und der ersten Produktgeneration sind identisch.

Bei 54 Bauteilarten ist eine Optimierung gelungen. Dies wird zum einen durch Eliminierung von Produktvarianten realisiert. Zum anderen werden Bauteilvarianten so optimiert, dass sie produktübergreifend zum Einsatz kommen können. Wo in der ersten Generation noch zwei Bauteilvarianten erforderlich waren, ist in der zweiten Generation nur noch eine Bauteilvariante erforderlich. Dennoch besteht im Überlappungszeitraum eine Variantenmehrung. Die Montageplanung muss analysieren, ob diese Variantenmehrungen zu Konflikten führen. Sollte dies der Fall sein, muss sie Lösungen für diese Konflikte erarbeiten.

Bei 12 Bauteilarten entsprach die Anzahl der Bauteilvarianten der zweiten Produktgeneration derjenigen der ersten Produktgeneration. Je Generation ist die Bauteilvariantenzahl zwar identisch, in der Überlappung muss jedoch die doppelte Anzahl verarbeitet werden.

Bei 10 Bauteilarten kommen mehr neue Bauteilvarianten hinzu, als in der ersten Produktgeneration vorhanden sind. Hier ist auch nach Auslauf der ersten Generation noch eine Variantenmehrung vorhanden.

Bild 5.13 fasst die genannten Zahlen in einem Ringdiagramm gemäß Bild 4.15 zusammen, um die beschriebenen Verhältnisse kompakt darzustellen.

Das Ringdiagramm gibt den Planern und dem Management einen ersten Eindruck, ob bei der Integration der neuen Generation mit großen oder geringen Planungsaufwänden und Investitionen zu rechnen ist. Fallen viele Bauteilarten in die Kategorien *Übernahme & Rückwärtskompatibel* oder *Optimierung*, wie in Bild 5.13 beispielhaft dargestellt, so sollte der Aufwand vergleichsweise gering sein.

Die Aussage über die Entwicklung der Bauteilvariantenvielfalt aus Bild 5.13 und Bild 5.12 gibt einen ersten Hinweis, wo Probleme entstehen können. Jedoch lässt sich eine verlässliche Aussage nur treffen, wenn die Vielfalt mit der installierten Kapazität abgeglichen wird.

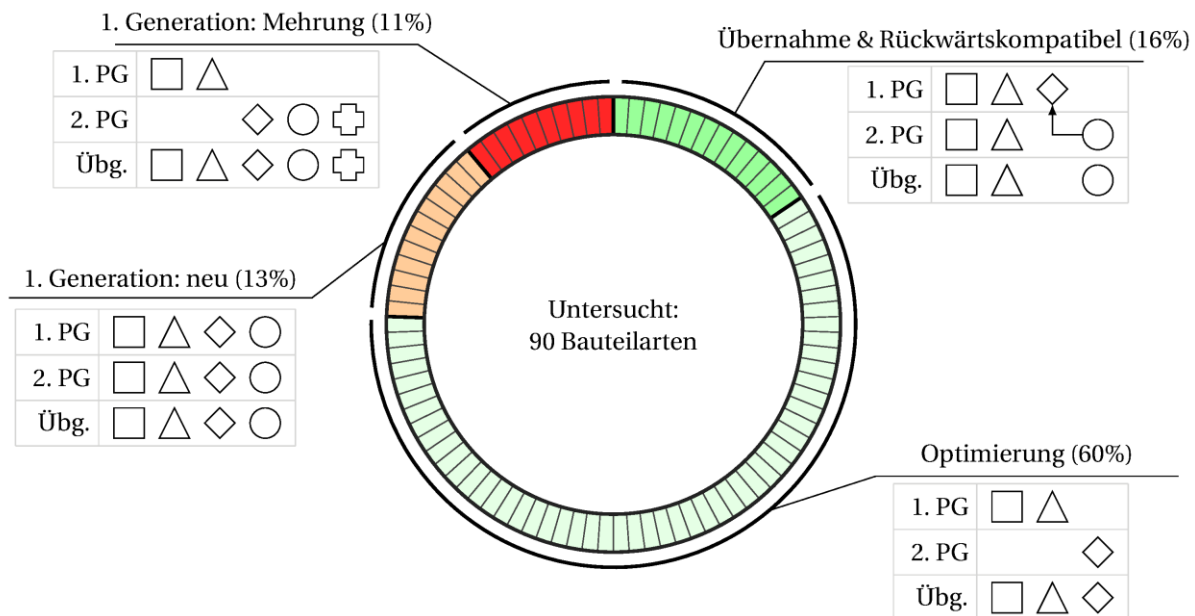


Bild 5.13: Entwicklung der Bauteilvariantenvielfalt

Evolution of the variety of parts

Hierfür sind zwei weitere Informationen notwendig: Die Kapazität der Stationen muss bekannt sein und es muss eine Linienbelegung definiert werden. Als Kapazität werden die aktuell an den Linien vorhandenen Kapazitäten übernommen, als Linienbelegung wird angenommen, dass diese analog der ersten Produktgeneration sein wird. Somit lassen sich erste Untersuchungen anstellen. Sollten sich die beiden getroffenen Annahmen später als falsch herausstellen, so lassen sie sich im Modell leicht anpassen.

Ein einfaches Beispiel für eine Kapazitätsauswertung enthält Tabelle 5.15. Hierin sind zwei Linien – Linie A und Linie B – dargestellt. Entlang der X-Achse werden verschiedene Bauteilarten aufgeführt und nach den Gewerken in denen sie verbaut werden sortiert. Für jede Generation gibt es innerhalb jeder Linie eine eigene Zeile. In diesen Zeilen ist die Anzahl an Bauteilvarianten für diese Generation dargestellt – unter Berücksichtigung der Linienbelegung für die entsprechende Linie.

Die ersten drei Bauteilarten sind beispielhaft befüllt, um die Logik erläutern zu können. Für Bauteilart 1 ergibt sich in Linie A in der ersten Generation eine Bauteilvariante und in der zweiten Generation ebenfalls eine Bauteilvariante. Beide Bauteilvarianten sind unterschiedlich, weshalb sich im Zeitraum der Überlappung zwei Bauteilvarianten ergeben. Da in der Linie an dieser Stelle nur die Kapazität für eine Bauteilvariante vorhanden ist, ist das Feld in der Zeile der Überlappung rot eingefärbt (für die Legende s. Tabelle 5.9).

Bei der zweiten Bauteilart gibt es 6 Bauteilvarianten in der ersten Generation und 5 in der zweiten Generation. Jedoch ist eine Bauteilvariante ein Übernahmeteil und folglich in beiden Generationen enthalten, weshalb sich als Summe der Bauteilvarianten für beide Generationen 10 und nicht 11 ergibt. Noch stärker ist dieser Effekt bei Bauteilart

3. Hier gibt es je Generation 2 Bauteilvarianten. Da es sich jedoch um dieselben 2 Bauteilvarianten handelt, liegen auch während der Überlappungsphase 2 Bauteilvarianten vor.

Diese Auswertung wird für jede Linie vorgenommen. Anhand der Farben und der rechten Spalte *Konflikte* lassen sich erkennen, welche Probleme noch zu lösen sind. Als Lösungsoptionen stehen wieder die Stellhebel aus Abschnitt 4.1 zur Verfügung.

Tabelle 5.15: Stationsauslastung zweier Linien

Station occupancy of two assembly lines

		Zylinderkopf			Short-Block			Rumpfmotor		Fertigmotor			Konflikte
		Zylinderkopf			Kurbelgehäuse				Ölpumpe				
Linie A		ZAA010	SAA010	RAA110
	1. Gen.	3	6	2	0
	2. Gen.	3	5	2	0
	Überlapp.	6	10	2	1
	Max. Kapazität	5	10	4	
Linie B		RBA110
	1. Gen.	2	8	2	0
	2. Gen.	1	9	2	1
	Überlapp	3	13	2	1
	Max. Kapa	3	8	2	

5.3.5 Anpassung der Linienbelegung

Adjustment of line allocation

Die Anpassung der Linienbelegung ist einer der Stellhebel der Integrationsplanung. (s. Schritt 5 aus Bild 4.1 und Abschnitt 4.1).

Wie in Abschnitt 4.2.3 beschrieben, ordnet die Linienbelegung bzw. -befähigung die Produkte den Linien zu. Für den konkreten Anwendungsfall wird hierin festgelegt, welche Montagelinie welchen Baukastenmotor herstellt. Dies hat weitreichende Konsequenzen, da mit jedem neuen Produkte potenziell neue Bauteilarten und insbesondere zusätzliche Bauteilvarianten an die Linie kommen.

Aufgrund der Baukastenlogik der in dieser Arbeit betrachteten Produkte und der zahlreichen Abhängigkeiten ist es ohne softwarebasierte Modellunterstützung nur mit erheblichem Aufwand möglich die tatsächlichen Auswirkungen der Änderungen der Linienbelegung zu untersuchen. Infolgedessen wurden große Anpassungen an der Linienbelegung in der Vergangenheit vermieden. Das Anpassen der Linienbelegung kann jedoch häufig eine sinnvolle und kostengünstige Lösung sein. Daher ist es ein wichtiger Mehrwert des entwickelten Modells, dass die Auswirkungen der Anpassung der Linienbelegung vollumfänglich betrachtet werden können.

