





# **TECHNISCHE UNIVERSITÄT MÜNCHEN**

Lehrstuhl für  
Werkzeugmaschinen und Fertigungstechnik

## **Teilautomatische Generierung von Simulationsmodellen für den entwicklungsbegleitenden Steuerungstest**

**Alexander Lindworsky**

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Gunther Reinhart

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Michael Zäh
2. Univ.-Prof. Dr.-Ing. Birgit Vogel-Heuser

Die Dissertation wurde am 13.01.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 14.06.2011 angenommen.



Alexander Lindworsky

**Teilautomatische Generierung  
von Simulationsmodellen für  
den entwicklungsbegleitenden Steuerungstest**



Herbert Utz Verlag · München

## **Forschungsberichte IWB**

Band 249

Zugl.: Diss., München, Techn. Univ., 2011

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf fotomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben – auch bei nur auszugsweiser Verwendung – vorbehalten.

Copyright © Herbert Utz Verlag GmbH · 2011

ISBN 978-3-8316-4125-3

Printed in Germany  
Herbert Utz Verlag GmbH, München  
089-277791-00 · [www.utzverlag.de](http://www.utzverlag.de)

## Geleitwort der Herausgeber

Die Produktionstechnik ist für die Weiterentwicklung unserer Industriegesellschaft von zentraler Bedeutung, denn die Leistungsfähigkeit eines Industriebetriebes hängt entscheidend von den eingesetzten Produktionsmitteln, den angewandten Produktionsverfahren und der eingeführten Produktionsorganisation ab. Erst das optimale Zusammenspiel von Mensch, Organisation und Technik erlaubt es, alle Potentiale für den Unternehmenserfolg auszuschöpfen.

Um in dem Spannungsfeld Komplexität, Kosten, Zeit und Qualität bestehen zu können, müssen Produktionsstrukturen ständig neu überdacht und weiterentwickelt werden. Dabei ist es notwendig, die Komplexität von Produkten, Produktionsabläufen und -systemen einerseits zu verringern und andererseits besser zu beherrschen.

Ziel der Forschungsarbeiten des *iwb* ist die ständige Verbesserung von Produktentwicklungs- und Planungssystemen, von Herstellverfahren sowie von Produktionsanlagen. Betriebsorganisation, Produktions- und Arbeitsstrukturen sowie Systeme zur Auftragsabwicklung werden unter besonderer Berücksichtigung mitarbeiterorientierter Anforderungen entwickelt. Die dabei notwendige Steigerung des Automatisierungsgrades darf jedoch nicht zu einer Verfestigung arbeitsteiliger Strukturen führen. Fragen der optimalen Einbindung des Menschen in den Produktentstehungsprozess spielen deshalb eine sehr wichtige Rolle.

Die im Rahmen dieser Buchreihe erscheinenden Bände stammen thematisch aus den Forschungsbereichen des *iwb*. Diese reichen von der Entwicklung von Produktionssystemen über deren Planung bis hin zu den eingesetzten Technologien in den Bereichen Fertigung und Montage. Steuerung und Betrieb von Produktionssystemen, Qualitätssicherung, Verfügbarkeit und Autonomie sind Querschnittsthemen hierfür. In den *iwb* Forschungsberichten werden neue Ergebnisse und Erkenntnisse aus der praxisnahen Forschung des *iwb* veröffentlicht. Diese Buchreihe soll dazu beitragen, den Wissenstransfer zwischen dem Hochschulbereich und dem Anwender in der Praxis zu verbessern.





## **Vorwort**

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) der Technischen Universität München.

Herrn Prof. Dr.-Ing. Michael Zäh und Herrn Prof. Dr.-Ing. Gunther Reinhart, den Leitern dieses Instituts, gilt mein besonderer Dank für die wohlwollende Förderung und großzügige Unterstützung meiner Arbeit.

Bei Frau Prof. Dr.-Ing. Birgit Vogel-Heuser, der Leiterin des Lehrstuhls für Automatisierungstechnik und Informationssysteme der Technischen Universität München, möchte ich mich für die Übernahme des Korreferates und die aufmerksame Durchsicht der Arbeit sehr herzlich bedanken.

Darüber hinaus bedanke ich mich bei allen Mitarbeiterinnen und Mitarbeitern des Instituts sowie allen Studenten, die mich bei der Erstellung meiner Arbeit unterstützt haben, recht herzlich.

Ferner geht mein Dank an Herrn Matthias Reihle, Herrn Frédéric Lacour und Herrn Fabian Meling, die diese Arbeit kritisch gelesen und viele hilfreiche Anregungen gegeben haben.

Des Weiteren danke ich Herrn Detlef Hagemann vom Verein Deutscher Werkzeugmaschinenfabriken (VDW) für die Unterstützung bei der Beantragung und der Durchführung des Forschungsprojektes AutoVIBN.

Herrn Dr. Benjamin Hummel danke ich für die sehr gute Zusammenarbeit im Projekt AutoVIBN sowie für die zahlreich geführten und für mich sehr interessanten Diskussionen, die für neue Denkanstöße gesorgt haben.

Besonderen Dank möchte ich meiner Familie aussprechen, der ich diese Arbeit widme. Meiner Mutter Christa und meinem Vater Günther möchte ich für die verständnisvolle Unterstützung während meines gesamten akademischen Werdeganges danken. Vor allem aber möchte ich meiner Frau Katrin danken, die mir während der Erstellung dieser Arbeit immer liebevoll zur Seite gestanden und mir auch in schwierigen Situationen stets Rückhalt gegeben hat.



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis.....</b>	<b>I</b>
<b>Abbildungsverzeichnis.....</b>	<b>IX</b>
<b>Tabellenverzeichnis.....</b>	<b>XIX</b>
<b>Abkürzungsverzeichnis.....</b>	<b>XXI</b>
<b>Formelverzeichnis .....</b>	<b>XXVII</b>
<b>Glossar .....</b>	<b>XXXI</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Gegenwärtige Situation im Maschinen- und Anlagenbau .....	1
1.2 Resultierende Problemstellung .....	2
1.3 Zielsetzung und Vorgehensweise .....	4
1.4 Gliederung der Arbeit .....	8
<b>2 Grundlagen .....</b>	<b>11</b>
2.1 Kapitelüberblick.....	11
2.2 Begriff der Mechatronik.....	11
2.3 Entwicklungsprozesse im Maschinen- und Anlagenbau .....	13
2.3.1 Entwicklungs- und Konstruktionsmethoden.....	13
2.3.2 Phasen des Entwicklungsprozesses.....	17
2.3.3 Inbetriebnahme von Maschinen und Anlagen .....	18
2.4 Grundlagen der Steuerungstechnik .....	20
2.4.1 Klassifizierung von Steuerungen .....	20
2.4.2 Speicherprogrammierbare Steuerungen .....	23

2.4.3	Dezentralisierung und Feldbussysteme .....	24
2.5	Systeme, Modelle und Simulation .....	25
2.5.1	Systembegriff und Klassifikation .....	25
2.5.2	Modellbegriff .....	27
2.5.3	Modellierungstechniken .....	28
2.5.4	Simulation .....	38
2.5.5	Echtzeitfähigkeit .....	38
2.6	Virtuelle Inbetriebnahme .....	39
2.6.1	Grundprinzip .....	39
2.6.2	Ausprägungen .....	40
2.6.3	Modelle für die Virtuelle Inbetriebnahme .....	41
2.6.4	Modellerstellungsprozess .....	43
2.6.5	Kommerzielle Simulationssysteme .....	45
2.6.6	Nutzen und Aufwand .....	45
<b>3</b>	<b>Stand der Forschung und der Technik .....</b>	<b>47</b>
3.1	Kapitelüberblick .....	47
3.2	Prüfmethoden für Steuerungssoftware .....	47
3.2.1	Klassifikation der Prüfmethoden .....	47
3.2.2	Formale Verifikation .....	48
3.2.3	Analytische Prüfmethoden .....	49
3.2.4	Dynamischer Test .....	49
3.3	Softwarebasierte Simulations- und Testmethoden .....	51
3.3.1	Forschungsarbeiten .....	51

3.3.2	Industrielle Ansätze.....	58
3.3.3	Handlungsbedarf .....	59
3.4	Planung und Konzeption im Maschinen- und Anlagenbau .....	62
3.4.1	Interdisziplinärer Systementwurf.....	62
3.4.2	Modelle zur Unterstützung des Entwicklungsprozesses ...	63
3.4.3	Fazit.....	70
<b>4</b>	<b>Konzept .....</b>	<b>71</b>
4.1	Kapitelüberblick.....	71
4.2	Anforderungen an die Methode .....	71
4.3	Integration der Simulation in den Entwicklungsprozess .....	72
4.3.1	Modellgetriebene Vorgehensweise .....	72
4.3.2	Grobplanung.....	74
4.3.3	Feinplanung.....	75
4.3.4	Unterstützung der Konstruktionsphase .....	75
4.4	Modellierungstechnik.....	76
4.4.1	Disziplinenübergreifendes Beschreibungsmittel.....	76
4.4.2	Fehlermodellierung .....	77
4.4.3	Simulation hybrider Systeme .....	78
4.5	Einbindung vorhandener Modelldaten.....	78
4.5.1	Funktionsmodell und CAD-Daten .....	78
4.5.2	Methode für die Datenintegration .....	79
4.6	Automatisierte Ableitung von Simulationsmodellen.....	80
4.6.1	Prinzip der Modellgenerierung .....	80

4.6.2	Berücksichtigung von Änderungen .....	81
4.7	Zusammenfassung .....	81
<b>5</b>	<b>Funktionsmodellierung von Maschinen und Anlagen.....</b>	<b>83</b>
5.1	Kapitelüberblick.....	83
5.2	Beschreibungsmittel und Modellierungstechniken .....	83
5.2.1	Anforderungsanalyse .....	83
5.2.2	Bewertung .....	84
5.3	Methode für die Funktionsmodellierung .....	86
5.3.1	Allgemeines .....	86
5.3.2	Bestandteile des Metamodells .....	87
5.3.3	Modellierung der Struktur .....	88
5.3.4	Spezifikation des Verhaltens .....	89
5.3.5	Geometriemodell.....	95
5.3.6	Kinematische Beziehungen.....	97
5.3.7	Materialfluss .....	98
5.3.8	Bedienerschnittstelle .....	100
5.3.9	Bibliotheksmechanismus .....	100
5.3.10	Simulation des Funktionsmodells .....	101
5.4	Bezug zur Konstruktionsebene .....	101
5.4.1	Technisches Ressourcenmodell .....	101
5.4.2	Aktoren .....	102
5.4.3	Sensoren .....	103
5.4.4	Kopplung zur Steuerungsebene .....	104

5.5	Vorgehensmodell .....	106
5.5.1	Konkretisierung des Vorgehensmodells .....	106
5.5.2	Modellbildung in der Grobplanungsphase .....	107
5.5.3	Modellbildung in der Feinplanungsphase .....	112
5.6	Zusammenfassung .....	116
<b>6</b>	<b>Teilautomatisierte Modellerstellung .....</b>	<b>117</b>
6.1	Kapitelüberblick .....	117
6.2	Konzeptbeschreibung .....	117
6.3	Nutzung von CAD-Daten .....	119
6.3.1	Grundsätzliches zu CAD-Systemen .....	119
6.3.2	Nutzbare Informationen der CAD-Systeme für die Simulation .....	120
6.3.3	Datenaustausch zwischen den CAD-Systemen und dem Funktionsmodell .....	121
6.3.4	Definition eines Austauschformates .....	123
6.3.5	Konstruktionsrichtlinien .....	125
6.3.6	Querbeziehungen zwischen Funktions- und Konstruktionsebene .....	127
6.3.7	Konsistenz von Funktions- und Konstruktionsmodellen .....	129
6.4	Automatisierte Erweiterung des Funktionsmodells mit MCAD- Daten .....	130
6.4.1	Konzept für die Integration der MCAD-Daten .....	130
6.4.2	Modelltransformation mit Hilfe der MCAD-Daten .....	132
6.4.3	Übernahme von Modellparametern .....	134

6.4.4	Kinematische Freiheitsgrade.....	138
6.4.5	Detektoren und Bindings .....	141
6.4.6	Visualisierung und Geometriemodell .....	145
6.5	Automatisierte Erweiterung des Funktionsmodells mit E/FCAD-Daten.....	152
6.5.1	Konzept für die Integration der E/FCAD-Daten.....	152
6.5.2	Modelltransformation mit Hilfe der E/FCAD-Daten .....	154
6.5.3	Integration der Aktordaten.....	155
6.5.4	Integration der Sensordaten .....	167
6.5.5	Anpassung des Technischen Ressourcenmodells .....	168
6.6	Automatische Generierung des VIBN-Modells .....	170
6.6.1	Ansatz zur Modellgenerierung.....	170
6.6.2	Ausleitung des VIBN-Modells .....	171
6.6.3	Steuerungskopplung, Visualisierung und Bedienung.....	172
6.6.4	Änderungen am Modell .....	174
6.7	Modelltest .....	174
6.7.1	Testmethoden.....	174
6.7.2	Test der Simulationsmodelle .....	176
6.8	Generierung rudimentärer Konstruktionsmodelle.....	179
6.9	Zusammenfassung .....	181
<b>7</b>	<b>Prototypische Umsetzung und Wirtschaftlichkeitsbetrachtung .....</b>	<b>183</b>
7.1	Kapitelüberblick.....	183
7.2	Entwicklungs- und Simulationsumgebung.....	183



7.2.1	Funktionsmodellierung .....	183
7.2.2	CAD-Datenintegration und VIBN-Simulator .....	185
7.3	Anwendungsbeispiele .....	187
7.3.1	Allgemeines.....	187
7.3.2	Hauptzeitparalleles Rüsten einer Werkzeugmaschine ....	187
7.3.3	Be- und Entladesystem einer Schleifmaschine .....	191
7.4	Wirtschaftlichkeitsbetrachtung .....	194
7.4.1	Allgemeines.....	194
7.4.2	Bewertung des wirtschaftlichen Aufwandes.....	194
7.4.3	Bewertung des wirtschaftlichen Nutzens .....	195
7.5	Zusammenfassung.....	199
<b>8</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>201</b>
8.1	Zusammenfassung.....	201
8.2	Ausblick .....	203
<b>9</b>	<b>Literaturverzeichnis.....</b>	<b>205</b>
<b>10</b>	<b>Anhang .....</b>	<b>227</b>
10.1	XML-Austauschformat für CAD-Daten .....	227
10.1.1	Mechanik.....	227
10.1.2	Elektrotechnik und Fluidik.....	229
10.2	Firmen- und Herstellerverzeichnis.....	231



## Abbildungsverzeichnis

Abbildung 1.1:	Grundprinzip der Virtuellen Inbetriebnahme (VIBN).....	3
Abbildung 1.2:	Fehler und Kosten bei der Steuerungssoftwareentwicklung nach OESTREICHER (1986).....	4
Abbildung 1.3:	Ansatz zur Integration und Teilautomatisierung der VIBN...	6
Abbildung 1.4:	Vergleich der Arbeitsabläufe zwischen der Methode auf der Basis des Funktionsmodells und einer konventionellen VIBN .....	7
Abbildung 1.5:	Gliederung und Aufbau der vorliegenden Arbeit .....	10
Abbildung 2.1:	Aufbau eines mechatronischen Systems gemäß VDI- RICHTLINIE 2206 (2004) .....	12
Abbildung 2.2:	Generelles Vorgehen beim Entwickeln und Konstruieren nach VDI-RICHTLINIE 2221 (1993) .....	14
Abbildung 2.3:	Vorgehensmodell nach VDI/VDE-RICHTLINIE 2422 (1994) .....	15
Abbildung 2.4:	Vorgehen beim Entwickeln mechatronischer Systeme nach VDI-RICHTLINIE 2206 (2004).....	16
Abbildung 2.5:	Wasserfallmodell nach ROYCE (1970) und Spiralmodell in Anlehnung an BOEHM (1988) bzw. vereinfachte Darstellung von LEVI & REMBOLD (2003) .....	17
Abbildung 2.6:	Vereinfachte Darstellung des Entwicklungsablaufes in der Werkzeugmaschinenindustrie in Anlehnung an GRÄTZ (2005) .....	18
Abbildung 2.7:	Aufgaben und Ziele der Inbetriebnahme nach WEBER (2006) .....	20
Abbildung 2.8:	Ebenen in der Produktionstechnik – Automatisierungs- pyramide in Anlehnung an VOGEL-HEUSER & WANNAGAT (2009) .....	21

Abbildung 2.9:	Einteilung von elektrischen Steuerungen in Anlehnung an WECK & BRECHER (2006) .....	22
Abbildung 2.10:	Kontinuierliche und diskrete Signale in Anlehnung an FREY & BOSSERT (2008) .....	26
Abbildung 2.11:	Beispiel eines endlichen Automaten nach HOFFMANN (2009) .....	30
Abbildung 2.12:	Beispiel für ein Petri-Netz in Form eines S/T-Netzes .....	31
Abbildung 2.13:	Beispiele für Statecharts in Anlehnung an HAREL (1987) ..	32
Abbildung 2.14:	Hybrider Automat in Anlehnung an HENZINGER (1996) ....	34
Abbildung 2.15:	Grundsätzliches Funktionsprinzip der VIBN .....	39
Abbildung 2.16:	Unterschiede zwischen Hardware-, Software- und Model-in-the-Loop-Simulation .....	41
Abbildung 2.17:	Aufbau von klassischen VIBN-Modellen .....	42
Abbildung 2.18:	Aufbau von Aktor- und Sensormodellen für die VIBN .....	42
Abbildung 2.19:	Modellerstellungsprozess der VIBN .....	44
Abbildung 3.1:	Einteilung von Softwareprüftechniken in Anlehnung an LIGGESMEYER (2009) .....	48
Abbildung 3.2:	Beispiel für ein Funktionsobjekt gemäß der Richtlinie für die Funktionsbeschreibung des VDW (VDW-RICHTLINIE 2001) .....	64
Abbildung 3.3:	Funktionsmodell nach VDI-RICHTLINIE 2222 BLATT 1 (1997) .....	65
Abbildung 3.4:	Methode der „Prinziplösung“ nach KALLMEYER (1998) ....	66
Abbildung 4.1:	Ableitung von Anforderungen für die Methode aus den Zielen der vorliegenden Arbeit .....	72
Abbildung 4.2:	Parallelisierung des Entwicklungsprozesses durch den Einsatz eines Funktionsmodells zur verbesserten disziplinenübergreifenden Kommunikation und Abstimmung .....	74

Abbildung 4.3:	Berücksichtigung unterschiedlicher Anforderungen in der Grob- und der Feinplanungsphase .....	75
Abbildung 4.4:	Nutzung von CAD-Daten zur teilautomatisierten Erweiterung der VIBN-Modelle .....	79
Abbildung 4.5:	Prinzip der automatisierten VIBN-Modellgenerierung auf der Basis des Funktionsmodells .....	81
Abbildung 4.6:	Konzeptbausteine und Inhalte der vorliegenden Arbeit.....	82
Abbildung 5.1:	Definition der statischen Anteile einer Komponente durch die syntaktische Schnittstelle (AUTOVIBN 2010) .....	87
Abbildung 5.2:	Definition der dynamischen Anteile einer Komponente durch die semantische Schnittstelle (AUTOVIBN 2010) .....	88
Abbildung 5.3:	Die Komponente als Strukturierungsmittel der Modellierungstechnik zur Beschreibung von Systemen .....	89
Abbildung 5.4:	Metamodell der in dieser Arbeit verwendeten Zustandsmaschine (AUTOVIBN 2010) .....	90
Abbildung 5.5:	Hybrider Automat zur Verhaltensmodellierung eines Antriebs .....	92
Abbildung 5.6:	Zustandsmaschine einer Förderbandsteuerung mit zwei Transportstrecken .....	93
Abbildung 5.7:	Fehlermodellierung mit Hilfe von Zustandsmaschinen.....	94
Abbildung 5.8:	Abbildung der Geometrie im Funktionsmodell .....	95
Abbildung 5.9:	Einfluss der Geometrie auf die Funktionalität .....	96
Abbildung 5.10:	Modellierung von Verbahrbewegungen durch Achsen .....	97
Abbildung 5.11:	Serielle Kinematik eines Portalsystems .....	98
Abbildung 5.12:	Materialquellen und -senken im Funktionsmodell .....	99
Abbildung 5.13:	Technisches Ressourcenmodell als Bindeglied zur Konstruktionsebene.....	101

Abbildung 5.14:	Spezifikation von Aktoren im Technischen Ressourcenmodell .....	103
Abbildung 5.15:	Einteilung von Sensoren im Technischen Ressourcenmodell .....	104
Abbildung 5.16:	Zuordnung der Schnittstellen des Funktionsmodells zu den realen Steuerungsein- und -ausgängen .....	105
Abbildung 5.17:	Vorgehensmodell zum Einsatz der Funktionsmodellierung .....	106
Abbildung 5.18:	Aufbau der Steuerkette eines Pneumatikzylinders .....	108
Abbildung 5.19:	Modellierung von Aktoren in der Grobplanungsphase .....	109
Abbildung 5.20:	Funktionsmodell der Grobplanungsphase eines Transportsystems mit drei Förderstrecken und mit Teilausschleusung .....	109
Abbildung 5.21:	Stark abstrahiertes Grobplanungsmodell eines Greifers ...	110
Abbildung 5.22:	Modellierung von Wirkketten in der Grobplanungsphase am Beispiel einer einfachen Förderstrecke .....	111
Abbildung 5.23:	Geometriemodell der Feinplanungsphase .....	113
Abbildung 5.24:	Modellierung von Aktoren in der Feinplanungsphase .....	113
Abbildung 5.25:	Modell der Feinplanungsphase eines pneumatischen Greifers .....	114
Abbildung 5.26:	Komponentenstruktur des Förderbandes in der Feinplanung .....	115
Abbildung 6.1:	Konzept zur Generierung von VIBN-Modellen .....	118
Abbildung 6.2:	Kopplungsmöglichkeiten von Funktionsmodell und CAD .....	121
Abbildung 6.3:	Integration der CAD-Daten in das Funktionsmodell über Zwischenformate .....	122
Abbildung 6.4:	XML-Schema für den Datenaustausch zwischen Funktionsmodell und MCAD-Systemen .....	124

Abbildung 6.5:	XML-Schema für den Datenaustausch zwischen Funktionsmodell und E/FCAD-Systemen .....	125
Abbildung 6.6:	Anlegen von Achsen, Detektoren und Bindings im CAD .....	127
Abbildung 6.7:	Modell der Beziehungen zwischen Funktionsmodell und CAD .....	128
Abbildung 6.8:	Konsistenzprüfung der Aktoren und der Sensoren .....	129
Abbildung 6.9:	Automatisierte Integration von MCAD-Daten .....	130
Abbildung 6.10:	Erweiterung von Aktoren im Funktionsmodell mit Informationen aus dem MCAD-System .....	131
Abbildung 6.11:	Prinzip der Modellmodifikation nach STAHL & VÖLTER (2005) .....	132
Abbildung 6.12:	Prinzip der Modelltransformation nach STAHL & VÖLTER (2005) .....	133
Abbildung 6.13:	Prinzip des Model-Weavings nach Stahl & Völter (2005) .....	133
Abbildung 6.14:	Transformation des Funktionsmodells durch Integration der MCAD-Daten .....	134
Abbildung 6.15:	Integration von technischen Parametern aus dem MCAD-Modell in das Funktionsmodell .....	135
Abbildung 6.16:	Transformationsregeln zum Einbinden der Parameter aus dem MCAD-Modell in die Variablenliste der hybriden Automaten – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011) .....	137
Abbildung 6.17:	Integration der Bewegungsachsen aus dem MCAD-Modell in das Funktionsmodell .....	139

Abbildung 6.18:	Transformationsregeln zum Anpassen der Bewegungsachsen und -pfade mit Daten aus dem MCAD-Modell – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011) .....	140
Abbildung 6.19:	Integration der Detektoren aus dem MCAD-Modell in das Funktionsmodell .....	142
Abbildung 6.20:	Transformationsregeln zum Anpassen der Detektoren und der Bindings mit Daten aus dem MCAD-Modell – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010) .....	144
Abbildung 6.21:	Integration der Geometrie aus dem MCAD-Modell in die Parts des Funktionsmodells .....	146
Abbildung 6.22:	Transformationsregeln der geometriebeschreibenden Bauteildaten aus dem MCAD-Modell – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010).....	148
Abbildung 6.23:	Eingesetzte Algorithmen zur Polygonreduktion – Vertex Clustering und Pair Contraction .....	151
Abbildung 6.24:	Polygonreduktion mit Hilfe des Vertex Clustering am Beispiel des Gehäuses eines Pneumatikzylinders .....	151
Abbildung 6.25:	Automatisierte Integration von E/FCAD-Daten in das Funktionsmodell .....	153
Abbildung 6.26:	Erweiterung von Aktoren im Funktionsmodell mit Informationen aus dem E/FCAD-System.....	154
Abbildung 6.27:	Transformation des Funktionsmodells durch Integration der E/FCAD-Daten .....	155
Abbildung 6.28:	Automatisierte Erweiterung des Arbeitsgliedes am Beispiel eines doppeltwirkenden Pneumatikzylinders .....	157
Abbildung 6.29:	Automatisierte Erweiterung des Arbeitsgliedes im Funktionsmodell mit Hilfe der E/FCAD-Daten .....	158



Abbildung 6.30:	Transformationsregeln zur Detaillierung des Arbeitsgliedes eines Aktors mit Hilfe der E/FCAD-Daten – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011).....	160
Abbildung 6.31:	Automatisierte Erweiterung der Steuer- und Stellkomponente am Beispiel eines doppeltwirkenden Pneumatikzylinders .....	163
Abbildung 6.32:	Automatisierte Erweiterung der Steuer- und Stellkomponente im Funktionsmodell mit Hilfe der E/FCAD-Daten .....	164
Abbildung 6.33:	Transformationsregeln zur Erweiterung der Steuer- und Stellkomponente mit Hilfe der E/FCAD-Daten.....	166
Abbildung 6.34:	Erzeugen von Kanälen zum Verbinden von Ports – Ausschnitt des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011).....	167
Abbildung 6.35:	Transformationsregeln für das Technische Ressourcenmodell – Metamodell nach AUTOVIBN (2010).....	169
Abbildung 6.36:	Generierung des VIBN-Modells als C++-Programm auf der Basis des Funktionsmodells .....	171
Abbildung 6.37:	Berücksichtigung der Maschinenanteile bei der Generierung des VIBN-Modells .....	172
Abbildung 6.38:	HiL-Kopplung mit Hilfe einer Hardware-Simulationskarte .....	173
Abbildung 6.39:	Einpflegen von Änderungen am VIBN-Modell durch Neugenerierung .....	174
Abbildung 6.40:	Funktionsorientiertes und strukturorientiertes Testen von Simulationsmodellen in Anlehnung an LIGGESMEYER (2009) .....	175
Abbildung 6.41:	Teststrategie und Vorgehen beim Prüfen der unterschiedlichen Modelle .....	178
Abbildung 6.42:	Generierung eines MCAD-Modells mit Hilfe der Daten aus dem Funktionsmodell.....	180

Abbildung 7.1:	Editor des Werkzeugprototyps zur Funktionsmodellierung .....	184
Abbildung 7.2:	Simulationsmodus des Funktionsmodells.....	184
Abbildung 7.3:	Prinzipielle Funktionsweise der Softwareprototypen zur Erweiterung des Feinplanungsmodells mit Hilfe von CAD-Daten.....	186
Abbildung 7.4:	Hauptzeitparalleles Rüsten einer Werkzeugmaschine der Firma Gebr. Heller Maschinenfabrik GmbH.....	188
Abbildung 7.5:	Grobplanungs-, Feinplanungs- und detailliertes VIBN-Modell des hauptzeitparallelen Rüstens .....	188
Abbildung 7.6:	Versuchsaufbau bei der Firma Heller – HiL-Teststand.....	189
Abbildung 7.7:	Modellierung komplexer Verfahrprofile und Materialflüsse .....	189
Abbildung 7.8:	Vergleich eines konventionellen Simulationsansatzes mit der hier vorgestellten Methode .....	190
Abbildung 7.9:	Be- und Entladesystem einer Schleifmaschine für Zahnräder .....	191
Abbildung 7.10:	Grobplanungs-, Feinplanungs- und detailliertes VIBN-Modell des Be- und Entladesystems einer Schleifmaschine .....	192
Abbildung 7.11:	Kulisse zur Falschteilerkennung und mechanischer Kipphebel zur Rohteilidentifikation .....	193
Abbildung 7.12:	Nutzen der entwickelten Methode im Vergleich mit einem konventionellen VIBN-Ansatz .....	200
Abbildung 10.1:	XML-Schema der Mechanik für Volumenprimitive .....	227
Abbildung 10.2:	XML-Schema der Mechanik für Punkte und Vektoren im dreidimensionalen Raum .....	227
Abbildung 10.3:	XML-Schema der Mechanik für Linien und Kreisbögen..	228

Abbildung 10.4:	XML-Schema der Mechanik für Attribute und Detektoren .....	228
Abbildung 10.5:	XML-Schema der Mechanik für Bewegungsachsen und -pfade.....	229
Abbildung 10.6:	XML-Schema der Elektrotechnik und der Fluidik für Attribute .....	229
Abbildung 10.7:	XML-Schema der Elektrotechnik und der Fluidik für die Verschaltung von Komponenten.....	230



## Tabellenverzeichnis

Tabelle 2.1:	Klassifizierung von Systemen nach FREY & BOSSERT (2008) .....	27
Tabelle 2.2:	Notwendige Daten für die Modellerstellung der VIBN.....	44
Tabelle 2.3:	Nutzen und Aufwand einer VIBN .....	46
Tabelle 3.1:	Bewertung des Standes der Forschung und der Technik.....	60
Tabelle 5.1:	Bewertung von Modellierungstechniken anhand relevanter Anforderungen hinsichtlich der angestrebten Methode.....	85
Tabelle 6.1:	Nutzbare Informationen in den CAD-Systemen.....	120
Tabelle 6.2:	Übersicht der Transformationsregeln für die einzelnen Elemente eines Arbeitsgliedes .....	161



## Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
AiF	Arbeitsgemeinschaft industrieller Forschungsvereinigungen „Otto von Guericke“ e. V.
API	Application Programming Interface (Programmierschnittstelle)
AS	Ablaufsprache
AWL	Anweisungsliste
AQUIMO	Adaptierbares Modellierungswerkzeug und Qualifizierungsprogramm für den Aufbau firmenspezifischer mechatronischer Engineeringprozesse (Forschungsprojekt)
AutoVIBN	Automatische Generierung von Verhaltensmodellen aus CAD-Daten für die qualitätsorientierte Virtuelle Inbetriebnahme (Forschungsprojekt)
AVILUS	Angewandte Virtuelle Technologien im Produkt- und Produktionsmittelbenszyklus (Forschungsprojekt)
BDE	Betriebsdatenerfassung
B/E-Netz	Bedingungs-Ereignis-Netz
bzw.	beziehungsweise
ca.	circa
CAD	Computer Aided Design (rechnergestützte Konstruktion)
CAE	Computer Aided Engineering (rechnergestützte Entwicklung)
CASE	Computer Aided Software Engineering (rechnergestützte Softwareentwicklung)
CAX	Computer Aided x (x stellvertretend für weitere Akronyme, siehe auch CAD, CAE und CASE)

COM	Component Object Model (Technologie der Firma Microsoft für die Kommunikation von Softwarekomponenten)
CPU	Central Processing Unit (Hauptprozessor)
CSP	Communicating Sequential Processes (Prozessalgebra zur Beschreibung der Kommunikation zwischen Prozessen)
DFG	Deutsche Forschungsgemeinschaft
d. h.	das heißt
DIN	Deutsches Institut für Normung e. V.
DP	dezentrale Peripherie
E/A	Steuerungsein- und -ausgänge
EA	endlicher Automat
ECAD	Elektro-CAD
EFS	Erweiterte Funktionsstruktur (Modellierungsmethode nach BATHELT (2006))
E/FCAD-Modell	Modell der Elektrik und der Fluidik, das in einer gemeinsamen Entwicklungsumgebung projiziert wird und deren Querbeziehungen enthält
engl.	englisch
EN	Europäische Norm
ERP	Enterprise Resource Planning (Planung der Unternehmensressourcen)
et al.	et alii (Maskulinum), et aliae (Femininum)
etc.	et cetera
e. V.	eingetragener Verein
FBS	Funktionsbausteinsprache
FCAD	Fluid-CAD



FEM	Finite-Elemente-Methode
HiL	Hardware-in-the-Loop (Simulationsmethode, bei der die reale Steuerungshardware zum Testen verwendet wird)
HMI	Human Machine Interface (Mensch-Maschine-Schnittstelle)
HPN	Hybrides Petri-Netz
Hrsg.	Herausgeber
ID	Identifikationsnummer
IEC	International Electrotechnical Commission
INCOSE	International Council on Systems Engineering
IP	Internet Protocol (Netzwerkprotokoll – Implementierung der Vermittlungsschicht des ISO/OSI-Referenzmodells)
ISO	International Organization for Standardization
<i>iwb</i>	Institut für Werkzeugmaschinen und Betriebswissenschaften der Technischen Universität München
KONDISK	Analyse und Synthese kontinuierlich-diskreter technischer Systeme (DFG-Schwerpunktprogramm)
KOP	Kontaktplan
MaSiEd	Maschinen-Simulator/Editor
MCAD	Mechanik-CAD
MDA	Model Driven Architecture (modellgetriebener Ansatz für die Softwareentwicklung der OMG)
MDSD	Model Driven Software Development (modellgetriebene Softwareentwicklung)
MES	Manufacturing Execution System (Produktionsleitsystem)
MiL	Model-in-the-Loop (Simulationsmethode, bei der die reale Steuerung durch ein Modell ersetzt wird)
MIT	Massachusetts Institute of Technologie

MKS	Mehrkörpersimulation
MSC	Message Sequence Chart (Nachrichten-Reihenfolge-Diagramm)
NC	Numerical Control (Numerische Steuerung)
OMG	Object Management Group
OPC	OLE for Process Control (Softwareschnittstelle in der Automatisierungstechnik)
PC	Personal Computer (Einzelplatzrechner)
PCI	Peripheral Component Interconnect (Bus-Standard zur Verbindung von Peripheriegeräten mit dem Chipsatz eines Prozessors)
PIM	Platform Independant Model (plattformunabhängiges Modell)
PN	Petri-Netz
PROFIBUS	Process Field Bus (weit verbreiteter Feldbus-Standard in der Automatisierungstechnik)
PSM	Platform Specific Model (plattformspezifisches Modell)
P/T-Netz	Prädikat-Transitions-Netz
PZD	Prozessdatenwort
ROOM	Real-Time Object Oriented Modeling (Modellierungssprache für den Entwurf von Echtzeitsystemen)
RT	Real-Time (Echtzeit)
S.	Seite
SiL	Software-in-the-Loop (Simulationsmethode, bei der die reale Steuerung durch einen Emulator ersetzt wird)
SPIN	Steuerungstechnisch interpretiertes Petri-Netz
SPS	Speicherprogrammierbare Steuerung
ST	Strukturierter Text

STEM	Spatio-Temporal Engineering Models (Entwicklungsumgebung für die Funktionsmodellierung und die VIBN des Forschungsprojektes AutoVIBN)
STEP	Standard for the Exchange of Product Model Data (Standard für den Produktdatenaustausch)
S/T-Netz	Stellen-Transitions-Netz
STW	Steuerwort
SysML	Systems Modeling Language (Modellierungssprache für die Systementwicklung)
TCP	Transmission Control Protocol (Übertragungssteuerungsprotokoll – Transportschicht des ISO/OSI-Referenzmodells)
TD	Technisches Detail
TRM	Technisches Ressourcenmodell
TUM	Technische Universität München
UML	Unified Modeling Language (Modellierungssprache für die Softwareentwicklung und weitere Anwendungsbereiche)
UML-PA	UML for Process Automation (UML für die Prozessautomatisierung)
UML-RT	UML for Real-Time (UML für Echtzeitanwendungen)
USD	US Dollar
VDE	Verband der Elektrotechnik Elektronik Informationstechnik e. V.
VDI	Verein Deutscher Ingenieure e.V.
VDMA	Verband Deutscher Maschinen- und Anlagenbau e. V.
VDW	Verein Deutscher Werkzeugmaschinenfabriken e. V.
VIBN	Virtuelle Inbetriebnahme
VR	Virtual Reality (Virtuelle Realität)

VRML	Virtual Reality Modeling Language (Beschreibungssprache für 3D-Modelle)
W3C	World Wide Web Consortium
X3D	Extensible 3D (Beschreibungssprache für 3D-Modelle)
XML	Extensible Markup Language (Auszeichnungssprache für hierarchisch strukturierte Textdaten)
z. B.	zum Beispiel

## Formelverzeichnis

### Griechische Buchstaben

Zeichen	Bedeutung
$\Gamma$	Endliche Menge an Ausgabesymbolen eines Transduktors
$\delta$	Übertragungsfunktion eines endlichen Automaten
$\Sigma$	Endliche Menge von Eingabesymbolen eines endlichen Automaten
$\omega$	Ausgabefunktion eines Transduktors

### Lateinische Buchstaben

Zeichen	Bedeutung
$a_0 \dots a_n$	Koeffizienten des Nennerpolynoms einer Übertragungsfunktion
$A$	Zeitbedingung
$b_0 \dots b_m$	Koeffizienten des Zählerpolynoms einer Übertragungsfunktion
$\mathbb{C}$	Menge der komplexen Zahlen
$d$	Zeitpunkt, zu welchem eine betrachtete Aufgabe abgeschlossen sein muss
$\Delta t$	Zeit, die zur Ausführung einer Aufgabe benötigt wird
$\mathbb{E}^3$	Dreidimensionaler euklidischer Raum
$EA$	Endlicher Automat in Tupel-Notation
$f()$	Funktion

$F$	Kantenfunktion eines Petri-Netzes
$G(s)$	Übertragungsfunktion
$i$	Zählervariable
$k$	Diskrete Zeitvariable
$K$	Kapazität einer Stelle eines Petri-Netzes
$m$	Zählervariable
$M_0$	Anfangsmarkierung eines Petri-Netzes
$n$	Zählervariable
$n$	Grad der Ableitung
$N(s)$	Nennerpolynom einer Übertragungsfunktion
$PN$	Petri-Netz in Tupel-Notation
$Q$	Menge finaler oder akzeptierender Zustände eines endlichen Automaten
$r$	Zeitpunkt, zu welchem eine betrachtete Aufgabe beginnt
$\mathbb{R}$	Menge der reellen Zahlen
$s$	Variable der Laplace-Transformierten
$s_0$	Startzustand eines endlichen Automaten
$S$	Endliche Menge von Zuständen eines endlichen Automaten
$S$	Endliche Menge von Stellen eines Petri-Netzes
$S\{ \}$	Abbildung der Eingangs- auf die Ausgangssignale eines Systems
$t$	Kontinuierliche Zeitvariable
$T$	Endliche Menge von Transitionen eines Petri-Netzes
$W$	Gewichtungsfunktion einer Kante eines Petri-Netzes
$x$	Allgemeine Variable einer Differenzialgleichung

$x_A$	Systemausgänge der Übertragungsfunktion
$x_E$	Systemeingänge der Übertragungsfunktion
$x_i$	Eingaben eines endlichen Automaten
$x(k)$	Zeitdiskretes Eingangssignal eines Systems
$x(t)$	Zeitkontinuierliches Eingangssignal eines Systems
$y$	Funktion, die n Mal stetig differenzierbar ist
$y_i$	Ausgaben eines endlichen Automaten
$y(k)$	Zeitdiskretes Ausgangssignal eines Systems
$y(t)$	Zeitkontinuierliches Ausgangssignal eines Systems
$\mathbb{Z}$	Menge der ganzen Zahlen
$Z(s)$	Zählerpolynom einer Übertragungsfunktion

## Mathematische Zeichen

Zeichen	Bedeutung
$\neg$	Negation
$\wedge$	Oder-Verknüpfung (Disjunktion)
$\vee$	Und-Verknüpfung (Konjunktion)
$\equiv$	Kongruenz





## Glossar

Im Rahmen der vorliegenden Dissertation wurde eine neue Modellierungstechnik erarbeitet, für die eine Reihe von Begriffen eingeführt wurde. Im Glossar sind Bezeichnungen, die sich auf diese Modellierungstechnik beziehen, explizit gekennzeichnet (\*).

### Binding\*

Das Binding beschreibt einen geometrisch definierten Einflussbereich einer Komponente (siehe Begriff *Komponente*), in welchem unterschiedliche Operationen mit den sich darin befindlichen Materialobjekten durchgeführt werden können. Die Bezeichnung *Binding* ist ein Oberbegriff für die zwei existierenden Ausprägungen: das PortBinding (siehe Begriff *PortBinding*) und das MoverBinding (siehe Begriff *MoverBinding*).

### Bondgraph

Der Bondgraph ist eine Modellierungstechnik zur Darstellung kontinuierlicher Systeme, die von PAYNTER (1959) entwickelt wurde. Es handelt sich dabei um gerichtete Graphen, die zur Simulation von Energieflüssen verwendet werden können.

### Detektor\*

Die Detektoren dienen zur Erkennung von Objekten in einem geometrisch definierten Raum. Sie können informationstechnisch durch die hybriden Automaten (siehe Begriff *hybrider Automat*) ausgewertet werden.

### Endlicher Automat

Der endliche Automat wird auch als Zustandsmaschine bezeichnet und ist eine Technik zur Modellierung ereignisdiskreter Systeme. Er verfügt über eine endliche Anzahl an Zuständen, die über Transitionen (siehe Begriff *Transition*) gewechselt werden können.

### FOCUS

FOCUS ist eine Methode zur Modellierung verteilter, reaktiver Systeme, die von BROY & STØLEN (2001) entwickelt wurde.

### Hybrider Automat\*

Der hybride Automat ist eine Variante des endlichen Automaten, bei welcher der Zustandsraum um kontinuierliche Variablen erweitert ist. Mit Hilfe von Differenzialgleichungen können die Änderungen dieser Variablen beschrieben werden. Der im Rahmen dieser Arbeit entwickelte hybride Automat verfügt über Mechanismen zur Interaktion mit dem Materialfluss, den Movern (siehe Begriff *Mover*), den Detektoren (siehe Begriff *Detektor*) sowie den PortBindings (siehe Begriff *PortBinding*) und den MoverBindings (siehe Begriff *MoverBinding*).

### Inspektion, Review und Walkthrough

Inspektionen, Reviews und Walkthroughs sind analytische Prüfmethode für Software, die sich anhand ihrer formalen Ausprägung unterscheiden lassen (Reihenfolge mit abnehmender Formalität: Inspektion → Review → Walkthrough).

### Kanal\*

Mit Hilfe der Kanäle können die Ports (siehe Begriff *Port*) einer Komponente (siehe Begriff *Komponente*) verbunden werden. Dadurch ist ein Informationsaustausch möglich.

### Komponente\*

Die Komponente ist ein Mittel zur hierarchischen Strukturierung von technischen Systemen. Sie kann weitere Komponenten oder andere Elemente der im Rahmen dieser Arbeit entwickelten Modellierungstechnik (z. B. einen hybriden Automaten oder Ports) enthalten.

### Mover\*

Die Mover dienen zur Festlegung der Freiheitsgrade einer Komponente (siehe Begriff *Komponente*). Für die Definition stehen Linearachsen, Rotationsachsen und interpolierte Achsen zur Verfügung.

### MoverBinding\*

Das MoverBinding beschreibt einen geometrisch definierten Einflussbereich einer Komponente (siehe Begriff *Komponente*), in welchem Materialobjekte verfahren werden können. Die Bewegungsrichtung wird mittels der Mover (siehe Begriff *Mover*) bestimmt, die Geschwindigkeit wird durch den zugehörigen hybriden Automaten (siehe Begriff *hybrider Automat*) der Komponente berechnet.

**Part\***

Ein Part repräsentiert die Geometrie eines Maschinenbauteils und ist einer Komponente (siehe Begriff *Komponente*) zugeordnet. Ferner können Parts über kinematische Freiheitsgrade verfügen, die durch die Mover (siehe Begriff *Mover*) festgelegt sind. Die Berechnung der Geschwindigkeit erfolgt durch den hybriden Automaten (siehe Begriff *hybrider Automat*) der zugehörigen Komponente.

**Petri-Netz**

Das Petri-Netz ist eine Modellierungstechnik, die von PETRI (1962) entwickelt wurde und welche die Automatentheorie um den Aspekt der Nebenläufigkeit erweitert.

**Port\***

Die Ports sind die Kommunikationsschnittpunkte einer Komponente (siehe Begriff *Komponente*). Ein Port verfügt über eine Richtung (eingehend oder ausgehend) sowie über einen Datentyp. Die Versendung von Nachrichten kann ereignis- oder zustandsbasiert erfolgen.

**PortBinding\***

Das PortBinding beschreibt einen geometrisch definierten Einflussbereich einer Komponente (siehe Begriff *Komponente*), in welchem Nachrichten über die Ports (siehe Begriff *Port*) an Materialobjekte, die wiederum selbst als Komponente modelliert sind, versendet werden können.

**Statecharts**

Statecharts erweitern Zustandsdiagramme um ein hierarchisches Konzept, um parallele Zustände sowie um einen Kommunikationsmechanismus (HAREL 1987).