Die Bedeutung der Belegungsplanung für die Motormontage sei im Folgenden an einigen Beispielen erläutert, die im Rahmen dieser Arbeit untersucht wurden:

1. Verschiebung des End of Production (EOP) eines Motors

Es kommt häufig vor, dass der EOP eines Motors verschoben wird und der Motor somit länger als geplant produziert werden muss. Da es sich hierbei meist nur um einige Wochen oder Monate handelt sollen Investitionen vermieden werden. Eine Anpassung des Produktes ist kurz vor dem Ende von dessen Herstellung ebenfalls nicht mehr sinnvoll. Als vorrangig zu betrachtende Handlungsoption bleibt somit die Anpassung der Linienbelegung.

Wie bereits beschrieben muss zunächst untersucht werden, ob mit dem geänderten EOP überhaupt ein Konflikt vorliegt. Dazu wird im Modell die Linienbelegung an den neuen EOP angepasst und die Ergebnisse der zwei Konfliktauswertungen (Kapazität und Anforderungen/Fähigkeiten) ermittelt.

Ergeben sich Konflikte durch die Verschiebung des EOP, so müssen verschiedene Szenarien untersucht werden. Zum einen kann probiert werden, die Produktionsvolumen innerhalb einer Linie zu verschieben. Das heißt, dass bestimmte Produkte vorproduziert werden und ihre Produktion somit früher enden oder ggf. einen Monat aussetzen kann. Im Montagesystemverbund wäre es denkbar, ein bestimmtes Produkt

temporär in eine andere Linie zu verschieben. Idealerweise können die Linien im Montagesystemverbund ohnehin mehrere Motoren gleichzeitig herstellen. Somit sollte die Verschiebung bestimmter Volumina kein Problem darstellen.

Tabelle 5.16 verdeutlicht das Vorgehen. Beispielhaft sei angenommen, dass der 3-Zylinder-Längsmotor länger produziert werden soll. Da dieser nur in Linie A hergestellt wird, wirkt sich die Verlängerung zunächst auf dieser Linie aus. Um die Verlängerung realisieren zu können, könnten ein oder mehrere Derivate oder ein gewissen Anteil des Volumens des Quermotors auf Linie B verlagert werden. Wenn sich daraus in Linie B Konflikte ergeben, könnten Teilumfänge dieser Linie – also Derivate des 4-Zylinder-Quermotors – in Linie C verlagert werden. Wenn dieses wiederum nicht in der Lage ist, alle Derivate herzustellen, ließen sich noch 4-Zylinder-Längsmotor Derivate in Linie D verschieben.

Dieses Beispiel ist ein Extremfall, der in dieser Ausprägung in der Praxis sehr selten vorkommt. Dennoch zeigt es, wie mit Hilfe eines gut gestalteten Baukastennetzwerks auf Veränderungen mithilfe der Linienbelegung reagiert werden kann. Zur schnellen Untersuchung der Änderungen können die Auswertungen des entwickelten Modells verwendet werden.

Tabelle 5.16: Verlagerung von Derivaten oder Produktionsvolumen innerhalb eines Produktionsnetzwerkes

Shifting of derivatives or production volume within the production network

	Linie A	Linie B	Linie C	Linie D
3-Zylinder-Quermotor	X →	X		
3-Zylinder-Längsmotor	X			
4-Zylinder-Quermotor		X →	X	
4-Zylinder-Längsmotor			X →	X
6-Zylinder-Längsmotor				X

2. Integration eines weiteren Motors in die Linie

Zum Alltagsgeschäft der Montageplanung gehört die Integration neuer Motorderivate in die bestehenden Linien. Soll mithilfe des Modells ein neuer Motor integriert werden, müssen dessen Daten zunächst in die Datenbank eingepflegt werden. Wenn durch den Motor neue Bauteilvarianten oder Bauteilarten entstehen, müssen diese ebenfalls in der Datenbank angelegt werden. Hierzu wird der Motor zunächst mithilfe der Variantenregeln einer Linie zugewiesen. Treten in der anschließenden Konfliktanalyse

Konflikte auf, sollte zunächst untersucht werden, ob diese durch Anpassung der Linienbelegung verringert oder vollständig eliminiert werden können (vergleiche vorheriges Beispiel).

Sollten auch nach Ausschöpfung aller Möglichkeiten der Motorverlagerung in andere Linien noch Konflikte bestehen, so bleiben als Möglichkeiten die Produktbeeinflussung und die Anpassung der Ressource.

3. Entfall eines Motors

Eine Möglichkeit des Variantenmanagements, Kosten einzusparen, ist der Entfall von Varianten. Im konkreten Fall hat BMW auf Fahrzeugebene untersucht, welche Fahrzeuge und Motoren wenig nachgefragt werden. Die Aufgabe der Montageplanung war die Bewertung, was der Entfall dieser Motorvarianten an Einsparung bringen würde.

Hierbei müssen zwei Fälle unterschieden:

- a) Der Motor wurde in eine / mehreren Linien integriert und kann jetzt entfallen.
- b) Der Motor wurde bisher noch nicht integriert und entfällt.

Es sind auch Fälle denkbar, in denen die Linie teilweise für den entfallenden Motor befähigt wurde. Wenn der Motor bereits in die Linie integriert wurde, dann sind die einzusparenden Investitionen gering. Es entfallen lediglich Lagerhaltungskosten und Logistikkosten. Einen großen Vorteil kann der Entfall von Motorderivaten auch nach deren Integration bringen, wenn sich dadurch in der Zukunft liegende Konflikte vermeiden lassen.

Deutlich sinnvoller aus Perspektive der Montageplanung ist eine Streichung von Motorvarianten bevor die Investition für deren Integration getätigt wurde. Ohne das in dieser Arbeit entwickelte Modell lässt sich jedoch auch zu diesem Fall nur schwer eine Aussage treffen. Es ist denkbar, dass durch den Entfall eines Motorderivates keine einzige Bauteilvariante eingespart werden kann. Somit blieben als Einsparung nur noch administrative Kosten, wie z.B. für die Pflege einer Motorsachnummer und geringe Einsparungen für das Motorlager. Es kann aber auch sein, dass durch den Entfall eines Motors viele Bauteilvarianten nicht mehr benötigt werden.

Wie viele Bauteilarten durch den Entfall des Motors an der Linie entfallen können, lässt sich mithilfe des Modells schnell und automatisch ermitteln. Hierzu kann die Abfrage *Exklusivteile* aus Abschnitt 4.4 verwendet werden. Somit bleibt dem Montageplaner nur noch die finanzielle Bewertung durch den Entfall dieser Bauteilvarianten.

4. Neugestaltung Linienbelegung

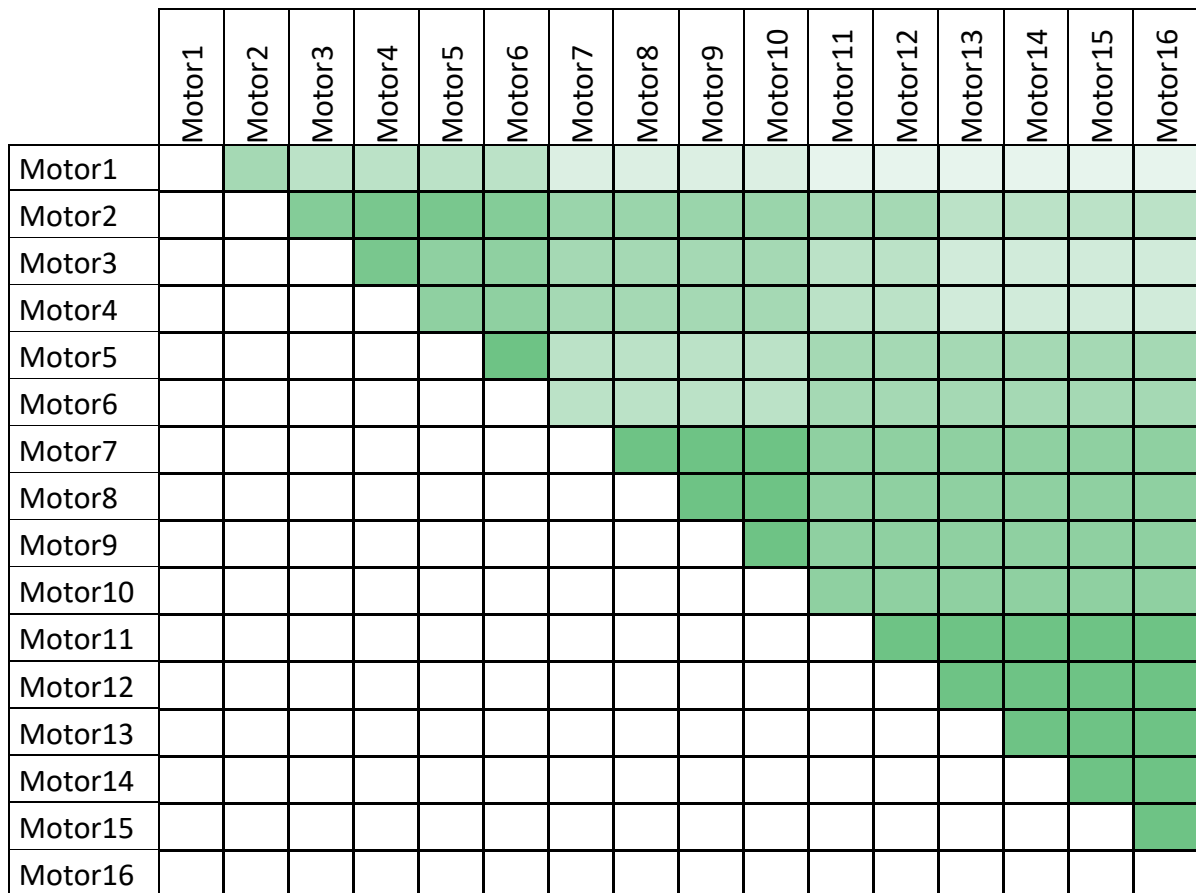
Je nachdem wie die aktuelle Linienbelegung entstanden ist oder wie viel sich seit deren Erstellung an den Produkten und deren Zusammensetzung aus Bauteilvarianten verändert hat, kann es sinnvoll sein, die Linienbelegung vollständig zu überdenken. Wie eingangs erwähnt sind hierbei nicht nur die Belange der Montageplanung relevant. Jedoch liefert sie einen wichtigen Input für deren Gestaltung.

Grundsätzlich sei davon ausgegangen, dass sich Produkte mit großer Ähnlichkeit günstiger in ein gemeinsames Montagesystem integrieren lassen als Produkte mit großen Unterschieden. Diese Annahme ist zwar nicht in jedem Fall, jedoch in den meisten Fällen, gültig.

Ein geeignetes Werkzeug für die Ermittlung der Gleichheit oder Kommunalität von Motoren ist die in Abschnitt 4.4 vorgestellte Heatmap (vgl. Tabelle 4.2). Aus ihr geht hervor, wie ähnlich sich die verschiedenen Motorvarianten in Bezug auf die in ihnen verbauten Bauteilvarianten sind (Stücklisten Ähnlichkeit). Ein Beispiel für eine solche Heatmap ist in Bild 5.14 dargestellt. Sehr ähnliche Motoren sollten in einer Linie hergestellt werden, Motoren, die große Unterschiede aufweisen, möglichst in verschiedenen. Übertragen auf das gezeigte Beispiel bedeutet dies, dass die Motoren 11 bis 16 in ein Montagesystem integriert werden sollten.

Wird die Kapazitätsgrenze in einer Linie nur für kurze Zeit überschritten oder tritt das Maximum der Auslastung nur für einen kurzen Zeitraum auf, so kann es ebenfalls sinnvoll sein, die Linienbelegung anzupassen. Die hierzu erforderlichen Schritte werden bereits ausführlich in Abschnitt 4.4, Bild 4.16, beschrieben. Sie gelten hier analog. Auf eine weitere Beschreibung wird daher verzichtet.

Eine weitere Möglichkeit zur effizienten Reduzierung von Konflikten in der Stations-Kapazitäts-Auswertung ist die Suche nach Exklusivbauteilvarianten, wie sie in Abschnitt 4.4 vorgestellt wird.



Legende: Hohe Übereinstimmung  Niedrige Übereinstimmung

Bild 5.14: Beispiel-Heatmap für einen Baukastenmotor

Exemplary Heatmap for a modular engine

5.3.6 Voruntersuchung Prozessabsicherung Montage

Ein wichtiger Prozessschritt für die Montageplanung ist die sogenannte Prozessabsicherung Montage. Hierin wird das virtuelle Produkt in einer frühen Phase auf seine Kompatibilität mit den bestehenden Montagesystemen untersucht.

Dazu werden Modelle der Betriebsmittel und Montageanlagen zusammen mit Produktmodellen in eine CAD Software geladen. Anschließend erfolgt manuell eine Untersuchung, ob Produkt und Ressource kompatibel sind.

An dieser Stelle kann mit dem Modell vorab eine wichtige Unterstützung geleistet werden, indem die Anforderungen und Fähigkeiten miteinander verglichen werden. Die hierin ermittelten Konflikte müssen nicht mehr manuell gefunden werden. Für eine Vordiskussion ist dies sehr hilfreich.

Je mehr Fähigkeiten und Anforderungen im Modell gepflegt sind, desto besser ist dessen Aussagekraft. Wie schwerwiegend die Überschreitung einer Fähigkeit ist, muss

durch den Montageplaner beurteilt werden. Anschließend kann eine der drei Handlungsoptionen der fähigkeitsbasierten Montageplanung (vgl. Abschnitt 4.1) gewählt werden.

5.4 Implementierung im Rahmen des SmartF-IT Projektes

Für diese Arbeit wurde das Modell in Form einer SQLite-Datenbank umgesetzt. Wie in Abschnitt 5.1.1 erläutert sind jedoch auch andere Umsetzungsalternativen möglich. Eine solche soll im Folgenden kurz vorgestellt werden.

Im Rahmen des SmartF-IT-Projektes, das vom Bundesministerium für Bildung und Forschung (BMBF) gefördert wurde, wurden „cyber-physische IT-Systeme zur Komplexitätsbeherrschung einer neuen Generation multiadaptiver Fabriken“ entwickelt. [SMAR16]. Im Rahmen dieses Forschungsprojektes wurde das Modell durch das Forschungsinstitut fortiss, einem „bayerischen Landesinstitut und An-Institut an der Technischen Universität München“ [FORT17], in einer eigenen Software umgesetzt.

Basis hierfür bot das von fortiss entwickelte Forschungstool Autofocus3, welches ursprünglich für das Systemdesign entwickelt wurde [FORT17]. Dies mündete in dem Tool SFIT, das auf der beschriebenen Methode und dem beschriebenen Modell basiert. SF-IT bietet eine grafische Oberfläche und erleichtert somit die Verwaltung und Anpassung der Daten. Zudem können gewisse Auswertungen direkt innerhalb der Software angezeigt werden. Das Tool kann auf der Homepage von fortiss bezogen werden. [FORT17, KON15]

5.5 Zusammenfassung Implementierung und Anwendungsbeispiel

Summary of implementation and example of application

In Kapitel 5 wurde das theoretische Modell aus Kapitel 4 in einem Software-Prototypen umgesetzt. Von allen denkbaren Umsetzungsalternativen ist eine Datenbank am zielführendsten für den in dieser Arbeit betrachteten Umfang. Um eine leichte Nachvollziehbarkeit zu gewährleisten wird das lokal laufende Datenbankmanagement-System SQLite verwendet.

Nach der Festlegung der zu verwendenden Software wurde die Datenstruktur ausdetailliert. Weiterhin wurde eine Variantenregel basierend auf dem Abgleich von Zeichenketten erläutert.