**Steuerungsintelligenz**

Der Begriff *Steuerungsintelligenz* beschreibt die Fähigkeit, Abläufe in technischen Systemen zu steuern. Sie kann zentral oder dezentral verteilt sein.

**Transition**

Der Übergang zwischen zwei Zuständen eines endlichen Automaten (siehe Begriff *endlicher Automat*) wird als Transition bezeichnet.



# 1 Einleitung

## 1.1 Gegenwärtige Situation im Maschinen- und Anlagenbau

Der Maschinen- und Anlagenbau zählt zu den wichtigsten Arbeitgebern in Deutschland. Mit deutlich über 900.000 Beschäftigten, und damit mehr als in allen anderen Industriezweigen, werden vorwiegend Einzelanfertigungen und Kleinserien hergestellt (VDMA 2010). Im Vergleich zur meist preisgünstigeren ausländischen Konkurrenz zeichnen sich die deutschen Produkte vor allem durch eine individuelle Anpassung an die Bedürfnisse des Kunden aus (TOMASZUNAS 1999, KOHRING 1993). Dies spiegelt sich im Entwicklungsaufwand wider, was durch zusätzliche Kosten und einen erhöhten Zeitdruck verdeutlicht wird. Dem gegenüber steht der Anspruch der Hersteller, die Gesamtentwicklungszeiten zu verkürzen, um Marktpositionen zu halten oder auszubauen (ZÄH ET AL. 2005A).

Darüber hinaus tritt ein weiterer grundlegender Trend im Maschinen- und Anlagenbau zutage. Die Produkte werden in zunehmendem Maße durch die Steuerungssoftware geprägt, deren Anteil in den vergangenen Jahrzehnten stetig angewachsen ist (GLAS 1993). Mechanische Wirkprinzipien werden vermehrt durch mechatronische Lösungen substituiert (BAUDISCH ET AL. 2007). Die Funktionserbringung ist nur noch im Zusammenwirken der Einzeldisziplinen Mechanik, Elektrotechnik und Software möglich. Eine Studie des Massachusetts Institute of Technology (MIT) sieht in der Mechatronik eine Zukunftstechnologie, der eine weiter steigende Bedeutung zugemessen wird (MIT 2003). Im Maschinenbau werden gemäß BENDER (2005) ca. 90 Prozent der wichtigen Innovationen, die nicht nur Detailverbesserungen darstellen, bereits in der Informationstechnik realisiert.

Die Gründe für den verstärkten Einsatz von mechatronischen Komponenten und speziell für den wachsenden Softwareanteil sind vielfältiger Natur. Dies können beispielsweise Kundenanforderungen oder Alleinstellungsmerkmale sein. Insbesondere ist aber die Flexibilität von Softwarelösungen zu nennen. Änderungen der Maschinenfunktionen können in der Steuerungssoftware einfacher als in der Maschinenhardware vorgenommen werden und implizieren keine konstruktiven Anpassungen. Deshalb werden häufig Fehler aus vorgelagerten Entwicklungsschritten softwareseitig ausgeglichen (BENDER 2005, THRAMOUBILDIS 2008).

### 1.2 Resultierende Problemstellung

Die Funktionsverlagerung in die Steuerungssoftware bringt zwar wesentliche Vorteile mit sich, erhöht allerdings im gleichen Maße die Komplexität der Programme. Abgesehen von der angestrebten Maschinenfunktionalität müssen zusätzlich Sicherheitsaspekte berücksichtigt werden. Hierzu gehören Fehler- und Ausfallszenarien, für die eine korrekte Handhabung zu gewährleisten ist. Der Test von Störfällen gestaltet sich in der industriellen Praxis schwierig, da diese nicht selten einen destruktiven Charakter haben und in Prüfläufen Schäden an der Anlage hervorrufen können. Aus Zeit- und Sicherheitsgründen unterbleibt daher meist eine systematische Untersuchung der Fehlerfälle, was sich unter Umständen negativ in der späteren Betriebsphase auswirkt. Deswegen ist die Software zu einem erheblichen Kostenfaktor herangewachsen, der mitunter einen Großteil der Entwicklungsaufwendungen ausmacht (BENDER 2005, KUPPINGER 2006).

Hierbei stellt sich die Frage, wie eine derart komplexe und umfangreiche Software noch ausreichend getestet werden kann. Erschwerend kommt hinzu, dass die Entwicklungsprozesse der Maschinen- und Anlagenhersteller klassischer Weise sequenziell aufgebaut sind (GRÄTZ 2005). Die Softwareentwicklung findet sehr spät, als nachgelagerte Phase der Mechanik- und der Elektrokonstruktion, statt. Während der Inbetriebnahme kann erstmals das Zusammenwirken der Einzeldisziplinen überprüft werden. Dabei müssen Fehler aus den vorangegangenen Entwicklungsschritten kompensiert werden. Dies erhöht den Zeitdruck beträchtlich, zumal auch organisatorische Belange eine wichtige Rolle spielen, da in der Regel vor Ort, beim Kunden, gearbeitet werden muss (WÜNSCH 2008). Darüber hinaus verursacht in diesem Stadium die fast fertiggestellte Maschine erhebliche Kapitalbindungskosten (EVERSHEIM & SOSSENHEIMER 1987).

Als Lösung der beschriebenen Problemstellung wurden in den letzten Jahren simulationsgestützte Ansätze entwickelt. Mit Hilfe virtueller Maschinenmodelle, die statt der realen Maschine an die Steuerung gekoppelt werden, lässt sich die Software frühzeitig im Entwicklungsprozess testen, siehe Abbildung 1.1. Die Methode ist unter dem Begriff *Virtuelle Inbetriebnahme (VIBN)* bekannt. Nach ZÄH ET. AL (2005B) besteht deren Grundprinzip aus einer Vorwegnahme der Steuerungsinbetriebnahme an einem virtuellen Modell der mechanischen, hydraulischen, pneumatischen und elektrischen Bestandteile einer Maschine.

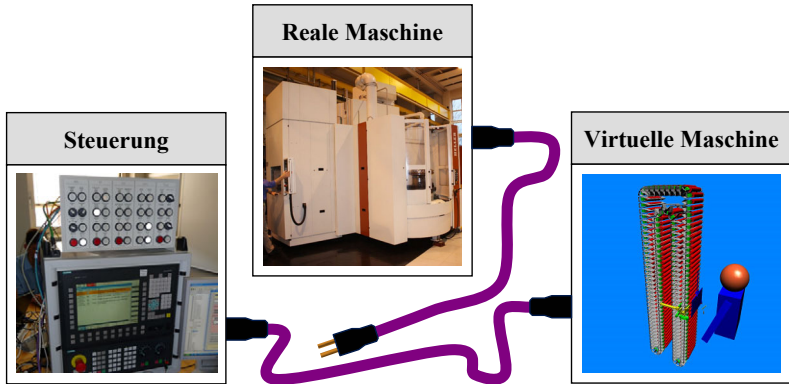


Abbildung 1.1: Grundprinzip der Virtuellen Inbetriebnahme (VIBN)

Trotz zahlreicher Vorteile konnte sich die VIBN bisher noch nicht als qualitätssichernde Maßnahme flächendeckend im industriellen Umfeld durchsetzen. Dies begründet sich vor allem in der aufwendigen Modellerstellung, die sich sehr zeit- und personalintensiv gestaltet (KIEFER 2007, SPITZWEG 2009). Der Modellierungsprozess ist durch eine manuelle Vorgehensweise geprägt und erfährt bestenfalls durch proprietäre Bibliotheksmechanismen Unterstützung. Zudem erweist sich die Integration der VIBN in den Entwicklungsprozess als problematisch. Da für die Erstellung der Simulationsmodelle detaillierte Informationen über die Maschine notwendig sind, wird damit erst nach Abschluss der Mechanik- und der Elektrokonstruktion begonnen. Eine automatisierte Auswertung dieser Daten findet in der Regel nicht statt. Die sequenziell aufgebauten Entwicklungsprozesse der Maschinen- und Anlagenbauer führen somit zu einem späten Einsatz der Simulationsmethode, was deren Nutzen erheblich verringert. Für sehr umfangreiche und materialflussintensive Systeme ist die Modellerstellung in einem derart kurzen Zeitrahmen teilweise nicht mehr möglich.

Des Weiteren kann die VIBN in den späten Phasen nur noch zum Test für eine vorgegebene Funktionalität verwendet werden. Ob sich jedoch die Funktion hätte einfacher oder sicherer gestalten lassen, ist nicht mehr Teil des Prüfverfahrens. Dies resultiert daraus, dass eventuell notwendige Konstruktionsänderungen auf Grund der hohen Folgekosten nicht mehr vollzogen werden. Ein Großteil der Fehler entsteht bei der Steuerungssoftwareentwicklung allerdings schon während der Spezifikation der Steuerungsaufgabe und nicht erst bei der Programmierung (WECK & BRECHER 2006). Daher würde sich ein frühestmöglichster Simulations-

einsatz wirtschaftlich lohnen. Abbildung 1.2 zeigt, dass durch die frühe Fehlerverursachung und die späte Beseitigung der Fehler die Kosten für den Test bzw. die Inbetriebnahme sehr hoch ausfallen (OESTREICHER 1986).

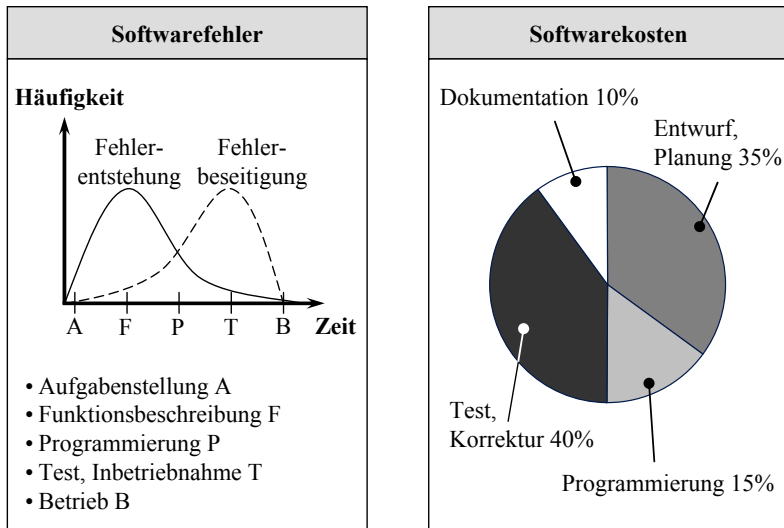


Abbildung 1.2: Fehler und Kosten bei der Steuerungssoftwareentwicklung nach OESTREICHER (1986)

### 1.3 Zielsetzung und Vorgehensweise

Basierend auf den im vorangegangenen Abschnitt erläuterten Problemstellungen wird für die vorliegende Arbeit die Zielsetzung formuliert. Es soll eine Methode entwickelt werden, die sich durch folgende Eigenschaften auszeichnet:

1. **Frühzeitige Integration der Modellbildung und der Simulation in den Entwicklungsprozess**
2. **Entwicklungsbegleitende Vorgehensweise**
3. **Teilautomatisierte Ableitung von VIBN-Modellen**
4. **Integration von Störszenarien**



Die Simulation ist frühzeitig in den Entwicklungsprozess zu integrieren, da sich ansonsten deren Vorteile durch die aufwendige Modellerstellung egalisieren. Dadurch lassen sich bereits in der Planungs- und der Projektierungsphase, noch bevor die Konstruktionsarbeit begonnen hat, anvisierte technische Lösungen auf deren prinzipielle Realisierbarkeit und Widerspruchsfreiheit prüfen. Somit können konzeptionelle Fehler rechtzeitig erkannt und beseitigt werden. Des Weiteren wird eine entwicklungsbegleitende Vorgehensweise angestrebt. Die Steuerungsoftwareentwicklung soll parallelisiert zur Mechanik- und zur Elektrokonstruktion stattfinden können, wobei dies durch zweckmäßige Modelle zu unterstützen ist. Das VIBN-Modell für den Steuerungstest soll weitestgehend automatisch erzeugt werden. Außerdem wird eine einfache Berücksichtigung von Störszenarien intendiert, die bisher wegen des hohen Aufwandes in der industriellen Praxis meist vernachlässigt wurden.

Um die oben genannten Ziele zu erfüllen, wird die in Abbildung 1.3 dargestellte Methode verfolgt. Das im Rahmen der Arbeit entwickelte Verfahren ist für den Maschinen- und Anlagenbau ausgelegt, wobei Systeme mit speicherprogrammierbaren Steuerungen (SPS) adressiert werden. Eine Übertragbarkeit auf andere Bereiche ist prinzipiell möglich, wurde aber nicht explizit betrachtet. Die Basis der Methode bilden die vier Modellkategorien Funktions-, Mechanik-CAD-(MCAD), Elektro-CAD-(ECAD) und VIBN-Modell. Da in modernen ECAD-Werkzeugen häufig auch die Fluidik projektiert werden kann, wird das dort hinterlegte Modell im weiteren Verlauf als E/FCAD-Modell bezeichnet. Das Grundprinzip der Methode besteht darin, eine Maschine zu Beginn des Entwicklungsprozesses durch ein abstraktes und schnell erstellbares Funktionsmodell zu beschreiben. Daraus können bereits erste, grobe VIBN-Modelle abgeleitet werden. Die im Laufe der Konstruktionsphase anfallenden Daten dienen im Anschluss der teilautomatisierten Erweiterung und werden in das Funktionsmodell rückgeführt, welches dann wiederum für die Generierung wesentlich detailgetreuerer VIBN-Modelle nutzbar ist.

Das Funktionsmodell wird in der Planungs- und der Projektierungsphase erstellt und umfasst sowohl das Steuerungs- als auch das Maschinenverhalten. Darüber hinaus können darin ebenfalls die Störszenarien spezifiziert werden. Mit Hilfe der Simulation ist eine Beurteilung des Gesamtverhaltens möglich. Des Weiteren fungiert das Funktionsmodell als Eingangsinformation für die disziplinspezifischen Entwicklungsstränge. Es kann entwicklungsbegleitend als Abstimmungs- und Kommunikationsmittel eingesetzt werden. KORN (1996) geht davon aus, dass 40 bis 50 Prozent der Arbeitszeit interdisziplinärer Entwicklungsprojekte

dazu aufgewendet werden, um Daten auszutauschen und Prozesse zu synchronisieren. Zur besseren Koordination und zur Diskussion zwischen den beteiligten Fachbereichen kann das Funktionsmodell einen Beitrag leisten, wodurch sich dessen Erstellung zusätzlich amortisiert.

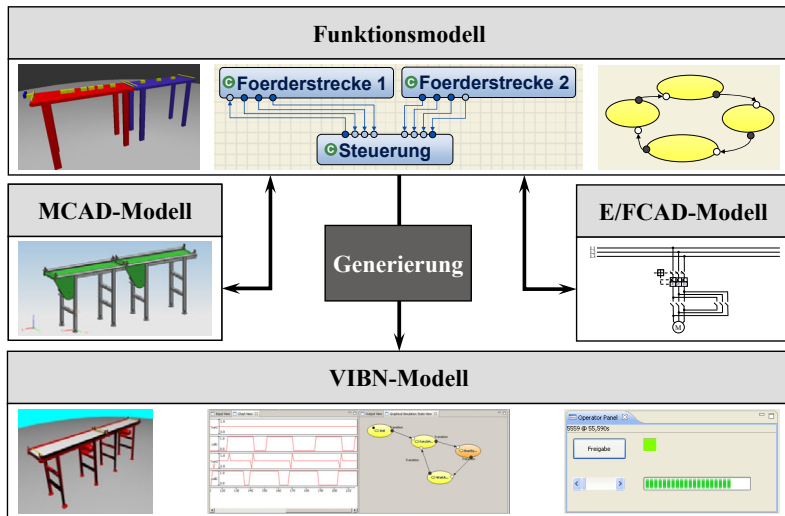


Abbildung 1.3: Ansatz zur Integration und Teilautomatisierung der VIBN

Durch den Einsatz des Funktionsmodells ergibt sich im Vergleich zur klassischen VIBN, wie sie in der Industrie zum Einsatz kommt, ein stark veränderter Arbeitsablauf. In Abbildung 1.4 sind die unterschiedlichen Abfolgen der einzelnen Schritte gegenübergestellt. Die in dieser Dissertation beschriebene Methode auf der Basis des Funktionsmodells kommt wesentlich früher zum Einsatz, als dies bei einer konventionellen VIBN der Fall ist, die eher einen nachgelagerten Prozess verkörpert. Dadurch können Änderungen rechtzeitig berücksichtigt werden. Diese Möglichkeit ist im Hinblick auf einen herkömmlichen VIBN-Ansatz in der Regel nicht mehr gegeben, da die Modellerstellung auf die Ergebnisse der vorangegangenen Konstruktionsarbeiten aufbaut. Bedingt durch die hohen Kosten werden Iterationsschleifen zur konstruktiven Optimierung einer Maschine in der Praxis vermieden. Ein wesentlicher Unterschied zwischen den beiden verglichenen Methoden besteht darin, dass das Funktionsmodell für weitere Entwicklungstätigkeiten (z. B. als übergreifender Informationsspeicher) verwendbar ist und daraus das VIBN-Modell automatisch generiert werden kann.

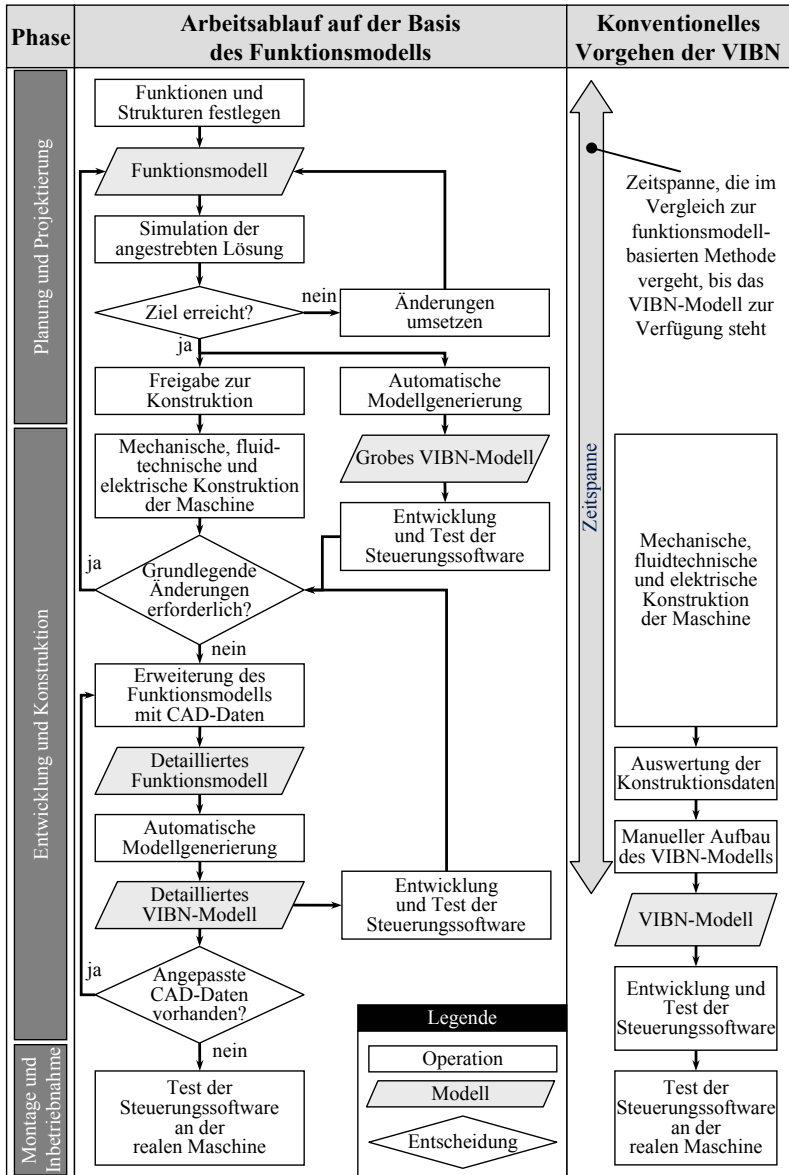


Abbildung 1.4: Vergleich der Arbeitsabläufe zwischen der Methode auf der Basis des Funktionsmodells und einer konventionellen VIBN

Zur Umsetzung der erläuterten Methode können vier wesentliche Handlungsbe-  
reiche ausgemacht werden, an denen sich der grundlegende Aufbau der vorlie-  
genden Arbeit orientiert:

- Modellierungstechnik

Für die Funktionsmodellierung ist eine geeignete Beschreibungstechnik  
erforderlich, mit Hilfe derer sich sowohl das Steuerungs- als auch das  
Maschinenverhalten abbilden lassen. Zudem müssen essenzielle Aspek-  
te, wie beispielsweise der Materialfluss, integriert werden können.

- Vorgehensmodell

Das Vorgehen unter Einsatz der Modellierungstechnik ist zu strukturie-  
ren. Dies umfasst die Gliederung in einzelne Phasen sowie die Definiti-  
on von Modellierungsrichtlinien.

- Integration von CAD-Daten

Das Funktionsmodell soll mit CAD-Daten angereichert werden. Hierzu  
sind Konstruktionsrichtlinien notwendig, die den Automatisierungsgrad  
der Datenübernahme erhöhen. Zudem ist die Konsistenz der Informati-  
onen sicherzustellen.

- Automatische Generierung des VIBN-Modells

Auf der Grundlage des Funktionsmodells sollen VIBN-Modelle abgelei-  
tet werden. Dafür sind zusätzliche Informationen notwendig, die bei der  
Generierung mit eingebunden werden müssen, wie z. B. die Kopplung  
zur Steuerung.

## 1.4 Gliederung der Arbeit

Die vorliegende Arbeit gliedert sich in acht inhaltliche Kapitel, welche in  
Abbildung 1.5 aufgelistet sind. In Kapitel 1 wurden zunächst die Ausgangssitua-  
tion und die Problemstellung im Maschinen- und Anlagenbau beschrieben. An-  
schließend wurde beruhend darauf die Zielstellung definiert und ein prinzipieller  
Lösungsansatz skizziert.

Das zweite Kapitel soll die Grundlagen im Hinblick auf die VIBN vermitteln.  
Dazu werden die Entwicklungsprozesse von Produktionsmaschinen analysiert,

das steuerungstechnische Basiswissen zusammengefasst sowie die Prinzipien der Modellbildung und der Simulation näher betrachtet. Darauf aufbauend wird die VIBN in allen Einzelheiten vorgestellt. Dies umfasst sowohl die wesentlichen Grundsätze als auch deren Spezifika in Bezug auf die Modellbildung.

In Kapitel 3 wird der Stand der Forschung und der Technik ausführlich dargestellt. Neben simulationsbasierten Ansätzen, die im Fokus der vorliegenden Arbeit stehen, werden zudem weitere Prüfmöglichkeiten für Steuerungssoftware geschildert. Beruhend auf den dort gewonnenen Erkenntnissen wird der Handlungsbedarf für die VIBN konkretisiert. Darüber hinaus wird als zweiter Schwerpunkt die abstrakte Modellierung von Maschinen und Anlagen in frühen Entwicklungsphasen unter funktionalen Gesichtspunkten aufgegriffen.

Die Beschreibung des konzeptionellen Hintergrundes der hier vorgestellten Methode ist in Kapitel 4 realisiert, welches als Überblick für die dedizierte Ausarbeitung in den sich anschließenden Kapiteln fungiert. Es beinhaltet die Integration des Ansatzes in den Entwicklungsprozess, die genauere Bestimmung einer geeigneten Modellierungstechnik, die automatisierte Nutzung vorhandener Konstruktionsdaten für die Modellbildung sowie die Ableitung von Simulationsmodellen für die VIBN.

Die detaillierte Darstellung und die Umsetzung der Methode in allen Einzelheiten findet sich in den Kapiteln 5 und 6. Dabei widmet sich das fünfte Kapitel der funktionalen Modellierung von Maschinen und Anlagen. Vorab werden die Anforderungen an die Modellierungstechnik definiert und gängigen Beschreibungsformen gegenübergestellt. Im Anschluss daran werden die hier entwickelte Modellierungstechnik und das zugehörige Vorgehensmodell präsentiert. Das sechste Kapitel behandelt die CAD-Datenerweiterung des Funktionsmodells und die automatisierte Ableitung des VIBN-Modells.

In Kapitel 7 werden die Werkzeugprototypen (Softwarewerkzeuge) und zwei industrielle Anwendungsbeispiele vorgestellt, welche zur Validierung der beschriebenen Methode dienen. Eine Wirtschaftlichkeitsbetrachtung rundet das Kapitel ab und soll eine Einordnung der erzielten Ergebnisse ermöglichen.

Das achte Kapitel fasst die Methode sowie die Ergebnisse der vorliegenden Arbeit zusammen und gibt Impulse für zukünftigen Forschungsbedarf.

<b>1 Einleitung</b>	<ul style="list-style-type: none"><li>• Ausgangssituation</li><li>• Problemstellung</li><li>• Zielsetzung und Lösungsansatz</li></ul>
<b>2 Grundlagen</b>	<ul style="list-style-type: none"><li>• Entwicklungsprozesse und Steuerungstechnik</li><li>• Modellbildung und Simulation</li><li>• Grundlagen der VIBN</li></ul>
<b>3 Stand der Forschung und der Technik</b>	<ul style="list-style-type: none"><li>• Prüfung von Steuerungssoftware</li><li>• Simulationsbasierte Testmethoden</li><li>• Planung und Konzeption im Maschinenbau</li></ul>
<b>4 Konzept</b>	<ul style="list-style-type: none"><li>• Integration in den Entwicklungsprozess</li><li>• Funktionsmodellierung</li><li>• Erweiterung und Ableitung des VIBN-Modells</li></ul>
<b>5 Funktionsmodellierung von Maschinen und Anlagen</b>	<ul style="list-style-type: none"><li>• Anforderungen an die Beschreibungstechnik</li><li>• Vorstellung der Modellierungstechnik</li><li>• Vorgehensmodell</li></ul>
<b>6 Teilautomatisierte Modellerstellung</b>	<ul style="list-style-type: none"><li>• Austauschformat für CAD-Daten</li><li>• Erweiterung des Funktionsmodells</li><li>• Ableitung des VIBN-Modells</li></ul>
<b>7 Prototypische Umsetzung und Wirtschaftlichkeitsbetrachtung</b>	<ul style="list-style-type: none"><li>• Entwickelte Softwarewerkzeuge</li><li>• Anwendungsbeispiele</li><li>• Wirtschaftlichkeitsbetrachtung</li></ul>
<b>8 Zusammenfassung und Ausblick</b>	<ul style="list-style-type: none"><li>• Zusammenfassung der Methodik</li><li>• Bewertung der Ergebnisse</li><li>• Weiterer Forschungsbedarf</li></ul>

Abbildung 1.5: Gliederung und Aufbau der vorliegenden Arbeit

## 2 Grundlagen

### 2.1 Kapitelüberblick

Das zweite Kapitel enthält die nötigen Grundlagen für die VIBN. Es soll ferner den in Kapitel 3 beschriebenen Stand der Forschung und der Technik ergänzen. Zunächst werden in Abschnitt 2.2 der Begriff und die Bedeutung der Mechatronik erklärt sowie der prinzipielle Aufbau eines mechatronischen Systems skizziert. Nachfolgend wird der Entwicklungsprozess von Produktionsmaschinen eingehend behandelt (Abschnitt 2.3). Dies umfasst verschiedene Konstruktionsmethoden sowie die einzelnen Entwicklungsphasen samt den Inhalten, wobei auf die Inbetriebnahme genauer eingegangen wird. Der Systembegriff und das allgemeine Basiswissen zur Modellbildung und zur Simulation stehen im Fokus von Abschnitt 2.5. Abschließend wird die VIBN detailliert in Abschnitt 2.6 betrachtet. Das Grundprinzip, die Ausprägungen sowie die Modellbildung mit konkretem Bezug zum simulativen Steuerungstest werden ausgiebig erläutert.

### 2.2 Begriff der Mechatronik

Der Begriff *mechatronics* (zu Deutsch: Mechatronik) wurde erstmals vom Präsident der YASKAWA Electric Corporation, Ko Kikuchi, im Jahre 1969 verwendet (HARASHIMA ET AL. 1996). Dieses Kunstwort setzt sich aus den Wörtern *mechanism* (Mechanismus), welches später durch *mechanics* (Mechanik) substituiert wurde, und *electronics* (Elektronik) zusammen.

Entsprechend der VDI-RICHTLINIE 2206 (2004) wird die Mechatronik wie folgt beschrieben, wobei sich die Definition an HARASHIMA ET AL. (1996) anlehnt:

*Mechatronik bezeichnet das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung.*

Eine ähnliche Definition formuliert SCHWEITZER (1986), der die Mechatronik als *ein interdisziplinäres Gebiet der Ingenieurwissenschaften (sieht), das auf den klassischen Disziplinen Maschinenbau, Elektrotechnik und Informatik aufbaut.*

*Ein typisches mechatronisches System nimmt Signale auf, verarbeitet sie und gibt Signale aus, die es z. B. in Kräfte und Bewegungen umsetzt.*

Die *International Federation for the Theory of Machines and Mechanisms* deutet die Mechatronik wie folgt (LOENDES 2000):

*Mechatronics is the synergistic combination of precision mechanical engineering, electronic control and systems thinking in the design of products and manufacturing processes.*

Es existieren zahlreiche weitere und sich davon unterscheidende Definitionen der Mechatronik, der gemeinsame Grundgedanke besteht jedoch bei allen im interdisziplinären Charakter. Die Funktionserbringung ist nur im Zusammenwirken der einzelnen Teilbereiche möglich.

Gemäß der VDI-RICHTLINIE 2206 (2004) besteht ein mechatronisches System aus dem Grundsystem, der Informationsverarbeitung sowie den Aktoren und den Sensoren, siehe Abbildung 2.1. Das Grundsystem setzt sich aus mechanischen, elektromechanischen, hydraulischen und/oder pneumatischen Komponenten zusammen. Über die Aktoren werden die Zustandsgrößen gezielt beeinflusst. Die Sensoren wandeln wiederum die Zustandsgrößen in rechnertechnisch interpretierbare Signale um. Die Informationsverarbeitung nimmt die Messwerte auf und legt die notwendigen Einwirkungen über die Aktoren fest (VDI-RICHTLINIE 2206 2004).

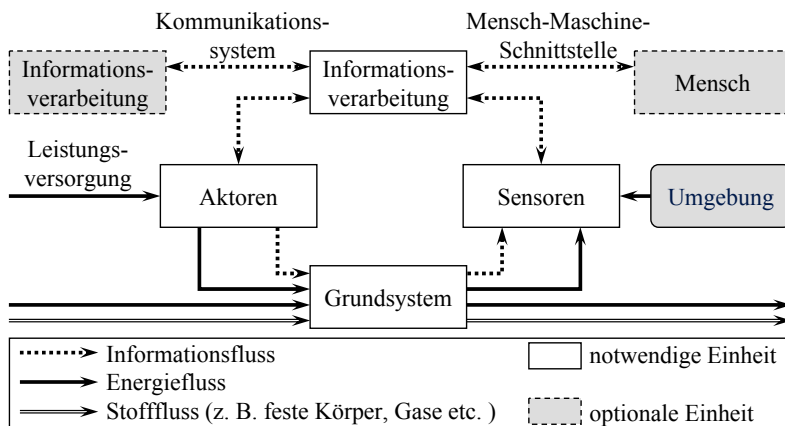


Abbildung 2.1: Aufbau eines mechatronischen Systems gemäß VDI-RICHTLINIE 2206 (2004)



## 2.3 Entwicklungsprozesse im Maschinen- und Anlagenbau

### 2.3.1 Entwicklungs- und Konstruktionsmethoden

Entwicklungs- und Konstruktionsmethoden dienen der systematischen Unterstützung von Entwicklungsprozessen. Durch die *Strukturierung wesentlicher Zusammenhänge und daraus ableitbarer Arbeitsanleitungen* soll diese Zielstellung erreicht werden (VDI-RICHTLINIE 2221 1993). Konstruktionsmethoden werden ebenso als Vorgehensmodelle bezeichnet, die nach BENDER (2005) den Entwicklungsprozess in einzelne Aktivitäten unterteilen und deren logische Abfolge einschließlich der Verantwortlichkeiten festlegen. Nachfolgend werden ausgewählte und für den Maschinenbau relevante Ansätze vorgestellt.

#### Allgemeine Grundlagen nach VDI-RICHTLINIE 2221 (1993)

Allgemeingültige Grundlagen des methodischen Entwickelns und Konstruierens werden in der VDI-RICHTLINIE 2221 (1993) behandelt. Dabei wird neben dem methodischen Vorgehen zum Lösen von Problemen auch die Strukturierung von Systemen in Teilsysteme zur Komplexitätsbeherrschung beschrieben. Ferner unterteilt die Richtlinie den Konstruktionsprozess in sieben generelle Arbeitsschritte und erhebt dadurch den Anspruch, das Entwickeln überschaubar, rationell und branchenunabhängig zu gestalten. Die einzelnen Schritte können in Abhängigkeit der Aufgabenstellung vollständig, nur teilweise oder iterativ durchlaufen werden, siehe Abbildung 2.2. Zunächst müssen die Anforderungen des Kunden oder der Produktplanung geklärt und präzisiert werden. Anschließend erfolgt das Ermitteln von Funktionen und von deren Strukturen, für die im Nachgang Lösungsansätze definiert werden müssen. In einem nächsten Schritt wird die prinzipielle Lösung in realisierbare Module gegliedert, die dann zu konkretisieren bzw. zu gestalten sind. Darauf aufbauend werden die Module mit weiteren Detailangaben ergänzt und zu einem Gesamtsystem verknüpft. Zuletzt ist die Produktdokumentation zu erstellen. Den genannten Arbeitsschritten ist laut Richtlinie gemein, *dass bei ihnen jeweils mehrere Lösungsvarianten untersucht, gegebenenfalls in Mustern bzw. Prototypen erprobt und anschließend beurteilt werden müssen*. Weiterhin sollen die einzelnen Arbeitsschritte nicht starr nacheinander ausgeführt werden. Iterationsschleifen sind möglich und dienen der schrittweisen Optimierung des Produkts. Einen umfassenden Überblick hinsichtlich einzelner Methoden geben Standardwerke des Konstruierens, wie z. B. ROTH (1982), KOLLER (1985) oder EHRENSPIEL ET AL. (1998).

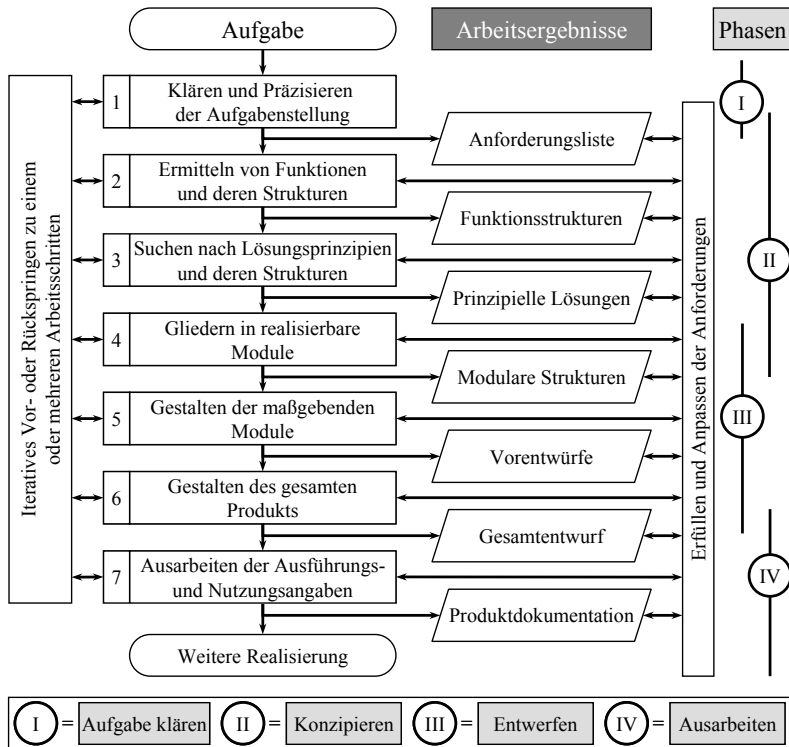


Abbildung 2.2: Generelles Vorgehen beim Entwickeln und Konstruieren nach VDI-RICHTLINIE 2221 (1993)

### Vorgehen nach PAHL & BEITZ (1993)

PAHL & BEITZ (1993) teilen den Konstruktionsprozess in die vier Hauptphasen *Planen*, *Konzipieren*, *Entwerfen* und *Ausarbeiten* ein. Den Ausgangspunkt stellt das *Planen* dar, wobei die Aufgabe definiert und der Entwicklungsauftrag festgelegt wird. Dem schließt sich das *Konzipieren* an. Dieses beinhaltet sowohl die Klärung der Aufgabenstellung als auch die Suche und die Kombination von Lösungsvarianten, die bewertet und ausgewählt werden müssen. Das *Entwerfen* umfasst die Erstellung der Baustruktur, die während des *Ausarbeitens* in eine entsprechende Dokumentation umzusetzen ist. Die beschriebenen vier Phasen können direkt den in der VDI-RICHTLINIE 2221 (1993) definierten Arbeitsschritten zugeordnet werden.

### Entwicklungsmethodik gemäß VDI/VDE-RICHTLINIE 2422 (1994)

Auf Grund des verstärkten Einsatzes der Mikroelektronik in der Gerätetechnik wurde die Richtlinie VDI/VDE-RICHTLINIE 2422 (1994) erlassen. Deren Vorgehensmodell ist in Abbildung 2.3 dargestellt. Ausgehend von der Aufgabenstellung wird ein Konzept definiert, welches disziplinspezifisch zu entwerfen und auszuarbeiten ist. Wie Abbildung 2.3 außerdem verdeutlicht, wird ein paralleliertes Prozedere hinsichtlich der Entwicklungsstränge der Software, der elektrischen Schaltung und der Elektromechanik vorgeschlagen. Als Grundlage dient das Vorgehen der VDI-RICHTLINIE 2221 (1993), welches an die Bedürfnisse der Konstruktion mikroelektronisch gesteuerter Geräte angepasst wurde.

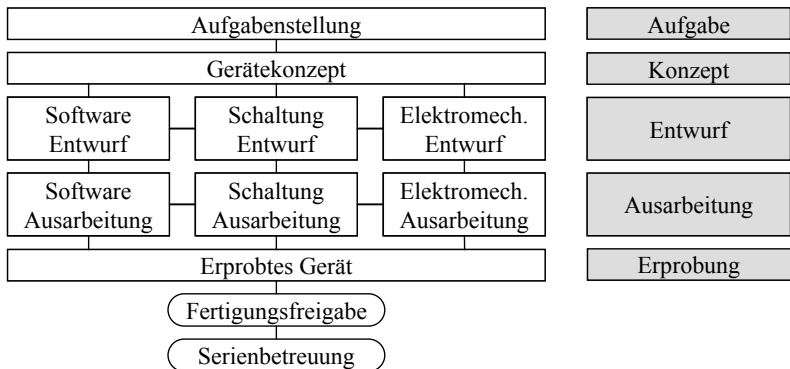


Abbildung 2.3: Vorgehensmodell nach VDI/VDE-RICHTLINIE 2422 (1994)

### Entwicklungsmethodik nach VDI-RICHTLINIE 2206 (2004)

Dem Trend von rein mechanischen hin zu mechatronischen Lösungen wurde mit der VDI-RICHTLINIE 2206 (2004) Rechnung getragen. Diese erhebt den Anspruch, das domänenübergreifende Entwickeln mechatronischer Systeme zu unterstützen. Des Weiteren sollen dadurch die VDI-RICHTLINIE 2221 (1993) und die VDI/VDE-RICHTLINIE 2422 (1994) ergänzt werden. Das Vorgehensmodell beruht im Wesentlichen auf den folgenden drei Elementen:

- Allgemeiner Problemlösungszyklus auf der Mikroebene
- V-Modell auf der Makroebene
- Vordefinierte Prozessbausteine zur Bearbeitung wiederkehrender Arbeitsschritte

Auf das V-Modell, das ursprünglich aus dem Bereich der Softwareentwicklung stammt (PONN & LINDEMANN 2008), soll an dieser Stelle genauer eingegangen werden. Es beschreibt den sogenannten Makrozyklus, der in Abbildung 2.4 dargestellt ist. Die Grundlage bildet zunächst der Systementwurf, welcher dazu dient, unter Berücksichtigung der Anforderungen ein domänenübergreifendes Lösungskonzept zu erstellen. Letzteres enthält die grundlegenden physikalischen und logischen Wirkprinzipien. Hierbei schlägt die Richtlinie eine Gliederung der Gesamtfunktion in Teilfunktionen vor, welchen dann geeignete Lösungen zugeordnet werden können. Das im Systementwurf skizzierte Lösungskonzept muss in den disziplinspezifischen Entwürfen weiter konkretisiert werden. Die Ergebnisse dieser Phasen werden zum Gesamtsystem integriert und im Zuge einer fortlaufenden Eigenschaftsabsicherung dem Lösungskonzept gegenübergestellt. Der Makrozyklus kann auch mehrmalig, über Zwischenstufen, durchlaufen werden, bis das fertige Produkt vorliegt.

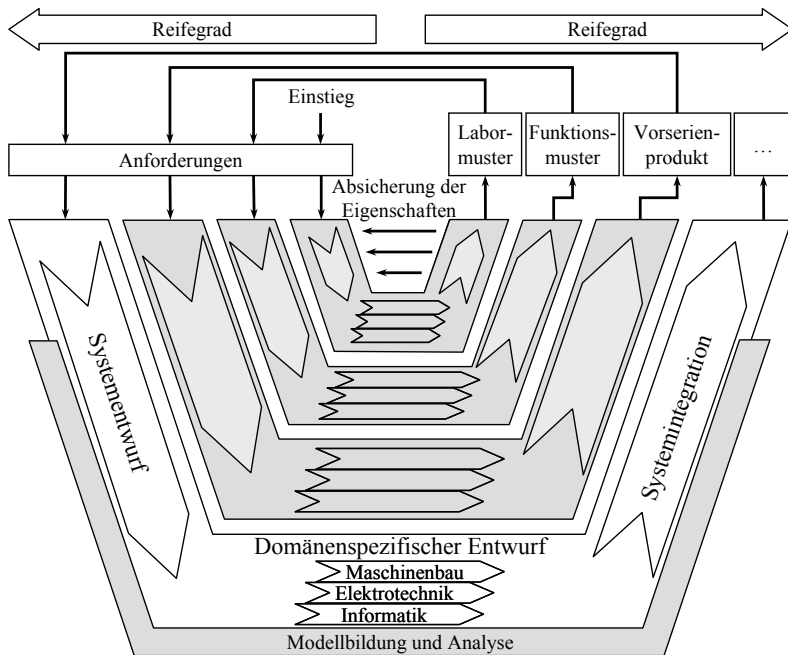


Abbildung 2.4: Vorgehen beim Entwickeln mechatronischer Systeme nach VDI-RICHTLINIE 2206 (2004)

### Methoden für die Softwareentwicklung

Die bisher vorgestellten Methoden entstammen der klassischen Konstruktionslehre und unterstützen die Entwicklung der Steuerungssoftware nur teilweise. Im Bereich der Informatik haben sich eigene Vorgehensmodelle etabliert. Abbildung 2.5 zeigt beispielsweise das Wasserfall- und das Spiralmodell (vereinfachte Darstellung). Von großer Bedeutung ist zudem das V-Modell (siehe oben), das auch in die VDI-RICHTLINIE 2206 (2004) Eingang gefunden hat.

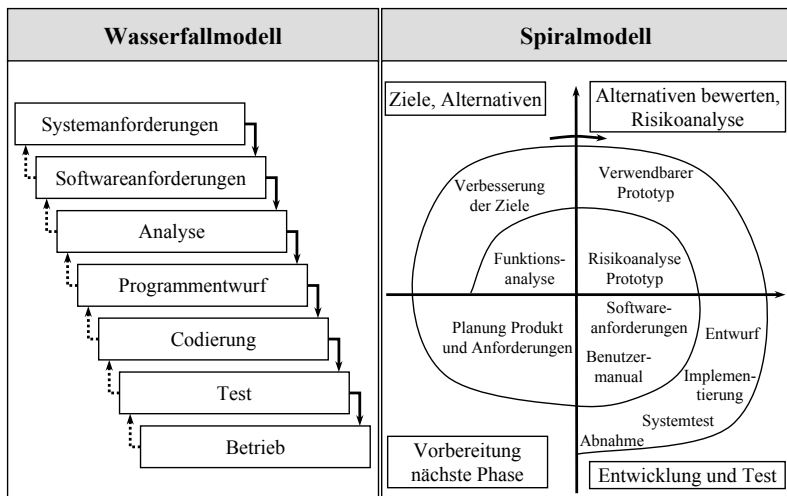


Abbildung 2.5: *Wasserfallmodell nach ROYCE (1970) und Spiralmodell in Anlehnung an BOEHM (1988) bzw. vereinfachte Darstellung von LEVI & REMBOLD (2003)*

### 2.3.2 Phasen des Entwicklungsprozesses

Die einzelnen Arbeitsschritte der unter Abschnitt 2.3.1 erläuterten Konstruktionsmethoden werden in der industriellen Praxis oft zu Entwicklungs- bzw. Konstruktionsphasen zusammengefasst, an denen sich terminlich und organisatorisch Ablaufpläne orientieren können (VDI-RICHTLINIE 2221 1993). Nach BENDER ET AL. (1999) besteht der *typische auftragsorientierte Entwicklungsprozess von Produktionsmaschinen* im Wesentlichen aus vier Phasen, die streng nacheinander ablaufen: die Mechanikkonstruktion, die Elektrokonstruktion, die Fertigung und Montage sowie die Inbetriebnahme.