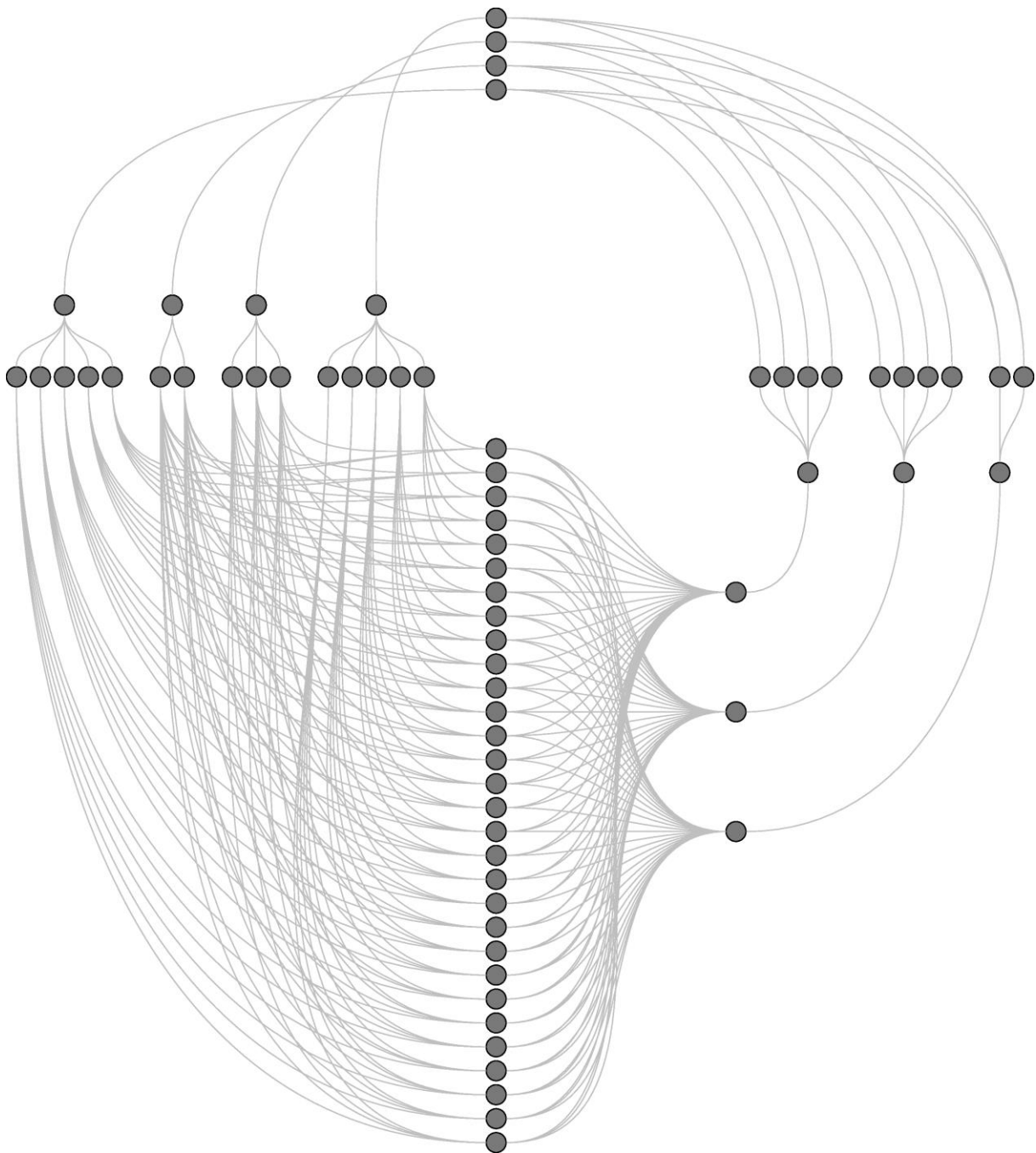


Bild 5.15: Graph zum erweiterten Kugelschreiber-Beispiel aus Abschnitt 5.2.7

Graph for the extended ballpoint pen example from section 5.2.7

Da das Modell für die Motormontage schnell eine unüberschaubare Größe erreicht und die Nachvollziehbarkeit der Ergebnisse somit kaum gegeben ist, wurde zur funktionalen Validierung des Modells eine fiktive Kugelschreiber-Fabrik untersucht. Anhand dieses Beispiels konnten verschiedene Problemfälle im Modell abgebildet und simuliert werden. Da sich die richtigen Ergebnisse für diese Fälle auch händisch bestimmen ließen, konnte validiert werden, dass das Modell bzw. der Softwareprototyp korrekt funktioniert. Es konnten alle relevanten Planungsentscheidungen im Modell abgebildet und deren Auswirkungen nachvollzogen werden.

Einen Graph über alle Zusammenhänge des Kugelschreiber-Beispiels zeigt Bild 5.15. In Tabelle 5.17 sind die Anzahl der Elemente für das Kugelschreiber-Beispiel und für die Motormontage gegenübergestellt. Daher ist es offensichtlich, dass ein entsprechender Graph für die Motormontage schnell sehr unübersichtlich werden würde und manuell nicht mehr nachvollzogen werden kann. Nach RENNER ist Varietät ein Komplexitätsmerkmal (vgl. Abschnitt 2.3.1), somit kann gefolgert werden, dass die Komplexität der Motormontage ebenfalls deutlich höher ist als die des Kugelschreiber-Beispiels.

Tabelle 5.17: Vergleich der Varietät der Anwendungsbeispiele aus Kapitel 5

Comparison of variety in the examples of application from chapter 5

	Bauteilart	Bauteilvarianten	Produkte	Linien	Stationen	Prozesse
Kugelschreiber-Einfach (vgl. Abschnitt 5.2.1)	4	6	2	2	8	4
Kugelschreiber-Erweitert (vgl. Abschnitt 5.2.7)	4	15	30	3	10	4
Motormontage (vgl. Abschnitt 5.3)	~230	~1000	~660	12	~830	~450

Im zweiten Teil der Validierung wurde das Modell anhand von realen Fragestellungen der Montageplanung im praktischen Einsatz erprobt und validiert. Die Abfragen waren identisch zu denen aus der funktionalen Validierung. Lediglich die Eingangsdaten wurden verändert. Die Beschaffung und Aufbereitung der Eingangsdaten war mit einigen Schwierigkeiten behaftet, für die jedoch Lösungen vorgestellt wurden.

Es konnte gezeigt werden, dass sowohl die Produktbeeinflussung als auch die Planung der Linienbelegung und die Gestaltung der Ressource durch den Einsatz des entwickelten Modells vereinfacht werden können. Ist das Modell initial eingerichtet und befüllt, so ist der Pflegeaufwand minimal. Selbe komplexe Änderungsszenarien und Fragestellungen ließen sich leicht in eine Abfrage übersetzen und auf die Datenbank anwenden.

Anforderungen und Fähigkeiten konnten miteinander verglichen werden und somit konnten bereits in einer frühen Phase erste Baubarkeitsuntersuchungen vorgenommen werden. Es konnte ebenfalls ermittelt werden, ob die durch die Linienbelegung bedingte Bauteilvariantenvielfalt die maximale Kapazität einer Linie oder einer Station zu einem bestimmten Zeitpunkt übersteigt.

6 Zusammenfassung, Fazit und Ausblick

Summary, conclusion and outlook

In dieser Arbeit wurde gezeigt, dass basierend auf bestehenden Datenmodellen und in Unternehmen vorhandenen Daten eine regel- und verknüpfungsbasierte Datenstruktur aufgebaut werden kann. Hierdurch wurden Produkt, Prozess und Ressource unter Berücksichtigung aller Einschränkungen miteinander verknüpft. Dadurch konnte der Montageplaner auf der Suche nach Auswirkungen von Änderungen, beispielsweise Nicht-Baubarkeiten, entlastet werden. In Kapitel 6 werden die gewonnenen Erkenntnisse zusammengefasst (vgl. Abschnitt 6.1) und basierend auf diesen wird ein Fazit gezogen (vgl. Abschnitt 6.2).

Durch das in dieser Arbeit entwickelte Modell werden weitere Potenziale bei der Optimierung und Automatisierung der Montageplanung eröffnet. Im Ausblick werden daher Forschungsfelder beleuchtet, die sich an diese Arbeit anschließen sollten (vgl. Abschnitt 6.3).

6.1 Zusammenfassung

Summary

Als wichtiger Bestandteil des Produktentstehungsprozesses hat die Montageplanung zwei Aufgaben: Einerseits ist sie verantwortlich für die Gestaltung des Montagesystems, in dem die Produkte montiert werden. Andererseits soll sie das Produkt beeinflussen, sodass dieses montagegerecht gestaltet wird und somit kosteneffizient hergestellt werden kann.

Eine Herausforderung bei dieser Aufgabe war die Variantenvielfalt, mit der sich die Montageplanung konfrontiert sieht. Diese führt zu einer steigenden Vielfalt in den Montagesystemen. Tritt diese Variantenvielfalt in einem Netzwerk von Montagesystemen auf, so entstehen hochvernetzte Systeme, da Änderungen an einer Stelle im System häufig Änderungsbedarf an anderer Stelle mit sich bringen. Zudem ändern sich die Montagesysteme über die Zeit, was zu einer Dynamik im Gesamtsystem führt. In der frühen Phase der Montageplanung sind die Informationen zudem noch mit einer hohen Unsicherheit behaftet und teilweise unvollständig. Die Montageplaner müssen daher eine hohe Komplexität beherrschen können.

Wirkzusammenhänge in solchen komplexen Systemen können mithilfe von Modellen abgebildet und untersucht werden. Solche Modelle werden unter dem Sammelbegriff der digitalen Fabrik zusammengefasst. Es gibt bereits mehrere Modelle die sich mit Montagesystemen und deren Planung beschäftigen. Jedoch zeigte die genauere Analyse dieser Modelle, dass keines ausreicht, um alle gestellten Anforderungen zu erfüllen. Somit kann keines der im Stand der Technik analysierten Modelle die beschriebene Komplexität und Dynamik ausreichend abbilden.

Die bestehenden Modelle wurden sinnvoll kombiniert und an mehreren Stellen erweitert, um somit ein Modell zu schaffen, das die gestellten Anforderungen erfüllt. Um das Modell in der Praxis einsetzen zu können, musste es implementiert werden. Dies geschah in Form einer Datenbank.

Ausgangspunkt für die Modellerstellung war die Methode der Integrationsplanung nach MÜLLER (s. Bild 2.13) [MUEL15, S. 554]. Nach dieser wird eine Montageplanungsaufgabe durch eine Änderung am Produkt angestoßen. Führt diese zu einem Konflikt, muss dieser aufgelöst werden. Dazu stehen drei Stellhebel zur Verfügung. Entweder muss die Linienbelegung angepasst, das Produkt umkonstruiert oder die Ressourcen des Montagesystems erweitert werden.

Es wurde ein Basis-Datenmodell entwickelt. Dieses übernimmt die grundsätzliche Gliederung in die Partialmodelle *Produkt*, *Prozess* und *Ressource* von den im Stand der Wissenschaft betrachteten Modellen. Die Erweiterung um Anforderungen und Fähigkeiten sowie weitere zusätzliche Klassen für die Kapazität und die Kommunikation definiert das vollständige Datenmodell.

Mit diesem vollständigen Datenmodell ließ sich das gesamte Spektrum eines Baukastenproduktes inklusive aller darin enthaltenen Varianz darstellen. Zudem ließ sich ein kompletter Montagesystemverbund darstellen und mit den Produkten auf einfache Art in Verbindung bringen. Durch den durchgängigen Einsatz von Verweisen anstelle von Kopien ist sichergestellt, dass in allen Modellteilen stets auf aktuelle Informationen zugegriffen wird. Die Verwendung von Regeln zur Verknüpfung großer Datenmengen verringert den Aufwand bei der Datenerhebung und -pflege erheblich.

Die Algorithmen wurden so entwickelt, dass bei jeder Änderung am System alle Beziehungen erneut ausgewertet werden. Dies stellt sicher, dass eine Änderung stets auf Ihre Auswirkungen um Gesamtsystem hin untersucht wird. Hierzu ist es erforderlich, dass sich die Rechenzeit des Algorithmus beschränkt ist. Durch die Verwendung eines Datenbankmanagementsystems und dessen Standardabfragen konnte dies sichergestellt werden.

Da die Zusammenhänge einer realen Motormontage sehr schwer nachzuvollziehen sind, wurde das Modell in einem ersten Schritt anhand eines einfachen, fiktiven Beispiels erprobt. Die Ergebnisse des Modells in diesem Teil ließen sich manuell nachvollziehen und somit auf Richtigkeit prüfen. Die Validierung zeigte, dass das Modell und dessen Implementierung die in Kapitel 2.5 definierten Anforderungen vollständig erfüllt.

Im Anschluss wurden daher reale Anwendungsfälle mithilfe des Modells untersucht. Basierend auf den entstehenden Daten ließen sich Montageplanungsentscheidungen treffen. Mithilfe des Modells können verschiedene Szenarien deutlich schneller untersucht werden, was auch dazu führt, dass mehrere verschiedene Szenarien berechnet werden und miteinander verglichen werden konnten.

6.2 Fazit

Conclusion

Die Forschungsfrage aus Abschnitt 1.2 lautete: Wie muss ein (Daten-)Modell zur Handhabung der steigenden Komplexität bei der Montagesystemplanung von Baukastenprodukten gestaltet sein?

Hierzu wurden in Abschnitt 2.5 insgesamt 10 Anforderungen an den praktischen Handlungsbedarf aufgestellt.

Mit dem in dieser Arbeit entwickelten Modell lassen sich Produkt, Prozess und Resource verknüpfen. Anforderung 1 ist somit erfüllt. Dies war jedoch auch schon bei den meisten Modellen aus dem Stand der Technik gegeben.

Im Modell werden ausschließlich Aggregationen verwendet. Somit werden keine Kopien von Informationen angelegt, sondern verweise auf ebendiese Informationen. Somit ist sichergestellt, dass stets mit aktuellen Informationen gearbeitet wird und keine Redundanzen entstehen, was Anforderung 2 entspricht.

Durch die Einführung und Verknüpfung der Klassen Produktvariante, Bauteilvariante und Bauteilart lassen sich Baukastenprodukte geschickt und mit überschaubarem Aufwand abbilden. Die eingeführten Variantenregeln, erleichtern das schnelle Anlegen und Verknüpfen einer Vielzahl von Produkt- und Bauteilvarianten zusätzlich. Die Anforderungen 3 und 4 wurden also ebenfalls umgesetzt.

Anforderung 5 beschreibt, dass sich Anforderungen der Produkte bzw. Bauteile mit den Fähigkeiten der Ressourcen des Montagesystems abgleichen lassen müssen. Dies konnte in mehreren Beispielen gezeigt werden und ist somit ebenfalls erfüllt.

Die Modellierung eines Montagesystemverbundes ist durch die Klassen Station und Linie gegeben. Um den Modellierungsaufwand zu verringern können mehrere Stationen einer Instanz einer Klasse Prozess verbunden werden. Finden also gleiche Prozesse in mehreren Linien statt, so muss dieser Prozess nur einmal angelegt werden. Ein weiterer Teil von Anforderung 6 war, dass sich die Montagesysteme innerhalb des Verbundes einfach und schnell mit Produktvarianten belegen lassen sollen. Dies konnte durch die Klasse Linienbelegung erreicht werden. Diese arbeitet nach denselben Variantenregeln, wie sie auch für die Zuordnung von Bauteilvarianten zu Produktvarianten verwendet werden. Durch diese durchgängige Verwendung von Aggregationen und Regeln zur Verknüpfung und die automatisiert ablaufenden Auswertungen kann auch bei hoher Dynamik und Veränderungsrate noch effizient mit dem Modell gearbeitet werden. Anforderung 7 konnte erfüllt werden.

Um die Kommunikation zu verbessern wurden Ansprechpartner sowohl auf Seiten der Produktentwicklung als auch bei den Montageplanern eingefügt. Auch diese sind per Aggregation verknüpft und können somit effizient angepasst werden. Das Ziele aus Anforderung 8 wurden erreicht.

Als Bestandteil des Modells wurden zahlreiche Auswertungen vorgestellt, die sich an realen Fragestellungen im Rahmen der Montagesystemplanung orientieren. Diese Auswertungen ermöglichen grobe Analysen in einer frühen Phase der Planung bis hin zu sehr detaillierten Analysen für die Feinplanung. Die Daten für diese Analysen lassen sich per Datenbankabfrage automatisiert ermitteln und können somit beliebig oft und für beliebig viele Szenarien effizient durchgeführt werden. Somit ist auch Anforderung 9 erfüllt.

In der 10. Anforderung sollte es ermöglicht werden, Fehler innerhalb des Modells zu identifizieren. Da die Modellbefüllung schlussendlich immer auf von Menschen erstellten Daten basiert sind Fehler nicht auszuschließen. Soweit möglich konnte dies im Modell umgesetzt werden. Ist eine beispielsweise keinem Montagesystem zugeordnet, so lässt sich dies mithilfe des Modells leicht ermitteln. Ist eine Bauteilvariante keiner Produktvariante zugewiesen ist dies ebenfalls leicht feststellbar, womit auch diese Anforderung erfüllt wurde.

Durch den Einsatz des Modells ließ sich die Dauer für das Finden von Konflikten im Rahmen der Montageplanung reduzieren. Die Befüllung des Modells bzw. der Datenbank war mit einem gewissen Initialaufwand verbunden. Da die benötigten Daten im Unternehmen aber ohnehin vorliegen müssen, lässt sich dieser Initialaufwand meist auf die Entwicklung einer Import-Schnittstelle reduzieren. Die dafür erforderliche systematische Strukturierung der Daten hilft, die Zusammenhänge der Klassen zu verstehen. Dies ist ein positiver Nebeneffekt.

Ist das Modell befüllt und sind die Zusammenhänge hergestellt, stellt das Modell eine große Unterstützung bei der Montageplanung dar. Viele Auswertungen, die die Basis von Entscheidungen in der Montageplanung sind, konnten mit dem im Rahmen dieser Arbeit entwickelten Modell schnell und aufwandsminimal innerhalb weniger Minuten erstellt werden. Die Alternative, eine manuelle Erstellung dieser Auswertungen, hätte mehrere Stunden bis Tage in Anspruch genommen.

Verschiedene Szenarien ließen sich mit geringem Zeitaufwand untersuchen und gegeneinander abwägen. Somit konnten Ideen zur Anpassung mittels der Stellhebel (vgl. Abschnitt 4.1) von Mitarbeitern oder Führungskräften schnell und effektiv untersucht werden. Es konnte effizient ermittelt werden, ob es sich um eine vielversprechende Idee handelt oder ob derart viele neue Konflikte entstehen, dass die Idee nicht weiterverfolgt werden sollte.

Beide Effekte, die hohe Zeitersparnis bei der Erstellung von Auswertungen und die Möglichkeit der Untersuchungen von vielen Szenarien, führten in Summe zu einem qualitativ höheren Ergebnis in der Montageplanung.

Das Modell kann mit unscharfen Daten genutzt werden und lässt sich im Nachhinein leicht anpassen und erweitern. Jede betroffene Schnittstelle kann in ihrem Partialmodell arbeiten, ohne dass andere Partialmodelle hierdurch gestört werden. Somit wird

das Parallele arbeiten durch Produktentwicklung und Montageplanung ermöglicht und begünstigt.

Die umsetzungsneutrale Beschreibung des Modells in Form eines UML-Klassendiagramms ermöglicht es, auch Implementierungen in anderen Programmiersprachen oder Softwaresystemen umzusetzen.

Das entwickelte Modell ermöglicht die Handhabung der Komplexität bedingt durch Vielfalt und Dynamik in der Montageplanung, sodass die Forschungsfrage als beantwortet betrachtet werden kann.

6.3 Ausblick und weiterer Forschungsbedarf

Outlook and further research

Der Ausblick und weitere Forschungsbedarf lässt sich in drei Hauptthemen einteilen: Es könnten weitere Bereiche und Schnittstellen in das Modell eingebunden werden, weitere Restriktionen wie beispielsweise die Geometrie oder Taktzeiten einbezogen werden oder Kosten als Parameter integriert werden, um somit eine Anwendbarkeit von Optimierungsverfahren zu ermöglichen. Auf diese drei Themen wird im Folgenden näher eingegangen.

Das Modell ist in seiner aktuellen Gestalt auf die Belange der Montageplanung ausgerichtet. Seine Grundsätze ließen sich aber auch auf mechanische Fertigungsbereiche erweitern. Es ist also denkbar, das Modell ebenfalls in der Planung der mechanischen Fertigung einzusetzen.

Weiterhin könnte das Modell auch für den Entwickler interessante Ergebnisse liefern, wenn dieser seine Produktideen bzw. deren Anforderungen direkt in das Modell einpflegt und somit eine Rückmeldung über die Montierbarkeit erhält. Dies unterstützt den Grundgedanken des Simultaneous Engineering.