Im Rahmen einer Studie unter den Mitgliedsfirmen des Vereins Deutscher Werkzeugmaschinenfabriken (VDW) konnte die sequenzielle Vorgehensweise ebenfalls bestätigt werden (GRÄTZ 2005), wie Abbildung 2.6 illustriert. Weiterhin ist charakteristisch, dass vor der Konstruktion die Maschine mit Hilfe einer sogenannten *Funktionsbeschreibung* spezifiziert wird. Als Hilfsmittel hierfür kommen in der Regel Skizzen sowie text- und tabellenbasierte Darstellungsformen zum Einsatz (MAURMAIER 2008, SCHEIFELE 2008, AUTOVIBN 2010).

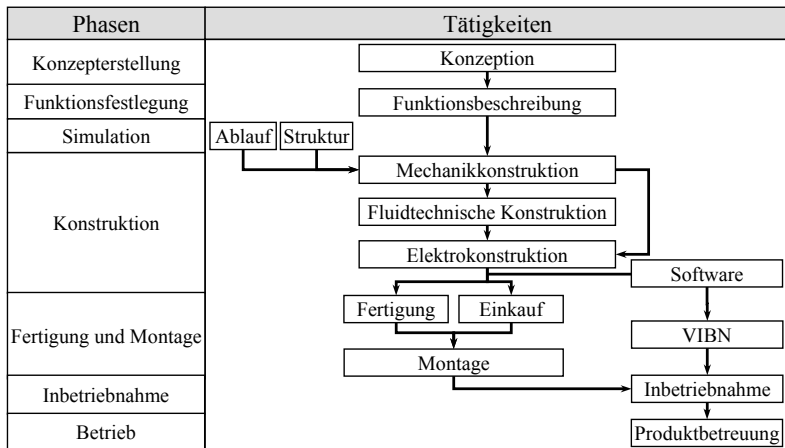


Abbildung 2.6: Vereinfachte Darstellung des Entwicklungsablaufes in der Werkzeugmaschinenindustrie in Anlehnung an GRÄTZ (2005)

### 2.3.3 Inbetriebnahme von Maschinen und Anlagen

Da der Schwerpunkt der vorliegenden Arbeit auf einer simulativen Unterstützung des Steuerungstests liegt, wird die Inbetriebnahmephase detailliert betrachtet. Dies umfasst zunächst die Klärung von Begrifflichkeiten und Definitionen, bevor auf die Bedeutung, die Zielstellungen und die Aufgaben eingegangen wird.

#### Begriff und Definition

Der Terminus der Inbetriebnahme ist nach WEBER (2006) sowohl in der Fachliteratur als auch in der Praxis unterschiedlich belegt, wobei dieser darunter die Überführung einer Anlage aus dem Ruhezustand in den Dauerbetriebszustand versteht.

EVERSHEIM (1990) sieht die Inbetriebnahme als termingerechte Versetzung der montierten Maschine in Funktionsbereitschaft mit anschließender Funktionsüberprüfung. Dabei entfallen lediglich ca. zehn Prozent des Aufwandes auf die Mechanik. Der Großteil der Tätigkeiten konzentriert sich auf die Elektrik und die Steuerung. Eine Abgrenzung zwischen Endmontage und Inbetriebnahme gestaltet sich in der betrieblichen Praxis schwierig, da zum Teil später unzugängliche Baugruppen bereits während der Montage in Betrieb genommen werden.

ZEUGTRÄGER (1998) definiert die Inbetriebnahme als *funktionsgerechtes Einschalten der Anlage in Verbindung mit dem Prozess und Hochfahren der Leistung auf das geforderte Niveau*. Weiterhin müssen die Funktionsbereitschaft und das Zusammenwirken der montierten Einzelkomponenten sichergestellt werden. Einzel- und Gesamtfunktionen sind hierfür zu prüfen.

### **Bedeutung der Inbetriebnahme**

Die Inbetriebnahme macht einen erheblichen Anteil an der Gesamtdurchlaufzeit aus. Studien zufolge wird von 15 bis 25 Prozent ausgegangen (VDW-BERICHT 1997). Fehler, die dort nicht erkannt werden, wirken sich unmittelbar im Betrieb beim Kunden aus. Erschwerend kommt hinzu, dass in dieser Phase erstmals das Zusammenwirken der domänenspezifischen Entwicklungsergebnisse getestet werden kann. Die nachfolgend aufgelisteten Aspekte beschreiben Herausforderungen und Probleme während des Inbetriebnahmeprozesses:

- Ausgleich zeitlicher Verzögerungen aus vorherigen Arbeitsschritten
- Hoher Zeitdruck durch feste Lieferfristen
- Großer Platzbedarf und Flächenengpässe
- Hohe Kapitalbindung durch die fast fertiggestellte Maschine
- Organisatorische Probleme auf Grund der räumlichen Distanz zwischen Baustelle und Herstellerstandort (vor allem im Ausland)

### **Aufgaben und Zielstellungen**

Die Aufgaben und Ziele einer Inbetriebnahme können vielschichtiger Natur sein, siehe Abbildung 2.7. Neben Gesichtspunkten, die sich konkret auf die Anlage oder die Maschine beziehen, wie etwa der Nachweis der Betriebssicherheit oder der Funktionsfähigkeit, werden ebenso organisatorische Ziele verfolgt. Hierzu zählen z. B. die Schulung des Personals oder der Wissensgewinn (WEBER 2006).

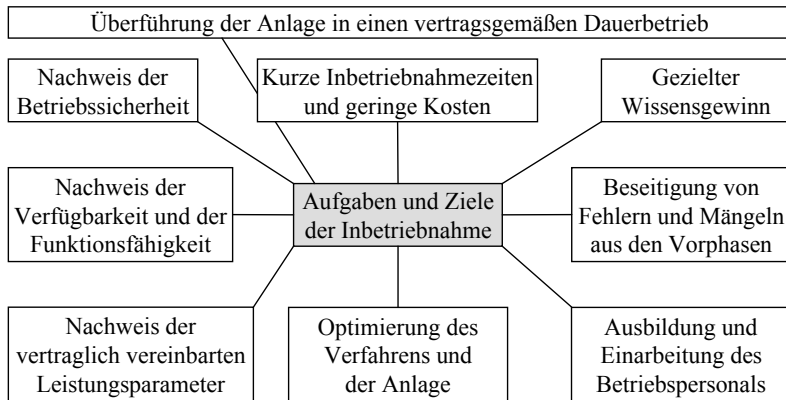


Abbildung 2.7: Aufgaben und Ziele der Inbetriebnahme nach WEBER (2006)

## 2.4 Grundlagen der Steuerungstechnik

### 2.4.1 Klassifizierung von Steuerungen

Das Steuern ist ein Vorgang, bei welchem die variablen Eingangsgrößen *auf Grund der dem System eigenen Gesetzmäßigkeiten* die variablen Ausgangsgrößen beeinflussen. Der grundlegende Unterschied zwischen einer Steuerung und einer Regelung besteht in der Rückführung der zu regelnden Größe und dem Vergleich mit dem Führungswert bei Letzterer, wodurch sich ein geschlossener Wirkungsablauf ergibt. Bei einer Steuerung ist dieser Abgleich nicht vorhanden (DIN IEC 60050 TEIL 351 2006).

Im Produktionsbetrieb lassen sich verschiedene Hierarchieebenen unterscheiden, die über eigene Rechner- bzw. Steuerungssysteme verfügen. Die in Abbildung 2.8 dargestellte Klassifizierung wird als Automatisierungspyramide bezeichnet (VOGEL-HEUSER & WANNAGAT 2009) und orientiert sich an den Ebenen, die in der DIN EN 62264 TEIL1 (2008) definiert sind. Ausgehend von der Unternehmensleitebene, die vorwiegend einen dispositiven Charakter aufweist, hin zur Prozessebene gestalten sich die Funktionen immer operativer (GÜNTHER & TEN HOMPEL 2010).



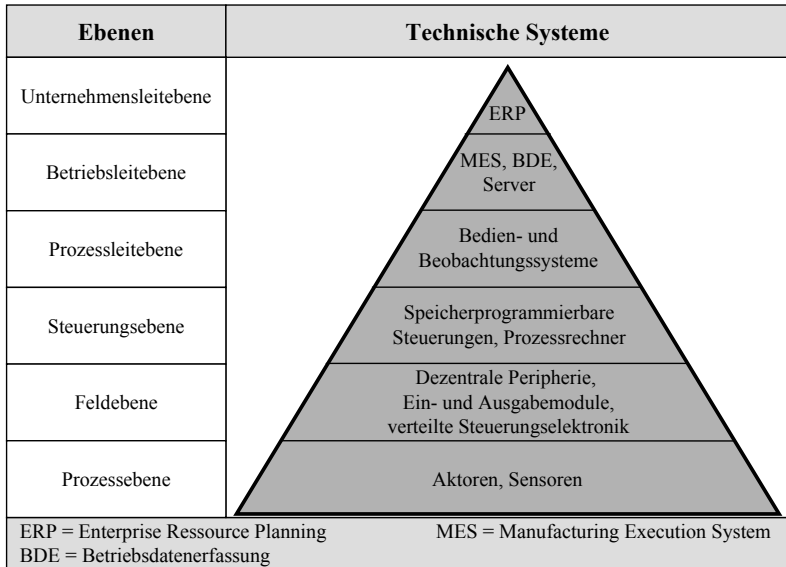


Abbildung 2.8: Ebenen in der Produktionstechnik – Automatisierungspyramide in Anlehnung an VOGEL-HEUSER & WANNAGAT (2009)

Die Strukturierung in autonome Teilsysteme ermöglicht eine Aufgabenverteilung, die zur Komplexitätsreduzierung beiträgt. Die funktionalen Steuerungsebenen müssen nach PRITSCHOW (2006) nicht unbedingt mit der gerätetechnischen Ebenenstruktur übereinstimmen. Mehrere Funktionsebenen können z. B. in einer Vorrichtung zusammengefasst sein.

### Klassifikation nach dem Steuerungsprinzip und der Realisierungsform

WECK & BRECHER (2006) kategorisieren Steuerungen nach dem Steuerungsprinzip und der Realisierungsform, siehe Abbildung 2.9. Die verwendeten Begrifflichkeiten sind auch größtenteils in der DIN IEC 60050 TEIL 351 (2006) zu finden. Eine Ablaufsteuerung verfügt stets über einen schrittweisen Ablauf, wobei die Übergänge von vorgegebenen Bedingungen abhängen. Es kann zwischen prozess- und zeitabhängigen Ausprägungen differenziert werden (DIN IEC 60050 TEIL 351 2006). Verknüpfungssteuerungen setzen Ein- und Ausgangssignale über boolesche Beziehungen zueinander in Relation. Weisen diese ein Speicherverhalten auf, so wird der Begriff des *Schaltwerks* verwendet. Der innere

Zustand des Systems hat somit Einfluss auf die Signalausgabe. Steuerungen ohne Speicherverhalten werden als *Schaltnetze* bezeichnet (WECK & BRECHER 2006).

Bei verbindungsprogrammierten Steuerungen ist das *Programm durch die Art der benutzten Funktionseinheiten und (durch) die gegenseitigen Verbindungen zwischen diesen Einheiten bestimmt* (DIN IEC 60050 TEIL 351 2006). WECK & BRECHER (2006) gliedern diese weiter in fest- und unprogrammierbare Steuerungen auf. Die DIN IEC 60050 TEIL 351 (2006) definiert speicherprogrammierbare Steuerungen (SPS) als *rechnergestützte Leiteinrichtung oder rechnergestütztes System, deren oder dessen logischer Ablauf über eine direkt oder über Fernsteuerung angeschlossene Programmiereinrichtung, z. B. ein Bedienfeld, einen Hilfsrechner oder ein tragbares Terminal, veränderbar ist*. Hiervon existieren weitere Varianten, die in Abbildung 2.9 dargestellt sind.

Darüber hinaus unterscheidet die DIN IEC 60050 TEIL 351 (2006) zwischen synchronen und asynchronen Steuerungen. Bei Ersterer erfolgt die *Bildung der Ausgangsgrößen aus den Eingangsgrößen synchron zu einem Taktsignal*. Die asynchrone Steuerung hingegen arbeitet ohne Taktsignal und löst *Änderungen der Ausgangsgröße nur durch Änderungen der Eingangsgrößen* aus.

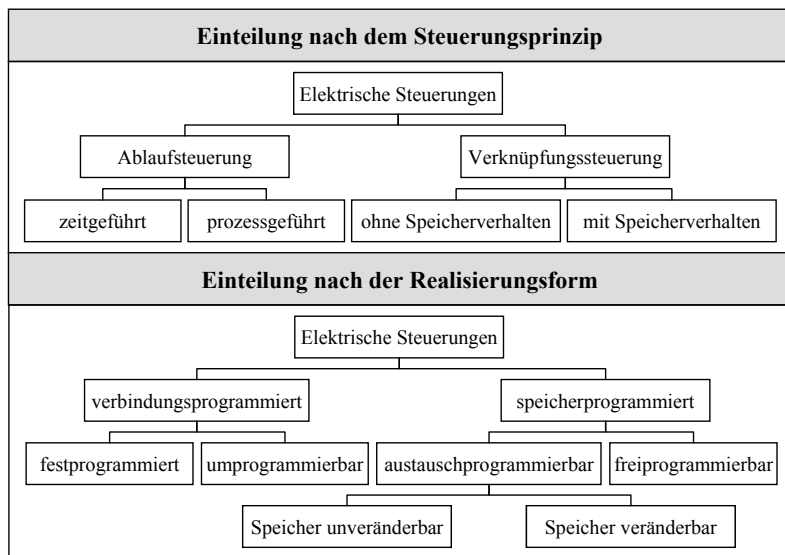


Abbildung 2.9: Einteilung von elektrischen Steuerungen in Anlehnung an WECK & BRECHER (2006)

### 2.4.2 Speicherprogrammierbare Steuerungen

Speicherprogrammierbare Steuerungen (SPS) finden heute in vielfältiger Weise Verwendung. Die Einsatzgebiete erstrecken sich von stückgut- und schüttgutverarbeitenden Maschinen über die Verfahrens- und Prozesstechnik bis hin zu sicherheitskritischen Bereichen, wie beispielsweise in Kraftwerken. Stückgutverarbeitende Produktionssysteme stehen im Rahmen der hier beschriebenen Forschungstätigkeiten im Fokus.

#### Aufbau

Eine SPS verfügt über einen modularen Aufbau. Entsprechend der umzusetzenden Aufgabe kann deren Konfiguration angepasst werden. Folgende Grundelemente sind dabei nötig (LERCH 2007):

- Stromversorgung
- Prozessor (CPU)
- Speichermodule
- Ein- und Ausgangsbaugruppen

Die zentrale Baugruppe ist die Central Processing Unit (CPU), welche die eigentliche Steuerungsfunktionalität realisiert. Über das darin enthaltene Steuerwerk wird die Programmbearbeitung zeilenweise durchgeführt. Neben der klassischen SPS existieren noch weitere, PC-basierte Varianten. Hierbei kann zwischen der Slot-SPS und der Soft-SPS abgegrenzt werden. Die Slot-SPS besteht aus einem PC und einer SPS-Karte, die unabhängig vom Betriebssystem des Rechners ist. Im Gegensatz hierzu läuft bei einer Soft-SPS die Software direkt auf dem PC. Die beiden Varianten unterscheiden sich in Bezug auf ihr Echtzeitverhalten (WELLENREUTHER & Zastrow 2005).

#### Arbeitsweise

Hinsichtlich der Programmbearbeitung einer SPS nennt PRITSCHOW (2006) insgesamt drei mögliche Betriebsarten, die nachfolgend genauer erläutert werden:

- Zyklisch
- Zeitgesteuert
- Interrupt-Bearbeitung (Unterbrechungsbetrieb)

Bei der zyklischen Programmbearbeitung werden zu Beginn einer Schleife die Steuerungseingänge abgefragt und als Prozessabbild gespeichert. Danach erfolgt das schrittweise Durchlaufen der Befehle des Steuerungsprogramms. Abschließend werden die Steuerungsausgänge aktualisiert. Da in Abhängigkeit der Eingangssignale und der Sprungbedingungen nur gewisse Programmteile durchlaufen werden müssen, ergeben sich in der Folge variable Zykluszeiten. Ein weiteres Charakteristikum dieser Betriebsart ist, dass Programme bzw. einzelne Teile davon immer mit der Gesamtzykluszeit abgearbeitet werden.

Im Gegensatz dazu erlaubt die zeitgesteuerte Programmbearbeitung die Definition einer festen Zeitspanne, in welcher bestimmte Programme ausgeführt werden sollen. Hierdurch ist es möglich, beispielsweise Sensorwerte in genau determinierten Zeitabständen zu ermitteln.

Die interruptgesteuerte Programmbearbeitung kann auf Ereignisse reagieren und somit die zyklischen oder die zeitgesteuerten Verfahren unterbrechen. Auf diese Art und Weise lässt sich die Reaktionszeit des Systems verkürzen. Nach der Unterbrechung wird die normale Programmbearbeitung wieder aufgenommen.

### **Programmiersprachen**

Im Rahmen der DIN EN 61131 TEIL 3 (2003) wird ein einheitlicher Standard für SPS-Programmiersprachen beschrieben, der sowohl die Syntax als auch die Semantik festlegt. Die Norm definiert insgesamt fünf Programmiersprachen:

- Anweisungsliste (AWL)
- Funktionsbausteinsprache (FBS)
- Strukturierter Text (ST)
- Ablaufsprache (AS)
- Kontaktplan (KOP)

Eine Einteilung kann nach textbasierten (AWL, ST) und grafischen (FBS, KOP) Sprachen vorgenommen werden. Der Ablaufsprache fällt nach WELLENREUTHER & ZASTROW (2005) eine übergeordnete Rolle zu, da sie grafische sowie textuelle Elemente enthält und zur internen Programmorganisation dient.

### **2.4.3 Dezentralisierung und Feldbussysteme**

Im Maschinen- und Anlagenbau hat sich in den letzten Jahren ein Trend zur Dezentralisierung von Steuerungszintelligenz abgezeichnet. Dies umfasst zum einen die Verlagerung von Steuerungsaufgaben in dezentrale Steuerungen (KLOCKE &

PRITSCHOW 2004), zum anderen einen Transfer in intelligente Aktoren und Sensoren (NYHUIS 2008). Die Zielstellung hierbei ist, mit Hilfe der Dezentralisierung und der Verteilung von Steuerungszintelligenz eine Komplexitätsreduzierung zu erreichen. Für die Kommunikation werden in der Regel Feldbussysteme genutzt, die eine Kopplung unterschiedlicher Steuerungen oder eine Anbindung von Feldgeräten ermöglichen. Auf der Feldebene können durch den Entfall der Parallelverdrahtung Kosten eingespart werden (SCHNELL & WIEDEMANN 2006). SCHMITT (1996) führt als Vorteil einer dezentralen Struktur neben dem geringeren Verkabelungsaufwand und den kürzeren Inbetriebnahmezeiten, die aus einer autarken Inbetriebnahme von einzelnen Funktionseinheiten herrühren, gleichermaßen eine bessere Erweiterbarkeit auf.

## 2.5 Systeme, Modelle und Simulation

### 2.5.1 Systembegriff und Klassifikation

Auf Grund einer Vielzahl von Definitionen in Bezug auf den Begriff des Systems, die maßgeblich von der Sichtweise auf den Untersuchungsgegenstand abhängen, soll an dieser Stelle eine Fassung für Produktionssysteme entsprechend der VDI-RICHTLINIE 3633 (1996) referenziert werden. Hiernach beschreibt ein System eine abgegrenzte Anordnung von Komponenten, die miteinander in Beziehung stehen. Es ist durch eine Systemgrenze gekennzeichnet, wobei über Schnittstellen Ein- und Ausgangsgrößen ausgetauscht werden können. Ein System lässt sich wiederum in Subsysteme und in nicht weiter zerlegbare Systemelemente aufgliedern. Ferner verfügt es über eine Aufbaustruktur, eine Ablaufstruktur sowie über eine Ablauflogik. Jedes System ist durch Zustandsgrößen und Zustandsübergänge gekennzeichnet.

#### Signale

Systemein- und -ausgänge werden in der Systemtheorie als Signale bezeichnet. Sie stellen *eine abstrakte Beschreibung einer veränderlichen Größe* dar (FREY & BOSSERT 2008). Die Zeitvariable kann dabei eine kontinuierliche oder eine diskrete Ausprägung aufweisen. Desgleichen kann der Wertebereich des Signals kontinuierlich oder diskret definiert sein, siehe Abbildung 2.10. Ereignisdiskrete Signale verfügen über keinen festen Zeitbezug, sondern sind vom Eintreten bestimmter Ereignisse abhängig.

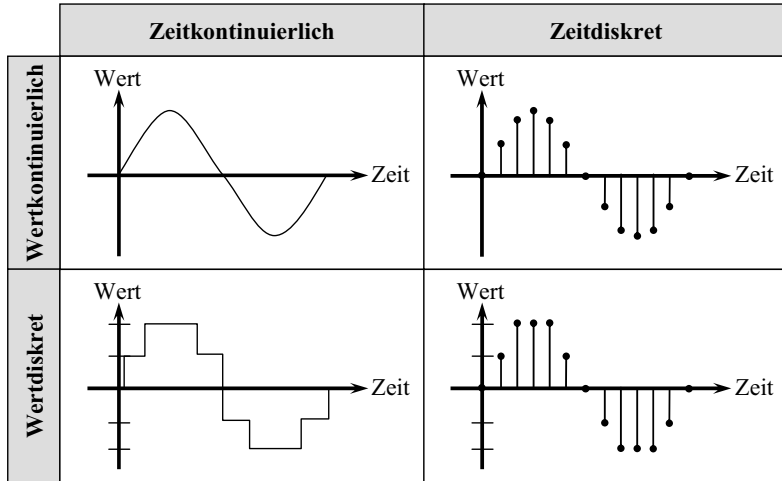


Abbildung 2.10: Kontinuierliche und diskrete Signale in Anlehnung an FREY & BOSSERT (2008)

### Klassifikation von Systemen

FREY & BOSSERT (2008) klassifizieren primär in kontinuierliche und diskrete Systeme. Unter einem (zeit-)kontinuierlichen System wird dabei eine Abbildung  $S$  verstanden, die einem zeitkontinuierlichen Eingangssignal  $x(t)$  ein zeitkontinuierliches Ausgangssignal  $y(t)$  zuordnet, wobei  $x(t), y(t) \in \mathbb{R}(\mathbb{C})$  und  $t \in \mathbb{R}$  gilt:

$$y(t) = S\{x(t)\} \quad (2.1)$$

Das (zeit-)diskrete System ist eine Abbildung  $S$ , die einem zeitdiskreten Eingangssignal  $x(k)$  ein zeitdiskretes Ausgangssignal  $y(k)$  zuweist, mit  $x(k), y(k) \in \mathbb{R}(\mathbb{C})$  und  $k \in \mathbb{Z}$ :

$$y(k) = S\{x(k)\} \quad (2.2)$$

Systeme sind durch weitere Eigenschaften, wie beispielsweise die Linearität oder die Zeitinvarianz, gekennzeichnet. Tabelle 2.1 gibt einen Überblick bezüglich einer möglichen Einordnung.

<b>Definitionsbereich</b>	<ul style="list-style-type: none"> <li>• (zeit-)kontinuierlich</li> </ul>	<ul style="list-style-type: none"> <li>• (zeit-)diskret</li> </ul>
<b>Wertebereich</b>	<ul style="list-style-type: none"> <li>• wertkontinuierlich</li> <li>• beschränkt</li> </ul>	<ul style="list-style-type: none"> <li>• wertdiskret, amplitudendiskret</li> <li>• nichtbeschränkt</li> </ul>
<b>Weitere Eigenschaften</b>	<ul style="list-style-type: none"> <li>• linear</li> <li>• zeitinvariant</li> <li>• kausal</li> <li>• stabil</li> <li>• dynamisch</li> <li>• deterministisch</li> </ul>	<ul style="list-style-type: none"> <li>• nichtlinear</li> <li>• zeitvariant</li> <li>• nichtkausal</li> <li>• nichtstabil</li> <li>• statisch</li> <li>• stochastisch</li> </ul>

Tabelle 2.1: Klassifizierung von Systemen nach FREY & BOSSERT (2008)

In den letzten Jahren hat die Modellvorstellung des hybriden Systems stark an Bedeutung gewonnen. Dieses ist nach LUNZE (2002) durch kontinuierliche Wirkungsweisen und sprunghafte Zustandsänderungen gekennzeichnet. Rein diskrete oder kontinuierliche Beschreibungsformen sind für dessen Darstellung nicht geeignet.

### 2.5.2 Modellbegriff

Der Begriff des Modells stammt etymologisch vom lateinischen *modulus* ab, was mit Maß oder Maßstab übersetzt werden kann (KASTENS & KLEINE 2008). Eine enge Verbindung besteht zum Systembegriff. Ein Modell ist laut BANKS (1998) die Abbildung eines Systems.

Gemäß der allgemeinen Modelltheorie nach STACHOWIAK (1973) ist ein Modell durch folgende drei Eigenschaften gekennzeichnet:

- Abbildung
- Minimierung
- Pragmatismus

Das Modell stellt ein Abbild dar, *eine Repräsentation natürlicher oder künstlicher, originärer Realitäten, die selbst wieder Modelle sein können*. Unter Minimierung wird verstanden, dass ein Modell nicht alle Attribute des Originals enthalten muss, *sondern nur diejenigen, die dem Modellierer bzw. dem Modellanwender relevant erscheinen*. Die für den Modellzweck wesentlichen Eigenschaf-

ten müssen hierbei berücksichtigt werden. Der Pragmatismus ist im Hinblick auf die Modellbildung durch eine *Orientierung am Nützlichen* charakterisiert. Dies bedeutet, dass *ein Modell [...] innerhalb einer bestimmten Zeitspanne und zu einem bestimmten Zweck für ein Original eingesetzt wird.*

BOSSEL (1994) unterscheidet zwischen zwei grundsätzlichen Möglichkeiten, das Verhalten eines Systems zu modellieren. Zum einen kann dieses durch eine Approximation nachgebildet werden. Dabei wird die innere Wirkstruktur des Originalsystems vernachlässigt und als sogenannte *Black Box* betrachtet. Zum anderen können die strukturellen Belange in das Modell mit einfließen. Das Verhalten ergibt sich dann aus der Wirkstruktur. Demnach handelt es sich um ein *Modell des Systems* und nicht um ein *Modell des Verhaltens*. Dieser Ansatz ist auch als *Glass-Box-Methode* bekannt (weiterer Begriff: *White Box*).

Black-Box- und Glass-Box-Modelle werden gleichermaßen als Verhaltens- bzw. Strukturmodelle betitelt. Außerdem sind nach BOSSEL (1994) Mischformen möglich, die häufig in der industriellen Praxis Anwendung finden, falls die Wirkstruktur oder Parameter nicht ermittelt werden können.

### 2.5.3 Modellierungstechniken

#### 2.5.3.1 Einordnung

Die Intention der Modellbildung ist, ein reales System zu abstrahieren und dem Modellzweck entsprechende Eigenschaften abzubilden. Dies erfordert eine adäquate Modellierungstechnik, die den Charakter des Systems berücksichtigt. Diskrete oder kontinuierliche Systeme verlangen nach geeigneten Beschreibungsmitteln. Deshalb spielt die Betrachtungsweise eine wesentliche Rolle. In der Automatisierungstechnik treten sowohl ereignisdiskrete als auch kontinuierliche Aspekte auf. Durch eine Abstraktion kann jedoch im Einzelfall auf eine Denkweise reduziert werden. Am Beispiel einer automatischen Schutztür lässt sich dies verdeutlichen. Sind lediglich die Zustände *Tür geöffnet* und *Tür geschlossen* von Interesse, so kann eine diskrete Modellierungstechnik zum Einsatz kommen. Die Vernachlässigung des kontinuierlichen Schließvorgangs ist jedoch nicht sinnvoll, wenn die Verfahrbewegung an sich mit Beschleunigungs- und Geschwindigkeitswerten Beachtung finden soll.

Die VDI/VDE-RICHTLINIE 3681 (2005) differenziert zwischen formalen, semi-formalen und informalen Beschreibungsmitteln. Formale und semiformale Mo-



dellierungstechniken verfügen über *eine definierte, vollständige Syntax sowie eine eindeutige semantische Interpretation*. Zudem besitzen formale Verfahren eine mathematische Basis. Nachfolgend wird eine Auswahl an formalen und semiformalen Techniken vorgestellt, die zum besseren Verständnis für den Stand der Forschung und der Technik in Kapitel 3 sowie für die Funktionsmodellierung in Kapitel 5 von Relevanz sind.

### 2.5.3.2 Ereignisdiskrete Modellierungstechniken

Im Rahmen der vorliegenden Arbeit sind insbesondere ereignisdiskrete Modellierungsmittel von Interesse, da sich diese für die Abbildung steuerungstechnischer Vorgänge eignen. ZEIGLER ET AL. (2000) definieren ein ereignisdiskretes System durch eine Menge von Zuständen, die zu beliebigen Zeitpunkten durch das Eintreten von Ereignissen gewechselt werden können.

#### Endliche Automaten

Der endliche Automat (EA), auch Zustandsmaschine genannt, ist eine gängige Technik zur Modellierung von ereignisdiskreten Systemen. Die Beschreibung erfolgt mit Hilfe von Zuständen und Transitionen (Zustandsübergänge). Durch die Zustände ist das System eindeutig charakterisiert, wobei immer nur ein Zustand gleichzeitig eingenommen werden darf. Da unterschiedliche Ausprägungen des endlichen Automaten existieren, soll hier eine Definition nach HOPCROFT ET AL. (2002) näher erläutert werden, wobei es sich um einen sogenannten Akzeptor handelt. Danach besteht ein endlicher Automat aus folgenden Elementen:

- Endliche Menge von Zuständen  $S$
- Startzustand  $s_0$
- Endliche Menge von Eingabesymbolen  $\Sigma$
- Menge finaler oder akzeptierender Zustände  $Q$
- Übergangsfunktion  $\delta$

Der oben erläuterte endliche Automat kann als 5-Tupel notiert werden:

$$EA = (S, \Sigma, \delta, s_0, Q) \quad (2.3)$$

Die sogenannten Transduktoren verfügen neben den Eingaben außerdem über Ausgaben. Die Tupel-Notation wird um eine endliche Menge von Ausgabesym-

bofen  $\Gamma$  und eine Ausgabefunktion  $\omega$  ergänzt. Hierdurch kann der Transduktor als 6-Tupel angegeben werden (HOFFMANN 2009):

$$EA = (S, \Sigma, \Gamma, \delta, s_0, \omega) \quad (2.4)$$

Transduktoren lassen sich ferner in Moore- und Mealy-Automaten einteilen. Die Ausgaben des Moore-Automaten hängen nur vom Zustand ab:

$$\Gamma = \omega(S) \quad (2.5)$$

Dagegen spielen beim Mealy-Automaten auch die Eingaben eine Rolle:

$$\Gamma = \omega(S, \Sigma) \quad (2.6)$$

Im Hinblick auf die Eindeutigkeit der Zustandsübergänge kann zwischen dem deterministischen und dem nichtdeterministischen Automaten unterschieden werden. Bei Letzterem können auf Grund einer konkreten Eingabe potenziell mehrere Folgezustände eingenommen werden. Da weiterhin eine Vielzahl an Varianten von endlichen Automaten existiert, sei an dieser Stelle auf entsprechende Fachliteratur verwiesen.

Als Darstellungsform von Zustandsmaschinen eignet sich der Zustandsgraph. Die Zustände werden als Knoten und die Transitionen als gerichtete Kanten visualisiert. Im Vergleich zu einer ebenso möglichen tabellarischen Beschreibung gestaltet sich der Zustandsgraph vor allem bei größeren Systemen übersichtlicher (HEDTSTÜCK 2007). In Abbildung 2.11 ist ein einfaches Beispiel eines endlichen Automaten jeweils in Mealy- und Moore-Notation aufgeführt.

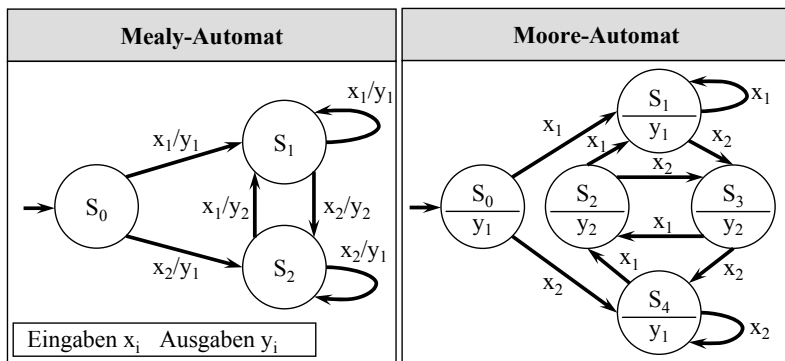


Abbildung 2.11: Beispiel eines endlichen Automaten nach HOFFMANN (2009)

## Petri-Netze

Das Petri-Netz (PN) ist eine Erweiterung des endlichen Automaten, welches diesen um den Aspekt der Nebenläufigkeit ergänzt (PETRI 1962). Im Bereich der Modellierung von Steuerungssystemen haben Petri-Netze eine breite Anwendung erfahren (VOGEL-HEUSER 2003). Grundsätzlich besteht ein derartiges Netz aus Stellen, Transitionen und einer Flussrelation (PRIESE & WIMMEL 2008):

- Endliche Menge an Stellen  $S$
- Endliche Menge an Transitionen  $T$
- Kantenfunktion  $F \subseteq (S \times T) \cup (T \times S)$

Somit kann ein Petri-Netz in allgemeiner Form als 3-Tupel definiert werden:

$$PN = (S, T, F) \quad (2.7)$$

Abbildung 2.12 zeigt ein einfaches Petri-Netz in Gestalt eines Stellen-Transitions-Netzes (S/T-Netz). Dieses ist als 6-Tupel definiert:

$$PN = (S, T, F, K, W, M_0) \quad (2.8)$$

Dabei ist  $K$  die Kapazität einer Stelle,  $W$  die Gewichtungsfunktion der Kanten und  $M_0$  die Anfangsmarkierung. Die zwei unterschiedlichen Arten von Knoten, die Stellen  $S$  und die Transitionen  $T$ , werden über gerichtete und gewichtete Kanten verbunden. Der Zustand des Systems ist über die Belegung der Stellen mit Marken definiert. Durch das Schalten einer Transition (auch Feuern genannt) ändert sich die Belegung der Stellen gemäß der Gewichtung. In Abbildung 2.12 ist verdeutlicht, wie sich beim Feuern der Transition  $t$  die Marken verschieben.

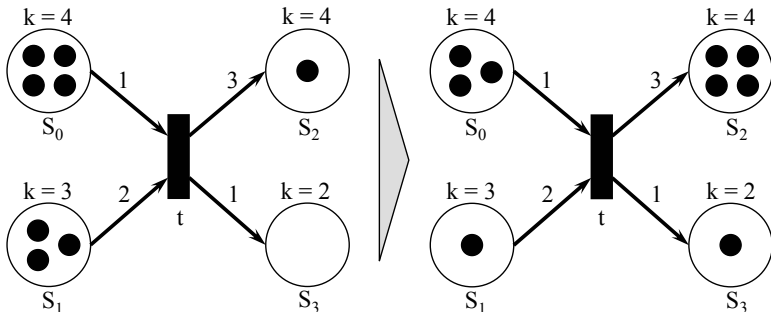


Abbildung 2.12: Beispiel für ein Petri-Netz in Form eines S/T-Netzes

Auf weitere Ausprägungen, wie beispielsweise Bedingungs-Ereignis-Netze (B/E-Netz) oder Prädikat-Transitions-Netze (P/T-Netz), wird hier nicht näher eingegangen. Für die Modellierung von Automatisierungssystemen sind noch die steuerungstechnisch interpretierten Petri-Netze (SIPN) von Interesse, die über Beziehungen zu Aktor- und Sensorsignalen verfügen (LUNZE 2006).

### Statecharts

Um komplexe, reaktive Systeme besser zu beherrschen, wurden von HAREL (1987) die Statecharts entwickelt. Diese erweitern Zustandsdiagramme um ein hierarchisches Konzept, um parallele Zustände sowie um einen Kommunikationsmechanismus. Abbildung 2.13 links zeigt, wie sich Zustände, die über abgerundete Rechtecke dargestellt werden, zu sogenannten Superstates zusammenfassen lassen. Darüber hinaus können mit Hilfe der Statecharts logische UND-Beziehungen abgebildet werden, siehe Abbildung 2.13, was die Modellierung von Nebenläufigkeiten erlaubt. Außerdem sind die Statecharts Teil der Unified Modeling Language (UML), die in Abschnitt 2.5.3.5 noch näher betrachtet wird.

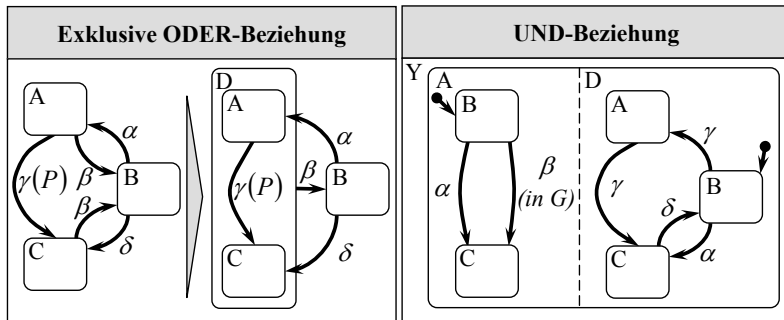


Abbildung 2.13: Beispiele für Statecharts in Anlehnung an HAREL (1987)

### 2.5.3.3 Kontinuierliche Modellierungstechniken

In den klassischen Ingenieurdisziplinen sind kontinuierliche Modelle von großer Bedeutung. Als Beschreibungstechniken kommen mathematische Ausdrucksformen zum Einsatz, wie beispielsweise algebraische Terme oder Differenzialgleichungen. Im Folgenden wird daher auf ausgewählte, kontinuierliche Modellierungstechniken eingegangen.

### Algebraische Gleichungen

Algebraische Gleichungen können zur Beschreibung von Beziehungen verwendet werden, wie z. B. für die Gleichgewichtsbedingungen in der Statik starrer Körper (GROSS ET AL. 2007).

### Differenzialgleichungen

In technischen Systemen treten zahlreiche kontinuierliche Abläufe auf, wie beispielsweise Beschleunigungen oder Ein- und Ausschaltvorgänge. Zu deren Abbildung können Differenzialgleichungen verwendet werden, bei denen die Zeit als Variable eingeht. Die gewöhnliche Differenzialgleichung lässt sich wie folgt darstellen (MEYBERG 2001):

$$F(x, y, y', \dots, y^{(n)}) = 0 \quad (2.9)$$

Dabei ist  $x$  die Variable und  $y$  eine Funktion, deren Ableitung bis zur  $n$ -ten Ordnung auftreten kann. In technischen Anwendungen spielen vor allem die linearen Differentialgleichungen zweiter Ordnung mit konstanten Koeffizienten eine wichtige Rolle, da sich damit mechanische und elektrische Schwingungen beschreiben lassen (MEYBERG 2001):

$$y'' + ay' + by = f(x) \quad (2.10)$$

In der Simulationstechnik werden Differenzialgleichungssysteme zur Lösung häufig in die Zustandsraumdarstellung überführt (RODDECK 2006). Hierbei werden Differenzialgleichungen höherer Ordnung in ein System erster Ordnung umgewandelt. Hängt eine Funktion von mehreren Variablen ab, beispielsweise von der Zeit und vom Ort, so können dafür partielle Differenzialgleichungen verwendet werden.

### Blockdiagramme

Eine grafische Modellierungstechnik zur Umsetzung von algebraischen Gleichungen oder Differenzialgleichungen sind die Blockdiagramme. Das System bzw. die Gleichungen werden mit Hilfe von einzelnen, miteinander verbundenen Blöcken dargestellt.

### Übertragungsfunktionen

In der Regelungstechnik kommt zur Spezifikation des Systemverhaltens linearer, zeitinvarianter Systeme die Übertragungsfunktion  $G(s)$  zum Einsatz. Diese setzt

sich aus dem Quotienten der Laplace-Transformierten von Systemausgängen  $x_A$  und von Systemeingängen  $x_E$  zusammen.  $Z(s)$  ist deren Zählerpolynom,  $N(s)$  deren Nennerpolynom.

$$\frac{x_A}{x_E} = \frac{b_0 + b_1 s + \dots + b_m s^m}{a_0 + a_1 s + \dots + a_n s^n} = G(s) = \frac{Z(s)}{N(s)} \quad (2.11)$$

## Bondgraphen

Eine weitere Möglichkeit zur Darstellung kontinuierlicher Systeme sind die von PAYNTER (1959) entwickelten Bondgraphen. Es handelt sich dabei um gerichtete Graphen, die zur Simulation von Energieflüssen verwendet werden können. Sie eignen sich besonders gut, um unterschiedliche physikalische Domänen miteinander zu verknüpfen. Die Knoten des Graphen repräsentieren Subsysteme, Komponenten oder Elemente, wie z. B. Kondensatoren oder Federn. Die Kanten definieren den Energiefluss (VAJNA ET AL. 2007), wobei dazu die Angabe einer Potenzial- und einer Flussvariable gehört, die miteinander multiplikativ verknüpft sind.

### 2.5.3.4 Hybride Modellierungstechniken

Hybride Modellierungstechniken können zur Beschreibung von Systemen verwendet werden, für die eine rein ereignisdiskrete oder kontinuierliche Darstellung unzureichend ist. Von den bereits oben vorgestellten ereignisdiskreten Verfahren existieren hybride Varianten, die nachfolgend genauer betrachtet werden.

#### Hybride Automaten

Der hybride Automat besteht, wie der endliche Automat, aus Zuständen und Transitionen, jedoch wird der Zustandsraum um kontinuierliche Variablen erweitert. Die zeitlichen Änderungen der Variablen in den einzelnen Zuständen werden mit Hilfe von Differenzialgleichungen ausgedrückt (HENZINGER 1996). In diesem Zusammenhang sind auch die Timed Automata zu nennen, die über eine Zeitkomponente verfügen (DILL & ALUR 1994). Abbildung 2.14 zeigt den hybriden Automaten eines Thermostaten, wobei  $x$  die Variable für die Temperatur ist.

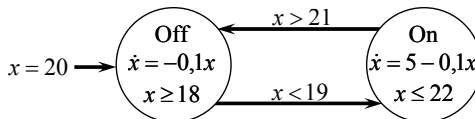


Abbildung 2.14: Hybrider Automat in Anlehnung an HENZINGER (1996)

### Hybride Petrinetze

Im Gegensatz zu diskreten Petri-Netzen verfügen die kontinuierlichen Varianten über Stellen, die anstatt der ganzzahligen Marken mit einer reellen positiven Zahl versehen sind. Das Feuern einer Transition erfolgt als kontinuierlicher Fluss (ALLA & DAVID 1998). Neben dem hybriden Petri-Netz (HPN) existieren weitere Formen, wie das Hybrid Dynamic Net (DRATH 1998) oder das Hybrid Functional Petri Net (MATSUNO ET AL. 2003)

### Hybride Statecharts

Hybride Statecharts erweitern den gewöhnlichen Statechart um Differenzialgleichungen und ermöglichen somit die Abbildung kontinuierlichen Verhaltens (KESTEN & PNUELI 1992).

#### 2.5.3.5 Umfassendere Modellierungstechniken

Standen bisher einzelne Modellierungstechniken im Fokus, so sollen nun umfangreichere Ansätze betrachtet werden, die sich aus mehreren Elementen bzw. Beschreibungsformen zusammensetzen.

#### Unified Modeling Language (UML)

Eine der wohl bekanntesten Modellierungssprachen ist die UML der Object Management Group (OMG). Deren hauptsächliches Einsatzgebiet ist im Bereich der Softwareentwicklung zu finden, die den Ursprung der UML darstellt (PILONE ET AL. 2006). Zur Spezifikation steht eine Vielzahl unterschiedlicher Diagrammtypen (aktuell 14 in der Version 2.3) zur Verfügung, die sich in Struktur- und Verhaltensdiagramme einteilen lassen. Für die Modellierung mechatronischer bzw. automatisierungstechnischer Systeme eignen sich vor allem folgende Diagrammtypen:

- Komponentendiagramm

Mit Hilfe von Komponenten kann der hierarchische Aufbau eines Systems beschrieben werden. Die Schnittstellen einer Komponente werden durch Ports charakterisiert, die über Konnektoren verbunden sind.

- Zustandsdiagramm

Die Zustandsdiagramme basieren auf den Statecharts von HAREL (1987) und können zur Verhaltensspezifikation herangezogen werden.

- Aktivitätsdiagramm

Die Aktivitätsdiagramme orientieren sich stark an den Petri-Netzen, deren Semantik sie weitestgehend übernehmen (OESTREICH 2006).

- Sequenzdiagramm

Die Abfolge von Nachrichten zwischen ausgewählten Teilnehmern kann durch Sequenzdiagramme modelliert werden. Diese stammen von den Message Sequence Charts (MSC) ab.