Zudem könnte das Modell in bestehende industrielle Softwarelösungen wie Siemens Teamcenter oder Dassault Systemes DELMIA integriert werden.

Aktuell betrachtet das Modell zur Ermittlung von Konflikten die Variantenzahl und -kapazität sowie lineare und diskrete Anforderungen und Fähigkeiten. Dies beschreibt aber nur einen Teil der Restriktionen eines Montagesystems. Einige Restriktionen lassen sich am besten geometrisch beschreiben und nur schwer in lineare oder diskrete Werte überführen. Die Klassen der Anforderungen und Fähigkeiten sollten deshalb derart erweitert werden, dass sich auch geometrische Anforderungen und Fähigkeiten definieren und miteinander vergleichen lassen.

Die Taktzeit kann ebenfalls eine Einschränkung für die Integrierbarkeit von Produktvarianten in ein Montagsystem darstellen. Diese hängt von vielen Faktoren, wie bspw. dem durchzuführenden Prozess, der Bauteilart und -variante sowie der eingesetzten

Ressource ab. Auch die Taktzeit sollte ähnlich der Kapazität als eigene Klasse in das Modell integriert und bei der Untersuchung auf Konflikte berücksichtigt werden.

Sind möglichst viele Restriktionen erfasst, können die verschiedenen Anpassungsmöglichkeiten im Rahmen der Integrationsplanung (Anpassung Linienbelegung, Anpassung Produkt, Anpassung Ressource) jeweils mit Kosten hinterlegt werden. Ist dies geschehen, könnte ein Optimierungsalgorithmus die kostenoptimale Lösung errechnen. Auch dies wäre eine sinnvolle Erweiterung für das entwickelte Modell.

Das entwickelte Modell stellt die Basis für die vollumfängliche Abbildung der für die Integrationsplanung relevanten Faktoren dar. Bereits in der aktuellen Form stellt es einen großen Mehrwert bei der Entscheidungsfindung in der Montageplanung dar, da es insbesondere die dynamischen und komplexeren Zusammenhänge miteinander verknüpft.

7 Literaturverzeichnis

References

- [ABEL11] Abele, R.; Reinhart, G.: *Zukunft der Produktion - Herausforderungen, Forschungsfelder, Chancen*. München: Carl Hanser Verlag, 2011.
- [ALDE06] Alders, K.: *Komplexitäts- und Variantenmanagement der AUDI AG*. In: Lindemann, U.; Reichwald, R.; Zäh, M. (Hrsg.): *Individualisierte Produkte - Komplexität beherrschen in Entwicklung und Produktion*. Berlin: Springer Verlag, 2006, S. 221-237
- [BECK12] Becker, J.; Probandt, W.; Vering, O.: *Grundsätze ordnungsgemäßer Modellierung - Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement*. Berlin: Springer Verlag, 2012
- [BOSS07] Bossmann, M.: *Feature-basierte Produkt- und Prozessmodelle in der integrierten Produktentstehung*. Diss. Universität des Saarlandes, 2007
- [BOSS92] Bossel, H.: *Modellbildung und Simulation - Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme*. Braunschweig: Vieweg Verlag, 1992
- [BRAE11] Braess, H.-H.; Seiffert, I.: *Hanfbuch Kraftfahrzeugtechnik*. 6. Aufl. Wiesbaden: Vieweg+Teubner
- [BREC11] Brecher, C.; Loosen, P.; Müller, R.; Schmitt, R.; Fayzullin, K.; Haag, S.; Malik, A.; Pavim, A.; Pyschny N.; Vette, M.: *Integratives Produkt- und Prozessdesign für die selbstoptimierende Montage*. In: Brecher, C. (Hrsg.): *Integrative Produktionstechnik für Hochlohnländer*. Berlin: Springer Verlag, 2011, S. 909 - 957
- [BRUN15] Brunner, N.: *Konzeption eines Plattformkonzeptes für die Planung von Montagesystemen*. Diss. TU Dortmund, 2015
- [BUCH12] Buchholz, M.: *Theorie der Variantenvielfalt - Ein produktions- und absatzwirtschaftliches Erklärungsmodell*. Diss. Technische Universität Ilmenau, 2012
- [CHEN76] Chen, P.: *The Entity-Relationship Model - Toward a Unified View of Data*. In: *ACM Transactions on Database Systems* Nr. 1, Jg. 1, 1976, S. 9-36
- [DEST17] <https://www.destatis.de/DE/ZahlenFakten/GesamtwirtschaftUmwelt/VGR/Inlandsprodukt/Tabelle n/BWSBereichen.html> [Stand: 23. Mai 2017]
- [DEUS12] Deuse, J.; Busch, F.: *Zeitwirtschaft in der Montage*. In: Lotter, B.; Wiendahl, H.-P.: *Montage in der industriellen Produktion*. 2. Aufl. Berlin: Springer Verlag, 2012, S. 79-107

- [DIET13] Dietz, C.; Richter, R.; Deuse, J.: *Variantenmanagement im Simultaneous Engineering - Ein Vorgehen zum Abgleich von Produktstruktur und Fertigungssystem*. In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb Nr. 5, Jg. 108, 2013, S. 325-329
- [DIN02] Norm DIN 199 Teil 1 (März 2002). Technische Produktdokumentation - CAD-Modelle, Zeichnungen und Stücklisten
- [DIN03] Norm DIN 8593 Teil 0 (September 2003). Fertigungsverfahren Fügen. Einordnung, Unterteilung, Begriffe
- [ECKS13] Eckstein, H.; Eichert J.: *Konstruktionsintegrierte Arbeitsvorbereitung*. In: Westkämper, E.; Spath, D.; Constantinescu, C.; Lentes, J. (Hrsg.): *Digitale Produktion*. Berlin: Springer Verlag, 2013, S. 201-221
- [EHRL07] Ehrlenspiel, K.; Kiewert, A.; Lindemann, U.: *Kostengünstig Entwickeln und Konstruieren - Kostenmanagement bei der integrierten Produktentwicklung*. 6. Aufl. Berlin: Springer Verlag, 2007
- [EIGN09a] Eigner, M.; Stelzer, R.: *Product Lifecycle Management - Ein Leitfaden für Product Development und Life Cycle Management*. 2. Aufl. Berlin: Springer Verlag, 2009
- [EIGN09b] Eigner, M.: *IT-Lösungen für den Produktentwicklungsprozess*. In: Bullinger, H.-J.; Spath, D.; Warnecke, H.-J.; Westkämper, E. (Hrsg.): *Handbuch Unternehmensorganisation - Strategien, Planung, Umsetzung*. 3. Aufl. Berlin: Springer Verlag, 2009
- [EILE15] Eilers, J.: *Methodik zur Planung skalierbarer und rekonfigurierbarer Montagesysteme*. Diss. RWTH-Aachen, 2015
- [EVER05] Eversheim W.; Schuh G.; Assmus D.: *Integrierte Produkt- und Prozessgestaltung*. In: Eversheim, W. (Hrsg.): *Integrierte Produkt- und Prozessgestaltung*. Berlin: Springer Verlag, 2008
- [EVER89] Eversheim, W.: *Organisation in der Produktionstechnik – Band 4, Fertigung und Montage*. 2. Aufl. Düsseldorf: VDI Verlag, 1989
- [FELD14] Feldmann, K.: *Einführung*. In: Feldmann, K.; Schöppner, V.; Spur, Günter (Hrsg.): *Handbuch Fügen, Handhaben, Montieren*. München: Hanser Verlag, 2014, S. 3-22
- [FIRC02] Firchau, N. L.; Franke, H.-J.; Huch, B.; Menge, M.: *Variantenmanagement: Variantenvielfalt in Produkten und Prozessen erfolgreich beherrschen*. In: Franke, H.-J.; Hesselbach, J.; Huch, B.; Firchau, N. L. (Hrsg.): *Variantenmanagement in der Einzel- und Kleinserienfertigung*. München: Hanser Verlag, 2002, S. 1-25
- [FORT17] <https://www.fortiss.org/home/> [Stand: 21 Jan. 2017]

- [FRAN03] Franke, C.: *Feature-basierte Prozesskettenplanung in der Montage als Basis für die Integration von Simulationswerkzeugen in der Digitalen Fabrik*. Diss. Universität des Saarlandes, 2003
- [GROS04] Große-Heitmeyer, V.; Wiendahl, H.-P.: *Einführung*. In: Wiendahl, H.-P.; Gerst, D.; Keunecke, L. (Hrsg.): *Variantenbeherrschung in der Montage*. Berlin: Springer Verlag 2004, S. 3-17
- [HART12] Hartel, M.; Lotter, B.: *Planung und Bewertung von Montagesystemen*. In Lotter, B.; Wiendahl, H.-P. (Hrsg.): *Montage in der industriellen Produktion*. 2. Aufl., Berlin: Springer Verlag 2012, S. 365-388
- [HESS12] Hesse, S.: *Montagegerechte Produktgestaltung*. In: Lotter, B.; Wiendahl, H.-P.: *Montage in der industriellen Produktion*. 2. Aufl. Berlin: Springer Verlag, 2012, S. 9-48
- [HOER19] Hörauf, L.: *Software-gestützte Planung cyber-physischer Montagesysteme mittels durchgängiger Informationsnutzung*. Diss. RWTH-Aachen, 2019
- [HUCK90] Huck, M.: *Produktorientierte Montageablauf- und Layoutplanung für die Robotermontage*, Diss. Univ. Karlsruhe, 1990
- [ISO04] Norm ISO 10303-11 (November 2011). Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.
- [ISO05] Norm ISO 19501 (Januar 2005). Unified Modeling Language Specification.
- [JONA00] Jonas, C.: *Konzept einer durchgängigen, rechnergestützten Planung von Montageanlagen*. Diss. TU München, 2000
- [KAMP11] Kampker, A.; Schuh, G.; Burggräf, P.; Franzkoch, B.; Nöcker, J. Rauhut, M.; Wesch-Potente, C.: *Die Fabrikplanung der Zukunft – Verbindung der Modellwelten*. In: Brecher, C. (Hrsg.): *Integrative Produktionstechnik für Hochlohnländer*. Berlin: Springer Verlag, 2011, S. 300-342
- [KATZ94] Katzy, B.: *Unternehmensplanung in produzierenden Unternehmen – Entwicklung einer Methode auf der Basis von Unternehmensmodellen*. Diss. RWTH-Aachen, 1994
- [KLAU02] Klauke, S.: *Methoden und Datenmodell der "Offenen Virtuellen Fabrik" zur Optimierung simultaner Produktionsprozesse*. Diss. Technische Universität Dresden, 2002

- [KOND15] Kondeva, A.; Aravantinos, V.; Hermanns, L.; Hörauf, L.: *The SFIT tool: Supporting assembly planners to deal with new product variants*. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2015
- [KONO03] Konold, P.; Reger, H.: *Praxis der Montagetechnik*. Berlin: Springer Verlag, 2003
- [KREI10] Kreibich, J.: *Using SQLite*. Sebastopol: O'Reilly, 2010
- [KUEH06] Kühn, W.: *Digitale Fabrik - Fabriksimulation für Produktionsplaner*. München: Hanser, 2006
- [LAND13] Landherr, M.; Neumann, M.; Wolkmann, J.; Constantinescu, C.: *Digitale Fabrik*. In Westkämper, E.; Spath, D.; Constantinescu, C.; Lentes, J. (Hrsg.): *Digitale Produktion*. Berlin: Springer Verlag, 2013, S. 107-131
- [LAND15] Landerl, C.; Mattes, W.; Rüllicke, M.; Durst, B.: *Der neue Reihensechszylinder-Ottomotor von BMW*. In: *MTZ Motortechnische Zeitschrift* Nr. 10, Jg. 76 (2015), S. 50-59
- [LIND09] Lindemann, U.: *Methodische Entwicklung technischer Produkte*. 3. Aufl. Berlin: Springer Verlag, 2009
- [LOTT12a] Lotter, B.: *Einführung*. In Lotter, B.; Wiendahl, H.-P. (Hrsg.): *Montage in der industriellen Produktion*. 2. Aufl. Berlin: Springer Verlag 2012, S. 1-8
- [LOTT12b] Lotter, B.: *Manuelle Montage von Großgeräten*. In Lotter, B.; Wiendahl, H.-P. (Hrsg.): *Montage in der industriellen Produktion*. 2. Aufl. Berlin: Springer Verlag, 2012, S. 147-165
- [LOTT12c] Lotter, E.: *Hybride Montagesysteme*. In Lotter, B.; Wiendahl, H.-P. (Hrsg.): *Montage in der industriellen Produktion*. 2. Aufl., Berlin: Springer Verlag, 2012, S. 167-193
- [LOTT92] Lotter, B.: *Wirtschaftliche Montage: ein Handbuch für Elektrogerätebau und Feinwerktechnik*. 2. Aufl. Düsseldorf: VDI Verlag, 1992
- [MOOS04] Moos, A.: *Datenbank-Engineering - Analyse, Entwurf und Implementierung objektrelationaler Datenbanken*. 3. Aufl. Wiesbaden: Vieweg Verlag, 2004
- [MOTU09] Motus, D.: *Referenzmodell für die Montageplanung in der Automobilindustrie*. Diss. Otto-von-Guericke-Universität Magdeburg, 2008

- [MUEL09] Müller, R.; Buchner, T.; Gottschalk, S.-F.; Fayzullin, K.; Herfs, W.; Hilchner, R.; Pyschny, N.: *Studie Strategien und Trends in der Montagetechnik und -organisation : cluster of excellence integrative production technology for high wage countries*. Aachen: Apprimus Verlag, 2009
- [MUEL13] Müller, R.: *Montagegerecht*. In Feldhusen, J.; Grote, K.-H. (Hrsg.): *Konstruktionslehre - Methoden und Anwendung erfolgreicher Produktentwicklung*. 8. Aufl. Berlin: Springer Verlag, 2013, S. 702-725
- [MUEL15] Müller, R.: *Modell zur fähigkeitsbasierten Montageplanung - Fähigkeitsbasierte Bewertung von Produktänderungen am Beispiel der Motormontage*. In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb Nr. 9, Jg. 110 (2015), S. 552-557
- [MUEL17] Müller, R.; Eilers, J.; Hermanns, L.; Gerdes, R.: *Modellgestützte Baubarkeitsprüfung in der Montageplanung - Planungsgenauigkeit bei der Integration von Bauteilvarianten erhöhen*. In: ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb Nr. 4, Jg. 107 (2015), S. 253-560
- [NEUH01] Neuhausen, J.: *Gestaltung modularer Produktionssysteme für Unternehmen der Serienproduktion*. Diss. RWTH Aachen, 2001
- [PAHL07] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H.: *Konstruktionslehre Grundlagen erfolgreicher Produktentwicklung*. 7. Aufl. Berlin: Springer Verlag, 2007
- [PATZ82] Patzak, G.: *Systemtechnik - Planung komplexer innovativer Systeme*. Berlin: Springer Verlag, 1982
- [PILL99] Piller, F. T.; Waringer, D.: *Modularisierung in der Automobilindustrie - neue Formen und Prinzipien. Modular Sourcing, Plattformkonzept und Fertigungssegmentierung als Mittel des Komplexitätsmanagements*. Aachen: Shaker Verlag, 1999
- [RENN07] Renner, I.: *Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil*. Diss. TU München, 2007
- [RICH09] Richter M.: *Zur Güte von Beschreibungsmodellen – eine erkenntnistheoretische Untersuchung*. In: Ilmenauer Schriften zur Betriebswirtschaftslehre, Nr. 8, 2015
- [RUMB05] Rumbaugh, J.; Jacobson, I.; Booch, G.: *The Unified Modeling Language Reference Manual*. Boston: Addison-Wesley, 2005

- [SCHN12] Schneider, S.; Fallböhrer, M.; Schallow, J.; Hartung, J.; Deuse, J.: *Agile Prozessplanung im Produktentstehungsprozess – Softwareentwicklung als Lösungshilfe für permanente Anpassungsbedarfe in der Prozessplanung*, In: ZWF - Zeitschrift für wirtschaftlichen Fabrikbetrieb, Nr. 3, Jg. 107 (2012), S. 158-162
- [SCHU04] Schubert, M.: *Datenbanken - Theorie, Entwurf und Programmierung relationaler Datenbanken*. Stuttgart: Teubner Verlag, 2004
- [SCHU05] Schuh, G.: *Produktkomplexität managen - Strategien, Methoden, Tools*. München: Carl Hanser Verlag, 2005
- [SCHU10] Schuh, G.; Lenders, M.; Arnoscht, J.; Rudolf, S.: *Effizienter innovieren mit Produktbaukästen - Studienergebnisse und Leitfaden - ein Beitrag zu Lean Innovation*. Aachen: WZL-Eigendruck, 2010
- [SCHU11] Schuh, G.; Behr, M.; Brecher, C.; Bührig-Polaczek, A.; Michaeli, W.; Arnoscht, J.: *Individualisierte Produktion*. In: Brecher, C. (Hrsg.): *Integrative Produktionstechnik für Hochlohnländer*. Berlin: Springer Verlag, 2011, S. 83 - 256
- [SCHU14] Schuh, G.; Hoppe, M.; Schubert, J.; von Mangoldt, J.: *Lieferantenauswahl*. In: Schuh, G.: *Einkaufsmanagement*. Berlin: Springer Verlag, 2014, S. 255 - 342
- [SCHU95] Schulte, C.: *Komplexitätsmanagement*. In: Corsten, H. (Hrsg.): *Handbuch Unternehmensführung*. Wiesbaden: Gabler Verlag, 1995, S. 757 - 765
- [SCHU97] Schuh, G.; Jonas, I.: *Beherrschung der Variantenvielfalt: Methoden zur Beherrschung der technologischen Konsequenzen der Ausrichtung auf den Markt*. In: *Technica* Nr. 7, Jg. 45 (1996), S. 23 - 32
- [SCHW97] Schweitzer, M.; Küpper, H.-U.: *Produktions- und Kostentheorie*. 2. Aufl. Wiesbaden: Gabler Verlag, 1997
- [SMAR16] <http://www.smartf-it-projekt.de/> [Stand: 16. Okt. 2016]
- [SNOW05] Snowden, D.J.: *Multi-ontology sense making: a new simplicity in decision making*. In: *Informatics in Primary Care* Nr. 13 (2005), S. 45 - 53
- [SPAT13] Spath, D.; Rally, P.: *Zukunftssicherung mit wandlungsfähigen Montagesystemen - Bedarf und Nutzen*. In: Spath, D.; Müller, R.; Reinhart, G. (Hrsg.): *Zukunftsfähige Montagesysteme - wirtschaftlich, wandlungsfähig und rekonfigurierbar*. Stuttgart: Fraunhofer Verlag, 2013, S. 9-19
- [SPUR86] Spur, G.; Stöferle, T.: *Handbuch der Fertigungstechnik. - 5: Fügen, Handhaben und Montieren*. München: Hanser Verlag, 1986