- Zeitverlaufsdiagramm

Zeitliche Verläufe können mit diesem Diagrammtyp dargestellt werden. Die Zeit ist auf der Abszisse angetragen, auf der Ordinate werden die Zustände der Objekte festgehalten. Durch Linien lassen sich die *Zusammenhänge zwischen Zustandswechseln im Verlauf der Zeit* kenntlich machen (OESTREICH 2006).

Darüber hinaus existiert eine Vielzahl an Profilen, welche die UML für unterschiedliche Verwendungszwecke anpassen.

### **System Modeling Language (SysML)**

Eine Erweiterung der UML für die Modellierung komplexer Systeme stellt die Systems Modeling Language (SysML) dar, die vom International Council on Systems Engineering (INCOSE) entwickelt wurde. Verglichen mit der sehr umfangreichen UML wurden nicht notwendige Elemente ausgeschlossen. Im Gegenzug umfasst die SysML dafür neue Diagrammtypen, wie beispielsweise das Anforderungsdiagramm zur Unterstützung der Anforderungsdefinition in frühen Entwicklungsphasen. Ferner ist die Abbildung kontinuierlicher Aspekte in die SysML mit eingeflossen (WEILKIENS 2006).

### **Real-Time Object Oriented Modeling (ROOM) und UML-RT (Real-Time)**

Die objektorientierte Modellierungssprache ROOM wurde von SELIC ET AL. (1994) zur Beschreibung reaktiver Systeme vorgestellt. Die hierarchische Strukturierung wird mittels sogenannter Aktoren (engl. *actors*) vorgenommen, die über Ports für die Kommunikation verfügen. Über Protokollklassen werden die zu versendenden Nachrichten definiert. Das Verhalten wird durch sogenannte *ROOMcharts* spezifiziert, die sich an HAREL (1987) anlehnen und in den Aktoren gekapselt sind. Die Konzepte von ROOM haben über das Profil UML-RT Ein-



gang in die UML gefunden. STAUNER ET AL. (2001) präsentieren mit HyROOM einen hybriden Ansatz, der kontinuierliche Anteile mit Hilfe von Blockdiagrammen abbildet.

## FOCUS

FOCUS ist eine Methode zur Modellierung verteilter, reaktiver System und wird durch den Werkzeugprototyp AutoFOCUS unterstützt (BROY & STØLEN 2001). Die zugrunde liegenden Konzepte können gemäß BROY ET AL. (1999) wie folgt zusammengefasst werden:

- *Komponenten* dienen zur hierarchischen Strukturierung von Systemen. Sie interagieren über Signale mit der Umwelt.
- *Datentypen* definieren die Datenstrukturen eines Systems.
- *Datenelemente* speichern Zustandsinformationen und sind in Komponenten gekapselt.
- *Ports* sind die Schnittstellen einer Komponente. Sie verfügen über einen Namen und einen Datentyp.
- *Kanäle* sind gerichtet und verbinden Komponenten über die Ports.
- *Verhalten* wird zustandsorientiert beschrieben.

Zur Notation wird auf folgende Elemente zurückgegriffen:

- *Systemstrukturdiagramme* beschreiben den Aufbau des Systems durch Komponenten, Ports und Kanäle. Eine Komponente kann wiederum weitere Komponenten oder eine Verhaltensbeschreibung beinhalten.
- *Datentypdefinitionen* werden mit einer textuellen Notationsform deklariert.
- *Zustandsübergangsdigramme* dienen der Verhaltensspezifikation mit Hilfe erweiterter hierarchischer Zustandsmaschinen. Die Transitionen weisen Vor- und Nachbedingungen auf sowie eine Menge von Ein- und Ausgabemustern zur Beschreibung der Nachrichten auf den Ports.
- *Ereignisdigramme* bilden das Kommunikationsverhalten ab und basieren auf den MSCs.

Die soeben erläuterten Beschreibungsformen ermöglichen unterschiedliche Sichtweisen auf ein System und unterstützen außerdem differenzierte Abstraktionsgrade.

### **Modelica**

Modelica ist eine objektorientierte Sprache für die Modellierung von physikalischen Systemen, wobei unterschiedliche Bereiche, wie etwa die Mechanik, die Elektrotechnik oder die Regelungstechnik, abgedeckt werden können. Die Modellbildung erfolgt mit Hilfe von Objektdiagrammen, die miteinander gekoppelt werden und z. B. Komponenten eines technischen Systems darstellen (ISERMANN 2008). Zusätzlich wird die Modellierungssprache durch eine frei verfügbare Bibliothek unterstützt. Mit dem Einsatz von Modelica geht in der Regel ein sehr hoher Detailgrad einher, der in der Konzeptionsphase unter Umständen noch nicht erreicht wird.

### **2.5.4 Simulation**

Die Simulationstechnik ist heute integraler Bestandteil der Entwicklung moderner mechatronischer Produkte. Die VDI-RICHTLINIE 3633 (1996) definiert den Begriff der Simulation wie folgt:

*Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.*

Als Simulationswerkzeuge eignen sich rechnergestützte Lösungen, auf welchen mit Hilfe sogenannter Simulatoren die Modelle ausgeführt werden können.

### **2.5.5 Echtzeitfähigkeit**

SCHNIEDER (1993) bezeichnet ein System als echtzeitfähig, wenn die Bedingungen der Rechtzeitigkeit, der Gleichzeitigkeit und der Vollständigkeit erfüllt sind. Dabei bedeutet rechtzeitig, dass die festgelegten maximalen Antwortzeiten nicht überschritten werden dürfen. Da ein Prozess aus mehreren Teilprozessen bestehen kann, müssen diese gleichzeitig ablaufen. In der Rechnertechnik wird das beispielsweise durch eine quasisimultane Bearbeitung erreicht. Das Kriterium der Vollständigkeit impliziert, dass alle Prozesse zu jeder beliebigen Zeit unter Einhaltung der Rechtzeitigkeit bearbeitet werden müssen.

ZÖBEL (2008) unterscheidet zwischen harter und weicher Echtzeit. Dabei wird eine harte Zeitbedingung wie folgt definiert. Sei  $r$  der Zeitpunkt, zu welchem eine betrachtete Aufgabe beginnt und  $d$  der Zeitpunkt, zu dem diese abgeschlossen sein muss. Die Zeitspanne, die für die Ausführung der Aufgabe in Anspruch genommen wird, sei  $\Delta e$ . Unter Einhaltung der Forderung der Rechtzeitigkeit ergibt sich die Zeitbedingung  $A$  zu:

$$A \equiv r + \Delta e \leq d \quad (2.12)$$

Harte Echtzeitsysteme sind dadurch charakterisiert, dass die Zeitbedingung  $A$  unbedingt, d. h. mit einer Wahrscheinlichkeit gleich eins, eingehalten wird. Im Gegensatz dazu werden bei weicher Echtzeit die Zeitbedingungen für *den überwiegenden Teil der Fälle* erfüllt oder geringfügig überschritten, was mit einer Wahrscheinlichkeit kleiner eins bezüglich der Einhaltung gleichzusetzen ist.

## 2.6 Virtuelle Inbetriebnahme

### 2.6.1 Grundprinzip

Die Bezeichnung *Virtuelle Inbetriebnahme (VIBN)* ist laut WÜNSCH (2008) um die Jahrtausendwende entstanden und beschreibt ein Vorgehen, das bereits aus früheren Forschungsansätzen bekannt ist. Wie schon in Kapitel 1 erläutert, beruht das Grundprinzip der VIBN auf einer Vorwegnahme der steuerungstechnischen Inbetriebnahme an einem virtuellen Modell der Maschine, siehe Abbildung 2.15.

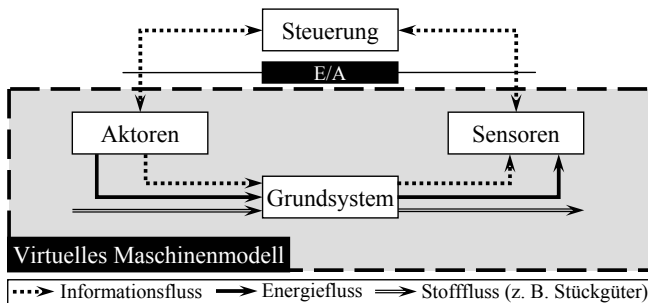


Abbildung 2.15: Grundsätzliches Funktionsprinzip der VIBN

Mit Hilfe der Simulation lässt sich der Steuerungscode inkrementell und entwicklungsbegleitend testen. Auf die Maschine als Testrahmen kann somit in den frühen Phasen zunächst verzichtet werden. Die Kopplung von Maschinenmodell und Steuerung erfolgt über das Prozessabbild, d. h. über die Steuerungseingangs- und -ausgangssignale (E/A). In der realen Anlage beeinflusst die Steuerung über die Aktoren die Zustandsgrößen des Grundsystems und erhält die Messgrößen über die Sensoren zurück. Die Wechselwirkungen zwischen Aktoren, Sensoren und Grundsystem müssen demnach im virtuellen Maschinenmodell für einen aussagekräftigen Test entsprechend abgebildet werden.

### 2.6.2 Ausprägungen

Eine VIBN lässt sich grundsätzlich als Hardware-in-the-Loop- (HiL-Simulation) oder als Software-in-the-Loop-Simulation (SiL-Simulation) durchführen. Die HiL-Simulation ist durch die Verwendung der originalen Steuerungstechnik und -software gekennzeichnet, siehe Abbildung 2.16. Die Maschine wird durch ein virtuelles Modell substituiert, welches auf einem Simulationsrechner in Echtzeit abläuft (ZÄH ET AL. 2004, PRITSCHOW ET AL. 2005). Dieser wird in der Regel über eine oder mehrere Feldbusanschlüsse mit der Steuerung verbunden. Prinzipiell bestehen weitere Kopplungsmöglichkeiten, z. B. über Einzelverdrahtungen, die in der Praxis aber keine Bedeutung haben. Da der HiL-Ansatz die gewohnte Entwicklungsumgebung samt Steuerungshardware in den Test mit einbindet, wurde dieser bislang im Unternehmensumfeld häufig bevorzugt.

Im Vergleich dazu wird bei der SiL-Simulation die *Steuerungssoftware [...] im Verbund mit einer Umgebungssimulation auf dem Host-Rechner, d. h. nicht der Zielhardware*, ausgeführt (BENDER 2005). Hierfür werden sogenannte Software-emulatoren verwendet (siehe Abbildung 2.16), wobei nach BENDER (2005) die Steuerung derart reproduziert wird, *dass sie sich so wie das [...] nachgebildete System* verhält. Der Steuerungsemulator, der nach RÖCK (2007) auch als *virtuelle Steuerung* bezeichnet wird, kann entweder mit der Maschinensimulation auf einem gemeinsamen oder auf einem über eine Schnittstelle gekoppelten Rechner ausgeführt werden. Weiterhin kann der Model-in-the-Loop-Ansatz (MiL) von der SiL-Simulation unterschieden werden. Beide Verfahren werden zwar im deutschen Sprachraum meist unter dem Oberbegriff SiL eingereiht (ISERMANN 2006). Jedoch kommt für die MiL-Simulation lediglich ein *Modell der Steuerungssoftware (zum Einsatz)*, z. B. in Form eines Zustandsautomaten, das mit der Maschinensimulation verbunden ist (BENDER 2006), siehe Abbildung 2.16.

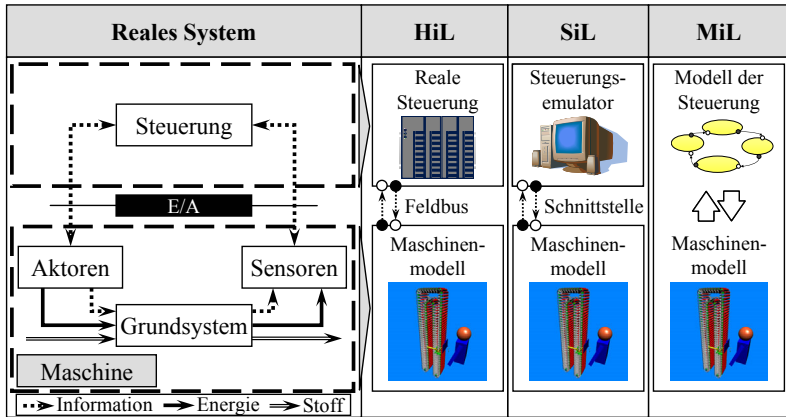


Abbildung 2.16: Unterschiede zwischen Hardware-, Software- und Model-in-the-Loop-Simulation

Da für die SiL- und die MiL-Simulation nicht die reale Steuerungshardware genutzt wird, besteht im Gegensatz zur HiL-Simulation nicht die Notwendigkeit, dass die verwendeten Simulationssysteme echtzeitfähig sein müssen (ISERMANN 2006). Dies impliziert allerdings, dass die Steuerungshardware und die Feldbusverbindungen nicht getestet werden können. Dafür lassen sich wesentlich detailliertere Modelle als mit dem HiL-Ansatz simulieren.

### 2.6.3 Modelle für die Virtuelle Inbetriebnahme

Ein klassisches VIBN-Modell setzt sich aus dem *Peripheriemodell*, dem *Materialflussmodell* und dem *Visualisierungs-* bzw. *Geometriemodell* zusammen (Abbildung 2.17). Diese Aufteilung ergibt sich auch aus der Historie, da früher für das Materialflussmodell und die Visualisierung teilweise externe Simulationswerkzeuge angebunden wurden. SPITZWEG (2009) definiert als zusätzliche Modellkategorie das *Kinematikmodell*, dessen Freiheitsgrade in der Regel im Visualisierungsmodell hinterlegt sind. Das Peripheriemodell wird ebenso als *Verhaltensmodell* oder als *Signalsimulation* bezeichnet und stellt die Schnittstelle über das Prozessabbild zur Steuerung dar (DIERSSSEN 2002). Es umfasst die Aktoren und die Sensoren sowie alle Wirkketten, die nicht im Materialfluss- oder im Visualisierungsmodell hinterlegt sind. Teilweise sind Sensoren im Materialfluss- oder Visualisierungsmodell abgebildet, wie z. B. Lichtschranken, und nur noch signaltechnisch mit der Peripheriesimulation verbunden.

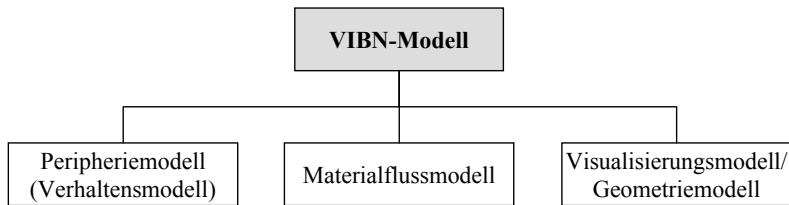


Abbildung 2.17: Aufbau von klassischen VIBN-Modellen

### Peripheriemodell

Den prinzipiellen Aufbau eines Aktor- oder Sensormodells zeigt Abbildung 2.18. Zunächst müssen die Steuerungsausgänge verarbeitet werden. Diese können, je nachdem, ob der Komponente eigene Steuerungsintelligenz innewohnt, unterschiedlich komplex ausgeprägt sein. Daraufhin wird das Verhalten berechnet, wobei eventuell weitere Daten über Schnittstellen zum Grundsystem oder anderen Aktoren und Sensoren mit einzubinden sind. Abschließend müssen die Zustandsgrößen ausgewertet und die Steuerungseingänge geschrieben werden. Aktoren oder Sensoren können gegebenenfalls nur über eine Teilmenge der beschriebenen Schnittstellen verfügen.

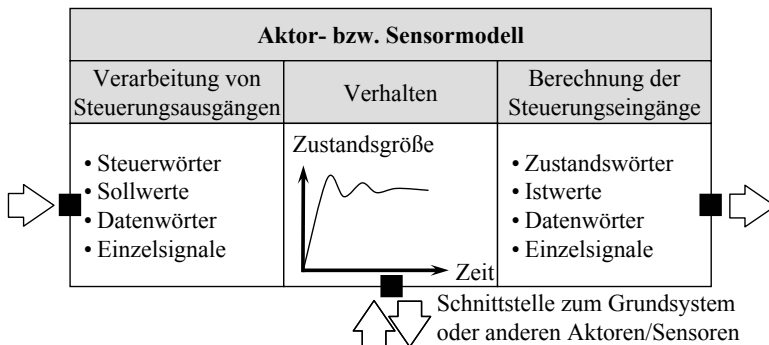


Abbildung 2.18: Aufbau von Aktor- und Sensormodellen für die VIBN

### Materialflussmodell

Für die Modellierung des Materialflusses werden üblicherweise logische Ersatzmodelle genutzt, die *unabhängig von den geometrischen Konstruktionsdaten* entwickelt werden (ZÄH ET AL. 2008). Die Erkennung von Kollisionen beruht

dabei auf der programmierten Logik und nicht auf der tatsächlichen räumlichen Ausprägung der Objekte. Einfache Beispiele für derartige Ansätze sind skriptbasierte Lösungen oder das sogenannte Schieberegister, bei dem das Material diskrete Positionen einnimmt und zeitlich getaktet zur nächsten Stelle weitergereicht wird. Zum Teil werden Ablauf- oder Logistiksimulationsprogramme mit der VIBN-Umgebung gekoppelt, um den Materialfluss einfacher beschreiben zu können. Den oben erläuterten Methoden ist gemein, dass das Verhalten sehr stark abstrahiert wird und Änderungen am Modell nur sehr umständlich umsetzbar sind. So bedingt z. B. ein zusätzlich eingeführtes Abgreifen von Bauteilen auf einem Förderband eine Anpassung am Modellcode bzw. an der Modellstruktur. Des Weiteren ist im Vergleich zu den Aktoren und Sensoren eine Modellwiederverwendung nur partiell möglich, da Transportsysteme meist kundenspezifisch an die Gegebenheiten vor Ort anzupassen sind. Aus den genannten Gründen macht der Materialfluss einen Großteil der Modellierungsarbeit aus.

Neue Verfahren gehen dazu über, die physikalischen Eigenschaften des Materialflusses abzubilden, um die Aussagekraft der Simulation deutlich zu erhöhen (SPITZWEG 2009). Das Stückgut wird als Starrkörper modelliert und ist durch die dreidimensionale Darstellung aus dem CAD repräsentiert. Auf jeden Körper können simulierte Kräfte wirken, wie z. B. die Schwerkraft oder die Reibung. Das Gesamtverhalten wird somit nicht durch das logische Ersatzmodell, sondern durch die auf die Starrkörper einwirkenden Kräfte bestimmt.

### **Visualisierungs- bzw. Geometriemodell**

Das Visualisierungsmodell dient zum Verfolgen der Maschinenbewegungen und -zustände. Meist kommen dreidimensionale Modelle zum Einsatz, die sich prinzipiell für die Kollisionsrechnung eignen. Hierfür hat sich z. B. der VRML-Standard (Virtual Reality Modeling Language) als zweckmäßig erwiesen, der aus gängigen MCAD-Werkzeugen exportiert werden kann. Mitunter werden auch zweidimensionale Visualisierungsbilder verwendet, die sich beispielsweise zur vereinfachten Darstellung schlecht einsehbarer Maschinenteile eignen.

## **2.6.4 Modellerstellungsprozess**

Der Modellerstellungsprozess für die VIBN setzt sich aus einer Analyse- und einer Modellierungsphase zusammen, siehe Abbildung 2.19.



Abbildung 2.19: Modellerstellungsprozess der VIBN

Die notwendigen Daten für die Modellerstellung der VIBN sind in Tabelle 2.2 zusammenfassend dargestellt. Eine Vielzahl der erforderlichen Informationen ist in CAD-Systemen hinterlegt. Neben der verbauten Aktorik und Sensorik können daraus Geometriekennwerte sowie kinematische Beziehungen extrahiert werden. Die genaue Ausprägung des Materialflusses einer Maschine ist allerdings nur begrenzt ersichtlich. Dieser kann rudimentär aus sogenannten Funktionsbeschreibungen entnommen werden. Die Adressbelegung der Steuerungsein- und -ausgänge ist in E/A-Listen oder im Steuerungsprojekt zu finden. Wie Tabelle 2.2 weiterhin verdeutlicht, ist eine Fülle an Informationen aus unterschiedlichen Bereichen zu analysieren, was den Prozess sehr zeitaufwendig gestaltet.

Notwendige Daten	Speicherform der Daten
Eingesetzte Aktorik und Sensorik	<ul style="list-style-type: none"> <li>• E/FCAD</li> <li>• MCAD</li> </ul>
Verhalten der Aktoren und Sensoren	<ul style="list-style-type: none"> <li>• Funktionsbeschreibungen</li> <li>• Herstellerspezifikationen</li> </ul>
Steuerungsein- und -ausgänge	<ul style="list-style-type: none"> <li>• E/A-Listen</li> <li>• Steuerungsprojekt</li> </ul>
Geometrie	<ul style="list-style-type: none"> <li>• MCAD</li> <li>• Technologieschemata</li> </ul>
Kinematik	<ul style="list-style-type: none"> <li>• MCAD</li> <li>• Funktionsbeschreibungen</li> </ul>
Wirkketten, insbesondere Materialfluss	<ul style="list-style-type: none"> <li>• Funktionsbeschreibungen</li> <li>• Skizzen</li> </ul>
Busprotokoll der verwendeten Aktoren und Sensoren	<ul style="list-style-type: none"> <li>• Herstellerspezifikationen</li> <li>• Bedienungsanleitungen</li> </ul>

Tabelle 2.2: Notwendige Daten für die Modellerstellung der VIBN

Die Modellierungsarbeit an sich ist in der industriellen Praxis durch eine manuelle Herangehensweise geprägt. Mit Hilfe der analysierten Daten wird die Maschine strukturiert und in Teilbereiche gegliedert. Für die Einzelelemente werden dann die zugehörigen Partialmodelle entwickelt. Die Maschinenmodule müssen



im Anschluss zum Gesamtsystem integriert werden. Unterstützt wird der Modellierer bestenfalls durch Bibliotheksmechanismen, die Standardmodule, wie z. B. elektrische Antriebe, enthalten.

### **2.6.5 Kommerzielle Simulationssysteme**

Simulationssysteme für die VIBN müssen spezielle Anforderungen erfüllen, wie etwa die Bereitstellung von Schnittstellen zur realen Steuerung oder eine geeignete Visualisierung. Des Weiteren sind sogenannte Bedienbilder oder Bedienpanels erforderlich, mit denen der Benutzer in das Modellgeschehen eingreifen kann.

Zwei in der Industrie häufig genutzte Simulationssysteme sind der Sinumerik Machine Simulator (Firma Siemens AG) und WinMOD (Firma Mewes und Partner GmbH). Die Modellbildung kann für beide Werkzeuge anhand von Blockschaltbildern vorgenommen werden, die sich mit Hilfe von Plänen organisieren lassen. Durch Bibliotheksmechanismen soll die Modellwiederverwendung verbessert werden. Der Sinumerik Machine Simulator verfügt zusätzlich über eine Programmierschnittstelle, die unter Verwendung eines Dialektes der Programmiersprache C die Erstellung von benutzerdefinierten Modellblöcken ermöglicht.

Ein anderer Vertreter kommerzieller VIBN-Lösungen ist die Simulationssoftware Virtuos der Firma ISG GmbH, einer Ausgründung der Universität Stuttgart. Die Modellierung basiert ebenfalls auf Blockschaltbildern. Prinzipiell kann auch das Produkt Matlab/Simulink der Firma The MathWorks für die VIBN genutzt werden, jedoch umfasst dieses keine Standardkopplung zur Steuerung. Außerdem gestaltet sich die Generierung des ablauffähigen Echtzeitcodes wesentlich zeintensiver im Vergleich zu Spezialwerkzeugen für die VIBN. Zusätzlich sind die Erstellung von Bedienbildern sowie die Integration der Bediensignale in die Simulation mit Programmieraufwand verbunden und dadurch mit einem größeren Zeitbedarf behaftet.

### **2.6.6 Nutzen und Aufwand**

Der Nutzen einer VIBN liegt hauptsächlich in der Reduzierung der Entwicklungs- und Inbetriebnahmezeiten sowie in der qualitativen Verbesserung des Steuerungs codes. ZÄH ET AL. (2006) quantifizierten im Rahmen einer Feldstudie die erzielbaren Zeiteinsparungen. So konnte die Inbetriebnahmezeit durch die Simulation um ca. 75 Prozent verkürzt werden. Gleichzeitig ließ sich eine deutli-

che Qualitätssteigerung der Steuerungssoftware, die in korrekt umgesetzten Steuerungsfunktionen gemessen wurde, statistisch nachweisen. Weiterhin ist mit Hilfe des Modells eine entwicklungsbegleitende Softwareerstellung möglich. Neben dem reinen Auffinden von Fehlern können Maschinen und Anlagen vorab oder während des laufenden Betriebs optimiert und getestet werden (HEINRICH & WORTMANN 2004). Ferner lassen sich destruktive Störszenarien risikofrei prüfen. Darüber hinaus kann die VIBN für Schulungszwecke oder für Präsentationen beim Kunden herangezogen werden.

Der dem Nutzen gegenzurechnende Aufwand besteht hauptsächlich in der Modellierungsarbeit, die von entsprechend geschultem Personal durchzuführen ist. Wie bereits in den vorangegangenen Kapiteln erläutert, gestaltet sich die Modellbildung äußerst arbeitsintensiv und langwierig. Zudem fallen Kosten für die Simulationsumgebung an. In Tabelle 2.3 sind der identifizierte Nutzen und der zu leistende Aufwand einer VIBN direkt gegenübergestellt.

<b>Nutzen</b>
<ul style="list-style-type: none"><li>• Verkürzung der Inbetriebnahme</li><li>• Höhere Softwarequalität</li><li>• Inkrementelle und entwicklungsbegleitende Softwareerstellung</li><li>• Optimierung</li><li>• Zerstörungsfreie Prüfung und Reproduktion von Fehlerszenarien</li><li>• Schulungen und Präsentationen</li></ul>
<b>Aufwand</b>
<ul style="list-style-type: none"><li>• Erstellungsaufwand des Simulationsmodells</li><li>• Einsatz von geschultem Personal</li><li>• Betrieb von Simulationssoftware und -hardware</li></ul>

*Tabelle 2.3: Nutzen und Aufwand einer VIBN*

Nachdem in Kapitel 2 die zum allgemeinen Verständnis notwendigen Grundlagen erläutert wurden, befasst sich das folgende Kapitel dediziert mit den wesentlichen technischen und wissenschaftlichen Entwicklungen in den Themenbereichen Steuerungssoftwaretest und Modellierung von mechatronischen Systemen in frühen Phasen.

## **3 Stand der Forschung und der Technik**

### **3.1 Kapitelüberblick**

Das dritte Kapitel behandelt den Stand der Forschung und der Technik. Hierzu werden in Abschnitt 3.2 Prüfmethode für die Steuerungssoftware betrachtet. Dies umfasst die formale Verifikation, die analytischen Verfahren und das Testen. In Abschnitt 3.3 werden simulationsbasierte Testmethoden untersucht, die für die vorliegende Dissertation von besonderer Bedeutung sind. Dabei stehen vor allem Ansätze im Vordergrund, die sich explizit mit der Modellerstellung beschäftigen bzw. diese beschleunigen. Neben rein wissenschaftlichen Methoden werden gleichermaßen Verfahren und Simulationssysteme aus dem industriellen Bereich vorgestellt. Abschließend wird der Handlungsbedarf für die VIBN zusammengefasst und konkretisiert. Da im Rahmen dieser Arbeit VIBN-Modelle auf der Grundlage eines disziplinenübergreifenden Funktionsmodells abgeleitet werden, sind in Abschnitt 3.4 relevante Modellierungsmethoden zur systematischen Unterstützung von Entwicklungsprozessen in frühen Phasen dargestellt.

### **3.2 Prüfmethode für Steuerungssoftware**

#### **3.2.1 Klassifikation der Prüfmethode**

Prüfverfahren für Software lassen sich grundsätzlich in statische und dynamische Methoden gliedern (SPILLNER & LINZ 2005). Diese unterscheiden sich dadurch, ob das zu prüfende Programm während der Untersuchung ausgeführt wird oder nicht. Den statischen Verfahren werden auch die analytischen Herangehensweisen zugerechnet, wie z. B. Inspektionen oder Reviews. Der dynamische Test wird in der Literatur häufig in Black-Box- und White-Box-Techniken gegliedert. Bei den Black-Box-Ansätzen findet die innere Struktur bzw. der Aufbau des Prüfprogramms keine Berücksichtigung. Im Gegensatz dazu bezieht der White-Box-Test die Programmstruktur mit ein. LIGGESMEYER (2009) beschreibt diese Einteilung des dynamischen Tests nach heutigem Wissensstand als zu grob und schlägt die in Abbildung 3.1 gezeigte Klassifikation vor, die sich an der Testreferenz ausrichtet, anhand welcher die Ergebnisse beurteilt werden.

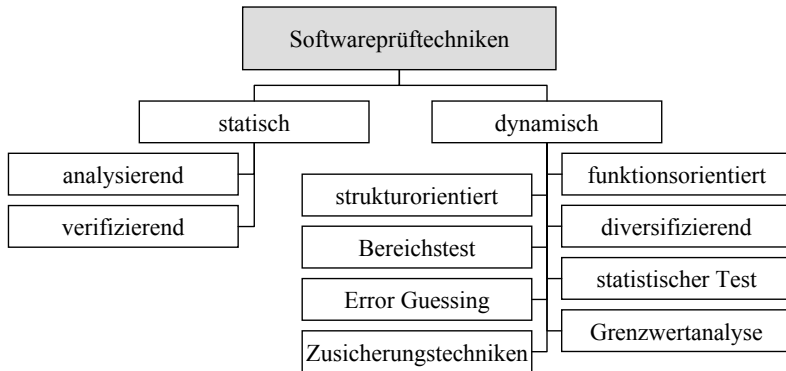


Abbildung 3.1: Einteilung von Softwareprüftechniken in Anlehnung an LIGGESMEYER (2009)

Da der funktionsorientierte und der strukturorientierte Test in der Praxis von besonderer Bedeutung sind, wird auf beide kurz eingegangen. Hinsichtlich der anderen dynamischen Verfahren sei auf entsprechende Fachliteratur verwiesen. Der funktionsorientierte Test ist eine Black-Box-Methode, bei welchem die Reaktionen des Prüfobjektes mittels einer Spezifikation bewertet werden. Es sind jedoch die Begriffe Black-Box-Test und funktionsorientierter Test einander nicht gleichzusetzen. Beispielsweise stellt der Zufallstest ebenfalls eine Black-Box-Technik dar, die allerdings nicht funktionsorientiert ist, da die Testfälle nicht aus einer Spezifikation abgeleitet werden. Die strukturorientierten Testverfahren *bewerten die Testvollständigkeit anhand der Abdeckung der Strukturelemente des Programmcodes*, z. B. Anweisungen oder Zweige, und werden demzufolge den White-Box-Methoden zugeordnet (LIGGESMEYER 2009).

### 3.2.2 Formale Verifikation

Die formale Verifikation kann unter den verifizierenden Methoden aus Abbildung 3.1 eingereiht werden. Mit Hilfe mathematischer Verfahren erfolgt dabei die Beweisführung, dass ein implementiertes Programm die Vorgaben einer formalen Spezifikation erfüllt (HOFFMANN 2008). Im Bereich der Softwareentwicklung für industrielle Automatisierungssysteme spielt die formale Verifikation dennoch eine untergeordnete Rolle. LIGGESMEYER (2009) sieht die Gründe für die geringe Akzeptanz derartiger Methoden in der Praxis vor allem in der *unzureichenden Anwendbarkeit dieser Techniken auf reale Software* sowie in den

mangelhaften Rahmenbedingungen, wie z. B. die *Verfügbarkeit einer hinreichend formalen Spezifikation*. Das notwendige Wissen zur Durchführung dieses Verfahrens ist in der Regel nicht Teil der Ausbildung des für die Steuerungsprogrammierung zuständigen Personals.

### 3.2.3 Analytische Prüfmethoden

Eine Möglichkeit, die Korrektheit von Software zu prüfen, sind sogenannte Inspektionen, Reviews oder Walkthroughs. Es handelt sich hierbei um manuelle Prüfmethoden, die BENDER ET AL. (1999) den analytischen Verfahren zurechnen. Das Prüfobjekt wird durch Experten gesichtet und bewertet. Organisatorische Maßnahmen, wie z. B. die Beteiligung mehrerer Gutachter, können sich positiv im Hinblick auf die Fehlersuche auswirken (OSKARSSON & GLASS 1997). Der Prozess läuft dabei mehr oder weniger formal ab, was gleichfalls die Unterschiede zwischen einer Inspektion, einem Review und einem Walkthrough ausmacht. Gemäß BALZERT (2009) ist das Vorgehen bei der Inspektion am stärksten und beim Walkthrough am schwächsten geregelt.

Analytische Verfahren sind geeignete und kostengünstige Mittel, um die Fehlerzahl im Steuerungscode zu reduzieren. FRÜHAUF (1990) geht bei Anwendung dieser Methoden von einer Fehlerentdeckungsrate von bis zu 80 Prozent aus, wobei lediglich 25 Prozent der gesamten Fehlerbehebungskosten verursacht werden.

### 3.2.4 Dynamischer Test

Dem dynamischen Test wird im Maschinen- und Anlagenbau eine große Bedeutung beigemessen. Im Gegensatz zu den statischen Methoden wird dabei das zu prüfende Programm ausgeführt. Das Testobjekt ist mit Eingangsdaten zu versorgen, wodurch konkrete Systemausgaben erzeugt werden. Anhand der Spezifikation lassen sich die Reaktionen bewerten. Zu diesem Zwecke ist die Definition unterschiedlicher Testfälle erforderlich. Auf Grund der Komplexität des Steuerungsprogramms kann in der Regel nur eine begrenzte Auswahl an Testszenarien berücksichtigt werden. Eine Abdeckung aller möglichen Kombinationen von Eingangsdaten wird *erschöpfender Test* genannt. Dieser spielt allerdings in der Praxis keine Rolle bzw. ist wegen der vorherrschenden Komplexität nicht realisierbar (ROITZSCH 2005). Verglichen mit der formalen Verifikation, die die Abwesenheit von Fehlern nachweisen soll, kann der Test nur die Anwesenheit von

Fehlern aufzeigen (DIJKSTRA 1969). Daher kommt der Auswahl geeigneter Testfälle eine tragende Rolle zu (VOGEL-HEUSER 2003). Als Testrahmen lassen sich unterschiedliche Werkzeuge verwenden, die das Testobjekt mit Eingabedaten versorgen und die gleichzeitig dessen Rückmeldungen aufzeichnen (SPILLNER & LINZ 2005). Im Folgenden werden diverse Ansätze vorgestellt, die ebenfalls im industriellen Einsatz vorzufinden sind. Dem Steuerungstest mit Softwaresimulatoren wird ein eigener Abschnitt gewidmet, siehe Abschnitt 3.3.

#### **Testwerkzeuge für Steuerungssoftware**

Der *Schalterkasten* ermöglicht einen manuellen, durch den Bediener zu verrichtenden Steuerungstest. Mit Hilfe von Tastern, Schaltern, Potentiometern oder Impulsgeneratoren lassen sich Stimuli für die Steuerung erzeugen. Lampen und Anzeigen dienen zur Visualisierung von aktuellen Zuständen und Werten (EVERSHEIM 1990). Schnelle und komplexe Signalverläufe können mit dieser Methode nur eingeschränkt dargestellt werden, da der Bediener aktiv in die Signalgenerierung integriert ist (WECK & BRECHER 2006).

Ein relativ einfacher Weg, Steuerungssoftware zu untersuchen, ist die Implementierung eines Testrahmens auf dem Zielsystem. Das Prinzip ähnelt dem des Schalterkastens, da das Verhalten der realen Maschine an der Schnittstelle zur Steuerung nachgebildet wird. Hierzu werden zusätzliche Programmteile zur Erzeugung von Eingangsdaten in der Steuerung hinterlegt und mit dem eigentlichen Steuerungsprogramm gekoppelt (WÜNSCH 2008). Der Vorteil dieses Verfahrens ist vor allem darin begründet, dass keine kostspielige Simulationshardware und -software erforderlich ist.

KOHRING (1993) beschreibt eine Testmethode, bei der zwei speicherprogrammierbare Steuerungen über die Ein- und Ausgabebaugruppen verbunden werden. Die eine der beiden Steuerungen fungiert als Testrahmen, um das Maschinenverhalten zu simulieren. Die andere Steuerung führt das zu testende Programm aus. Die Herangehensweise entspricht in den Grundzügen der heutigen VIBN und kann als deren Vorläufer angesehen werden.

Für den Softwaretest von Großserienprodukten ist der Einsatz sogenannter Hardwaresimulatoren sinnvoll. Proprietäre Hardwareschaltkreise ermöglichen die Erzeugung komplexer Signalverläufe (BENDER ET AL. 1999). Eine Verwendung im Bereich des Maschinen- und Anlagenbaus ist jedoch nicht zweckmäßig, da dort vorwiegend kleine Serien oder gar Einzelstücke gefertigt werden. Dementsprechend müssten für Neukonstruktionen die Hardwaresimulatoren modifiziert wer-

den, was wiederum mit Folgekosten verbunden wäre. Softwarebasierte Simulationsansätze, wie sie im nachfolgenden Abschnitt beschrieben werden, sind auf Grund von deren Flexibilität besser für Produktionssysteme geeignet (KAISER 2000).

## **3.3 Softwarebasierte Simulations- und Testmethoden**

### **3.3.1 Forschungsarbeiten**

#### **3.3.1.1 Kategorisierung**

Die VIBN zählt zu den dynamischen Prüfmethoden. Die erforderlichen Ein- und Ausgaben werden von einem softwarebasierten Simulator generiert bzw. verarbeitet. In Abschnitt 2.6.2 wurden bereits die Grundlagen für die HiL- und SiL-Simulation erörtert. Im vorliegenden Abschnitt sollen nun Forschungsarbeiten vorgestellt werden, die sich explizit mit der Modellierung bzw. mit der Reduzierung des dafür notwendigen Aufwandes auseinandersetzen. Die unterschiedlichen Lösungswege umfassen die nachstehend aufgeführten Leitgedanken:

- Geeignete und intuitive Modellierungstechniken
- Modularisierung und Standardisierung von Simulationsmodellen
- Modellvereinfachung durch Erhöhung des Abstraktionsgrades und Auswahl relevanter Maschinenbereiche
- Integration von Konstruktionsdaten

#### **3.3.1.2 Einsatz von adäquaten Modellierungstechniken**

Eine geeignete Beschreibungstechnik kann einen wichtigen Beitrag dazu leisten, die Modellierungsarbeit zu beschleunigen. Dieser Problemstellung widmete sich TOMASZUNAS (1998), der im Rahmen einer umfassenden Analyse vor allem die wenig effiziente Maschinenmodellierung sowie die schwierige Abbildung des Prozessgutes und der intelligenten Komponenten als Schwachpunkte identifizierte. Der Forderung nach einer zweckmäßigen und ökonomischen Modellierungstechnik wurde auf der Basis von ROOM begegnet. Diese objektorientierte Entwicklungsmethode für verteilte Echtzeitsysteme fand in diesem Zusammenhang

eine Anwendung auf die Maschinenmodellierung. Da es sich um eine ereignis-diskrete Technik handelt, wurden kontinuierliche Vorgänge durch eine Diskretisierung approximiert. Der Materialfluss wurde als Datenstruktur modelliert, die zwischen den einzelnen Akteuren (Akteur im Sinne der Modellierungssprache ROOM zur Strukturbeschreibung) weitergereicht wird, ähnlich einem Schieberegister. Die tatsächliche Geometrie der Materialobjekte fand jedoch keine Berücksichtigung, da es sich um ein logisches Ersatzmodell handelt. Eine Kollisionserkennung ist somit nicht möglich. Dies trifft ebenso auf die räumlichen Ausprägungen der Anlage zu, die vernachlässigt wurden. TOMASZUNAS (1998) erhebt den Anspruch, dass mit Hilfe der beschriebenen Methode eine intuitive und im Vergleich zu herkömmlichen Ansätzen eine schnellere Maschinenmodellierung möglich ist.

Das Ziel der Forschungsaktivitäten von ALBERT (1998) war die Entwicklung einer Beschreibungs- und Strukturierungstechnik, die eine wirtschaftliche Modellbildung gestattet. Hierzu wurde eine domänenübergreifende Softwarearchitektur für virtuelle Maschinen definiert, die sich aus unterschiedlichen Architekturteilen zusammensetzt und deren spezifische Vorteile vereint. Als Modellierungssprache wurde die UML gewählt und um zusätzliche Elemente erweitert. Einen Kernpunkt stellt die Abbildung von hybridem Maschinenverhalten dar. Der Materialfluss sowie die Geometrie und die Kollisionsrechnung waren nicht explizit Gegenstand der Untersuchungen.

#### **3.3.1.3 Modularisierung und Standardisierung von Modellen**

Die Modellwiederverwendung kann zu erheblichen Zeiteinsparungen führen (REINHART ET AL. 2007). Die Voraussetzung dafür sind standardisierte Module, mit deren Hilfe sich das Gesamtsystem aufbauen lässt. Der Ansatz kann besonders effizient auf Akteur- und Sensormodelle angewendet werden, für den Materialfluss ist dies nur begrenzt gültig. Diese Tatsache spiegelt sich in den nachfolgend beschriebenen Forschungsarbeiten wider, die den Materialfluss nur rudimentär abhandeln oder gänzlich aussparen.

Die Modularisierung und Bibliotheksbildung von Maschinenmodellen war auch Gegenstand der im vorangegangenen Abschnitt beschriebenen Verfahren von TOMASZUNAS (1998) und ALBERT (1998), die eine komponentenbasierte Systematik vorschlugen. Dies soll durch die verwendeten objektorientierten Methoden unterstützt werden.



MIN ET AL. (2002) beschäftigten sich mit der Modellierung von rekonfigurierbaren Produktionssystemen. Der Ansatz bezieht die Maschinensimulation in die übergeordnete Fabriksimulation ein. Mit Hilfe eines eigens entwickelten Softwarewerkzeugs kann das Gesamtsystem hierarchisch abstrahiert in Module zerlegt werden. Sogenannte *Manufacturing Devices* repräsentieren logische Einheiten, die den Steuerungen, beispielsweise von Werkzeugmaschinen oder Robotern, entsprechen. Die Manufacturing Devices sind durch Simulationsmodelle ersetzbar, womit sich die Möglichkeit eröffnet, auf die reale Steuerungshardware zu verzichten. Die Modellbildung auf der Maschinenebene ist modularisiert und wird durch Bibliotheksmechanismen unterstützt. Ein CASE-Werkzeug (Computer Aided Software Engineering) dient zur Erstellung der einzelnen Komponenten und zur Konfiguration der Gesamtmaschine. Hieraus kann anschließend der Modellcode für die Simulation abgeleitet werden. Die beschriebene Methode soll die Modellierung beschleunigen und gleichzeitig die Modellgüte erhöhen. Durch die Einbeziehung der Fabrikebene in die Simulation ist MIN ET AL. (2002) zufolge ein verbesserter Test auf der Maschinenebene darstellbar.

Die systematische Wiederverwendung von Modellen stand im Mittelpunkt der Forschungsaktivitäten von XU (2003). Prinzipien und Techniken aus dem Bereich der Softwareentwicklung wurden untersucht und auf die Maschinenmodellierung übertragen. Der zweigliedrige Lösungsansatz beinhaltet eine Komponentenbibliothek sowie einen Modulbaukasten. Für Erstere ist ein Modellierungsrahmen definiert, der allgemeingültige Konzepte und einheitliche Strukturen vorgibt. Während die Komponentenbibliothek generisch gestaltet sein kann, sind dagegen im Modulbaukasten maschinenfamilienspezifische Modelle abgelegt. Diese setzen sich wiederum aus Teilen der Komponentenbibliothek zusammen. XU (2003) stellt abschließend fest, *dass die Optimierung des Universalitätsgrades der Modellkomponenten, d. h. die Möglichkeit zur Nutzung der gleichen Komponenten in unterschiedlichen Kontexten, ein iterativer und relativ aufwendiger Prozess ist.* Der Sachverhalt trifft vor allem auf den Materialfluss zu, wobei als Beispiel Transportbänder genannt werden.

KIEFER (2007) beschreibt eine mechatronikorientierte digitale Planungsprozesskette, die auch die VIBN umfasst. Die Basis hierfür bilden mechatronische Komponenten, aus denen die Gesamtmaschine zusammengesetzt wird. Die Einzelmodule werden domänenübergreifend entwickelt und enthalten Informationen aus allen Fachbereichen, wie z. B. das Partialmodell für die VIBN. Ein Schwerpunkt liegt außerdem auf der organisatorischen Einbindung der entwickelten Vorgehensweise.

Ein anderer Ansatz, der auf Bibliothekskonzepte aufbaut, wurde von SPATH & LANDWEHR (2000) entworfen. Der Unterschied zu den oben erläuterten Methoden besteht darin, dass für die Modellbildung eine VR-Umgebung (Virtual Reality) zum Einsatz kommt. Des Weiteren ist eine Betriebsmittelbibliothek nutzbar, die sich aus geometrischen und funktionalen Teilen zusammensetzt. Die Erstellung neuer Komponenten verlangt zunächst die Festlegung der räumlichen Ausprägungen, bevor nachfolgend die funktionalen Eigenschaften zu definieren sind. Im Anschluss an den Modellaufbau der Produktionsanlage wird die Steuerungslogik grafisch-interaktiv programmiert. Dabei werden Sensoren, Merker und Aktoren zu kleinsten Steuerungsnetzwerken zusammengefügt. Zur Abbildung der Gesamtlogik werden diese wiederum über Petri-Netze miteinander verbunden. Den systeminternen Ablauf setzt ein Compiler in AWL-Code um, der auf einer SPS ablauffähig ist.

#### **3.3.1.4 Begrenzung des Simulationsbereichs und Erhöhung des Abstraktionsgrades**

Die bisher beschriebenen Verfahren haben zum Ziel, die Maschine oder Anlage ganzheitlich zu simulieren. Weitere Möglichkeiten, den Modellierungsaufwand zu verringern, ergeben sich aus einer stärkeren Abstraktion oder einer Begrenzung der Simulation auf Teile des Gesamtsystems.

Die Verwendung von Black-Box-Modellen reduziert ein System auf die wesentlichen, für die VIBN relevanten Belange (TOMASZUNAS 1998, XU 2003, WÜNSCH 2008, ZÄH & PÖRNBACHER 2008). DIERSSEN (2002) skizziert diesbezüglich ein Vorgehen, das auf einer sogenannten *mechatronischen Simulation* aufbaut. Ausgewählte Komponenten sollen dabei zunächst detailliert auf der Grundlage der physikalischen Gegebenheiten simuliert werden. Die somit erhaltenen Trajektorien werden dann in das VIBN-Modell integriert. Eine Entlastung hinsichtlich der Modellierungstätigkeiten soll zum einen durch die Wiederverwendung der Trajektorien und zum anderen durch die Nutzung von Parametern zur Anpassung der Verläufe erreicht werden.

WÜNSCH (2008) stellt eine skalierbare Methode vor, welche die Simulation auf wichtige Anlagenteile beschränkt, um dadurch den wirtschaftlichen Einsatz im Unternehmen zu gewährleisten. Zunächst ist der Aufwand bei der Modellierung zu bewerten. Größen, wie z. B. die Anzahl der Prozesse und der Materialflussverbindungen, gehen in die Begutachtung mit ein. Darauf folgend wird der Nut-

zen einer VIBN abgeschätzt. Die von WÜNSCH (2008) eingeführten Bewertungskriterien *Projektgröße*, *Bedeutung* und *Steuerungsanteil* werden evaluiert und daraus quantitativ der Mehrwert errechnet. Aufwand und Nutzen können anschließend in ein Wirtschaftlichkeitsportfolio eingetragen werden. WÜNSCH (2008) differenziert dabei zwischen vier Bereichen im Portfolio. Diese werden durch eine Komplexitäts- und eine Rentabilitätsgrenze abgetrennt und spiegeln unterschiedliche Wirtschaftlichkeiten von Simulationsprojekten wider.

Eine weitere Strategie, den Modellierungsaufwand zu minimieren, wird von DOMINKA (2007) in Form der *hybriden Inbetriebnahme* aufgezeigt. Das Grundprinzip beruht darauf, dass in die VIBN zunehmend reale Anlagenkomponenten integriert und gegen die simulierten Teile ausgetauscht werden. Für diese Zwecke erfolgt eine Parallelschaltung von Maschine und Simulation. Durch ein schrittweises Aktivieren der Aktorik kann der Betrachtungsschwerpunkt auf die entsprechenden Bereiche gelegt werden. Darüber hinaus wird eine Inbetriebnahme ohne Prozessgut beschrieben, bei der nur der Materialfluss simulativ abgebildet ist. Hierdurch können Beschädigungen an der Maschine vermieden werden. Je nach Anwendungsszenario kommuniziert die Steuerung entweder mit dem Simulationsmodell oder mit der Maschine bzw. mit beiden. Im Falle der Inbetriebnahme ohne Prozessgut wird die reale Aktorik angesteuert sowie diejenigen Sensoren eingebunden, die den mechanischen Zustand des Produktionssystems überwachen. Alle Sensoren zur Erfassung des Materialflusses werden simuliert. DOMINKA (2007) sieht den Aufwand der Modellerstellung für eine vollständig auf der Simulationsbasis durchgeführte Inbetriebnahme als sehr hoch an. Um Modellierungsarbeit einzusparen, wird als Option die Inbetriebnahme ohne Prozessgut vorgeschlagen. Da die reale Aktorik und Sensorik zur Erfassung der Maschinenzustände verwendet werden kann, müssen lediglich die Prozessgutsensoren und der Materialfluss modelliert werden. Dies setzt allerdings voraus, dass die Maschine physisch vorhanden ist, wodurch der Ansatz erst zu einem späten Zeitpunkt im Entwicklungsprozess angewendet werden kann.

Die Arbeiten von DOMINKA (2007) wurden von KAIN ET AL. (2010) im Rahmen des Verbundprojektes *AVILUS* aufgegriffen und zur sogenannten betriebsparallelen Simulation weiterentwickelt. Um die Verfügbarkeit von Produktionssystemen zu erhöhen, werden in der Regel Überwachungs- und Diagnoseverfahren eingesetzt. Damit sollen störende Einflüsse rechtzeitig identifiziert werden. Die für diese Zwecke zusätzlich verbaute Sensorik erhöht zum einen den Entwicklungsaufwand und zum anderen sind die verbauten Geräte selbst wieder mit einer gewissen Störanfälligkeit belastet. Daher wurde von KAIN ET AL. (2010) ein An-

satz entwickelt, bei dem mit Hilfe einer der Maschine parallel geschalteten Simulation auf zusätzliche Sensoren verzichtet werden kann. Das Prinzip ist an den regelungstechnischen Beobachter angelehnt. Das Gesamtmodell der Maschine besteht aus einem funktionalem und einem historienbasierten Anteil. Erstgenannter legt das Referenz- bzw. das Sollverhalten der Anlage fest, im zweiten Modell werden stochastische Einflüsse abgebildet, die über einen Signalabgleich mit der realen Prozesssensorik gewonnen werden.

#### 3.3.1.5 Integration von Konstruktionsdaten

Die im Entwicklungsprozess vorhandenen Konstruktionsdaten können dazu beisteuern, die Erstellung des VIBN-Modells zu beschleunigen (KLEMM 2007). Wesentliche Informationen sind im MCAD-Modell sowie in den Stromlauf- und Fluidplänen hinterlegt, siehe Abschnitt 2.6.4. Durch eine automatisierte Einbeziehung dieser Daten lässt sich nicht nur Modellierungsarbeit einsparen, sondern im gleichen Zuge die Modellqualität erhöhen. Bisherige Bemühungen, Konstruktionsdaten automatisiert zu integrieren, konzentrierten sich in der Praxis meist darauf, mit Hilfe von Stücklisten bibliotheksbasiert Elemente in der VIBN-Simulationsumgebung anzulegen, wie es beispielsweise SCHEIFELE (2008) erläutert. Dies umfasst in der Regel nur die Aktoren und die Sensoren, die lediglich einen Teil des VIBN-Modells ausmachen.

FREISEISEN ET AL. (2000) beschreiben einen Ansatz aus dem Bereich der Verfahrenstechnik, der Hydraulik- und Elektroschaltpläne in die objektorientierte Modellierungssprache Modelica überführt. Dabei kommt eine Bibliothek zum Einsatz, aus deren Bausteinen das Gesamtmodell der Maschine automatisiert aufgebaut und verschaltet wird. Das Vorgehen wird am Beispiel einer Spritzgussmaschine dargestellt, weshalb der Materialfluss keine Rolle spielt. Die MCAD-Daten bleiben unberücksichtigt, sowohl die Geometrie als auch die kinematischen Beziehungen werden vernachlässigt. Ebenso ergeben sich offene Fragestellungen hinsichtlich der Konsistenz zwischen den verwendeten CAD-Daten und dem Simulationsmodell. Auf stückgutverarbeitende Produktionssysteme ist das Verfahren angesichts der oben erläuterten Sachverhalte nicht übertragbar.

SCHOB ET AL. (2009) umschreiben ein Konzept zur Ableitung von VIBN-Modellen aus E/FCAD- und MCAD-Informationen. Wie auch bei FREISEISEN ET AL. (2000) bildet eine Modelica-Bibliothek die Basis, mit Hilfe derer die Modelle generiert werden. Elemente in den Stromlauf- und den Fluidplänen werden zunächst gruppiert und zu logischen Einheiten zusammengefasst, denen dann ent-

sprechende Simulationsmodelle in Modelica zugeordnet sind. Durch die automatisierte Verschaltung der Einzelteile entsteht eine sogenannte logische Struktur. Diese wird um informelle Daten manuell durch den Bediener erweitert. So müssen beispielsweise notwendige Parameter oder Verbindungen der Aktoren und der Sensoren zum MCAD eingegeben werden. Des Weiteren muss händisch eine funktionale Maschinenstruktur definiert werden, gegen welche die logische Struktur auf Konsistenz geprüft werden kann. Im Anschluss daran wird das Modelica-Modell abgeleitet. Simulator und Steuerung sind über *OLE for Process Control (OPC)* gekoppelt, eine offene und standardisierte Softwareschnittstelle für den Datenaustausch in der Automatisierungstechnik, wobei *OLE* für die Technologie *Object Linking and Embedding* der Firma Microsoft steht (SCHNELL & WIEDEMANN 2006). Da die realen Feldbusverbindungen nicht genutzt werden, erfordert dies Änderungen am Steuerungsprogramm. Das geschilderte Vorgehen wurde exemplarisch an einer Dreiachskinematik getestet. Eine Validierung der Methode im industriellen Maßstab steht noch aus. Der dargelegte Ansatz beschränkt sich zudem auf die Aktorik und die Sensorik, der Materialfluss bleibt außen vor. Ferner wird eine Reduzierung des unter Umständen sehr umfangreichen 3D-Modells aus der Mechanik bzw. eine Auswahl simulationsrelevanter Teile nicht angegangen. Aus den genannten Gründen ist die Methode eher für kleine und wenig materialflusslastige Maschinen geeignet.

SPITZWEG (2009) widmete sich intensiv der teilautomatisierten Modellerstellung für die physikalische Simulation des Materialflusses. Dabei werden CAD-Daten für die Ableitung des Geometriemodells genutzt. Für die Parametrierung der einzelnen physikalischen Objekte wird die Verwendung eines Produktdatenmanagementsystems vorgeschlagen, aus dem die notwendigen Informationen extrahiert werden können. Das Peripheriemodell, welches hauptsächlich die Aktoren und Sensoren umfasst, ist hierbei manuell zu erstellen und in die Simulation zu integrieren.

Das automatisierte Auswerten und Heranziehen von CAD-Modellen kann, wie oben gezeigt, Modellierungsarbeit einsparen helfen, jedoch bleiben wesentliche Aspekte unbeantwortet. Dazu zählt in den meisten Fällen der Materialfluss, der in der Regel einen Großteil der Modellerstellung ausmacht. Hierfür können, wie von SPITZWEG (2009) zumindest für die Geometrie verdeutlicht, die Informationen aus dem CAD unterstützend wirken. Zudem ist zu klären, wie mit kurzfristigen Änderungen umgegangen wird, die nicht im CAD abgebildet sind.

Ein wesentlicher Nachteil der oben beschriebenen Verfahren ergibt sich durch die Abhängigkeit der Modellerstellung von den CAD-Daten, die erst zu einem späten Zeitpunkt zur Verfügung stehen. Im Vergleich zur hier vorgeschlagenen Methode gestaltet sich die VIBN dadurch als singuläre Lösung, die den anderen Konstruktionsphasen hinten angestellt wird. Änderungen oder Optimierungen der Maschinenfunktionen sind auf Grund der konstruktiven Restriktionen und der damit verbundenen Kosten nur eingeschränkt möglich.

#### 3.3.2 Industrielle Ansätze

Dem Thema der effizienten und wirtschaftlichen Modellerstellung haben sich auch Anbieter kommerzieller Simulationssysteme verschrieben. Die Siemens AG griff im Rahmen eines Pilotprojektes das Prinzip der Modellwiederverwendung auf (BAUDISCH ET AL. 2007). Die Grundlage bildete der Simatic Automation Designer als prototypisches Entwicklungswerkzeug. Die Gesamtmaschine wird darin aus einzelnen Modulen, den sogenannten *Ressourcen*, konfiguriert. Diese lassen sich mit Hilfe von *Ports* verschalten. Jede Ressource kann wiederum unterschiedliche *Facetten* beinhalten, die disziplinspezifische Belange darstellen, wie z. B. Teile des Steuerungsprogramms oder das VIBN-Modell. Auf der Basis des Informationsmodells im Simatic Automation Designer kann das Maschinenmodell für das Simulationswerkzeug Sinumerik Machine Simulator (siehe Abschnitt 2.6.5) generiert werden. Die entsprechenden Teilmodelle werden automatisch aus der Bibliothek der Simulationsumgebung entnommen und verknüpft.

Ein weiteres Werkzeug im gleichen Kontext stellt *Process Simulate* der Firma Siemens PLM Software GmbH dar. Die Konfiguration von Fertigungszellen kann komponenten- bzw. modulbasiert geschehen. Die Erweiterung *Process Simulate Commissioning* dient zur simulativen Inbetriebnahme der projektierten Anlage. Neben den Standardelementen, die bereits über ein spezifisches Verhalten verfügen, können außerdem benutzerdefinierte Komponenten programmiert werden. Aus dem Informationsmodell in *Process Simulate* lassen sich zudem Steuerungsprogramme ableiten.

Das Entwicklungswerkzeug DELMIA Automation der Firma Dassault Systèmes bedient sich eines ähnlichen, komponentenbasierten Ansatzes. Die Verhaltensbeschreibung erfolgt mit Hilfe des sogenannten Smart Device Builders. Die Arbeiten werden direkt in einer 3D-Umgebung durchgeführt. Des Weiteren können darin systemneutrale Steuerungsprogramme erstellt und getestet werden, die sich

über Postprozessoren in herstellerspezifische Programmiersysteme importieren lassen (KIEFER 2007).

Die oben erörterten Systeme reduzieren den Modellerstellungsaufwand durch die Möglichkeit zur Standardisierung und zur Modularisierung. Für Anpasskonstruktionen oder Maschinen, die über Baukästen konfiguriert werden können, ist dieser Weg geeignet. Neukonstruktionen können damit nur bedingt zeitsparender abgebildet werden.

### **3.3.3 Handlungsbedarf**

Im vorliegenden Abschnitt soll der Stand der Technik im Hinblick auf die VIBN zusammengefasst und der konkrete Handlungsbedarf daraus abgeleitet werden. Wie Tabelle 3.1 verdeutlicht, wird die Simulation erst spät in den Entwicklungsprozess integriert. Somit ist ein entwicklungsbegleitendes Vorgehen und folglich eine frühzeitige Softwareprogrammierung nicht möglich. Meist wird exklusiv das Maschinenverhalten betrachtet und die Abbildung der Steuerung im Modell vernachlässigt. Dadurch kann die Gesamtfunktionalität in frühen Phasen, in denen noch kein fertiges Steuerungsprogramm für Testzwecke existiert, nicht simuliert werden. Es ist daher eine Vorgehensweise notwendig, die sich an den aktuellen Entwicklungsstand anpassen lässt.

Um die Modellerstellung zu beschleunigen, wurden unterschiedliche Wege beschritten. Intuitive grafische Modellierungstechniken können dazu einen Beitrag leisten. Allerdings vernachlässigen nach derzeitigem Stand diese Hilfsmittel wesentliche Kernpunkte, wie z. B. die Geometrie, den Materialfluss oder die Fehler-spezifikation. Ferner sind die Modellwiederverwendung und abhängig vom Verwendungszweck die Abstraktion von Nutzen, jedoch als alleiniges Konzept unzureichend. Weiterhin kann durch die automatisierte Einbindung der CAD-Daten nicht nur Modellierungsarbeit eingespart, sondern auch die Modellgüte verbessert werden. Einzelne Schritte dazu wurden bereits unternommen, weisen allerdings noch Verbesserungspotenzial auf und kommen zudem erst gegen Ende des Entwicklungsprozesses zum Tragen. Daher sind Lösungskonzepte zu entwickeln, welche die spät zur Verfügung stehenden CAD-Daten wirtschaftlich integrieren.

Legende:		I		II				III				IV		
● Anforderung erfüllt		TOMASZYNAS 1998	ALBERT 1998	MIN 2002	XU 2003	KIEFER 2007	SPATH & LANDWEHR 2000	DIERSSEN 2002	DOMINKA 2007	WUNSCH 2008	KAIN ET AL. 2010	FREISEISEN ET AL. 2000	SCHOB ET AL. 2009	SPITZWEG 2009
◐ Anforderung teilweise erfüllt														
○ Anforderung nicht erfüllt														
I	Methoden aus Abschnitt 3.3.1.2													
II	Methoden aus Abschnitt 3.3.1.3													
III	Methoden aus Abschnitt 3.3.1.4													
IV	Methoden aus Abschnitt 3.3.1.5													
Prozess	Frühe Integration der Simulation	○	○	○	○	◐	○	○	○	○	◐	○	○	○
	Entwicklungsbegleitendes Vorgehen	○	○	○	○	◐	○	○	○	◐	◐	○	○	○
	Modellbasierte Softwareentwicklung	○	○	◐	◐	◐	●	○	○	○	○	○	○	○
Aufwands- reduzierung	Intuitive Modellierungstechnik	●	●	○	◐	◐	◐	○	○	○	○	○	○	○
	Modellwiederverwendung	●	●	●	●	●	●	◐	◐	◐	◐	◐	◐	○
	Abstraktion, Reduktion auf Teilbereiche	◐	◐	○	◐	○	○	●	●	●	●	○	○	○
	Automatisierte Integration CAD-Daten	○	○	○	○	○	○	○	○	○	○	◐	●	◐
Modellierungs- technik	Abbildung von hybridem Verhalten	◐	●	●	◐	●	●	●	●	●	●	●	●	●
	Abbildung des Materialflusses	◐	○	○	◐	○	○	○	◐	◐	◐	○	○	●
	Geometrie, Kollisionserkennung	○	○	●	○	●	●	●	◐	○	◐	○	●	●
	3D-Visualisierung	○	○	●	○	●	○	●	●	●	●	○	●	○
	Fehlerspezifikation	○	○	○	○	○	○	○	○	○	○	○	○	○

Tabelle 3.1: Bewertung des Standes der Forschung und der Technik

Aus dem Stand der Technik wird für die vorliegende Arbeit der Handlungsbedarf wie nachfolgend beschrieben abgeleitet. Es ist eine Methode zu entwickeln, welche die unten aufgelisteten Anforderungen berücksichtigt und die durch ein geeignetes Softwaresystem unterstützt wird:

- Frühzeitige Integration der Simulation in den Entwicklungsprozess
- Entwicklungsbegleitender Ansatz, der die Abbildung von Maschinen- und Steuerungsverhalten gestattet
- Unterstützung der Modellwiederverwendung durch Bibliotheksmechanismen
- Hybride Modellierungstechnik, die wichtige Aspekte, wie etwa den Materialfluss oder die reale 3D-Geometrie, berücksichtigt
- Grafische und intuitive Modellierungstechnik, die eine an den Entwicklungsstand angepasste Abstraktion ermöglicht



- Fehlermodellierung, die parallel zum Gutablauf erstellt werden kann
- Teilautomatische Auswertung und Einbeziehung vorhandener Konstruktionsmodelle
- Teilautomatisierte Ableitung der VIBN-Modelle

## 3.4 Planung und Konzeption im Maschinen- und Anlagenbau

### 3.4.1 Interdisziplinärer Systementwurf

Den Ausgangspunkt für eine Maschinenentwicklung bilden die Anforderungen des Kunden, die normalerweise in Form eines Lastenhefts festgelegt werden. Die Überführung der Anforderungen in ein technisches Produkt verläuft über unterschiedliche Phasen. In Kapitel 2 wurden für den Maschinen- und Anlagenbau gängige Entwicklungs- und Konstruktionsmethoden vorgestellt, die diesen Prozess durch strukturierte Vorgehensweisen unterstützen können. Den Ansätzen ist gemein, dass in den frühen Entwicklungsstadien eine Planungs- und Konzeptionsphase zu durchlaufen ist. Diese dient dazu, die Grundlagen für die sich anschließende technische Detaillierung der Maschine zu liefern. So müssen z. B. gemäß VDI-RICHTLINIE 2221 (1993) die Aufgaben geplant und geklärt, Funktionen und Funktionsstrukturen ermittelt sowie Lösungsprinzipien gesucht werden. Die VDI/VDE-RICHTLINIE 2422 (1994) schlägt die Erstellung eines Gerätekonzeptes vor, welches im Anschluss disziplinspezifisch zu vertiefen ist. Ebenso empfiehlt die VDI-RICHTLINIE 2206 (2004) zunächst die Ausarbeitung eines Systementwurfs. Dieser soll auf der Basis der definierten Anforderungen die Umsetzung eines domänenübergreifenden Lösungskonzeptes ermöglichen, *das die wesentlichen physikalischen und logischen Wirkungsweisen [...] beschreibt*.

Der VDW hat eine Richtlinie für die Funktionsbeschreibung erlassen, die insbesondere die Planungs- und die Konzeptionsphase adressiert (VDW-RICHTLINIE 2001). Die Richtlinie erhebt den Anspruch, Kosten und Durchlaufzeiten zu reduzieren. Dabei steht die Funktionsbeschreibung als zentrales Element im Mittelpunkt des Entwicklungsprozesses und bildet die Grundlage für die nachfolgende Entwicklung und Konstruktion in allen Disziplinen. Eine erste grobe Funktionsbeschreibung sollte demnach bereits in der Angebotsphase bzw. der Projektvorbereitung erstellt werden, um eine Abstimmung mit dem Kunden zu gestatten.

### 3.4.2 Modelle zur Unterstützung des Entwicklungsprozesses

#### 3.4.2.1 Allgemeines

Modelle können den Entwicklungsprozess in vielfältiger Weise unterstützen. Mit Hilfe der Abstraktion von der eigentlichen Problemstellung lässt sich das Systemverständnis verbessern. Gleichzeitig können Modelle für die interdisziplinäre Kommunikation herangezogen werden. Eine modellbasierte Vorgehensweise kann vor allem in der Phase des Systementwurfs einen erheblichen Mehrwert generieren, da dort wichtige Konzeptentscheidungen getroffen werden, die maßgeblichen Einfluss auf die Gesamtentwicklungszeit und auf die Kosten haben. Sind die verwendeten Modelle zudem von formaler Natur, so lassen sich daraus weitere Artefakte ableiten, wie z. B. Software oder Entwicklungsdokumente.

Im nachfolgenden Abschnitt stehen für den Maschinen- und Anlagenbau relevante Richtlinien im Fokus, die Vorschläge hinsichtlich einer funktionalen Modellbildung unterbreiten. Im Anschluss daran wird in den Abschnitten 3.4.2.3 und 3.4.2.4 eine Auswahl an Forschungsarbeiten dargelegt, die für die vorliegende Arbeit von Bedeutung sind. Hierbei besteht kein Anspruch auf Vollständigkeit, es sollen vielmehr die wesentlichen Herangehensweisen verdeutlicht werden. Die Verfahren werden in zwei Bereiche eingeteilt. Die in Abschnitt 3.4.2.3 erläuterten Methoden unterscheiden zwischen einer Funktions- und einer Wirkstruktur, die über Beziehungen verbunden sind. Diese Praktiken sind teilweise weniger formal gestaltet und eher auf der konzeptionellen, methodischen Ebene angesiedelt. Die in Abschnitt 3.4.2.4 präsentierten Ansätze konzentrieren sich stärker auf die modellierungstechnischen Belange und die Simulation. Darüber hinaus findet sich in Abschnitt 3.4.2.5 ein kurzgefasster Exkurs in die modellgetriebene Softwareentwicklung.

#### 3.4.2.2 Richtlinien mit Modellcharakter

Die Richtlinie zur Funktionsbeschreibung des VDW (VDW-RICHTLINIE 2001) enthält neben organisatorischen Maßnahmen zusätzlich einen Vorschlag bezüglich der funktionalen Modellierung von Maschinen. Es wird eine Komposition der Anlage aus sogenannten *Funktionsobjekten* bzw. *Funktionseinheiten* empfohlen, siehe Abbildung 3.2. Diese vereinigen technische Daten und Funktionen der Mechanik, der Fluidtechnik, der Elektrik und der Steuerungssoftware. Die Funktionalität ist über definierte Schnittstellen nach außen gekapselt. Einzelne Funk-

tionsobjekte können wiederum zu *Funktionsuntergruppen* oder *Funktionsgruppen* zusammengefasst werden, wobei die *Funktionsuntergruppen* zur Strukturierung der *Funktionsgruppen* dienen.

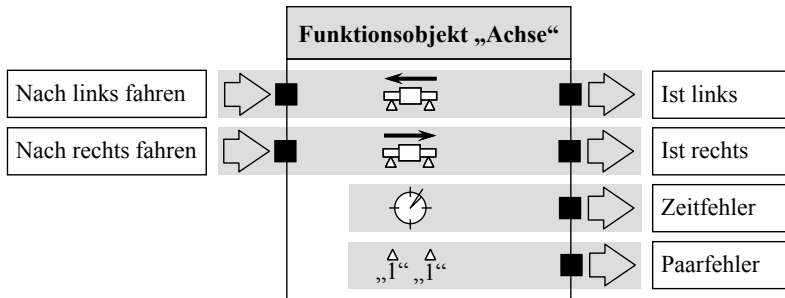


Abbildung 3.2: Beispiel für ein Funktionsobjekt gemäß der Richtlinie für die Funktionsbeschreibung des VDW (VDW-RICHTLINIE 2001)

Als Beschreibungsmittel für das funktionale Verhalten werden mehrere Techniken genannt, wobei eine Unterscheidung zwischen diskreten und kontinuierlichen Größen stattfindet. So werden beispielsweise für die Darstellung diskreter Abläufe Zustandsgraphen oder Schrittketten vorgeschlagen, wohingegen kontinuierliche Funktionen mit Hilfe von Diagrammen abzubilden sind. Ergänzend kommen Tabellenwerke zum Einsatz, die z. B. Informationen über Störungen, Parameter sowie Aktoren und Sensoren beinhalten. Werkzeugseitig wird der Modellierungsansatz nicht unterstützt, wodurch sich die praktische Anwendung als schwierig erweist.

#### Richtlinien des VDI

Die VDI-RICHTLINIE 2222 BLATT 1 (1997) stellt eine Konkretisierung der VDI-RICHTLINIE 2221 (1993) dar und konzentriert sich hauptsächlich auf die Ermittlung prinzipieller Lösungen sowie die Funktionsmodellierung. Die Formulierung der Aufgabenstellung geschieht mit Hilfe von Objekt-Prädikat-Bezeichnungen, wie z. B. *Wagen heben*. Deren weitere Abstraktion dient dazu, um die Allgemeingültigkeit zu erhöhen. So wird etwa die Objekt-Prädikat-Beziehung *Wagen heben* nach mehreren Abstraktionsschritten in *Stoff leiten* umdeklariert. Aus den Formulierungen können bestimmte Funktionen direkt abgeleitet werden. Diese lassen sich gemäß der Richtlinie für alle Maschinen mit Hilfe der drei sogenannten *allgemeinen Größen* *Stoff*, *Energie* und *Information* beschreiben. Die Größen

können für die Funktionsspezifikation mit Zustandsänderungen kombiniert werden. Als notwendig und hinreichend erachtet die Richtlinie die Zustandsänderungen *Speichern*, *Leiten*, *Umformen*, *Wandeln* und *Verknüpfen*. Zudem wird eine Notation bzw. Darstellungsform vorgeschlagen, wie sie sich Abbildung 3.3 entnehmen lässt. Jede Zustandsgröße verfügt über ein Symbol, z. B. für das Speichern ist das ein Kreis, das die allgemeine Größe in Form eines Buchstaben beinhaltet.

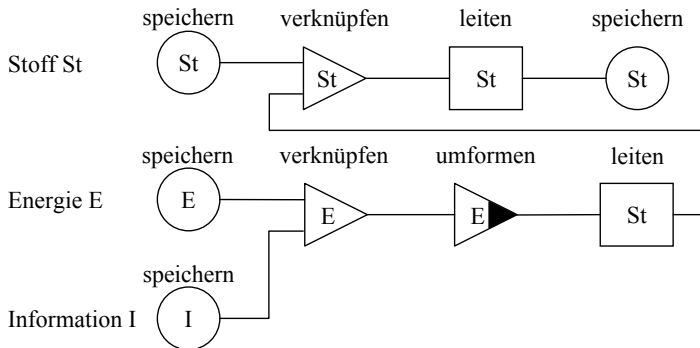


Abbildung 3.3: Funktionsmodell nach VDI-RICHTLINIE 2222 BLATT 1 (1997)

Die VDI/VDE-RICHTLINIE 2422 (1994) umfasst ebenfalls die Thematik der Spezifikation von Funktionen. Es wird eine Funktionsstrukturierung vorgeschlagen sowie eine Schnittstellenbeschreibung eingefordert. Die Richtlinie bleibt hierbei insgesamt sehr allgemein und benennt unterschiedliche Techniken, wie z. B. Funktions- und Signalflusspläne oder Zustandsgraphen.

Der Systementwurf spielt auch in der VDI-RICHTLINIE 2206 (2004) eine bedeutende Rolle. Auf der Basis der Anforderungen soll ein domänenübergreifendes Lösungskonzept erstellt werden. Hierzu ist die Gesamtfunktionalität in Teilfunktionen aufzugliedern, für die dann Wirkprinzipien und Lösungselemente zu definieren sind. Für konkrete Maßnahmen wird auf weitere Richtlinien verwiesen.

### 3.4.2.3 Integrierte Funktions- und Wirkstrukturmodellierung

Die im vorliegenden Abschnitt dargestellten Forschungsansätze nutzen zur Modellierung technischer Systeme eine abstrakte Funktionsstruktur sowie eine konkrete Wirkstruktur. Die Verfahren haben einen interdisziplinären Entwicklungs-

prozess zum Ziel, was durch ein methodisches Vorgehen erreicht werden soll. Auf Grund der teilweise weniger formalen Beschreibungsformen weisen sie einen eher dokumentativen Charakter auf. Eine simulative Absicherung ist somit nicht primäre Intention.

KALLMEYER (1998) stellt unter dem Begriff der Prinziplösung eine Modellierungstechnik für mechatronische Systeme vor, siehe Abbildung 3.4. Im Zuge der Funktionsmodellierung werden zunächst die Funktionen hierarchisiert und deren Zusammenwirken über eine Flussverkettung wiedergegeben. Systemelemente erfüllen die spezifizierten Funktionen. Ein Systemelement repräsentiert wiederum Wirkprinzipien, Lösungselemente oder Subsysteme, die weitere Systemelemente aggregieren. Einzelne Lösungselemente stehen über Assoziationsbeziehungen miteinander in Verbindung. Die Verhaltensspezifikation wird über domäneneigene Aspektmodelle realisiert, was durch eine grafische Notation in den Systemelementen dargestellt ist.

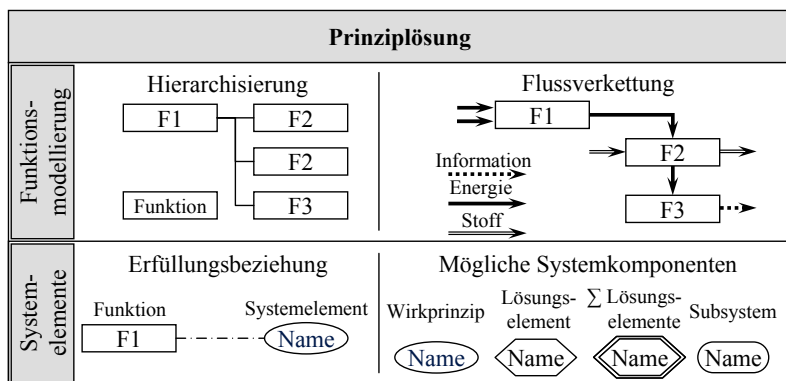


Abbildung 3.4: Methode der „Prinziplösung“ nach KALLMEYER (1998)

FLATH (2001) und GEHRKE (2005) griffen die Arbeiten von Kallmeyer auf und erweiterten diese. FLATH (2001) führt zusätzlich Zustände und Stözzustände in die Systembeschreibung ein, wobei die Funktionen als Transitionen fungieren und den Übergang zwischen den Zuständen bilden. Des Weiteren können Zustandsparameter definiert werden, die funktionale Zielgrößen repräsentieren. GEHRKE (2005) verwendet für seine Forschungsarbeiten die Systemstruktur von KALLMEYER (1998) sowie die Funktionshierarchie der VDI-RICHTLINIE 2222 BLATT 1 (1997). Beide Beschreibungsmittel werden mit Hilfe der UML formalisiert, wobei vornehmlich Klassendiagramme zum Einsatz kommen.

Unter dem Begriff des Konzeptmodells beschreibt LIPPOLD (2001) eine Methode zur Unterstützung des Entwicklungsprozesses für mechatronische Systeme. Als sogenannte Konzeptionselemente werden Funktionselemente, Wirkelemente, Verhaltenselemente und Assoziationselemente festgelegt. Dabei stellen die Assoziationselemente eine Beziehung zwischen den Funktions- und den Wirkelementen her. Zudem werden die Verhaltenselemente nichthierarchisch an die Wirkelemente gebunden. Als Modellierungstechniken für die Verhaltensbeschreibung werden Bondgraphen, Blockdiagramme und Statecharts genannt.

Das mechatronische Systemmodell von DOHMEN (2002) umfasst unterschiedliche Sichtweisen für Funktionen, für Komponenten und für das Verhalten. Dabei dient die Komponentenstruktur als Grundlage für die Modellierung des Systemverhaltens. Dessen Spezifikation basiert auf der Modellierungssprache ROOM, wodurch eine bessere Vernetzung von Mechanik- und Elektrokonstruktion erreicht werden soll. Auch eine mögliche Ableitung von Steuerungscode aus dem Systemmodell wird angedeutet. Der beschriebene Ansatz ist in ein Vorgehensmodell nach VDI-RICHTLINIE 2221 (1993) eingebettet, wobei die Lösungsfindung sich an der VDI-RICHTLINIE 2222 BLATT (1997) orientiert. Zusätzlich wird eine exemplarische Einbindung von CAX-Werkzeugen geschildert.

BATHELT (2006) greift die VDI-RICHTLINIE 2206 (2004) auf und definiert eine sogenannte *Erweiterte Funktionsstruktur (EFS)*, die zu Beginn des Entwicklungsprozesses eine Einbeziehung der Steuerungstechnik ermöglichen soll. Die Methode ist auf SPS-gesteuerte Systeme ausgerichtet und erlaubt eine automatisierte Ableitung von Steuerungscode. Die Basis bildet die ablaufbezogene Funktionsstruktur, wie sie auf eine ähnliche Art und Weise von KALLMEYER (1998) unter dem Begriff der Flussverkettung verwendet wird, siehe Abbildung 3.4. Die neu implementierten Übergangsbedingungen zwischen den Funktionen lassen sich den Eingangssignalen der Steuerung zuweisen. Funktionen, die einen Aktor repräsentieren, bezeichnet BATHELT (2006) als Aktorfunktionen und stellt sie den Ausgangssignalen der SPS gegenüber.

Ein *integriertes Produktmodell* wurde von REINHART ET AL. (2002) entwickelt. Ein Teil davon ist das Funktionsmodell, das sich unter anderem an KALLMEYER (1998) orientiert und Schnittstellen von den Typen *Stoff*, *Energie* und *Signal* besitzt. Der Ansatz ist mit unterschiedlichen Sichtweisen auf das zu entwickelnde System gekoppelt, die in einem Metamodell spezifiziert sind (LERCHER 2008). Jede Disziplin verfügt dabei über eine funktions- und eine strukturorientierte Betrachtungsweise.

#### 3.4.2.4 Objektorientierte und komponentenbasierte Systemmodellierung

Die im vorangegangenen Abschnitt vorgestellten Methoden beschreiben das zu entwickelnde Produkt auf einem abstrakten Niveau. Zwar wird ein Bezug von der Funktionsebene zur Wirkstruktur erstellt und beispielsweise über Flussverkettungen die Schnittstellen detailliert, allerdings gehen daraus die Wechselwirkungen zwischen den disziplinspezifischen Bereichen nur schwer hervor. Diese lassen sich mit Hilfe einer Simulation abschätzen, welche die ereignisdiskreten und kontinuierlichen Anteile mechatronischer Systeme berücksichtigt. Dazu sind formale Beschreibungstechniken erforderlich, die eine Ausführung des Modells auf einem Simulationsrechner erlauben. Die nachfolgend diskutierten Ansätze greifen auf objektorientierte oder komponentenbasierte Modellierungsmechanismen zurück.

Ein disziplinenübergreifendes Informationsmodell zur Beschreibung von Produktionsmaschinen war Gegenstand der Forschungsarbeiten von SCHAICH (2001). Zentrale Aspekte waren die Entwicklung eines Referenzmodells zur Verhaltensbeschreibung sowie dessen Umsetzung in einem Softwarewerkzeug. Hierfür wurde unter anderem auf endliche Automaten zurückgegriffen.

TILBURY ET AL. (2001) nutzten zur Modellierung der Steuerungslogik Petri-Netze. Mit Hilfe eines modularen Ansatzes sollen zum einen die Rekonfigurierbarkeit und zum anderen die automatische Code-Generierung unterstützt werden. Das Maschinenverhalten wird als sogenannter *Event Generator* abgebildet.

Im Rahmen des DFG-Schwerpunktprogrammes KONDISK widmeten sich BENDER ET AL. (2002) der hybriden Modellierung technischer Systeme. Das prototypische CASE-Werkzeug MaSiEd (Maschinen-Simulator/Editor) wurde mit Hilfe der eigens entwickelten Beschreibungstechnik HyROOM erweitert. Die formale Semantik basiert auf den HyCharts (STAUNER 2002).

BONFÉ ET AL. (2005) erweiterten den objektorientierten Ansatz der UML-RT um Bondgraphen als Beschreibungsmittel. Damit lassen sich kontinuierliche Aspekte der Mechanik und der Elektrotechnik im Modell berücksichtigen.

Auf der Grundlage der UML wurde von VOGEL-HEUSER & KATZKE (2005) eine Erweiterung bzw. ein Profil für die Prozessautomatisierung (UML-PA) geschaffen (siehe auch KATZKE 2008). Akteure, Sensoren sowie Steuerungs- und Rege-



lungsanteile werden in stereotypischen Klassen gekapselt, die Kommunikation erfolgt über Ports. Das Verhalten wird durch Timed State Machines festgelegt.

Ein Ansatz zur Entwicklung mechatronischer Systeme wurde von BURMESTER ET AL. (2007) präsentiert. Der Schwerpunkt lag auf der Abbildung von hybridem Verhalten. Hierzu wurde ein Modellierungswerkzeug für Echtzeitsysteme, die *Fujaba Real-Time Suite*, mit einer Entwicklungsumgebung für die Elektromechanik, *CAMeL-View*, gekoppelt. Die *Fujaba Real-Time Suite* basiert auf der UML, die zur Mechatronic UML erweitert wurde (BURMESTER ET AL. 2005).

Die UML an sich stellt ein mächtiges Beschreibungsmittel dar, das über eine Vielzahl an Modellierungskonstrukten verfügt. Der Hauptanwendungsbereich liegt in der klassischen Softwareentwicklung (LINHARES ET AL. 2006). Aus diesem Grund wurde von der Object Management Group (OMG) die SysML entwickelt, die auf der UML aufbaut und speziell auf die Systemmodellierung abzielt. Ebenso wie die UML ist die SysML Gegenstand zahlreicher Forschungsarbeiten (LINHARES ET AL. 2006, SCHÜTZ & WANNAGAT 2009, BRECHER & KARLBERGER 2009). Die genannten Arbeiten sind lediglich ein kleiner Auszug an Vorhaben, die sich mit der SysML beschäftigen und diese für die Systemspezifikation nutzen.

ZÄH & PÖRNBACHER (2008) beschreiben eine Möglichkeit zur automatischen Ableitung von SPS-Code für Werkzeugmaschinen. Die Methode basiert auf einem abstrakten Modell des Gesamtsystems, für welches zwei Detaillierungsstufen unterschieden werden. Auf der obersten Ebene (High-Level-Modeling) werden Sequenzdiagramme eingesetzt, welche die Kommunikation zwischen einzelnen Elementen beinhalten. Die darunterliegende Ebene (Detail-Level-Modeling) erfordert für jede Komponente die Definition von spezifischem Verhalten. Zur Abbildung steuerungstechnischer, ereignisdiskreter Anteile werden endliche Automaten vorgeschlagen. Für das Maschinenverhalten wird eine Reihe von Modellierungstechniken angegeben, wie z. B. Tabellen oder algebraische Gleichungen.

Ein interdisziplinäres Entwicklungswerkzeug steht im Mittelpunkt des Forschungsvorhabens AQUIMO (WÜRLIN & WASCHNER 2008). Maschinen werden dabei mit Hilfe sogenannter Basiskomponenten gegliedert, denen ein charakteristisches Verhalten innewohnt. Das kumulierte Gesamtverhalten ergibt sich aus dem Zusammenwirken der Einzelbestandteile. Aus der Entwicklungsumgebung können Modelle für das VIBN-Werkzeug *Virtuos* erzeugt werden. Die Methode ist allerdings dadurch eingeschränkt, dass lediglich vordefinierte Standardkom-

ponenten miteinander automatisch verschaltet werden. Aus diesem Grund lässt sich das Maschinenverhalten bzw. das Verhalten der Basiskomponenten nicht direkt in der mechatronischen Entwicklungsumgebung spezifizieren.

#### 3.4.2.5 Modellgetriebene Softwareentwicklung

Im Gegensatz zum Maschinenbau, bei dem modellgetriebene Ansätze zur Ableitung von Steuerungscode nicht sehr weit verbreitet sind, finden diese in der klassischen Softwareentwicklung starken Anklang. Die modellgetriebene Softwareentwicklung (engl. Model Driven Software Development, kurz MDSD) nutzt die abstrakte Modellbeschreibung, um daraus Programmcode zu erzeugen. In dem Zusammenhang ist auch die Model Driven Architecture (MDA) der OMG zu nennen. Diese unterscheidet zwischen einem plattformunabhängigen Modell (engl. Platform Independent Model, kurz PIM) der Applikationslogik und dem plattformspezifischen Modell (engl. Platform Specific Model, kurz PSM), wobei Letzteres *mittels Modell-zu-Modell-Transformationen hergeleitet wird*. Aus dem PSM kann schließlich über eine Modell-zu-Text-Transformation Code für die jeweilige Programmiersprache generiert werden (BALZERT 2009).

#### 3.4.3 Fazit

Die beschriebenen Ansätze sind häufig sehr abstrakt gehalten. Dies trifft vor allem auf die Methoden in Abschnitt 3.4.2.3 zu, die außerdem meist nicht genügend formal gestaltet sind, um für Simulationszwecke eingesetzt werden zu können. Teilweise beschäftigen sich die vorgestellten Herangehensweisen mit mechatronischen Produkten im Allgemeinen, was impliziert, dass für die Automatisierungstechnik relevante Belange nicht genügend berücksichtigt werden. Die in Abschnitt 3.4.2.4 vorgestellten Verfahren beziehen die Simulation zwar in der Regel ein, jedoch liegen deren Schwerpunkte im Bereich der Steuerungssoftwareentwicklung. Um einen Mehrwert für die funktionale Modellierung von Maschinen in frühen Phasen zu erzielen, müssen die Geometrie und der Materialfluss in das Verhaltensmodell mit eingebunden werden. Ebenso ist eine Schnittstelle zum Bediener (Human Machine Interface, kurz HMI) vorzusehen.

Im vorliegenden Kapitel wurde der Stand der Forschung und der Technik beschrieben. Für die identifizierten Problemstellungen soll nun im folgenden Kapitel ein Lösungsansatz präsentiert werden.

## 4 Konzept

### 4.1 Kapitelüberblick

In Kapitel 4 wird das dieser Arbeit zugrunde liegende Konzept vorgestellt. Zunächst werden die zu erfüllenden Anforderungen an die Methode festgehalten (Abschnitt 4.2). Die Basis des Gesamtverfahrens bildet das sogenannte Funktionsmodell, welches als übergeordnete Instanz fungiert und den Entwicklungsprozess frühzeitig unterstützen kann (Abschnitt 4.3). Die für die Funktionsmodellierung notwendigen Eigenschaften der Beschreibungstechnik werden in Abschnitt 4.4 spezifiziert, wobei dies auch die Abbildung von Störszenarien beinhaltet. Durch eine teilautomatisierte Integration von CAD-Daten wird das abstrakte Funktionsmodell erweitert (Abschnitt 4.5) und das VIBN-Modell für den Steuerungstest vollständig aus dem Funktionsmodell abgeleitet (Abschnitt 4.6).