- [SPUR97] Spur, G.; Krause, F.-L.: *Das virtuelle Produkt - Management der CAD-Technik*. München: Hanser Verlag, 1997
- [SQLA17] sqlabs.com [Stand: 10. Okt. 2017]
- [SQLI17] Sqlitebrowser.org [Stand: 10. Okt. 2017]
- [STAC73] Stachowiak, H.: *Allgemeine Modelltheorie*. Wien: Springer Verlag, 1973
- [STEI14] Steinpatzer, F.; Brüner, T.; Schwarz, C.; Rüllicke, M.: *Die neuen drei- und vierzylinder Ottomotoren von BMW*. In: MTZ Motortechnische Zeitschrift Nr. 06, Jg. 75 (2014), S. 30 - 37
- [STEI97] Steinwasser, P.: *Modulares Informationsmanagement in der integrierten Produkt- und Prozeßplanung*. Diss. Universität Erlangen-Nürnberg, 1997
- [TOEL10] Töllner, A.; Jungmann, T.; Bücken M.; Brutscheck, T.: *Modelle und Modellierung - Terminologie, Funktionen und Nutzung*. In: Bandow, G.; Holzmüller H. H. (Hrsg.): *Das ist gar kein Modell! Unterschiedliche Modelle und Modellierungen in Betriebswirtschaftslehre und Ingenieurwissenschaften*. Wiesbaden: Gabler Verlag, 2010, S. 3-21
- [TUEC11] Tücks, G.; Eilers, J.: *Komplexitätsbeherrschung durch wandlungsfähige Produktion*. In: Complexity Management Journal Nr. 1, 2011, S. 4-8
- [ULRI76] Ulrich, P.; Hill, W.: *Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre (Teil I)*. In: WiSt Wirtschaftswissenschaftliches Studium Nr. 7, 1976, S. 304-309
- [ULRI81] Ulrich, H.: *Die Betriebswirtschaftslehre als anwendungsorientierte Sozialwissenschaft*. In: Geist, M.N.; Köhler R. (Hrsg.): *Die Führung des Betriebs*, Stuttgart: Schäffer-Poeschel Verlag 1981, S. 1-25
- [VDI03] Richtlinie VDI 2218 (2003): Informationsverarbeitung in der Produktentwicklung: Feature-Technologie
- [VDI08] Richtlinie VDI 4499-1 (2008): Digitale Fabrik, Teil 1: Grundlagen
- [VDI78] Richtlinie VDI 2815 (1987): Begriffe für die Produktionsplanung und -steuerung - Einführung, Grundlagen
- [VDI87] Richtlinie VDI 2235 (1987): Wirtschaftliche Entscheidungen beim Konstruieren; Methoden und Hilfen
- [VDI90] Richtlinie VDI 2860 (1990): Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen

- [VIET13] Vietor T.; Stechert, C.: *Produktarten zur Rationalisierung des Entwicklungs- und Konstruktionsprozesses*. In: Feldhusen, J.; Grote, K.-H.: *Konstruktionslehre - Methoden und Anwendung erfolgreicher Produktentwicklung*. 8. Aufl. Berlin: Springer Verlag, 2013, S. 817 - 871
- [WARN75] Warnecke, H.-J.; Löhr, H.-G.; Kiener, W.: *Montagetechnik - Schwerpunkt der Rationalisierung*. Mainz: Krausskopf, 1975
- [WERN13] Werners, B.: *Grundlagen des Operations Research*. 3. Aufl. Berlin: Springer Verlag, 2013
- [WIEN04] Große-Heitmeyer, V.; Wiendahl, H.-P.: *Einführung*. In: Wiendahl H.-P.; Gerst, D.; Keunecke, L. (Hrsg.): *Variantenbeherrschung in der Montage*. Berlin: Springer Verlag, 2004, S. 3-20
- [WIEN10] Wiendahl, H.-P.: *Betriebsorganisation für Ingenieure*. München: Carl Hanser Verlag, 2010
- [ZAGE06] Zagel, M.: *Übergreifendes Konzept zur Strukturierung variantenreicher Produkte und Vorgehensweise zur iterativen Produktstruktur-Optimierung*. Diss. TU Kaiserslautern, 2006

8 Anhang

Im Anhang finden Sie den SQLite Quellcode für die Erstellung der Tabellen. Zusammen mit den Quellcodes aus der Arbeit und diesen Datensätzen lassen sich Ergebnistabellen generieren. Die Diagramme können mit einer beliebigen Software erstellt werden.

```
CREATE TABLE IF NOT EXISTS tab_anforderungen (  
ID_Anforderung INTEGER PRIMARY KEY,  
ID_Bauteil INTEGER,  
ID_Prozesstyp INTEGER,  
Anforderung TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS tab_bauteilarten (  
ID_Bauteilart INTEGER PRIMARY KEY,  
Nr TEXT,  
Bezeichnung TEXT,  
ID_Ansprechpartner INTEGER,  
Anstellungsart INTEGER,  
Ordnername TEXT,  
Dateiname TEXT,  
Modul TEXT,  
Gewerk TEXT,  
Verbauort TEXT,  
Auswertbar INTEGER,  
Neu_einlesen INTEGER,  
Eingelesen_am INTEGER  
);
```

```
CREATE TABLE IF NOT EXISTS tab_bauteilvariante (  
ID_Bauteilvariante INTEGER PRIMARY KEY,  
Bezeichnung TEXT,  
ID_Bauteilart INTEGER,  
Variantencode TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS tab_faehigkeiten (  
ID_Faehigkeit INTEGER PRIMARY KEY,  
ID_Station INTEGER,  
ID_Prozesstyp INTEGER,  
Faehigkeit_min TEXT,  
Faehigkeit_max TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS tab_kapazitaeten_station (  
ID_Station INTEGER,  
ID_Bauteilart INTEGER,  
Kapazitaet INTEGER,  
PRIMARY KEY(ID_Station, ID_Bauteilart)  
);
```

```
CREATE TABLE IF NOT EXISTS tab_kapazitaeten_linie (  
ID_Linie INTEGER,  
ID_Bauteilart INTEGER,  
Kapazitaet INTEGER,  
PRIMARY KEY (ID_Linie, ID_Bauteilart)  
);
```

```
CREATE TABLE IF NOT EXISTS tab_linien (  
ID_Linie INTEGER PRIMARY KEY,  
Bezeichnung TEXT,  
Standort TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS tab_linienbelegung (  
ID_Linienbelegung INTEGER PRIMARY KEY,  
ID_Linie INTEGER,  
Variantencode TEXT,  
PVL INTEGER,  
SOP INTEGER,  
EOP INTEGER  
);
```

```
CREATE TABLE IF NOT EXISTS tab_personen (  
ID_Person INTEGER PRIMARY KEY,  
Vorname TEXT,  
Nachname TEXT,  
Abteilung TEXT,  
Email TEXT,  
Telefon TEXT,  
Weitere_Informationen TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS tab_produkte (  
ID_Produkt INTEGER PRIMARY KEY,  
Bezeichnung TEXT,  
Variantencode TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS tab_prozesse (  
ID_Prozess INTEGER PRIMARY KEY,  
ID_Prozesstyp INTEGER,  
Bezeichnung TEXT,  
Reihenfolge INTEGER  
);
```

```
CREATE TABLE IF NOT EXISTS tab_prozesstypen (  
ID_Prozesstyp INTEGER PRIMARY KEY,  
Bezeichnung TEXT,  
Abgleichart INTEGER  
);
```

```
CREATE TABLE IF NOT EXISTS tab_stationen (  
ID_Station INTEGER PRIMARY KEY,  
Nr TEXT,  
Bezeichnung TEXT,  
Beschreibung TEXT,  
Reihenfolge INTEGER,  
ID_Ansprechpartner INTEGER,  
Gewerk TEXT,  
Typ TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS tab_zeitraeume (  
ID_Zeitraum INTEGER PRIMARY KEY,  
Datum INTEGER  
);
```

```
CREATE TABLE IF NOT EXISTS lnk_linien_stationen (  
ID_Linie INTEGER,  
ID_Station INTEGER,  
Reihenfolge INTEGER,  
PRIMARY KEY (ID_Station, ID_Linie)  
);
```

```
CREATE TABLE IF NOT EXISTS lnk_prozesse_bauteilarten (  
ID_Prozess INTEGER,  
ID_Bauteilart INTEGER,  
Konsumiert INTEGER,  
PRIMARY KEY (ID_Prozess, ID_Bauteilart)  
);
```

```
CREATE TABLE IF NOT EXISTS lnk_stationen_prozesse (  
ID_Station INTEGER,  
ID_Prozess INTEGER,  
PRIMARY KEY (ID_Station, ID_Prozess)  
);
```