### 4.2 Anforderungen an die Methode

EHRLENSPIEL (2009) sieht Ziele als Sollvorstellungen, die durch die daraus resultierenden Anforderungen konkretisiert werden. Auf der Grundlage der in Kapitel 1 definierten Ziele werden in diesem Abschnitt Anforderungen an die zu entwickelnde Methode abgeleitet, siehe Abbildung 4.1. Um eine frühzeitige Integration der Simulation zu ermöglichen, wird ein disziplinenübergreifendes Modell gefordert, das der Gesamtentwicklung vorangestellt ist. Dieses soll zudem koordinierend den Konstruktionsprozess unterstützen, indem einzelne Arbeitsschritte durch die Möglichkeit zur Abstimmung am Modell parallelisiert werden können. Für diese Zwecke ist zum einen eine geeignete Modellierungstechnik nötig, die abteilungsübergreifend einsetzbar ist und die ferner eine einfache und schnelle Fehlerspezifikation erlaubt. Zum anderen muss der Ansatz für ein entwicklungsbegleitendes Vorgehen unterschiedliche Phasen bzw. Abstraktionsniveaus berücksichtigen. Darüber hinaus sollen im Entwicklungsprozess vorhandene Daten, wie z. B. CAD-Daten, für die teilautomatisierte Generierung der VIBN-Modelle genutzt werden. Dies dient erstens der Reduzierung des Modellierungsaufwandes und zweitens der Verbesserung der Modellqualität.

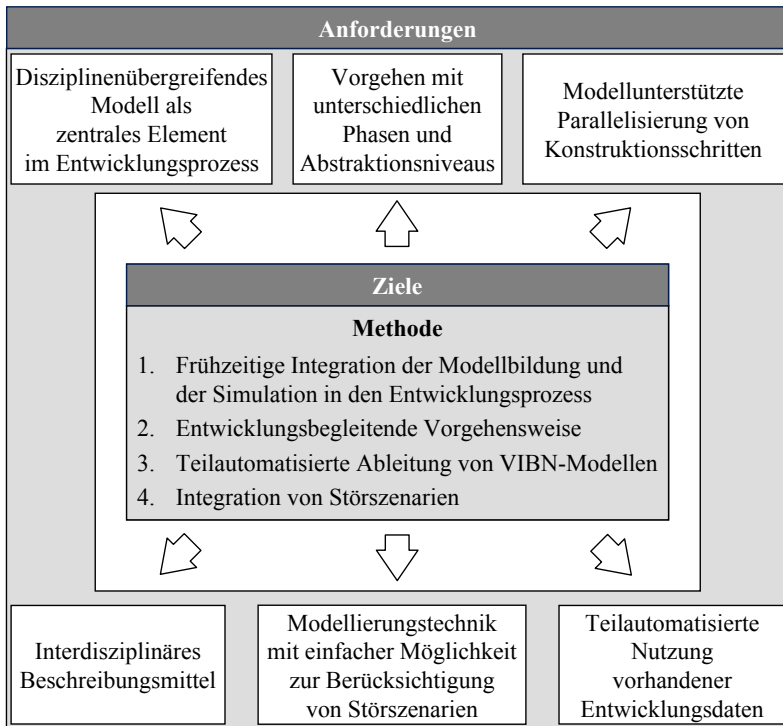


Abbildung 4.1: Ableitung von Anforderungen für die Methode aus den Zielen der vorliegenden Arbeit

Das Konzept der vorliegenden Arbeit, das die oben aufgeführten Anforderungen umsetzt, wird in den sich anschließenden Abschnitten näher erläutert.

## 4.3 Integration der Simulation in den Entwicklungsprozess

### 4.3.1 Modellgetriebene Vorgehensweise

In der Literatur wird zwischen modellbasiertem und modellgetriebenem Vorgehen differenziert. Der Unterschied zwischen beiden besteht nach ALT (2009) darin, dass bei einem modellgetriebenen Ansatz das hinterlegte Modell nicht nur Dokumentationscharakter hat, sondern ausreichend formal ist, um daraus auto-

*matisch Artefakte für den Entwicklungsprozess ableiten zu können. Bei einem modellbasierten Prozess existiert lediglich eine gedankliche Verbindung zwischen Modell und Softwareimplementierung* (STAHL & VÖLTER 2005).

Im Rahmen dieser Arbeit soll eine modellgetriebene Methode zum Einsatz kommen. Auf der Basis des formalen Funktionsmodells wird dabei das VIBN-Modell abgeleitet. Weitere Generierungsmechanismen, z. B. zur automatisierten Erstellung von Teilen des Steuerungscode oder von rudimentären CAD-Modellen, sind prinzipiell umsetzbar, allerdings nicht explizit Gegenstand der vorliegenden Dissertation.

Das im Laufe dieser Arbeit vorgestellte Funktionsmodell umfasst sowohl das Steuerungs- als auch das Maschinenverhalten auf einem simplifizierten Niveau. Durch den hohen Abstraktionsgrad ist das Modell sehr einfach und zeitsparend zu erstellen. Dennoch kann bereits in frühen Phasen die Maschine bzw. das dafür angestrebte Lösungskonzept simuliert und somit besser analysiert werden. Darüber hinaus fungiert das Funktionsmodell als Kommunikations- und Synchronisationsmittel zwischen den verschiedenen Fachabteilungen. Dadurch sollen Entwicklungsschritte parallelisiert werden, da sich gemeinsame Grundlagen vorab im übergreifenden Modell klären lassen. In der Folge können sich kürzere Entwicklungszeiten und, durch die engere Verzahnung der Prozesse, qualitativ verbesserte Gesamtergebnisse einstellen.

In Abbildung 4.2 sind der klassisch sequenzielle und der hier angestrebte, idealisierte Entwicklungsprozess vergleichend dargestellt. Auf der Basis der Kundenanforderungen ist zunächst das Funktionsmodell zu erarbeiten. Die Überführung von Anforderungen in konkrete Funktionsprinzipien wird im Maschinen- und Anlagenbau häufig als *Funktionsbeschreibung* bezeichnet (VDW-RICHTLINIE 2001). Es sind jedoch die in der Praxis anzutreffenden Darstellungsformen üblicherweise nicht formal und deshalb nicht ausführbar. Dadurch müssen Daten später manuell in die Simulationsumgebung übertragen werden, was schnell Fehler nach sich ziehen kann (MAURMAIER 2008). Dem hier vorgestellten Ansatz liegt dagegen ein formales Modell zugrunde, welches zu jedem beliebigen Zeitpunkt simulierbar ist. Die Modellbildung gliedert sich in zwei Phasen, die Grob- und die Feinplanung. Die nachfolgende parallelisierte Konstruktion greift deren Ergebnisse auf und erarbeitet die technischen Details. Die daraus resultierenden CAD-Daten werden dann teilautomatisiert zur Erweiterung des Funktionsmodells genutzt.

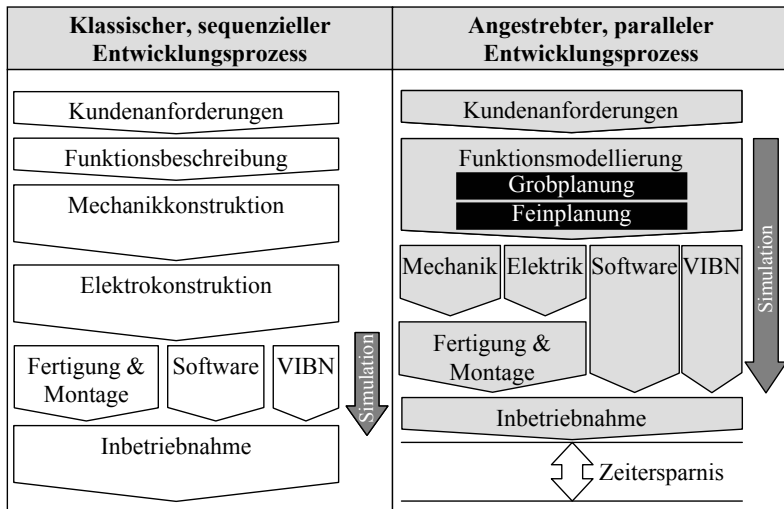


Abbildung 4.2: Parallelisierung des Entwicklungsprozesses durch den Einsatz eines Funktionsmodells zur verbesserten disziplinenübergreifenden Kommunikation und Abstimmung

### 4.3.2 Grobplanung

Den Ausgangspunkt der Entwicklungsaktivitäten bildet die Erstellung des Grobmodells, das lediglich die grundlegenden Funktionen und noch keine technischen Details enthält. So können beispielsweise im Falle einer automatischen Schutztür zu diesem Zeitpunkt nur die Kundenanforderungen von Interesse sein, dass eine Schutztür vorhanden und eventuell Verfahrensgeschwindigkeiten vorgegeben sind. Ob zum Verfahren ein elektrischer Antrieb oder ein Pneumatikzylinder verwendet wird, ist später zu klären. Weiterhin lässt sich eine grobe Maschinengeometrie und -kinematik hinterlegen, die eine Abschätzung der räumlichen Dimensionen sowie eine Kollisionsbetrachtung erlaubt. Die Grobplanungsphase wird idealerweise zusammen mit dem Kunden, dem firmeneigenen Vertrieb sowie einer technisch ausgebildeten Arbeitsgruppe durchgeführt, siehe Abbildung 4.3 links. In Folge dessen können die Anforderungen aus unterschiedlichen Bereichen von Beginn an berücksichtigt und deren Erfüllung durch das Simulationsmodell überprüft werden.

### 4.3.3 Feinplanung

Das Modell der Grobplanungsphase stellt die Eingangsinformation für die Feinplanung dar (Abbildung 4.3). In diesem Schritt erfolgen eine strukturelle und eine verhaltensspezifische Detaillierung der Maschine sowie eine Erweiterung mit technischen Informationen. Für das im vorangegangenen Abschnitt erläuterte Beispiel der Schutztür muss demnach das Wirkprinzip (elektrisch, hydraulisch oder pneumatisch) festgelegt und das Ansteuerungsverhalten definiert werden. Weiterhin ist die Geometrie der projektierten Anlage zu verfeinern, um eine verbesserte Kollisionsbetrachtung zu ermöglichen. Die Erstellung des Modells sollte einer mechatronischen Projektgruppe obliegen, die disziplinenübergreifend die technischen Anforderungen bündelt. Dadurch kann die Softwareentwicklung frühzeitig an der Gestaltung der Maschine beteiligt werden. Die integrierte Vorgehensweise hat das Gesamtoptimum der Maschinen zum Ziel, was sich nicht zwangsläufig aus den Optima der einzelnen Fachrichtungen zusammensetzt.

Das Modell der Feinplanungsphase lässt sich zur Generierung erster VIBN-Modelle heranziehen. Durch die Integration der anfallenden CAD-Daten kann es später sukzessive erweitert werden. Daraus können wiederum detaillierte Modellversionen für den Steuerungstest abgeleitet werden.

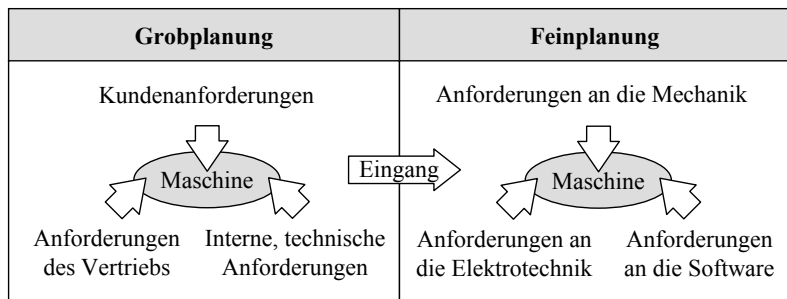


Abbildung 4.3: Berücksichtigung unterschiedlicher Anforderungen in der Grob- und der Feinplanungsphase

### 4.3.4 Unterstützung der Konstruktionsphase

Im Funktionsmodell sind disziplinenübergreifende Informationen enthalten, die eine Parallelisierung der mechanischen, der elektrischen und der softwaretechnischen Entwicklungsstränge prinzipiell gestattet. Üblicherweise dienen die jewei-

ligen Konstruktionsmodelle zur Verständigung und zur Abstimmung zwischen den Fachbereichen. Domänenspezifische Modelle weisen jedoch für derartige Zwecke den Nachteil auf, dass sie einerseits eine große Fülle an Informationen beinhalten, die für die anderen Disziplinen nicht von Interesse sind, andererseits aber wesentliche Aspekte, wie z. B. das Funktionsprinzip sowie das Zusammenwirken von Anlagenteilen, aussparen.

Die Mechanikkonstruktion kann aus dem Funktionsmodell den strukturellen Aufbau der Maschine sowie Geometrieinformationen aufgreifen und weiterverarbeiten. Dabei ist auch eine automatische Generierung von Grobmodellen im MCAD möglich, siehe Abschnitt 6.8. Weiterhin sind im Funktionsmodell generische Akteur- und Sensordaten hinterlegt, die sich zusammen mit den Geschwindigkeits- und den Beschleunigungswerten für eine Antriebsauslegung heranziehen lassen. Ebenso kann die elektrische Konstruktion diese Informationen nutzen. Eine automatische Ableitung von Teilen der Stromlauf- und der Fluidpläne ist denkbar, siehe Abschnitt 6.8. Desgleichen kann die Softwareentwicklung von der vorgestellten Methode profitieren, indem die festgelegten Maschinenabläufe aufgenommen und weiter detailliert werden. Um ein Divergieren der unterschiedlichen Entwicklungsstände in den Abteilungen zu verhindern, muss die Konsistenz zwischen den CAD-Modellen sichergestellt werden. Dies ist zudem im Hinblick auf die teilautomatische Integration der Daten in das Funktionsmodell von Relevanz. Daher wird im Rahmen dieser Arbeit ein Mechanismus zur Konsistenzprüfung entworfen.

## 4.4 Modellierungstechnik

### 4.4.1 Disziplinenübergreifendes Beschreibungsmittel

Die Funktionsmodellierung erfordert ein geeignetes und formales Beschreibungsmittel, mit dem ausführbare Modelle erstellt werden können. Ingenieurwissenschaftliche Grundprinzipien, wie z. B. die Aufgliederung des Gesamtsystems in Teilsysteme (Dekomposition bzw. Komposition im Falle einer Aggregation von Elementen) oder die Abstraktion, gilt es dabei zu berücksichtigen. Zudem soll lediglich ein gemeinsames Beschreibungsmittel für alle am Entwicklungsprozess beteiligten Disziplinen zum Einsatz kommen. Dies impliziert, dass sowohl das Maschinen- als auch das Steuerungsverhalten mit einer Modellierungstechnik definiert werden müssen. Auf Grund dessen ist eine ereignisdiskret-



kontinuierliche Abbildungsform zu wählen. Weiterhin sind die im Stand der Technik geforderten Aspekte, wie z. B. der Materialfluss oder die Geometrie, darin zu integrieren.

Da bisher existierende Modellierungstechniken nicht alle der genannten Kernpunkte enthalten, wird im Rahmen dieser Arbeit eine neue Methode vorgestellt. Es handelt sich um ein grafisch-interaktives Beschreibungsmittel, bei dem die praktischen Erfahrungen mehrerer Anlagen- und Maschinehersteller mit eingeflossen sind. Die gewählten Ausdrucksformen sollen für Entwickler unterschiedlicher Fachbereiche verständlich sein. Die Strukturierung der Maschine beruht auf einem komponentenbasierten Ansatz. Für die Verhaltensspezifikation wird eine Variante der hybriden Automaten verwendet, die über Erweiterungsmechanismen für den Materialfluss und für die räumliche Bewegungen von Maschinenteilen verfügt. Darüber hinaus wird ein Vorgehensmodell für die Funktionsmodellierung erarbeitet, das die Verwendung der Beschreibungstechnik regelt. Dies umfasst Modellierungsrichtlinien für die Grob- und die Feinplanungsphase. Dadurch soll ferner der Import der CAD-Daten unterstützt bzw. der Automatisierungsgrad der Datenübernahme erhöht werden.

#### **4.4.2 Fehlermodellierung**

Die Qualität von Steuerungssoftware hängt in entscheidendem Maße von der Berücksichtigung möglicher Störfälle ab. Häufig werden diese auf Grund des destruktiven Charakters nicht getestet, da im Laufe des Tests Schädigungen an der Maschine verursacht werden können. Im Simulationsmodell sind die Fehlerszenarien zwar gefahrlos analysierbar, erhöhen allerdings den Modellierungsaufwand beträchtlich. Der Ablauf des Gutmodells muss für jeden Einzelfall oder für komplexe Ausfallsituationen spezifisch angepasst werden.

Für die hier verfolgten Zwecke soll eine orthogonale Fehlermodellierungstechnik genutzt werden. Dies bedeutet, dass das Störverhalten unabhängig vom Gutablauf in einem eigenen Modell abgebildet werden kann. Das Grundprinzip basiert auf einer Verfälschung der Ergebnisse des Gutmodells. Beispielsweise wird durch das Fehlermodell die Geschwindigkeit einer Achse zu null gesetzt, wodurch sich ein Verklemmen darstellen lässt. Eine solche Herangehensweise hat den Vorteil, dass zum einen die Beschreibung des Normalfalles nicht modifiziert werden muss. Zum anderen wird die Wiederverwendbarkeit von Modellbausteinen verbessert. Die Fehlerspezifikation ist Teil des Funktionsmodells und somit schon zum Beginn des Entwicklungsprozesses integriert.

### 4.4.3 Simulation hybrider Systeme

Im Rahmen der vorliegenden Arbeit werden Maschinen und Anlagen als hybride Systeme betrachtet. Dies resultiert aus der ereignisdiskreten Arbeitsweise der Steuerung und dem kontinuierlichen Wirken der Maschine. Daher wurde in Abschnitt 4.4.1 ein ereignisdiskret-kontinuierliches Beschreibungsmittel für die Funktionsmodellierung gefordert. Auf diese Art und Weise sollen die Reaktionen des Gesamtsystems mit Hilfe der Simulation bewertbar gestaltet werden.

Eine Ablaufsimulation, wie sie häufig in frühen Entwicklungsphasen für die Anlagenplanung zum Einsatz kommt, wird für die hier verfolgten Zwecke nicht als zielführend erachtet. Dies begründet sich in dem dabei verwendeten zeitdiskreten Ansatz, bei welchem im Gegensatz zur ereignisdiskreten Herangehensweise die Reaktionen des Systems unberücksichtigt bleiben. Die Signale sind in diesem Falle zeitgesteuert und hängen nicht vom Eintreten bestimmter Ereignisse ab.

Durch die Verwendung einer hybriden Modellvorstellung kann die angestrebte Funktionslogik aussagekräftig in der Simulation getestet werden. Als Beispiel sei das Auslösen einer Lichtschranke genannt, die einen kontinuierlichen Anlaufvorgang eines Förderbandes bedingt. Dabei wird der Motor nur dann gestartet, wenn die Lichtschranke durch ein Materialobjekt passiert wurde und nicht auf Grund einer zeitlichen Ersatzlogik.

## 4.5 Einbindung vorhandener Modelldaten

### 4.5.1 Funktionsmodell und CAD-Daten

Die teilautomatisierte Integration vorhandener CAD-Daten kann für die VIBN von großem Nutzen sein. Die Ableitung von VIBN-Modellen allein aus CAD-Informationen, wie in ersten Ansätzen praktiziert, greift zum einen im Entwicklungsprozess zu spät und vernachlässigt zum anderen bis dato wesentliche Charakteristika der Maschine. Daher wird an dieser Stelle eine Methode vorgeschlagen, deren Grundlage das Funktionsmodell bildet, welches mit CAD-Daten angereichert wird, siehe Abbildung 4.4. Aus dem Modell der Feinplanungsphase lassen sich bereits erste VIBN-Modelle ableiten. Die im Laufe des Konstruktionsprozesses anfallenden Details aus dem CAD werden teilautomatisiert in das Feinplanungsmodell eingebunden. Auf der Basis des erweiterten Funktionsmodells können im Anschluss VIBN-Modelle mit höherer Detailtreue generiert wer-

den. Mit der beschriebenen Herangehensweise ist ein früher und entwicklungsbegleitender Simulationsansatz möglich.

Durch die formale Beschreibungstechnik des Funktionsmodells können die im CAD-System fehlenden Informationen, wie z. B. der Materialfluss, ergänzt werden. Informelle Eingaben durch den Entwickler, wie teilweise im Stand der Technik erläutert, sind nicht erforderlich.

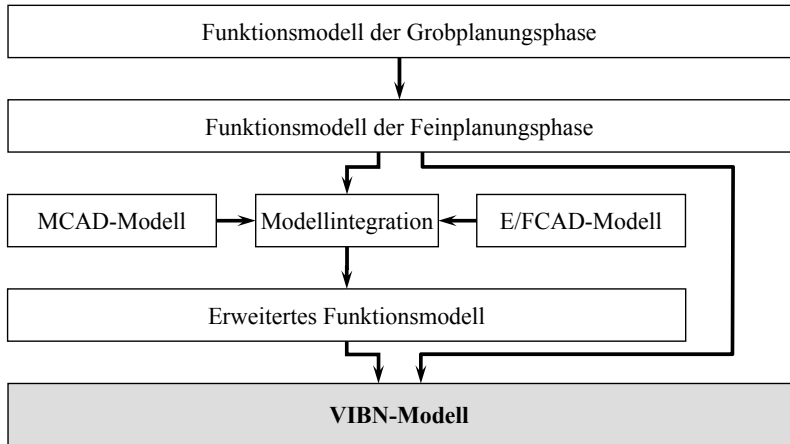


Abbildung 4.4: Nutzung von CAD-Daten zur teilautomatisierten Erweiterung der VIBN-Modelle

#### 4.5.2 Methode für die Datenintegration

Hinsichtlich der teilautomatisierten Einbeziehung der Modelldaten aus dem CAD soll eine Unabhängigkeit vom eingesetzten Werkzeug gewährleistet werden. Dadurch ist die Methode auf eine beliebige CAD-Softwarelandschaft anwendbar, solange sich die benötigten Informationen über geeignete Schnittstellen extrahieren lassen.

Weiterhin ist eine eindeutige Zuordnung zwischen den Elementen im Funktionsmodell und den Objekten in den CAD-Systemen notwendig. Eine Herausforderung stellen dabei die unterschiedlichen hierarchischen Strukturen im MCAD und im E/FCAD dar. Deshalb wird in das Funktionsmodell ein Mechanismus zur Beschreibung der Querbeziehungen installiert. Dieser wird zum einen für die Integration der Konstruktionsdaten benötigt, zum anderen lässt sich hierdurch

eine Konsistenzprüfung und somit ein Vergleich der jeweiligen Entwicklungsbereiche verwirklichen. Die Integration der MCAD- und der E/FCAD-Daten in das Funktionsmodell kann unabhängig voneinander geschehen, da sich die einzubindenden Informationen auf verschiedene Elemente der Modellierungstechnik auswirken. Unterstützt wird der Vorgang durch Modellierungsrichtlinien, welche Teil des Vorgehensmodells sind.

Um den Automatisierungsgrad für die Anreicherung mit CAD-Details zu erhöhen, werden außerdem Konstruktionsrichtlinien erlassen. Im Falle, dass diese von Beginn an Beachtung finden, ergibt sich nur ein geringer Mehraufwand. Beispielsweise erfordert die Einhaltung einer gewissen Modellstruktur im MCAD bei frühzeitiger Berücksichtigung deutlich weniger Ressourcen, als wenn dies im Nachhinein am fertigen Modell geschehen muss.

## **4.6 Automatisierte Ableitung von Simulationsmodellen**

### **4.6.1 Prinzip der Modellgenerierung**

Das VIBN-Modell wird vollständig aus dem übergeordneten Funktionsmodell abgeleitet. Da kommerzielle Simulationswerkzeuge nicht alle Bestandteile der hier entwickelten Modellierungstechnik unterstützen und darüber hinaus nur einen beschränkten Zugang über Schnittstellen für den Generierungsprozess zur Verfügung stellen, wurde im Rahmen der vorliegenden Arbeit ein eigener Simulator erstellt. Das Funktionsmodell wird dazu in ein C++-Programm überführt, welches sich mit Hilfe des prototypischen Simulationswerkzeuges für HiL- oder SiL-Versuchsläufe nutzen lässt. Dadurch ist gewährleistet, dass die wesentlichen Kernpunkte aus dem Funktionsmodell verlustfrei übernommen werden können.

Des Weiteren stimmen die Schnittstellen im Funktionsmodell nicht unbedingt mit den realen Steuerungsein- und -ausgängen überein. Aus diesem Grund müssen für die Generierung des VIBN-Modells zusätzliche Informationen eingebunden werden, die diese Beziehungen beinhalten (Abbildung 4.5). Dafür wird im Funktionsmodell eine weitere Beschreibungsform eingeführt, die während des Transformationsvorgangs mit einbezogen wird. Ferner dürfen für die VIBN nur die maschinenbeschreibenden Teile des Funktionsmodells ausgeleitet werden, da ansonsten eine Kopplung mit der Steuerung nicht möglich wäre.

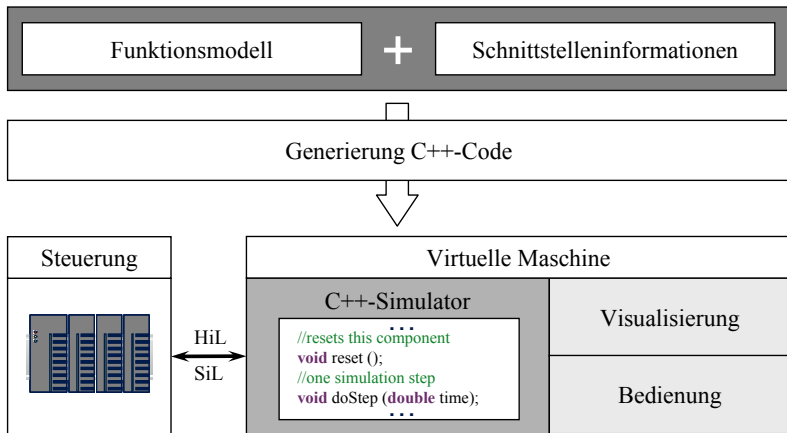


Abbildung 4.5: Prinzip der automatisierten VIBN-Modellgenerierung auf der Basis des Funktionsmodells

#### 4.6.2 Berücksichtigung von Änderungen

Da das VIBN-Simulationsmodell lediglich als C++-Programm existiert, ist eine Verständlichkeit des generierten Codes nur bedingt gegeben. Daher ist es nicht zweckdienlich, Änderungen direkt im Modellcode vorzunehmen. Bei Modifikationsbedarf sind Anpassungen auf der Ebene des Funktionsmodells zu bewerkstelligen. Anschließend muss das VIBN-Modell neu generiert werden. Dieses Vorgehen begründet sich zum einen dadurch, dass Änderungen im Funktionsmodell wesentlich schneller verwirklicht und simuliert werden können. Zum anderen hat das Funktionsmodell den Charakter einer entwicklungsübergreifenden Instanz, weshalb Umgestaltungen oder Verbesserungen mit den anderen Disziplinen abzustimmen sind.

### 4.7 Zusammenfassung

Das wesentliche Konzept und die Inhalte dieser Arbeit sind in Abbildung 4.6 zusammenfassend veranschaulicht und gliedern sich in insgesamt vier Blöcke. Die Modellierungstechnik sowie das zugehörige Vorgehensmodell werden in Kapitel 5 detailliert erläutert. Die Erweiterung des Funktionsmodells mit CAD-Daten und die Generierung des VIBN-Modells sind Teil des sechsten Kapitels.

Kapitel 5	
<b>Modellierungstechnik</b> <ul style="list-style-type: none"><li>• Hybride Beschreibungsform</li><li>• Grafisch und intuitiv</li><li>• Berücksichtigung der Geometrie</li><li>• Abbildung des Materialflusses</li><li>• Schnittstelle zum Benutzer</li></ul>	<b>Vorgehensmodell</b> <ul style="list-style-type: none"><li>• Entwicklungsbegleitendes Vorgehen</li><li>• Modellierungsrichtlinien</li><li>• Möglichkeit zur Parallelisierung von Prozessen</li></ul>
Kapitel 6	
<b>Integration von CAD-Daten</b> <ul style="list-style-type: none"><li>• Werkzeugunabhängige Erweiterung des Funktionsmodells</li><li>• Konstruktionsrichtlinien</li><li>• Konsistenzprüfung</li></ul>	<b>Ableitung des VIBN-Modells</b> <ul style="list-style-type: none"><li>• Vollständige Generierung des VIBN-Modells</li><li>• VIBN-Simulator</li><li>• HiL- und SiL-Kopplung</li></ul>

Abbildung 4.6: Konzeptbausteine und Inhalte der vorliegenden Arbeit

## **5 Funktionsmodellierung von Maschinen und Anlagen**

### **5.1 Kapitelüberblick**

Das vorliegende Kapitel ist der Funktionsmodellierung von Maschinen und Anlagen gewidmet. Aus Verständnisgründen soll zunächst auf die Modellierungstechnik eingegangen werden (Abschnitte 5.3 und 5.4), da deren Kenntnis für das gedankliche Nachvollziehen des Vorgehensmodells (Abschnitt 5.5) notwendig ist. Vorab werden in Abschnitt 5.2 die Anforderungen an eine geeignete Modellierungstechnik gesammelt und mit bestehenden Beschreibungsformen verglichen.

### **5.2 Beschreibungsmittel und Modellierungstechniken**

#### **5.2.1 Anforderungsanalyse**

Die Modellierung von Maschinen und Anlagen in einem Entwicklungsprozess-übergreifenden Informationsmodell erfordert eine adäquate Modellierungstechnik. Deren notwendige Eigenschaften sind nachfolgend in Form einer Anforderungsliste zusammengefasst, die als zu erfüllende Randbedingungen in die hier vorliegende Arbeit einfließen:

- Abbildung von Struktur und Hierarchie

Eine Strukturierung des Systems sowie eine beliebig tiefe Hierarchisierung müssen möglich sein. Durch Kompositions- bzw. Dekompositionsprozesse kann das Systemverständnis erhöht und die Komplexität beherrschbar gestaltet werden.

- Abbildung des Steuerungs- und des Maschinenverhaltens

Für eine Gesamtsimulation in frühen Entwicklungsphasen muss neben dem Maschinenverhalten ebenso die Steuerungslogik berücksichtigt werden. Um den Einarbeitungsaufwand für den Entwickler zu reduzieren, soll in der hier beschriebenen Methode lediglich ein gemeinsames Beschreibungsmittel Verwendung finden.

- Abstraktionsfähigkeit

Der Wirkungsbereich des verfolgten Ansatzes erstreckt sich vom Beginn des Entwicklungsprozesses bis hin zur VIBN. Daher ist durch angepasste Abstraktionsniveaus die Einsetzbarkeit zu gewährleisten. Mit Hilfe der Modellierungstechnik müssen folglich unterschiedliche Detaillierungsstufen darstellbar sein.

- Berücksichtigung der Geometrie

Die tatsächliche Geometrie einer Maschine spielt für die Funktionalität eine wichtige Rolle. Daher soll die Maschinengeometrie neben der Visualisierung als Bestandteil der Verhaltenssimulation in die Anforderungsliste mit einfließen.

- Abbildung des Materialflusses

Der Materialfluss macht in Automatisierungssystemen einen erheblichen Anteil des Modellierungsaufwandes aus. Die Geometrie des Stückgutes ist bei der Modellbildung zu berücksichtigen, was eine verbesserte Kollisionsbetrachtung erlaubt.

- Darstellung von Störszenarien

Im Rahmen dieser Arbeit soll eine orthogonale Modellierungstechnik für Störfälle zum Einsatz kommen. Dies bedeutet, dass das Fehlermodell unabhängig vom Modell des Gutablaufs und mit diesem kombinierbar ist. Dadurch kann die Fehlermodellierung nach der Festlegung des fehlerfreien Verlaufs erfolgen, ohne das Gutmodell ändern zu müssen.

- Bedienerschnittstelle

Da im Funktionsmodell auch die Steuerung modelliert ist, muss über ein virtuelles HMI eine Einflussnahme möglich sein. Des Weiteren sind Vorgänge, wie z. B. das Einlegen und die Entnahme von Material, abzubilden. Hierfür sind entsprechende Eingriffsmechanismen vorzusehen.

### 5.2.2 Bewertung

In Kapitel 2 wurden in Abschnitt 2.5.3 eine Reihe von Modellierungstechniken erörtert, die zur Beschreibung technischer Systeme Verwendung finden. Für die vorliegende Arbeit sollen die genannten Modellierungstechniken den in Ab-



schnitt 5.2.1 definierten Anforderungen gegenübergestellt werden, siehe Tabelle 5.1. Da die ausschließlich ereignisdiskreten oder kontinuierlichen Verfahren auf Grund der hier angewendeten Betrachtungsweise einer Maschine als hybrides System generell nicht geeignet erscheinen, sind diese in der Bewertung ausgenommen. Wegen der unzureichenden Strukturierungsmechanismen weisen die hybriden Varianten der Automaten und der Petri-Netze Schwächen hinsichtlich der Gliederung von Systemen auf. Die hybriden Statecharts profitieren diesbezüglich von ihrem hierarchischen Konzept. Die übrigen zitierten Beschreibungstechniken stellen für diese Zwecke geeignete Mittel, wie z. B. die Zerlegung in Komponenten, zur Verfügung. Allerdings sind deren Modellierungsmöglichkeiten in Bezug auf das hybride Gesamtverhalten von Maschinen und Anlagen eingeschränkt. Die Ausnahme bildet dabei Modelica, was allerdings mit einem höheren Detaillierungsgrad erkaufte werden muss. Informationen, wie z. B. Reibungskoeffizienten, stehen zu Beginn der Entwicklung noch nicht zur Verfügung bzw. sind da noch nicht von Interesse. Für frühe Phasen ist eine gewisse Abstraktionsfähigkeit erforderlich, um die Komplexität zu reduzieren. Ebenso erschwert die sehr große Vielfalt an Modellierungskonstrukten der UML deren Verwendung (FRIEDRICH 2009), was einer ihrer Hauptkritikpunkte ist (WEILKIENS 2006).

Die Einbeziehung der Geometrie in das Verhaltensmodell, der Materialfluss, die Modellierung von Störszenarien und die Integration von Bedienerschnittstellen bleiben bei allen betrachteten Beschreibungsformen entweder unberücksichtigt oder können nur unbefriedigend und umständlich realisiert werden.

<b>Legende:</b> ● Anforderung erfüllt ◐ Anforderung teilweise erfüllt ○ Anforderung nicht erfüllt	Hybride Automaten	Hybride Petri-Netze	Hybride Statecharts	UML	SysML	ROOM/UML-RT	Focus	Modelica
Abbildung von Struktur und Hierarchie	○	○	●	●	●	●	●	●
Abbildung von Steuerungs- und Maschinenverhalten	●	●	●	◐	◐	◐	◐	●
Abstraktionsmöglichkeit	●	●	●	●	●	●	●	◐
Geometrie, Kollisionserkennung	○	○	○	○	○	○	○	●
Realistische/detaillierte Einbindung des Materialflusses	○	○	○	○	○	○	○	◐
Spezifikation von Störszenarien	◐	◐	◐	◐	◐	◐	◐	◐
Bedienerschnittstelle	○	○	○	○	○	○	○	◐

*Tabelle 5.1: Bewertung von Modellierungstechniken anhand relevanter Anforderungen hinsichtlich der angestrebten Methode*

Die untersuchten Modellierungsansätze erfüllen die gestellten Anforderungen nur teilweise. Eine Beschreibungstechnik für materialflussintensive Maschinen und Anlagen, die alle der hier formulierten Belange berücksichtigt, existiert derzeit nicht. Daher wird in Abschnitt 5.3 eine neue Modellierungstechnik präsentiert, die den oben genannten Gesichtspunkten gerecht werden soll.

### 5.3 Methode für die Funktionsmodellierung

#### 5.3.1 Allgemeines

Der Begriff des Funktionsmodells ist in der Literatur sehr weit gefasst. Zur Darstellung solcher Modelle können unterschiedliche Hilfsmittel herangezogen werden. Die einfachste Form ist die Aufzählung der Funktionen in einer Liste (PONN & LINDEMANN 2008). Komplexere Verfahren stellen die funktionalen Zusammenhänge formal und modellbasiert dar, wodurch sie besser zur Analyse und zur Simulation geeignet sind. Der gemeinsame Grundgedanke besteht darin, ein technisches System frühzeitig auf einem abstrakten Niveau zu beschreiben.

In den sich anschließenden Abschnitten wird der Ansatz zur Funktionsmodellierung dieser Arbeit vorgestellt. Die modellierungstechnischen Grundlagen wurden im von der AiF geförderten Projektvorhaben AUTOVIBN (2010) erarbeitet und greifen Teile der FOCUS-Theorie auf (BROY & STØLEN 2001), siehe Abschnitt 2.5.3.5. Das Werkzeug AutoFOCUS wurde dabei wie folgt zum Prototyp *STEM* (*Spatio-Temporal Engineering Models*) adaptiert bzw. ergänzt:

- Erweitertes hybrides Automatenmodell
- Orthogonales Modell zur Fehlerspezifikation
- Geometrie, Kinematik und Kollisionserkennung
- Materialquellen und -senken, Materialfluss
- Bedienerschnittstelle
- Bibliotheksmechanismus mit parametrischer Funktion
- Modell der Querbeziehungen zur Konstruktionsebene
- Konsistenzprüfung zwischen Funktionsmodell und CAD-Modellen

### 5.3.2 Bestandteile des Metamodells

Dieser Abschnitt behandelt die wichtigsten Teile des Metamodells. In den sich anschließenden Abschnitten werden dann Einzelaspekte davon aufgegriffen und umfassender erläutert. Hierfür werden mitunter UML-Klassendiagramme eingesetzt, wobei diese der Übersicht wegen nicht alle Details enthalten. Da die Funktionsmodellierung in englischer Sprache verfasst ist, werden die Namen der einzelnen Modellelemente im weiteren Verlauf übersetzt bzw. eingedeutscht. Das zentrale Element der Beschreibungstechnik ist die Komponente (engl. *component*), wobei diese als logische Einheit zu betrachten ist und das Verhalten nach außen kapselt. Damit können sowohl Software- als auch Hardwarefunktionen der Maschine abgebildet werden.

Die statischen Anteile einer Komponente sind in Abbildung 5.1 dargestellt. *Ports* verkörpern die Kommunikationsschnittstellen. Sie können eingehender oder ausgehender Art sein. Die geometrischen Ausprägungen der Bauteile einer Maschine werden durch *Parts* repräsentiert. *Detektoren* (engl. *detector*) besitzen eine räumliche Gestalt und dienen der Erkennung und Weiterverarbeitung von Kollisionen, womit sich beispielsweise Einflussbereiche von Sensoren modellieren lassen. Mit Hilfe der *Bindings* werden Materialflussoperationen definiert. Sie sind ähnlich aufgebaut wie die Detektoren, können allerdings Material fest an sich binden bzw. erlauben eine Relativbewegung in einem geometrisch festgelegten Bereich. Achsen und Verfahrenskurven werden durch die sogenannten *Mover* beschrieben.

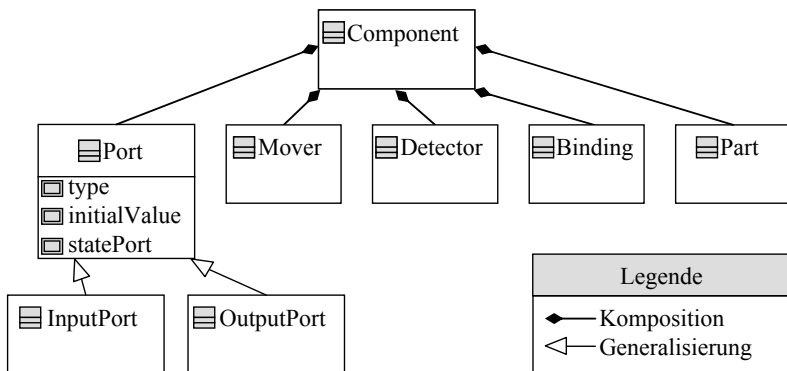


Abbildung 5.1: Definition der statischen Anteile einer Komponente durch die syntaktische Schnittstelle (AUTOVIBN 2010)

Die dynamischen Anteile einer Komponente zeigt Abbildung 5.2. Die Ports sind über Kanäle verbunden, über welche Nachrichten ausgetauscht werden. Nichthierarchische Komponenten, die folglich keine weiteren Komponenten enthalten, verfügen über genau eine Verhaltensspezifikation (*ComponentBehaviourSpec*). Im Regelfall wird das Verhalten mit den hybriden Automaten (*AutomatonSpec*) beschrieben. Alternativ können Bedienpanels (*OperatorPanel*) oder durch Funktionen definierte Verfahrprofile (*CurveSpec*) zum Einsatz kommen. Zudem ist eine orthogonale Verhaltenserweiterung möglich, mit der Fehlerszenarien (*ErrorModeSpec*) abgebildet werden können.

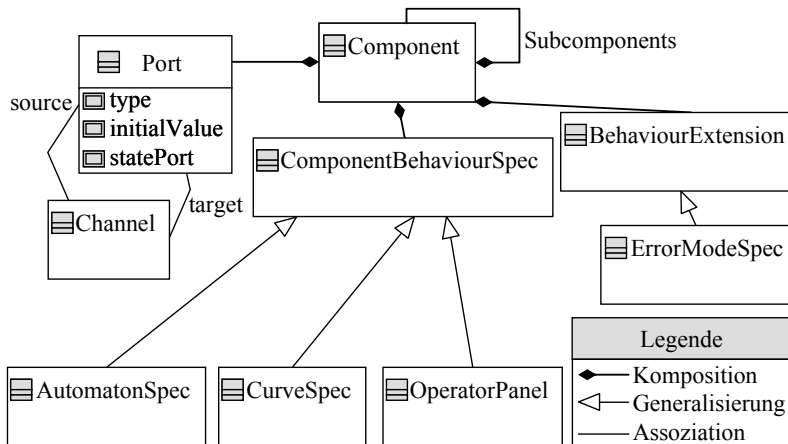


Abbildung 5.2: Definition der dynamischen Anteile einer Komponente durch die semantische Schnittstelle (AUTOVIBN 2010)

Im Gegensatz zur strikten Trennung von Verhaltens-, Geometrie- und Materialflussmodell, wie sie z. B. bei klassischen VIBN-Ansätzen üblich ist, wird hier die Komponente als integrierendes Element verwendet. Dadurch kann eine aufwendige Kopplung unterschiedlicher Simulationsprogramme, wie ansonsten meist erforderlich, vermieden werden.

### 5.3.3 Modellierung der Struktur

Zur Gliederung von Systemen dienen die Komponenten. Hiermit können sowohl einzelne Bestandteile als auch das Gesamtsystem beschrieben werden. Eine Komponente kann wiederum weitere Komponenten enthalten, wodurch sich

komplexe, hierarchische Strukturen verwirklichen lassen. Mittels Dekompositions- bzw. Aggregationsprozessen verbessert sich die Übersichtlichkeit. Dies trägt somit zum besseren Verständnis bei, siehe Abbildung 5.3.

Die eingehenden und die ausgehenden Ports einer Komponente verfügen über einen Namen und einen Datentyp. Primitive Datentypen, wie z. B. *Boolean* oder *Double*, sind gegeben. Durch den Einsatz von komplexen, benutzerdefinierten Typen lässt sich die Anzahl der Schnittstellenpunkte reduzieren. Die Verbindung von Ports ist mittels Kanälen realisiert, über welche Nachrichten zwischen Komponenten ausgetauscht werden können, siehe Abbildung 5.3. Der Versand ist ereignisbasiert. Für die Modellierung ist es allerdings teilweise erforderlich, dass Signale konstant anliegen. Daher können Ports wahlweise zustandsbasiert sein, wodurch der aktuelle Wert zyklisch versendet wird.

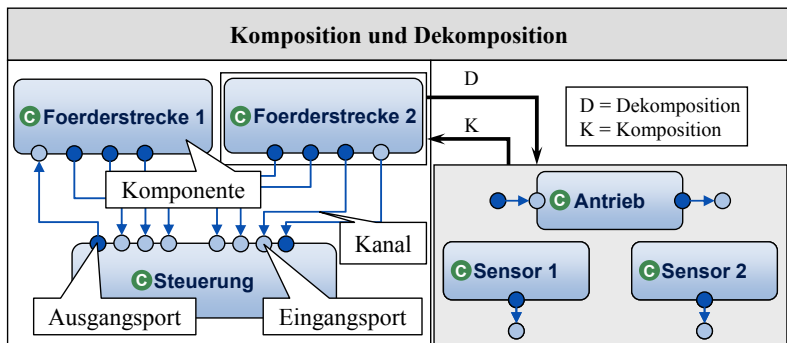


Abbildung 5.3: Die Komponente als Strukturierungsmittel der Modellierungstechnik zur Beschreibung von Systemen

### 5.3.4 Spezifikation des Verhaltens

#### 5.3.4.1 Modell für den Gutablauf

Die Abbildung des Steuerungs- und des Maschinenverhaltens stellt einen Schwerpunkt für die Funktionsmodellierung dar und beansprucht zudem einen Großteil der zeitlichen Ressourcen. Daher ist eine geeignete Modellierungstechnik an dieser Stelle von besonderer Bedeutung. Demzufolge bieten sich vor allem grafische Beschreibungsmittel an, da diese übersichtlicher gestaltet und somit verständlicher für den Entwickler sind. In der VDW-RICHTLINIE (2001) zur

Funktionsbeschreibung werden unter anderem die Zustandsgraphen als adäquate Darstellungsform definiert. Ebenso schlägt die VDI/VDE-RICHTLINIE 2422 (1994) diese zur Modellierung von Steuerungsaufgaben vor. Die Forschungsarbeiten von TOMASZUNAS (1998), siehe Abschnitt 3.3.1, haben verdeutlicht, dass auch das Maschinenverhalten mit Hilfe von Zustandsmaschinen definiert werden kann. Dadurch soll die Verhaltensmodellierung wesentlich intuitiver und effizienter durchzuführen sein.

Für die Verhaltensspezifikation im Rahmen dieser Arbeit kommen hybride Automaten zum Einsatz. Häufig wird für derartige Zwecke eine Fülle an Modellierungskonstrukten vorgeschlagen, wie es z. B. im Hinblick auf die UML der Fall ist. Dies impliziert wiederum, dass der Entwickler über einen ausreichenden Kenntnisstand für alle erforderlichen Beschreibungsmittel verfügen muss. Der hier vorgestellte Ansatz zielt bewusst darauf ab, die Elemente der Modellierungstechnik zu beschränken, um schon in frühen Entwicklungsphasen einfach und zeitnah Modelle erstellen zu können. Das Metamodell der verwendeten Zustandsmaschine ist in Abbildung 5.4 illustriert. Durch die Zustände (engl. *states*) und die Belegung der Variablen ist das abgebildete System eindeutig charakterisiert. Für die Automaten lassen sich ferner kontinuierliche Variablen definieren, deren Änderungen durch Differenzialgleichungen in jedem einzelnen Zustand beschrieben werden können. Die Zustandswechsel erfolgen über Transitionen.

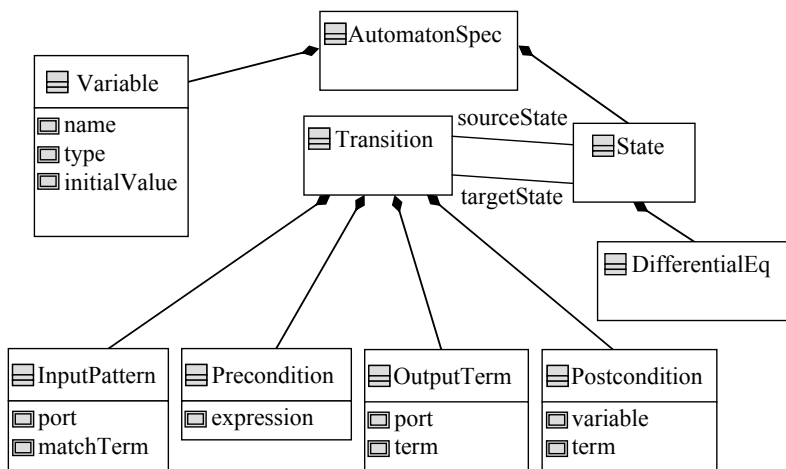


Abbildung 5.4: Metamodell der in dieser Arbeit verwendeten Zustandsmaschine (AUTOVIBN 2010)

Der Mechanismus zum Ausführen von Transitionen basiert auf folgenden Elementen:

- Eingabemuster (*InputPatterns*)
- Vorbedingungen (*Preconditions*)
- Ausgaben (*OutputTerms*)
- Nachbedingungen (*Postconditions*)

Entsprechen die an den Eingangsports anliegenden Nachrichten den in den Eingabemustern definierten logischen Ausdrücken und sind die formulierten Vorbedingungen, z. B. eine Ungleichung, erfüllt, so wird der Zustand gewechselt. Dabei werden entsprechende Ausgaben generiert und in den Nachbedingungen die Zustandsvariablen des Automaten angepasst.

Wie im vorangegangenen Abschnitt beschrieben, ist der Automat Teil einer Komponente. Durch die daraus resultierende Kapselung lassen sich parallele Abläufe erzeugen, wie es z. B. für ROOM auf eine ähnliche Art und Weise umgesetzt ist. Zudem verfügt der verwendete hybride Automat über Erweiterungen, um Kollisionen räumlicher Objekte zu erfassen (Abschnitt 5.3.5), Bewegungen von Maschinenteilen zu berechnen (Abschnitt 5.3.6) und mit dem Materialfluss (Abschnitt 5.3.7) zu interagieren.

Anhand der in Abbildung 5.5 illustrierten Zustandsmaschine soll die Modellierungstechnik exemplarisch erläutert werden. Es handelt sich hierbei um das stark abstrahierte Modell eines Antriebes, dessen technische Ausprägungen noch nicht näher spezifiziert sind. Die Komponente *Antrieb* verfügt über einen Eingangsport *Freigabe* vom Typ *Boolean* und einen Ausgangsport *Position* vom Typ *Double*, der aktuelle Positionswerte zur Weiterverarbeitung liefert. Auf den Port *Fehler* wird im Rahmen der Fehlerspezifikation in Abschnitt 5.3.4.2 eingegangen. Die Übersetzungsverhältnisse des Antriebsstranges sind bereits berücksichtigt und eingerechnet. Weiterhin wurden die Variablen *\_pos*, *\_v* und *\_vsoll* angelegt (Variablen sind durch einen vorangestellten Unterstrich gekennzeichnet), die alle den Typ *Double* aufweisen. Die beiden Ersten wurden initial zu null gesetzt und *\_vsoll* mit dem Wert für die gewünschte Sollgeschwindigkeit versehen. Die Zustände werden durch Ovale dargestellt. Zu Beginn befindet sich der Automat im Zustand *Stopp*, was durch einen Punkt symbolisiert ist. In diesem Zustand wird die zeitliche Ableitung der Variablen *\_pos* gleich der Geschwindigkeit *\_v* und somit zu null gesetzt. Sobald auf dem Eingangsport *Freigabe* der Wert *true* (Ein-

gabemuster) anliegt und die Vorbedingung  $\_vsoll > 0$  zutrifft, wird die Transition *Starten* geschaltet. Die Notation für die Ein- und die Ausgaben ist an die Prozessalgebra *Communicating Sequential Processes (CSP)* von HOARE (1985) angelehnt. Zur Abfrage der Eingänge wird ein Fragezeichen und zum Schreiben der Ausgänge ein Ausrufezeichen verwendet. In der Nachbedingung wird festgelegt, dass die Variable  $\_v$  den Wert von  $\_vsoll$  annimmt. Auf den Ausgangsport *Position* wird als Ausgabe der aktuelle Wert der Variable  $\_pos$  geschrieben. Das System befindet sich nun im Zustand *Verfahren*, wobei die zeitliche Ableitung von  $\_pos$  gleich dem Wert von  $\_v$  gesetzt wird. Der Antrieb würde nun gemäß der Modellvorstellung eine Verfahrbewegung mit der vorgegebenen Sollgeschwindigkeit ausführen. Bei einer Negierung der Verfahrfreigabe wird über die Transition *Anhalten* in den Zustand *Stopp* gewechselt. Hierbei wird zunächst der Port für die Positionsmeldung beschrieben und in der Nachbedingung der Variablen  $\_v$  der Wert null zugewiesen. Somit ist die Ableitung der Größe  $\_pos$  gleich null, wodurch der Antrieb steht. Die Selbsttransitionen der beiden Zustände dienen zum Versenden des aktuellen Positionswertes, da Ein- und Ausgabeänderungen nur mit Hilfe der Zustandsübergänge möglich sind.

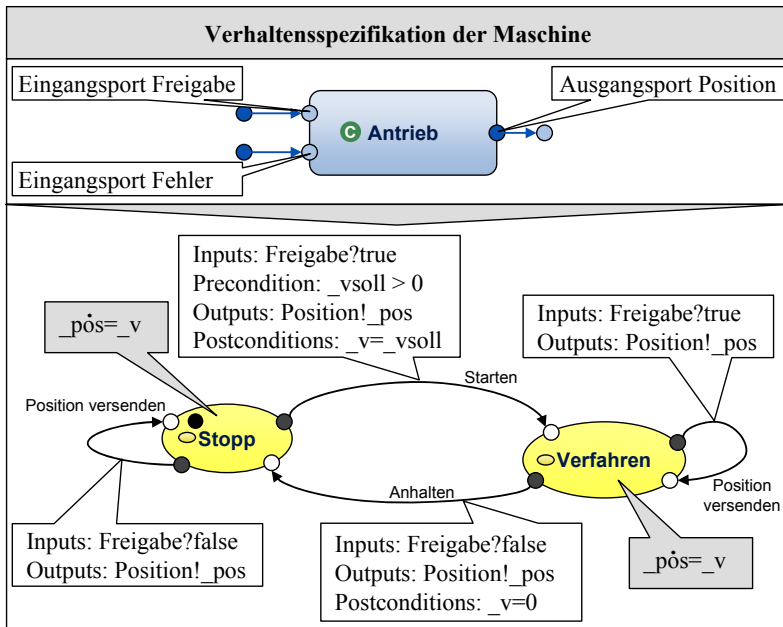


Abbildung 5.5: Hybrider Automat zur Verhaltensmodellierung eines Antriebs



Die Hinterlegung von Differenzialgleichungen in den Zuständen ist optional. Dadurch lassen sich mit dieser hybriden Modellierungstechnik ebenfalls rein ereignisdiskrete Systeme abbilden, wie es z. B. für Steuerungsfunktionen notwendig ist. Abbildung 5.6 zeigt das Steuerungsmodell einer Anlage, das aus zwei Förderbändern besteht. Zufällig verteiltes Stückgut auf dem ersten Förderband kann unter Zuhilfenahme zweier Lichtschranken und einer Wegmesseinrichtung mit äquidistanten Abständen auf dem zweiten Förderband gruppiert werden.

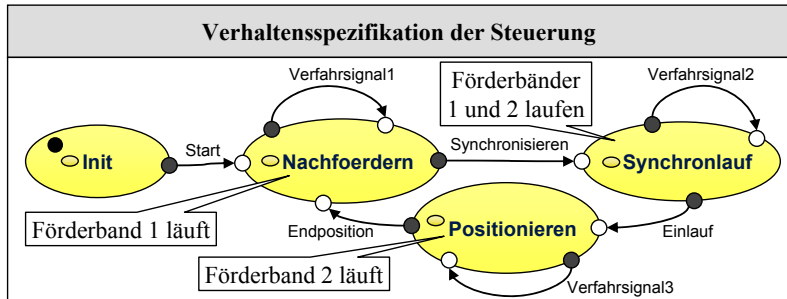


Abbildung 5.6: Zustandsmaschine einer Förderbandsteuerung mit zwei Transportstrecken

#### 5.3.4.2 Integration von Störszenarien und Fehlerfällen

Die steigende Komplexität von Produktionssystemen korreliert nach MILBERG & EBNER (1994) mit einer Reduzierung der Verfügbarkeit. REINHART ET AL. (2002) nennen in diesem Zusammenhang die zunehmende Störanfälligkeit und fordern eine frühzeitige Berücksichtigung von Fehlerfällen im Entwicklungsprozess. Daher wird hier die Integration von Störszenarien im Zuge der Funktionsmodellierung angegangen. Diese ist somit von Beginn an Bestandteil des Gesamtvorgehens. Die Einbindung von Fehlerfällen in die Simulation ist ein zeitaufwendiger Prozess, da das Modell des Gutablaufs dazu normalerweise geändert werden muss. Des Weiteren ist für jeden zusätzlichen Störfall das Modell entsprechend zu erweitern, was dieses wiederum sehr komplex und schwer verständlich gestaltet. Aus den genannten Gründen soll eine Modellierungstechnik für Störszenarien zum Einsatz kommen, die auf dem Modell des Gutablaufs aufbaut, dieses jedoch nicht strukturell verändert. Dabei finden sogenannte Filter Verwendung, welche die Ein- und Ausgangsports sowie die Ergebnisse der Differenzialgleichungen in den Zuständen der hybriden Automaten manipulieren. Als Beschreibungsform der Fehlerfälle wird ebenfalls auf endliche Automaten zurückgegriffen.

Am Beispiel des in Abschnitt 5.3.4.1 erläuterten Modells eines Antriebes soll nun eine Fehlerspezifikation vorgenommen werden, siehe Abbildung 5.7. Das Modell des Störfalls wird wie der hybride Automat des Verhaltensmodells einer Komponente zugewiesen. Somit ist das Normal- vom Fehlverhalten abgetrennt, wobei das Fehlermodell die Werte des Gutmodells verfälscht. Wie Abbildung 5.7 verdeutlicht, verfügt der Automat des Fehlermodells über die Zustände *Normal*, *Signalverfälschung*, *Positionsfehler* und *Klemmen*. Fehlerzustände sind im Modell durch Ovale dargestellt, der Initialzustand ist durch einen Punkt markiert. Die Transitionen werden in Abhängigkeit des am Eingangsport *Fehler* anliegenden Signals vorgenommen. Dazu wurde ein Datentyp definiert, der sich wie folgt zusammensetzt:  $\{\text{Signalfehler}, \text{Positionsfehler}, \text{Geschwindigkeitsfehler}\}$ . Sobald einer der drei Fehler anliegt, wird in den entsprechenden Zustand gewechselt.

Die Signalverfälschung ist über drei Filtermechanismen realisiert, die in jedem Fehlerzustand zur Verfügung stehen. So wird beispielsweise im Zustand *Signalverfälschung* der am Eingangsport *Freigabe* anliegende Wert über den *Eingabefilter* auf *false* gesetzt. Dadurch wird ein Anfahren des Antriebes trotz Verfahrfreigabe verhindert. Der Ausgangsport *Position* wird im Zustand *Positionsverfälschung* unter Anwendung des *Ausgabefilters* mit einem festen Wert versehen. Somit können falsche Positionswerte übermittelt werden. Ein direkter Zugriff auf die verhaltensbeschreibenden Differenzialgleichungen wird im Zustand *Klemmen* vorgenommen. Der Ableitung der Variablen *\_pos* können beliebige Werte oder Gleichungen zugewiesen werden, wodurch ein Verklemmen oder eine Schwergängigkeit simulierbar ist. Durch Kombinationen lassen sich zudem komplexere Fehlerszenarien nachbilden.

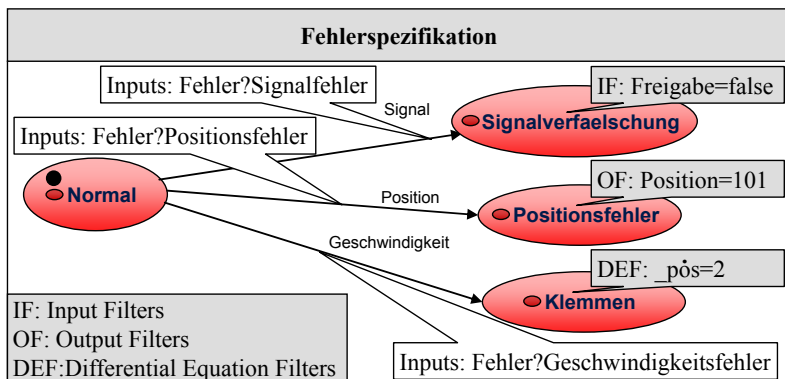


Abbildung 5.7: Fehlermodellierung mit Hilfe von Zustandsmaschinen

### 5.3.5 Geometriemodell

Die Funktionalität einer Maschine hängt in wesentlichen Teilen von deren äußerer Gestalt und somit von der Geometrie ab. Dies umfasst z. B. die potenziellen Kollisionen von Achsen oder die räumliche Anbringung und die Ausrichtung von Sensoren. Das Verschieben oder Drehen einer Lichtschranke kann bereits gravierende Auswirkungen auf das Verhalten der Anlage haben. Daher ist die Geometrie im Rahmen der hier vorgestellten Methode Bestandteil des Verhaltensmodells und dient nicht nur zu Visualisierungszwecken.

Zunächst ist ein Raumbegriff notwendig, der die Grundlage für die Darstellung und Einbindung von geometrischen Objekten in das Funktionsmodell liefert. In der Mathematik existiert eine Vielzahl an Definitionen, auf die an dieser Stelle aber nicht näher eingegangen werden soll. Der euklidische Raum  $\mathbb{E}^3$  gilt nach ESCHENBURG & JOST (2007) als der Raum der *alltäglichen geometrischen Vorstellung*, in welchem sich *die den Gesetzen der Newtonschen Mechanik gehorchenden physikalischen Prozesse* abspielen. Der dreidimensionale euklidische Raum bildet gleichermaßen die Grundlage für das Funktionsmodell.

Das hier eingesetzte Geometriemodell besteht aus drei Elementen (siehe Metamodell in Abschnitt 5.3.2): den *Parts*, welche die Maschinegeometrie repräsentieren, den *Detektoren*, die für die Erfassung und Weiterverarbeitung von Kollisionen verwendet werden, und den *Bindings*, die der Handhabung von Materialobjekten dienen. Alle drei Konstrukte werden, wie schon die Zustandsmaschinen, einer Komponente zugeordnet. Die Modellierung erfolgt in einem ersten Schritt mit den Volumenprimitiven *Quader*, *Zylinder* und *Kugel*, siehe Abbildung 5.8. Zugelassen sind nur formerhaltende Transformationen der Geometrie.

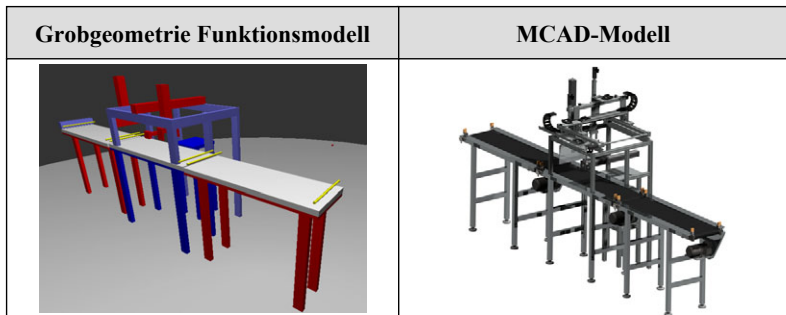


Abbildung 5.8: Abbildung der Geometrie im Funktionsmodell

Im späteren Verlauf des Entwicklungsprozess können zudem 3D-Daten aus dem MCAD-System in Form von VRML-Dateien integriert werden, siehe Abschnitt 6.4.6. Bereits die grobe Geometrie lässt erste Rückschlüsse auf das Systemverhalten zu.

Während die Verwendung der Parts als strukturbeschreibende Elemente selbsterklärend ist, soll auf die Detektoren und die Bindings genauer eingegangen werden. Die Detektoren sind zur Abbildung von Sensoren vorgesehen, die zur Erkennung von Objekten in einem geometrisch definierten Raum benötigt werden. Am Beispiel einer Lichtschranke, die als Zylinder modelliert wird, lässt sich dies verdeutlichen (siehe Abbildung 5.9). Sobald Material mit dem Zylinder kollidiert, wird die Information im zugehörigen Automaten verarbeitet. Hierfür ist lediglich ein Zustand nötig, der über eine Selbsttransition die aktuelle Kollisionsabfrage des Detektors Lichtschranke auf den Ausgangsport LS schreibt (Detektoren und Bindings sind wie die Variablen mit einem vorangestellten Unterstrich gekennzeichnet und können wie diese durch einen Automaten informationstechnisch verarbeitet werden). Ein Verschieben oder Drehen des Sensors erfordert keine Änderungen am Verhaltensmodell, wie es etwa bei logik- oder skriptbasierten Ansätzen der Fall ist, da hier die Geometrie Teil des Funktionsmodells ist.

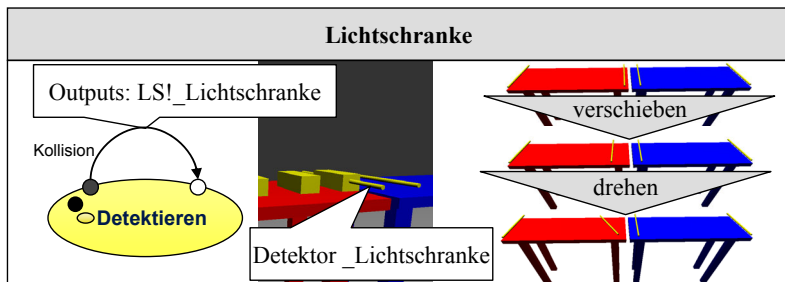


Abbildung 5.9: Einfluss der Geometrie auf die Funktionalität

Bindings stellen im Prinzip erweiterte Detektoren dar, die über zusätzliche Eigenschaften verfügen. Die räumliche Ausprägung beschreibt den Bereich, in welchem diese Einfluss auf Materialobjekte nehmen können. Grundsätzlich können zwei Arten von Bindings unterschieden werden, das *PortBinding* und das *MoverBinding*. Ersteres assoziiert bei einer Kollision mit einem Materialobjekt die Ports der Komponente, die das Binding enthält, mit den Ports der Komponente des Stückgutes (Material wird ebenfalls als Komponente modelliert, siehe Abschnitt 5.3.7), wodurch ein Informationsaustausch stattfinden kann. Das Mover-

Binding hingegen ermöglicht bei Kollision mit Materialobjekten deren Transport relativ zur geometrischen Repräsentation entlang einer definierten Achse (Mover). Über den verhaltensbeschreibenden hybriden Automaten können die bindenden Eigenschaften aktiviert oder deaktiviert werden, so dass sich das Binding wie ein gewöhnlicher Detektor auswirkt. Des Weiteren verfügt jedes Binding über eine Priorität, die im Falle von Mehrfachbindungen den Zugriff regelt.

### 5.3.6 Kinematische Beziehungen

Die räumliche Bewegung von Parts, Detektoren, Bindings oder Materialobjekten erfordert die Definition von Freiheitsgraden. Für diese Zwecke können einer Komponente Achsen in Form sogenannter Mover zugewiesen werden. Neben den trivialen translatorischen und rotatorischen Operationen, die durch die Linearachsen (*LinearAxis*) und die Drehachsen (*RotatoryAxis*) beschrieben werden (siehe Abbildung 5.10 links und Mitte), lassen sich außerdem komplexere Verfahrbewegungen erzeugen. Hierzu stehen Mechanismen zur Interpolation von Bewegungspfaden (*InterpolatedAxis*) zur Verfügung, welche diese mit Hilfe von Stützpunkten berechnen (Abbildung 5.10 rechts).

Die festgelegten Freiheitsgrade können anschließend einzelnen Parts, Detektoren oder Bindings direkt zugewiesen werden, so dass diese entsprechend den vorgegebenen Richtungen verfahren bzw. sich drehen. Wie bereits in Abschnitt 5.3.5 angedeutet, können zusätzlich Relativbewegungen in den MoverBindings für Materialobjekte erzeugt werden, die durch Mover zu spezifizieren sind.

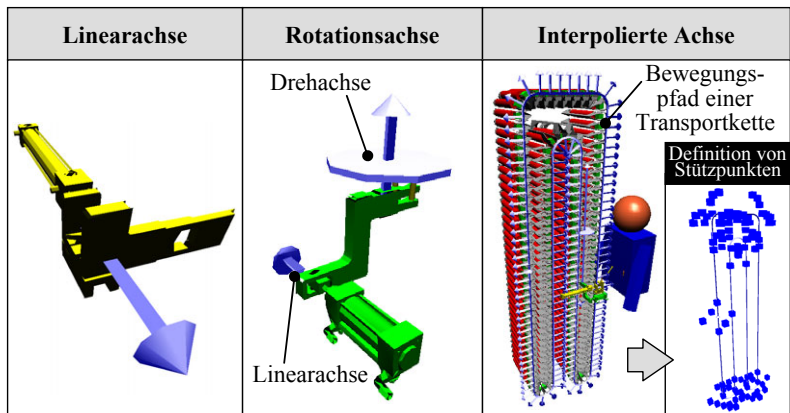


Abbildung 5.10: Modellierung von Verfahrbewegungen durch Achsen

Die Berechnung der Beschleunigungen und der Geschwindigkeiten erfolgt in den Zuständen der hybriden Automaten. Die Ergebnisse der Differenzialgleichungen werden direkt den zugehörigen Achsen zugeteilt.

Der Aufbau von seriellen Kinematiken ist durch eine Superposition von Freiheitsgraden möglich. Hierfür wird der Mover der kinematisch übergeordneten Komponente der mitzubewegenden Komponente auferlegt (siehe Abbildung 5.11 links, Zuweisung durch gestrichelte Linie geregelt). Ein Beispiel dafür zeigt Abbildung 5.11, in welcher ein einfaches Portalsystem dargestellt ist. Der translatorische Freiheitsgrad des Verfahrantriebs in Richtung der x-Achse wird dem Hubantrieb (y-Achse) aufgezwungen, so dass dieser der Bewegung folgen muss. Prinzipiell können hiermit beliebig komplexe Szenarien modelliert werden. Allerdings ist die Verwendung der interpolierten Achsen für gewisse Anwendungsfälle, wie z. B. Kettenmagazine oder mäanderförmige Transportbänder, wesentlich einfacher und intuitiver umzusetzen.

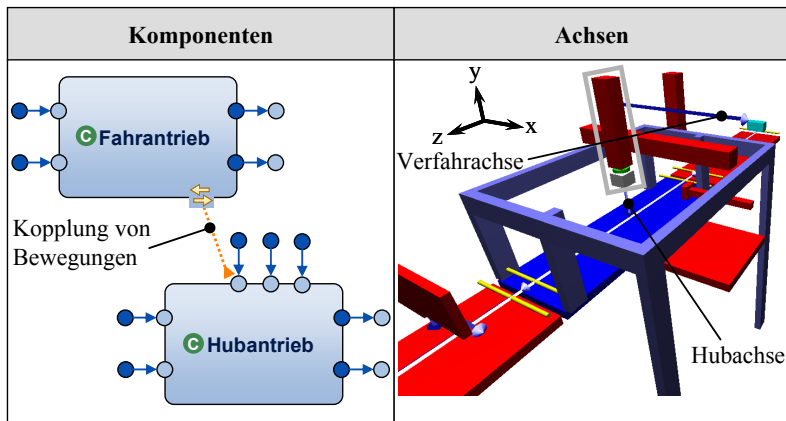


Abbildung 5.11: *Serielle Kinematik eines Portalsystems*

### 5.3.7 Materialfluss

Im Rahmen dieser Arbeit steht der Materialfluss in Form von Stückgütern im Vordergrund. Schütt- und Fließgüter wurden nicht explizit betrachtet. Die Grundlagen zur Abbildung des Materialtransportes wurden in den Abschnitten 5.3.5 und 5.3.6 durch die MoverBindings und die Mover geschaffen. Die einzelnen Materialobjekte werden als Komponente beschrieben und setzten sich somit aus

deren Modellelementen, wie z. B. den Parts oder den Detektoren, zusammen. Dadurch steht der volle Modellierungsumfang zur Verfügung, wodurch Material mit Eigenverhalten erzeugt werden kann, das sich durch die hybriden Automaten beschreiben lässt. Zunächst ist ein Referenzteil zu definieren, welches dann über eine Materialquelle, den sogenannten *Entry*, im Maschinenmodell beliebig oft erzeugt wird, siehe Abbildung 5.12 links. Die Anzahl und die Zeitpunkte der Generierung werden mit Hilfe einer Zustandsmaschine beschrieben. Die Materialsenken, auch als *Exits* bezeichnet, dienen zum Löschen des Stückgutes, siehe Abbildung 5.12 rechts. Entries und Exits verfügen über eine geometrische Repräsentation in Form von Quadern. Über die Position im Raum können somit die Orte festgelegt werden, an denen Material in das System ein- bzw. austreten soll. Darüber hinaus müssen für die Entries Verteilungsfunktionen, z. B. eine Normalverteilung, zur Materialgenerierung innerhalb dieses definierten Bereichs angegeben werden. Das Aktivieren der Exits ist wiederum über die Automaten geregelt, wobei die Kollision des Stückgutes mit der räumlichen Darstellung der Materialsenke als Voraussetzung gilt.

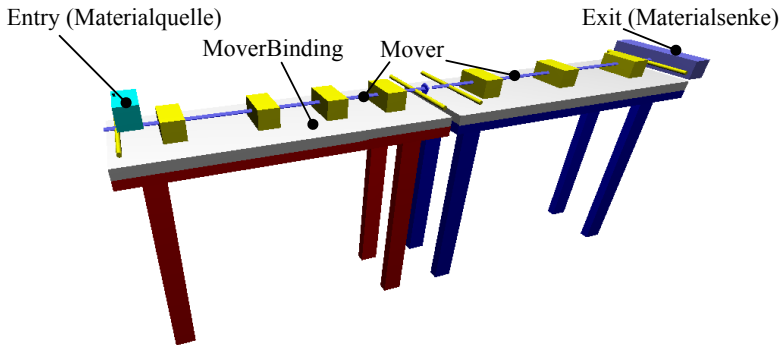


Abbildung 5.12: Materialquellen und -senken im Funktionsmodell

Der Transport des Materials ist, wie bereits erläutert, über die MoverBindings und die Mover realisiert. Dabei kann das Stückgut zum einen an das MoverBinding fixiert werden, indem die Relativgeschwindigkeit zu null gesetzt wird. Das MoverBinding selbst kann sich samt Materialobjekt bewegen, falls entsprechende Achsen definiert sind. Hierdurch lässt sich z. B. ein verfahrbarer Greifer modellieren. Zum anderen kann die Relativbewegung im MoverBinding zur Darstellung von Transportstrecken verwendet werden. In Abbildung 5.12 ist das Beispiel eines Förderbandes verdeutlicht. Für dessen Modellierung sind lediglich das

MoverBinding, der Mover zur Definition der Verfahrbewegung und ein hybrider Automat, der die Geschwindigkeiten berechnet, erforderlich. Da die Geometrie und die Kollisionserkennung Bestandteil des Modells sind, ergeben sich beispielsweise das Aufstauen des Stückgutes oder das Auslösen von Lichtschranken aus dem Modellcharakter. Logische Ersatzmodelle sind dafür nicht erforderlich. Mit Hilfe der interpolierten Achsen und der Einbindung der 3D-Geometrie können Szenarien mit einer sehr hohen Detailgüte abgebildet werden.

### 5.3.8 Bedienerschnittstelle

Die Integration des Bedieners umfasst im Bezug auf die hier vorgestellte Modellierungstechnik zwei Aspekte. Erstens soll die Schnittstelle zwischen Mensch und Steuerung, auch HMI genannt, im Funktionsmodell abgebildet werden. Der Bediener kann dadurch Verfahrfreigaben erteilen oder Bearbeitungsprogramme auswählen. Zweitens ist eine Schnittstelle zur Maschinensimulation erforderlich. Häufig ist der Anlagenbediener durch manuelle Tätigkeiten in den Gesamttablauf mit einbezogen. Beispielsweise sind neue Bauteile einzulegen oder mechanische Hebelmechanismen zu betätigen. Dafür sind entsprechende Möglichkeiten im Modell vorzusehen. Für die Funktionsmodellierung ist demzufolge eine alternative Verhaltensspezifikationstechnik möglich (siehe Metamodell in Abschnitt 5.3.2). Anstatt der hybriden Automaten können Bedienpanels (OperatorPanel) herangezogen werden. Diese stellen z. B. Taster, Schieberegler oder Anzeigeelemente zur Verfügung, die entweder die Ports beschreiben oder lesend darauf zugreifen können.

### 5.3.9 Bibliotheksmechanismus

Die Wiederverwendung von Komponenten oder von deren Bestandteilen, wie z. B. die hybriden Automaten, kann die Modellierungsarbeit deutlich beschleunigen. Aus diesem Grund ist für die hier vorgestellte Funktionsmodellierung ein parametrischer Bibliotheksmechanismus integriert. Neben der trivialen Funktion als Modellspeicher (Komponenten, Automaten, Bedienpanels etc.) zeichnet sich dieser dadurch aus, dass bestimmte Muster in Abhängigkeit von Variablen erzeugt werden können. Als Beispiel sei ein Palettenförderer genannt, der sich aus einer festen Zahl gleicher Elemente zusammensetzt. Das Referenzteil muss lediglich einmal in der Bibliothek abgelegt werden. Im Anschluss lassen sich die Einzelpaletten entlang einer Achse (Mover) gemäß des vorgegeben Parameters generieren.



### 5.3.10 Simulation des Funktionsmodells

Grundsätzlich existieren zwei unterschiedliche Wege, um das Modell auszuführen. Zum einen kann dieses direkt in der Entwicklungsumgebung für die Funktionsmodellierung interpretiert und simuliert werden, was einem MiL-Ansatz gleichkommt. Zum anderen lässt sich Code für die HiL- oder die SiL-Simulation generieren. Diese Herangehensweise hat den Vorteil, dass z. B. das Funktionsmodell schon zu Entwicklungsbeginn simuliert werden kann, selbst wenn noch nicht alle erforderlichen Daten für HiL- oder SiL-Testläufe vorhanden sind.

## 5.4 Bezug zur Konstruktionsebene

### 5.4.1 Technisches Ressourcenmodell

In Kapitel 4 wird die Projektierungs- und Planungsphase als eine der Konstruktion vorangestellte Phase beschrieben. Im Hinblick auf den Abstraktionsgrad bzw. das Detailniveau ist diese somit zwischen den Anforderungen und der Konstruktionsebene einzuordnen, siehe Abbildung 5.13. Die Umsetzung der Anforderungen in Konstruktionsdaten und -modelle kann durch das hier vorgeschlagene Funktionsmodell unterstützt werden. Da dieses die Maschine stark abstrahiert, wird das sogenannte *Technische Ressourcenmodell (TRM)* eingeführt, das einen Bezug zu den konkreten technischen Ausprägungen der Konstruktionsebene herstellt.

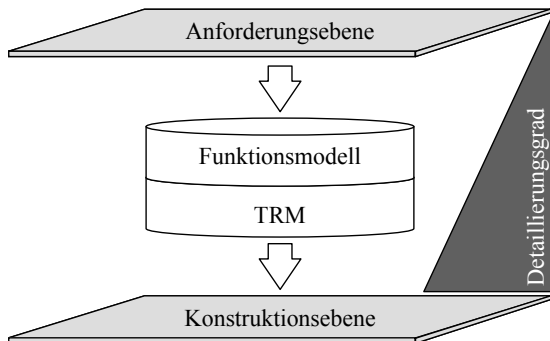


Abbildung 5.13: *Technisches Ressourcenmodell als Bindeglied zur Konstruktionsebene*

Das TRM ist wiederum Bestandteil einer Komponente im Funktionsmodell und dient der Erfüllung zweier Ziele. Zum einen sollen grundlegende Informationen für die sich anschließende Entwicklungsarbeit zur Verfügung gestellt werden. Auf Grund der Tatsache, dass alle am Prozess beteiligten Disziplinen im besonderen Maße von den Eigenschaften der Aktoren und der Sensoren abhängen, umfasst das TRM vor allem Daten über die charakteristischen Merkmale der beiden Letztgenannten. Zum anderen wird das TRM für die Generierung des VIBN-Modells herangezogen. Darin sind die Verbindungen zwischen den Ein- und den Ausgängen des Funktionsmodells und denen der realen Steuerung abgebildet.

Die Repräsentation von Aktor- und Sensordaten als Ausgangspunkt für die Konstruktion in den unterschiedlichen Fachbereichen erfordert eine geeignete Beschreibungsform. Für diese Zwecke muss der Informationsgehalt des TRMs derart gestaltet sein, dass keine Entwicklungstätigkeiten vorweggenommen werden, die sich wesentlich einfacher und schneller in den disziplinspezifischen Phasen umsetzen lassen. Es muss jedoch die Aussagekraft hinreichend sein, so dass eine gemeinsame Grundlage geschaffen und Entwicklungsschritte dadurch potenziell parallelisiert werden können. Daher ist zunächst eine adäquate Klassifikation von Aktoren und Sensoren notwendig, welche die oben genannten Aspekte berücksichtigt.

### 5.4.2 Aktoren

Nach RODDECK (2006) dienen Aktoren in mechatronischen Systemen zur *Erzeugung von Bewegungen* oder zum *Aufbringen von Kräften und Momenten*. Sie wandeln eine Hilfsenergie, wie etwa elektrische Energie oder Druckluft, in die angestrebte Energieform, beispielsweise mechanische Energie, um. Im Wesentlichen besteht ein Akteur aus zwei Elementen, dem Energiesteller, der über das Stellsignal angesprochen wird, und dem Energiewandler, der die gewünschte Ausgangsenergie liefert. ISERMANN (2008) bezeichnet diese auch als Primär- und Sekundäraktor.

Die Klassifikation des TRMs folgt hinsichtlich der Einteilung bzw. der Beschreibung von Aktoren einem Schema, das an RODDECK (2006) angelehnt ist und welches den hier verfolgten Ziele angepasst wurde. Der vorliegende Ansatz konzentriert sich hauptsächlich auf Systeme, deren Hilfsenergie in mechanische Energie gewandelt wird. Ein Akteur des TRMs kann anhand des Wirkprinzips, der Bewegungsart sowie der Regelungsart spezifiziert werden, siehe Abbildung 5.14. Das Wirkprinzip beschreibt, welche Hilfsenergie zugeführt und in mechanische

Energie gewandelt wird. Der Hauptanwendungsbereich liegt auf elektrischen, hydraulischen und pneumatischen Systemen, die für die Automatisierungstechnik eine hohe Relevanz aufweisen. Dieser Einteilung nicht entsprechende Aktoren, wie z. B. elektrochemische Ausführungen, werden unter dem Punkt *andere* zusammengefasst. Die durch den Aktor erzeugten Bewegungen können translatorischer oder rotatorischer Art sein. Ein weiteres Klassifikationsmerkmal ist der Typ der Regelung, wobei vor allem die Positions- und die Geschwindigkeitsregelung von Bedeutung sind. Alternative Regelungsarten, wie z. B. die Kraftregelung, werden wiederum zusammengefasst.

Wirkprinzip	elektrisch	hydraulisch	pneumatisch	andere
Bewegung	linear		rotatorisch	andere
Regelungsart	Position	Drehzahl/ Geschwindigkeit	ungeregelt	andere

Abbildung 5.14: Spezifikation von Aktoren im Technischen Ressourcenmodell

In bestimmten Fällen, z. B. bei Kundenvorgaben, muss es möglich sein, in frühen Phasen detaillierte Daten zu hinterlegen. Dies ist in Form sogenannter *Technischer Details* (engl. *Technical Detail*, kurz TD) realisiert, welche einem Aktor im TRM zugewiesen werden können und eine textuelle Beschreibung beinhalten.

### 5.4.3 Sensoren

Der Begriff des Sensors ist in der Automatisierungstechnik weit verbreitet, wobei als Synonyme ebenfalls die Ausdrücke Messfühler oder Messaufnehmer gebräuchlich sind (ISERMANN 2008). HESSE & SCHNELL (2009) definieren einen Sensor als *technisches Bauteil, das aus einem Prozess zeitvariable physikalische oder auch elektrochemische Größen erfasst und in ein eindeutiges elektrisches Signal umsetzt*. Für die Wandelung der Messgröße in das weiterverarbeitbare elektrische Ausgangssignal können unterschiedliche Prinzipien und Effekte zum Einsatz kommen, die häufig zur Klassifikation herangezogen werden. Darüber hinaus kann eine Betrachtung von Sensoren im Hinblick auf die zu erfüllenden Anforderungen sinnvoll sein, wie dies von FRADEN (2004) formuliert wird. HESSE & SCHNELL (2009) beschreiben eine ähnliche Möglichkeit zur Klassifizierung von Sensoren, in welcher die zu erfüllenden Aufgaben, wie z. B. die Feststellung der physischen Präsenz von Objekten, als charakteristisches Merkmal herangezogen werden.

Eine Einteilung von Sensoren im hier vorgestellten TRM soll die wesentlichen Bereiche der Automatisierungstechnik abdecken. Vor allem geregelte Antriebe spielen dahingehend eine wichtige Rolle, für die Positionen oder Geschwindigkeiten zu erfassen sind. Die Gliederung der Sensoren nach Messgrößen wird für das TRM allerdings nicht für sinnvoll erachtet, da sich hierdurch Problemstellungen, wie etwa das Erkennen von Material durch Lichtschranken, nur über Umwege darstellen lassen. Gerade diese Art von Sensoren findet in der Automatisierungstechnik umfassend Verwendung. Dabei ist weniger die Messgröße, beispielsweise im Fall der Lichtschranke die Lichtintensität, sondern eher die durch den Sensor zu erfüllende Aufgabe von Interesse. Daher wird eine Einteilung für Sensoren im TRM vorgeschlagen, wie sie in Abbildung 5.15 dargestellt ist. Das Schema soll eine einfache und schnelle Beschreibung des zu implementierenden Sensors in frühen Entwicklungsphasen ermöglichen. Hierfür ist es zunächst nicht von Bedeutung, ob der zugrunde liegende physikalische Effekt eines Sensors beispielsweise induktiv oder kapazitiv ist. Vielmehr steht die Funktion oder die Aufgabe, die damit gelöst werden soll, im Vordergrund. Es können jedoch Sensoren, wie schon für die Aktoren erläutert, bei Bedarf um TDs erweitert werden. Die primäre Einteilung für das TRM orientiert sich am Messproblem bzw. an der Aufgabenstellung. Als Kategorien werden die Erfassung der *Position* und der *Geschwindigkeit* bzw. der *Drehzahl*, die *Erkennung von Materialobjekten* sowie die Messung weiterer *physikalischer Größen* definiert. Falls erforderlich, können in den TDs Merkmale der technischen Ausprägung (z. B. Näherungssensor) und des Messprinzips (z. B. induktiv) hinterlegt werden.

Messproblem/ Aufgabe	Position	Drehzahl/ Geschwindigkeit	Materialerkennung	Erfassung physikalischer Größen
Technische Details	Technische Ausprägung		Messprinzip	

Abbildung 5.15: Einteilung von Sensoren im Technischen Ressourcenmodell

#### 5.4.4 Kopplung zur Steuerungsebene

Die Schnittstellen auf der Ebene des Funktionsmodells entsprechen nicht zwangsläufig den realen Steuerungsein- und -ausgängen. Daher ist eine Zuordnung über Eins-zu-eins-Beziehungen nicht immer möglich, was beispielsweise für komplexe Steuerwörter von umrichterregerten Antrieben zutrifft: Reicht auf der Modellebene für die Verfahrfreigabe unter Umständen ein boolesches Signal aus, so müssen im Zustandswort meist mehrere Bits gesetzt sein. Ein wei-

teres Beispiel sind Prozessdatenwörter, bei denen einzelne Bitbereiche für unterschiedlichen Informationsgehalt stehen und die zudem für die Einbindung in das Modell umgerechnet bzw. übersetzt werden müssen. Aus diesem Grund ist eine Zuordnung zwischen den Ein- und Ausgängen auf der Steuerungsseite und den Schnittstellen der Komponenten vorzunehmen.

Im Rahmen des hier vorgestellten Ansatzes ist die Problemstellung mit Hilfe des TRMs gelöst. Aktoren und Sensoren können um Schnittstellen zur Steuerung erweitert werden, siehe Abbildung 5.16. Hierzu muss der betrachtete Adressbereich, z. B. ein Datenwort, zunächst in sogenannte Felder zerlegt werden. Die Größe des Feldes kann von einem Bit bis zu einem Doppelwort reichen. Mit Hilfe von Adaptern lassen sich dann die Felder wiederum mit den Ports verknüpfen.

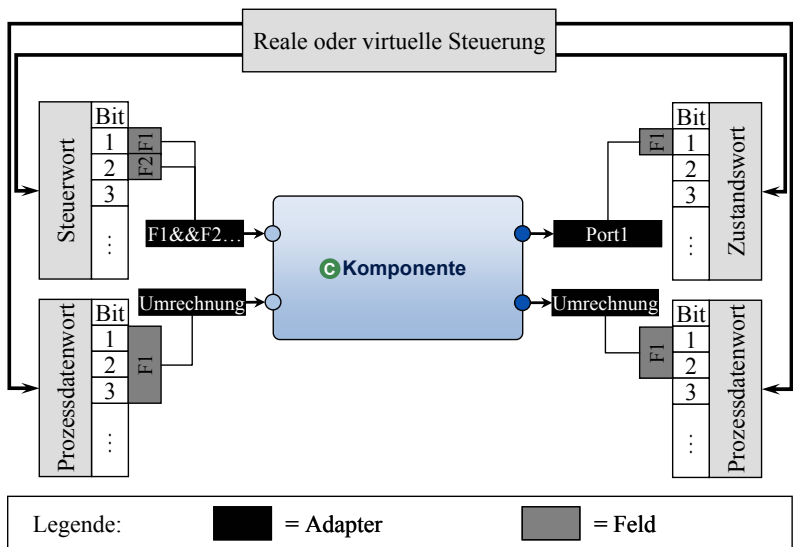


Abbildung 5.16: Zuordnung der Schnittstellen des Funktionsmodells zu den realen Steuerungsein- und -ausgängen

## 5.5 Vorgehensmodell

### 5.5.1 Konkretisierung des Vorgehensmodells

*In Vorgehensmodellen [...] werden wichtige Elemente einer Handlungsfolge abgebildet, die als Hilfsmittel zum Planen und Kontrollieren von Prozessen dienen können (LINDEMANN 2009).*

Das Vorgehensmodell dieser Arbeit wurde in den Grundzügen bereits in Kapitel 4 skizziert. Im vorliegenden Abschnitt soll nun eine detaillierte Beschreibung erfolgen. Der wesentliche Inhalt umfasst die nachstehend genannten Aspekte:

- Phasen und deren Inhalte
- Beteiligte Personen und Fachbereiche
- Modellierungsrichtlinien

Die einzelnen Phasen verdeutlicht Abbildung 5.17, wobei diese auf unterschiedlichen Ebenen angesiedelt sind und beginnend von der Anforderungsebene über die Funktionsebene bis hin zur Konstruktionsebene durchlaufen werden. Der Schwerpunkt dieser Dissertation liegt auf der Funktionsmodellierung. Die Anforderungen dienen als Eingangsinformation und werden als gegeben vorausgesetzt. Wie Anforderungen erfasst und in welcher Form diese festgehalten werden, ist nicht Teil der Betrachtungen. Ebenso wird das Vorgehen in der Konstruktionsphase nicht weiter ausgeführt.

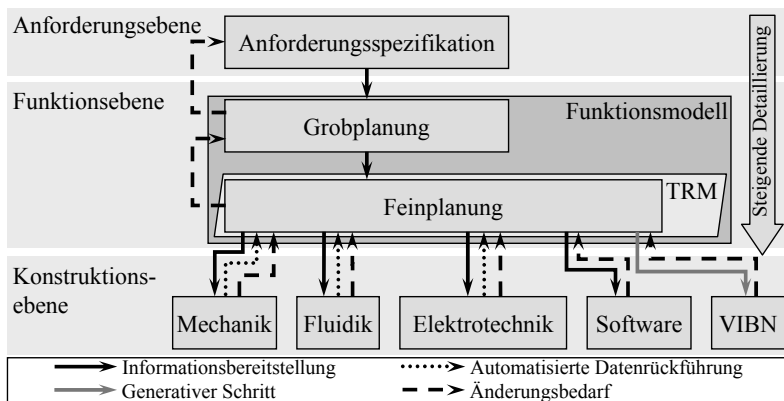


Abbildung 5.17: Vorgehensmodell zum Einsatz der Funktionsmodellierung

Abbildung 5.17 hebt hervor, dass eine iterative Vorgehensweise angestrebt wird. Somit lassen sich Ergebnisse schnell abgleichen und notwendige Anpassungen zeitnah umsetzen. Dadurch soll verhindert werden, dass Änderungsbedarf in späten Phasen zu einem nochmaligen Durchlaufen des Gesamtzyklus führt. Dies ist häufig bei sequenziellen Entwicklungsprozessen anzutreffen, wo beispielsweise während der steuerungstechnischen Inbetriebnahme identifizierte Probleme Modifikationen in der Mechanik- oder der Elektrokonstruktion nach sich ziehen können.

Das Funktionsmodell der Feinplanungsphase bildet die Grundlage für die Entwicklungstätigkeiten auf der Konstruktionsebene. Dies umfasst sowohl generative Schritte, wie z. B. für die VIBN, als auch die Aufgabe, die Kommunikation und die Abstimmung zwischen den Disziplinen zu verbessern. Eine automatische Ableitung von vereinfachten Konstruktionsmodellen ist grundsätzlich möglich, siehe Abschnitt 6.8, jedoch kein vordergründiges Ziel der vorliegenden Arbeit. Es soll lediglich die prinzipielle Umsetzbarkeit gezeigt werden. Die teilautomatisierte Rückführung von Informationen aus den CAD-Systemen dient zur Anreicherung des Funktionsmodells mit den für die VIBN essenziellen Daten. Grundlegende Änderungen sind idealerweise erst im Funktionsmodell zu implementieren, um eine gemeinsame Basis für alle Disziplinen zu schaffen. Das VIBN-Modell wird vollständig aus dem übergeordneten Modell generiert. Ein automatischer Informationsrückfluss vom VIBN-Modell zum Funktionsmodell findet nicht statt.

Die folgenden Abschnitte widmen sich der ausführlichen Vorstellung der Grob- und der Feinplanungsphase. Am Beispiel eines einfachen Transportsystems wird der Einsatz der Modellierungsmethode verdeutlicht.

### **5.5.2 Modellbildung in der Grobplanungsphase**

Die Grobplanungsphase folgt unmittelbar auf die Anforderungsspezifikation. Das Funktionsmodell kann in diesem Abschnitt dazu herangezogen werden, um mit dem Kunden, dem firmeneigenen Vertrieb und der Konstruktion ein gemeinsames Grundverständnis der zu projektierenden Maschine zu entwickeln und die Belange der beteiligten Interessensgruppen ausreichend zu berücksichtigen. Das zugehörige Modell zeichnet sich durch einen sehr hohen Abstraktionsgrad aus und kann dadurch sehr schnell und aufwandsarm erstellt werden. Aufgrund dessen lassen sich bereits in diesem frühen Stadium zeitnah unterschiedliche Szenarien erarbeiten und mit Hilfe der Simulation bewerten. Hierbei spielt die genaue tech-

nische Ausprägung eine untergeordnete Rolle, da sie erst im Rahmen der Feinplanung genauer zu bestimmen ist. Lediglich die Funktionen und das Zusammenwirken der einzelnen Systembestandteile sind von Interesse. Im Folgenden wird dargestellt, wie die Funktionen von Aktoren und Sensoren sowie deren verbindende Wirkketten mit Hilfe der in Abschnitt 5.2 vorgestellten Modellierungstechnik abzubilden sind.

### Aktoren

Auf das Verhalten bzw. den Arbeitsablauf einer Maschine wird mit Hilfe der Aktoren Einfluss genommen. Dabei lösen die Signale der SPS über eine Steuerkette entsprechende Reaktionen aus. Der grundsätzliche Aufbau von Steuerketten wird in Abbildung 5.18 gezeigt, wobei dies am Beispiel eines pneumatischen doppeltwirkenden Zylinders verdeutlicht ist, aber gleichfalls für die Elektrotechnik Gültigkeit besitzt. Steuerketten bestehen demzufolge aus Signalgliedern für Eingaben, Steuergliedern zur Signalverarbeitung und Stellgliedern zur Signalausgabe für die Arbeitsglieder (LANGMANN 2004). Des Weiteren können Signalglieder bei einfachen Systemen auch direkt auf die Stellglieder wirken oder es können gar alle Elemente der Steuerkette in einem Bauteil vereint sein (BAUER 2009).

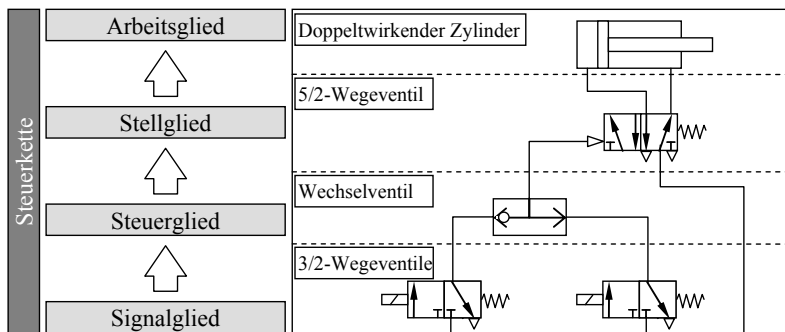


Abbildung 5.18: Aufbau der Steuerkette eines Pneumatikzylinders

Zudem ist es nach PRITSCHOW (2006) möglich, dass einzelne Glieder einer Steuerkette geregelt betrieben werden. Das ist z. B. häufig bei elektrischen Antrieben in der Automatisierungstechnik der Fall. Die SPS steuert den Prozess und generiert Sollwertvorgaben, die über den Feldbus an die Regler der Elektromotoren gesendet werden.



Da Steuerketten unter Umständen sehr komplex ausgeprägt sein können, ist in der Grobplanungsphase eine Darstellung in dieser detaillierten Form nicht zielführend. Darüber hinaus sind grundlegende Informationen noch nicht vorhanden, weil diese erst im Laufe des Entwicklungsprozesses erarbeitet werden müssen. Daher wird die gesamte Steuerkette, vom Signalglied bis zum Arbeitsglied, durch eine einzelne Komponente im Funktionsmodell abstrahiert, der ein stark vereinfachtes Verhalten innewohnt, siehe Abbildung 5.19. Damit ist allerdings noch keine Festlegung auf ein Wirkprinzip verbunden. Bei der Abstraktion des Verhaltens steht lediglich die angestrebte Funktion im Vordergrund, strukturelle Aspekte werden vernachlässigt bzw. vereinfacht.

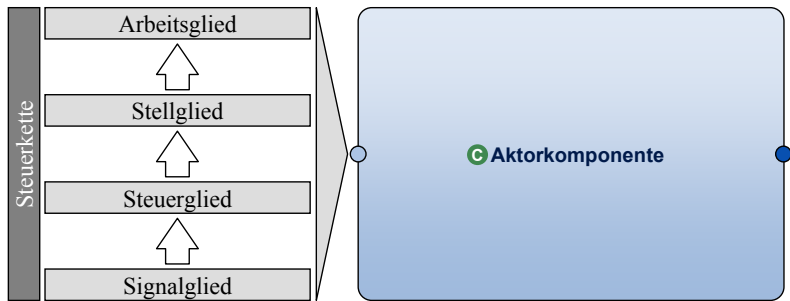


Abbildung 5.19: Modellierung von Aktoren in der Grobplanungsphase

Am Beispiel eines Transportsystems soll die Modellbildung in der Grobplanungsphase erläutert werden. Das zugehörige Funktionsmodell ist in Abbildung 5.20 visualisiert. Wie deutlich erkennbar ist, handelt es sich um eine stark reduzierte Maschinengeometrie, die jedoch für erste Analysen ausreicht. Im Detail soll der Mechanismus zum Abgreifen des Stückgutes von der zweiten Förderstrecke betrachtet werden.

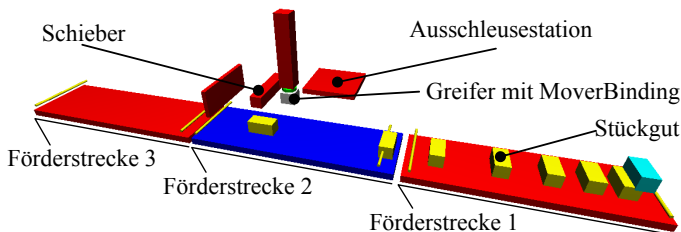


Abbildung 5.20: Funktionsmodell der Grobplanungsphase eines Transportsystems mit drei Förderstrecken und mit Teileausschleusung

Die Komponente *Greifer* verfügt lediglich über den booleschen Eingangsport *Fixieren*, mit welchem das Festhalten von Materialobjekten gesteuert wird, siehe Abbildung 5.21. Das gesamte Greifersystem wird durch das MoverBinding *\_Fixierbereich* abstrahiert. Dabei ist das Verhalten so stark verallgemeinert, dass es sich beispielsweise sowohl um einen konventionellen pneumatischen Greifer als auch um ein mit Unterdruck arbeitendes Saugsystem handeln könnte. Auf diese Art und Weise wird der Lösungsraum für die spätere technische Detaillierung nicht eingeschränkt. Der zugehörige hybride Automat verfügt nur über die zwei Zustände *Lösen* und *Halten*. Der initiale Zustand *Lösen* wird bei Anliegen der Nachricht *true* am Eingangsport *Fixieren* verlassen. Im Zustand *Halten* wird das MoverBinding *\_Fixierbereich* aktiviert. Da keine Differenzialgleichungen für eine Relativbewegung im MoverBinding angegeben sind, wird das Materialobjekt statisch an den Greifer gebunden. Durch eine weitere Transition in den Zustand *Lösen* kann das Stückgut wieder freigegeben werden.

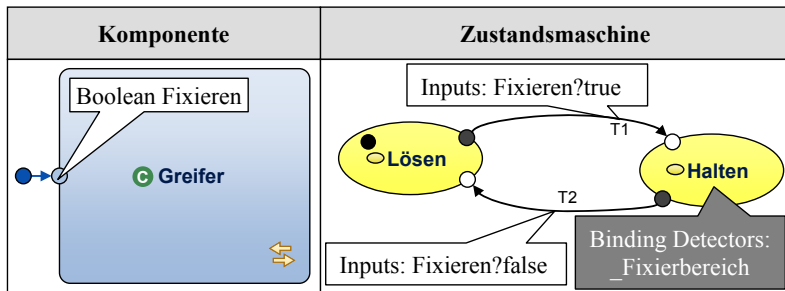


Abbildung 5.21: Stark abstrahiertes Grobplanungsmodell eines Greifers

Ein anderes Beispiel für die funktionale Modellierung von Aktoren in der Grobplanungsphase ist der Schieber in Abbildung 5.20, der zum Entfernen von Material auf der zweiten Transportstrecke dient. Dieser ist durch eine logische Komponente abgebildet, die ein kontinuierliches Verfahren zwischen zwei Endlagen ermöglicht. Die Bewegung könnte beispielsweise durch einen Arbeitszylinder oder einen Elektromotor hervorgerufen werden, was aber in diesem Entwicklungsstadium nicht zu spezifizieren ist.

## Sensoren

Die Erfassung von Zustandsgrößen über Sensoren lässt sich ebenso stark abstrahieren und auf den wesentlichen Funktionszweck vereinfachen, wie es bereits für die Aktoren gezeigt wurde. Ein Sensor wird hierzu entweder als eigenständige

Komponente modelliert oder, falls von Vorteil, in die Komponente eines Aktors integriert. An zwei einfachen Beispielen soll dies verdeutlicht werden.

Besteht die Aufgabe des Sensors in der Erfassung von Materialobjekten in einem geometrisch definierten Einflussbereich, so kann er durch einen Detektor abgebildet und in einer eigenen Komponente gekapselt werden. Kollisionen des Stückgutes mit dem Detektor werden durch eine Zustandsmaschine über den Ausgangsport versendet. Wie das Erfassen technisch bewerkstelligt wird, ob durch eine Lichtschranke oder einen Näherungssensor, ist somit noch nicht festgelegt. Dies hängt beispielsweise von der Größe des Einflussbereiches oder den Materialeigenschaften ab und ist in späteren Phasen zu klären.

Um die Positions- und die Geschwindigkeitsmeldung eines Sensors zu vereinfachen, kann dieser in die Komponenten des zugehörigen Aktors integriert werden. Dadurch besteht die Möglichkeit des direkten Zugriffs auf die Variablen der Differenzialgleichungen im hybriden Automaten des Aktors. Die entsprechenden Werte, wie z. B. der zurückgelegte Weg, können berechnet und über die Ports nach außen gemeldet werden.

### Wirkketten

Die genauen Ausprägungen der Wirkketten einer Maschine sind zu Entwicklungsbeginn zumindest für Neukonstruktionen nicht bekannt. Daher ist es sinnvoll, diese zu abstrahieren. Am Beispiel einer Förderstrecke soll dies verdeutlicht werden, siehe Abbildung 5.22. Das System besteht aus den drei Komponenten *Antrieb*, *Teileerkennung\_Einlauf* und *Teileerkennung\_Auslauf*, die eine vereinfachte Transportstrecke darstellen. Das Verhaltensmodell des Antriebs wurde bereits in Abschnitt 5.3.4.1 erläutert. Die Wirkkette zwischen dem Aktor und den Sensoren ist mit Hilfe eines MoverBindings und eines Movers modelliert.

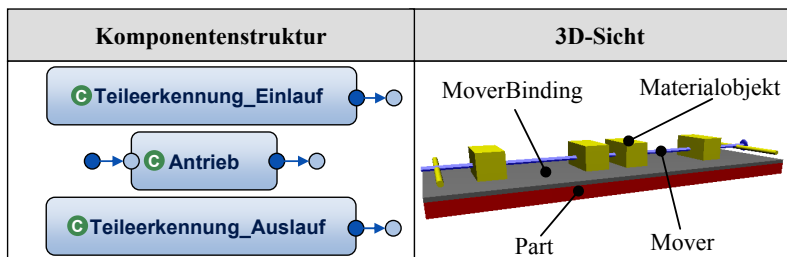


Abbildung 5.22: Modellierung von Wirkketten in der Grobplanungsphase am Beispiel einer einfachen Förderstrecke

Mechanische Aspekte, wie z. B. Übersetzungen, sind vernachlässigt bzw. schon in die Komponente des Antriebs integriert, der lediglich die Geschwindigkeit vorgibt. Die mögliche Lösungsvielfalt wird dadurch nicht reduziert. Das klassische Förderband wäre nur eine davon, ebenso könnten Paletten- oder gar Vibrationsförderer eingesetzt werden. Die genauere Spezifikation ist Teil der Feinplanungsphase.

### **Wiederverwendung von Modellen**

Die Modellerstellung der Grobplanung kann durch eine Bibliothek unterstützt werden. Hierin sind nicht Elemente enthalten, die direkt ein technisches Bauteil beschreiben, sondern logische Komponenten, die eine gewisse Funktionalität zur Verfügung stellen, wie z. B. das Verfahren zwischen zwei Endlagen.

### **5.5.3 Modellbildung in der Feinplanungsphase**

Der Informationsgehalt des Modells der Grobplanungsphase stellt die Grundlage für die Feinplanungsphase dar. Idealerweise wird das Modell dieses Abschnittes durch eine mechatronische Projektgruppe erstellt, die Mitglieder aus allen Entwicklungsdisziplinen enthält. Somit können die technischen Anforderungen aus allen Bereichen frühzeitig berücksichtigt werden. Das Optimum jeder einzelnen Disziplin führt nicht automatisch zu einem Gesamtoptimum, daher kann eine frühe Abstimmung zur bestmöglichen Lösungsfindung beitragen. Darüber hinaus sollten grundlegende Änderungen im Laufe des Konstruktionsprozesses zuerst mit allen beteiligten Gewerken am Feinplanungsmodell diskutiert und anschließend dort eingepflegt werden. Hierdurch lassen sich nicht nur Inkonsistenzen verhindern, sondern auch zeitaufwendige Iterationsschleifen verkürzen.

Im Zuge der Grobplanung wird die Funktionsweise der Maschine beschrieben, ohne auf deren technische Umsetzung näher einzugehen. Die Abbildung der Wirkstrukturen ist Teil der Feinplanungsphase. Es handelt sich allerdings dabei noch immer um ein Funktionsmodell, das strukturelle Aspekte vereinfacht darstellt. Dennoch kann es bereits zur Ableitung grober VIBN-Modelle herangezogen werden. Zu diesem Zwecke sind weitere Informationen in den Generierungsprozess mit einzubeziehen, wobei an dieser Stelle auf Abschnitt 6.6 verwiesen sei. Abbildung 5.23 zeigt die Geometrie in der Feinplanungsphase des bereits aus der Grobplanung bekannten Beispiels eines Transportsystems. Deutlich zu erkennen sind die im Vergleich detaillierten Strukturen, welche ebenso für die anschließende mechanische Konstruktion von Nutzen sind.

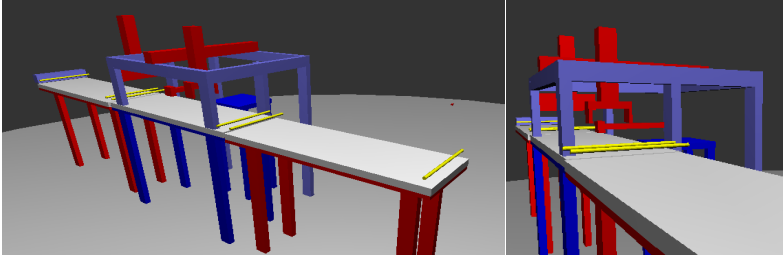


Abbildung 5.23: Geometriemodell der Feinplanungsphase

### Aktoren

Das Aktormodell der Feinplanungsphase gliedert die Steuerkette in zwei Komponenten auf, siehe Abbildung 5.24. Das Arbeitsglied, beispielsweise ein Pneumatikzylinder, wird explizit in einer eigenen Komponente abgebildet, wodurch die technische Lösung im Wesentlichen festgelegt ist. Signal-, Steuer- und Stellglieder werden abstrahiert und zu einer weiteren Komponente zusammengefasst. Diese wird hier als *Steuer- und Stellkomponente* bezeichnet. Dadurch sind das Wirkprinzip und die Schnittstellen des Aktors definiert, ohne auf die genaue technische Ausprägung eingehen zu müssen, die in der sich anschließenden Konstruktionsphase geklärt werden kann. Das Funktionsmodell der Feinplanungsphase bestimmt vielmehr das Sollverhalten, das durch entsprechende Bauteile, wie z. B. Wegeventile oder Schütze, unter den gegebenen Randbedingungen umzusetzen ist.

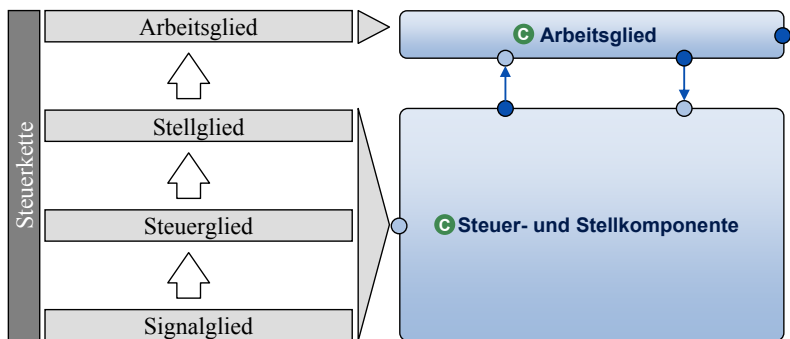


Abbildung 5.24: Modellierung von Aktoren in der Feinplanungsphase

Die Modellierung in dieser Phase soll am Beispiel des Greifers aus Abschnitt 5.5.2 erläutert werden. Als Lösungselement des in der Grobplanung stark abstrahierten Systems kommt ein pneumatischer Greifer zum Einsatz. Dieser erhält das boolesche Eingangssignal *Fixieren*, das durch die Steuer- und die Stellkomponente verarbeitet und in die zwei booleschen Ausgangssignale *Ein* und *Aus* umgewandelt wird. Die Aufgabe der Fluidkonstruktion besteht darin, dieses logische Verhalten durch passende Ventile abzubilden. Die Ausgänge der Steuer- und der Stellkomponenten bilden wiederum die Eingänge für den pneumatischen Greifer, was auf einen doppeltwirkenden Kolbenantrieb deutet. Abbildung 5.24 zeigt den zugehörigen hybriden Automaten. Die Endlagen sind durch die Zustände *Eingefahren* und *Ausgefahren* definiert, die Dynamik ist in den Zuständen *Einfahren* und *Ausfahren* hinterlegt. Es handelt sich noch immer um ein funktionales Modell, was sich an den booleschen Eingängen und der Tatsache manifestiert, dass außer den Endlagen keine statische Zwischenposition eingenommen werden kann, da dies funktional nicht vorgesehen ist.

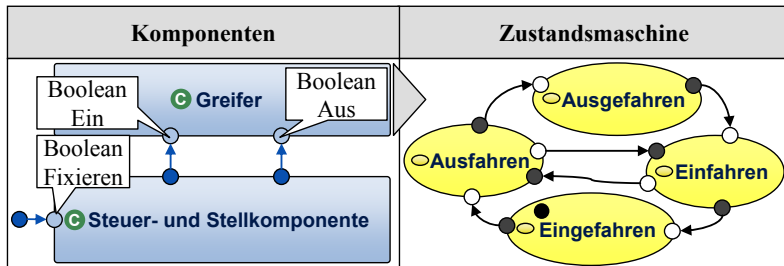


Abbildung 5.25: Modell der Feinplanungsphase eines pneumatischen Greifers

## Sensoren

Für die Abbildung von Sensoren ist im Modell der Feinplanungsphase nunmehr jeweils eine eigenständige Komponente vorgesehen. Als Beispiel dient wiederum die Positions- und die Geschwindigkeitsmessung. In der Grobplanungsphase wurde diese Sensorfunktion in die Aktorkomponente integriert. Hierfür ist jetzt eine entsprechende Komponente anzulegen, die über ein verfeinertes Verhalten verfügt und die Funktion des technischen Bauteils abstrakt beschreibt. In Abbildung 5.26 wird die Komponentenstruktur eines Förderbandes gezeigt, wobei ein inkrementeller Drehgeber zur Geschwindigkeitsmessung dient. Dieser ist an der hinteren Umlenkrolle des Bandes installiert. Hinsichtlich des Verhaltens müssen z. B. Zustandsmeldungen und Resetfunktionen implementiert werden.

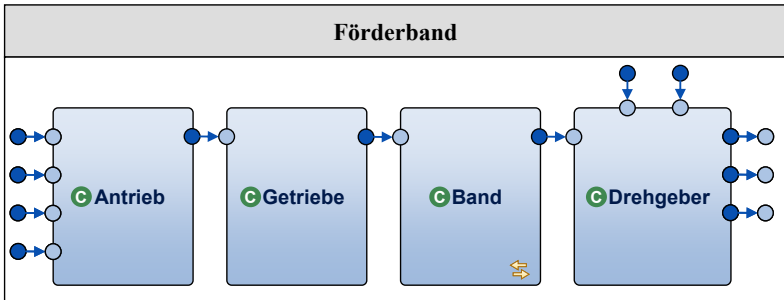


Abbildung 5.26: Komponentenstruktur des Förderbandes in der Feinplanung

### Wirkketten

Die Wirkketten sind in der Feinplanungsphase vollständig zu definieren, wie es Abbildung 5.26 des bekannten Förderbandbeispiels verdeutlicht. Der elektrische Antrieb wirkt über eine Getriebestufe auf die Antriebsrolle des Förderbandes, ebenso hat der Durchmesser der Umlenkrolle Einfluss auf die Geschwindigkeitsmessung des angeflanschten Drehgebers.

### Technisches Ressourcenmodell (TRM)

Das TRM muss in diesem Stadium dem Funktionsmodell hinzugefügt werden. Zum einen können dadurch die Aktoren und die Sensoren für die weitere Entwicklungsarbeit genauer spezifiziert werden. Zum anderen beinhaltet es die realen Steuerungssignale sowie die Felder und die Adapter zum Funktionsmodell. Somit können bereits vor der mechanischen und der elektrischen Konstruktion erste VIBN-Modelle für die Steuerungssoftwareentwicklung abgeleitet werden. Des Weiteren kann das Funktionsmodell die Aufgabe der sogenannten E/A-Listen übernehmen, welche die Adressvergabe für die einzelnen Aktoren und Sensoren festhalten.

### Integration von Fehlerszenarien

Die Modelle für die Störfälle, welche sich auf die Aktoren, auf die Sensoren und auf die Wirkketten auswirken, sind erstmalig in der Feinplanungsphase festzulegen. Dadurch kann das Verhalten einer technischen Lösung in Fehlersituationen besser abgeschätzt werden.

### **Steuerungslogik und Modellwiederverwendung**

Die für Simulationszwecke erforderliche Steuerungslogik im Funktionsmodell der Feinplanungsphase weist bereits charakteristische Merkmale des zu implementierenden Steuerungsprogrammes auf. Weiterhin hat sich gezeigt, dass es von Vorteil ist, auch den Aufbau des Steuerungsmodells hierarchisch zu gestalten und mit den zugehörigen Maschinenanteilen in mechatronischen Komponenten zusammenzuführen. Dies erleichtert die Wiederverwendung erheblich und zudem kann durch die Dezentralisierung der Steuerungsintelligenz die übergreifende Gesamtsteuerungskomponente wesentlich kompakter ausfallen. Die Modellbibliothek in dieser Phase orientiert sich an den technischen Komponenten.

### **5.6 Zusammenfassung**

Kapitel 5 widmet sich der Planungs- und der Projektierungsphase im Maschinen- und Anlagenbau. Zunächst wurden die Anforderungen für eine geeignete Modellierungstechnik formuliert und im Anschluss ein Vorschlag für deren konkrete Umsetzung präsentiert. Diese Beschreibungsform soll für alle am Entwicklungsprozess beteiligten Disziplinen verständlich sein und einen schnellen Modellaufbau ermöglichen. Ferner wurde ein Vorgehensmodell definiert, das den Einsatz der Modellierungstechnik mit Hilfe unterschiedlicher Phasen strukturiert und durch Richtlinien unterstützt. Es wird dabei zwischen einer Grobplanungs- und einer Feinplanungsphase unterschieden. Zu Beginn des Entwicklungsprozesses steht die Funktionalität der zu projektierenden Maschine im Mittelpunkt, die in einem nächsten Schritt erweitert und um technische Details angereichert wird. Das auf diese Art und Weise erstellte Funktionsmodell dient zum einen als Abstimmungs- und Kommunikationsmittel zwischen den verschiedenen Fachbereichen, zum anderen bildet es die Grundlage für die automatische Generierung des VIBN-Modells, die nachfolgend in Kapitel 6 behandelt wird.



## 6 Teilautomatisierte Modellerstellung

### 6.1 Kapitelüberblick

Nachdem in Kapitel 5 die Funktionsmodellierung detailliert beschrieben wurde, befasst sich das sechste Kapitel mit der teilautomatisierten Ableitung von Modellen für die VIBN. Hierzu ist in Abschnitt 6.2 der entwickelte Ansatz im Überblick erläutert. Abschnitt 6.3 beinhaltet grundlegende Betrachtungen zur Nutzung von CAD-Daten für die Simulation. Dies umfasst die verwertbaren Informationen in den CAD-Systemen sowie den Datenaustausch. Zudem wird eine Methode zur Konsistenzsicherung beschrieben. Die Erweiterung des Funktionsmodells um Details aus dem MCAD und dem E/FCAD ist Gegenstand der Abschnitte 6.4 und 6.5. Dazu werden die jeweiligen Konzepte zur Datenintegration sowie deren Umsetzung vorgestellt. Die Generierung der VIBN-Modelle ist Bestandteil von Abschnitt 6.6. Für den Steuerungstest können Modelle sowohl in HiL- als auch in SiL-Konfiguration abgeleitet werden. Dem Modelltest ist Abschnitt 6.7 gewidmet. Abschließend gibt Abschnitt 6.8 einen Ausblick über den Nutzen des Funktionsmodells für die Generierung von rudimentären CAD-Modellen.

### 6.2 Konzeptbeschreibung

Im Folgenden wird das Konzept zur Teilautomatisierung der Modellerstellung für die VIBN übergreifend beschrieben. Die Erläuterung der genauen Umsetzung des Ansatzes erfolgt im Anschluss separat in den jeweiligen Unterkapiteln. Die in der vorliegenden Arbeit vorgestellte Methode wird als teilautomatisch titulierte, da ein gewisser manueller Aufwand zur Erstellung des Funktionsmodells zu erbringen ist. Weiterhin sind Konstruktionsregeln einzuhalten, die einen automatisierten Import von Daten aus den CAD-Systemen zur Modellverfeinerung ermöglichen. Das VIBN-Modell an sich lässt sich jedoch vollständig aus dem Funktionsmodell generieren.

Die Grundlage bildet das Modell der Feinplanungsphase, siehe Abbildung 6.1. Dieses kann mit Hilfe der TRMs in ein VIBN-Modell überführt werden. Allerdings handelt es sich dabei noch immer um ein stark abstrahiertes Modell, das zwar die grundlegenden Wirkstrukturen beinhaltet, aber dennoch Aktoren und Sensoren sowie die Geometrie hauptsächlich unter funktionalen Gesichtspunkten

beschreibt. Daher werden CAD-Daten teilautomatisiert für dessen Detaillierung herangezogen. Hierfür wird ein Modell zur Verwaltung der Querbeziehungen zwischen Funktions- und Konstruktionsebene, ein sogenanntes Mapping-Modell, eingeführt. Dieses dient gleichzeitig zur Konsistenzprüfung.

In den CAD-Systemen sind vor allem statische Informationen gespeichert, jedoch lassen sich implizit über den Typ eines Bauteils bzw. eines Betriebsmittels Verhaltenscharakteristika bestimmen. Während sich die Daten aus dem MCAD überwiegend auf die Geometrie beziehen und somit direkt übernommen werden können, ist für die Integration der Details aus dem E/FCAD eine Modellbibliothek mit grundlegenden Elementen, wie z. B. Elektromotoren oder Wegeventilen, erforderlich. Das erweiterte Funktionsmodell ist für den Entwickler noch immer interpretierbar, da nur auf die Bestandteile des in Kapitel 5 erläuterten Metamodells zurückgegriffen wird. Somit sind Testläufe vorab in der Entwicklungsumgebung der Funktionsmodellierung möglich, ehe daraus das detaillierte VIBN-Modell abgeleitet wird.

Die Einbindung der CAD-Daten ist derart umgesetzt, dass diese getrennt und von der Modellkategorie (MCAD oder E/FCAD) unabhängig geschehen kann. Dies ist dadurch realisierbar, dass sich die Erweiterung auf unterschiedliche Modellelemente bezieht.

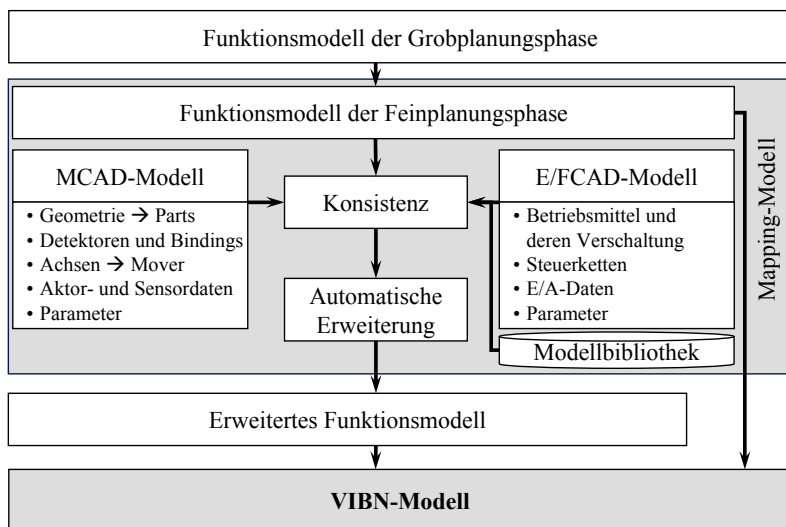


Abbildung 6.1: Konzept zur Generierung von VIBN-Modellen

## 6.3 Nutzung von CAD-Daten

### 6.3.1 Grundsätzliches zu CAD-Systemen

In den letzten Jahrzehnten hat im Bereich der Anlagen- und Maschinenentwicklung der Computer starken Einzug erhalten. Dies spiegelt sich in der mechanischen Konstruktionsarbeit wider. Die nachfolgend beschriebene Einteilung in 2D-, 2½D- und 3D-CAD-Systeme beruht auf HAMMEL & STRAUB (1988). Als Ersatz für die klassischen Zeichenbretter kamen zunächst 2D-Konstruktionswerkzeuge zum Einsatz. Dadurch ließ sich die Wiederverwendbarkeit von technischen Zeichnungen erheblich verbessern. Als nächste Stufe wurden 2½D-Systeme entwickelt, die eine Erstellung von räumlichen Körpern durch Vereinigungen oder Schnitte mit primitiven Hüllflächen ermöglichten (GRÄTZ 2006). Heute finden dagegen vorwiegend 3D-CAD-Systeme Verwendung, mit denen komplexe dreidimensionale Geometrien aufgebaut werden können. Nicht selten sind zusätzlich Mehrkörper- oder FEM-Simulationsprogramme integriert, die eine weiterführende Betrachtung und Bewertung des Entwicklungsobjektes gestatten. Mit Hilfe von CAD/CAM-Kopplungen können zudem Bearbeitungsprogramme für NC-Steuerungen abgeleitet werden. Die Unterstützung des Entwicklungsprozesses durch derartige Hilfsmittel wird als *Computer Aided Engineering (CAE)* bezeichnet.

Eine ähnliche Entwicklung ist für die Elektrokonstruktion feststellbar. SCHÄFER & ROLLER (2003) beschreiben den Weg vom Reißbrett über erste rechnerunterstützte Herangehensweisen und den Einsatz von Bauteildatenbanken bis hin zum heute üblichen objektorientierten Konstruieren. Wurden zu Beginn mit Hilfe des Computers lediglich Symbole auf einem Plan platziert und miteinander verbunden, so konnten durch die Verwendung von Bauteildatenbanken den einzelnen Elementen neben einer grafischen Repräsentation auch zusätzliche Informationen beigelegt werden (GRÄTZ 2006). Die neueste Generation an ECAD-Werkzeugen ist durch einen objektorientierten Ansatz geprägt. Diese Systeme erlauben eine durchgängige Projektierung der Elektrik und der Fluidik in einer Entwicklungsumgebung. Die darin enthaltenen Querbeziehungen können einfacher konsistent gehalten werden, da sich Änderungen auf beide Bereiche auswirken. Derartige Softwarewerkzeuge werden ebenfalls als CAE-Lösungen bezeichnet, da sie den Entwicklungsprozess umfassender unterstützen als nur zum Zwecke der Konstruktionsdokumentation.

6.3.2 Nutzbare Informationen der CAD-Systeme für die Simulation

In Abschnitt 2.6.4 wurden die für eine herkömmliche VIBN grundsätzlich notwendigen Daten beschrieben und welche Formen normalerweise zu deren Speicherung gebräuchlich sind. Dabei standen neben den Funktionsbeschreibungen in Prosa vor allem die CAD-Daten im Vordergrund. Im Folgenden soll nun verstärkt darauf eingegangen werden, welche Informationen aus den CAD-Systemen für die Simulation nutzbar sind. Allerdings umfasst dies nicht nur die bisher erläuterten Daten, sondern die Ausführung wird im Hinblick auf die Elemente der vorgestellten Modellierungstechnik gezielt erweitert. Als Voraussetzung wird für die Mechanik der Einsatz moderner 3D-Systeme angenommen. Des Weiteren wird davon ausgegangen, dass die Elektrotechnik und die Fluidik in einem gemeinsamen Werkzeug projiziert sind. In Tabelle 6.1 sind die nutzbaren Informationen getrennt nach MCAD und E/FCAD zusammengefasst.

Das 3D-Modell der Feinplanungsphase enthält lediglich grobe Strukturen, die mittels Volumenprimitiven abgebildet sind. Zur Detaillierung können die Geometrieinformationen des MCAD-Modells herangezogen werden, die sich im Hinblick auf die Parts, die Detektoren und die Bindings auswerten lassen. Ebenso können Bewegungsachsen oder -pfade übernommen werden. Eine Grundbedingung ist allerdings, dass die genannten Modellierungselemente auch im MCAD-Modell explizit hinterlegt sind, siehe Abschnitt 6.3.5 (Konstruktionsrichtlinien). Die Aktor- und die Sensordaten bilden die Schnittstelle zur Elektrotechnik sowie zur Fluidik und werden für die Konsistenzprüfung ausgewertet.

MCAD
<ul style="list-style-type: none"><li>• Bauteilhierarchie</li><li>• Parts: Geometrie, Position und Orientierung</li><li>• Detektoren und Bindings: Geometrie, Position und Orientierung</li><li>• Mover: Start- und Endpunkte, Stützpunkte, Kurven, Längen</li><li>• Aktor- und Sensordaten: Typ</li><li>• Technische Parameter</li></ul>
E/FCAD
<ul style="list-style-type: none"><li>• Betriebsmittel/Bauteile: Typ, Anschlusspunkte, Steuerketten</li><li>• Implizites Verhalten der Aktoren und der Sensoren</li><li>• Adressen der Steuerungsein- und -ausgänge</li><li>• Technische Parameter</li></ul>

Tabelle 6.1: Nutzbare Informationen in den CAD-Systemen

Während die Daten des MCAD-Modells eher statische Aspekte beinhalten, können die im E/FCAD projektierten Betriebsmittel bzw. Bauteile und deren Verschaltungen zu Steuerketten für die Verfeinerung des dynamischen Maschinenverhaltens herangezogen werden. Dabei ergeben sich Verhaltensmerkmale einer Einzelkomponente implizit aus dem Typ des verwendeten Bauteils.

Sowohl aus dem MCAD als auch aus dem E/FCAD können technische Parameter für die Simulation extrahiert werden. Diese sind in Form sogenannter Attribute in den Objekten der CAD-Modelle hinterlegbar. Dem Entwickler ist es dadurch möglich, zusätzliche Informationen zu hinterlegen.

### 6.3.3 Datenaustausch zwischen den CAD-Systemen und dem Funktionsmodell

Der Informationsaustausch zwischen den CAD-Systemen und dem Funktionsmodell kann grundsätzlich auf zwei unterschiedliche Arten geregelt sein, siehe Abbildung 6.2. Zum einen kann das Funktionsmodell direkt über eine Programmierschnittstelle, auch *Application Programming Interface (API)* genannt, mit dem CAD-Werkzeug gekoppelt werden (Abbildung 6.2 links). Zum anderen lassen sich die notwendigen Daten in ein Zwischenformat speichern, welches vom CAD-System geschrieben und anschließend in das Funktionsmodell eingelesen wird (Abbildung 6.2 rechts).

Die meisten modernen CAD-Systeme verfügen über Programmierschnittstellen, die einen umfassenden Zugriff auf das hinterlegte Modell erlauben. Dadurch können, sofern von der API unterstützt, ein Großteil der vorhandenen Informationen extrahiert werden. Nachteilig erweist sich allerdings die werkzeugspezifische Kopplung, da durch diese eine Austauschbarkeit nicht mehr gegeben ist.

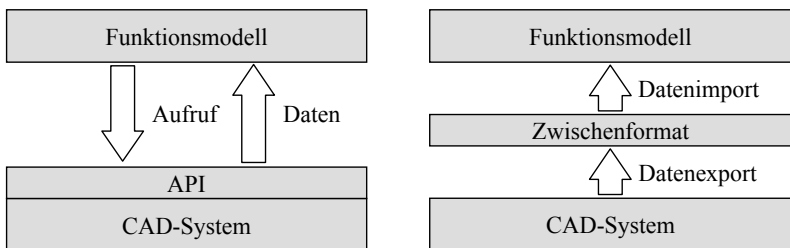


Abbildung 6.2: Kopplungsmöglichkeiten von Funktionsmodell und CAD

Diese Einschränkung besteht bei der Speicherung in einem Zwischenformat nicht. Im Falle einer allgemeingültigen Definition desselben können unterschiedliche CAD-Systeme verwendet werden, vorausgesetzt sie erfüllen die hier postulierten Anforderungen in Bezug auf die zu exportierenden Daten. Von dieser Variante lassen sich wiederum zwei Ausprägungen unterscheiden, siehe Abbildung 6.3. Zum einen können standardisierte Datenformate, die von gängigen CAD-Systemen beherrscht werden, herangezogen werden (Abbildung 6.3 links). Diesbezüglich wäre z. B. der *Standard for the Exchange of Product Model Data (STEP)* zu nennen (ISO 10303-1 1994). Zum anderen ist die Verwendung der Programmierschnittstelle zum Schreiben eines Zwischenformates möglich (Abbildung 6.3 rechts). Dabei kann mit Hilfe eines externen Programmes auf die API zugegriffen werden, die gewünschten Daten können ausgelesen und in einer neutralen Form gespeichert werden.

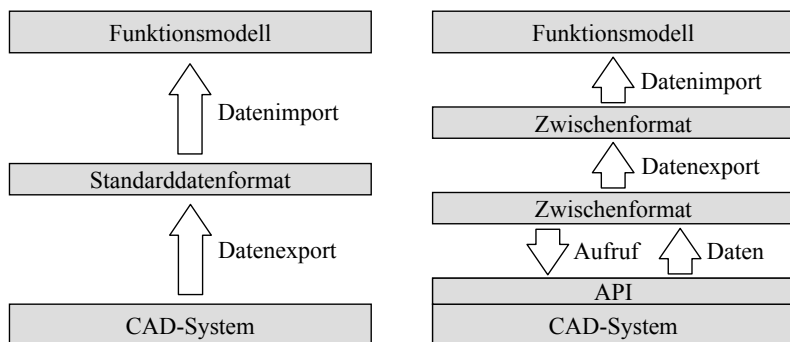


Abbildung 6.3: Integration der CAD-Daten in das Funktionsmodell über Zwischenformate

Im Rahmen dieser Arbeit wird der Ansatz in Abbildung 6.3 rechts über die API und das Zwischenformat verfolgt. Hierfür wurde ein eigens definiertes Austauschmedium für die Mechanik, die Elektrotechnik und die Fluidik entwickelt, siehe Abschnitt 6.3.4. Als Informationsträger kommt die *Extensible Markup Language (XML)* zum Einsatz. Einzige Ausnahme bilden die Geometriedaten der Parts, die zwar ebenso mit Hilfe der API ausgeleitet werden, allerdings als VRML-Modelle. Deren tessellierte Darstellung von 3D-Daten ist vor allem für die Visualisierung und die Kollisionsrechnung geeignet (CAMPAGNA 1999).

Der Grund für das gewählte Vorgehen ist die im Vergleich zu einer direkten Kopplung einfache Substitutionsmöglichkeit von CAD-Werkzeugen. Des Weiteren

ren haben sich Standarddatenformate als unzureichend für die Informationsspeicherung erwiesen, da nicht alle der benötigten Daten ausgeleitet werden können. Vor allem der in der Industrie weit verbreitete STEP-Standard wurde intensiv untersucht. Obwohl es sich dabei um ein umfangreiches Produktmodell handelt, dient dieser meist nur als Austauschmedium für Geometriedaten. In modernen CAD-Werkzeugen kommen aus strategischen Gründen noch immer hersteller-spezifische Datenformate zum Einsatz, da die *individuellen Leistungsmerkmale der Systeme mit dem STEP-Modell nicht unbedingt realisiert werden können* (CUIPER 2000).

### 6.3.4 Definition eines Austauschformates

Im Folgenden wird ein allgemeingültiges Austauschformat auf der Basis der XML vorgestellt, das für den Datentransfer vom CAD zum Funktionsmodell dient. Die vom *World Wide Web Consortium (W3C)* unterstützte XML definiert eine generische Syntax, die Daten durch sogenannte *Tags* auszeichnet (HAROLD & MEANS 2002). Von Vorteil für die hier verfolgten Zwecke sind die Plattform-unabhängigkeit und der hierarchische Aufbau von XML-Dokumenten.

Das Wurzelement der hier genutzten XML-Datei ist das Projekt (engl. *project*). Dieses verfügt über Eigenschaften zur Beschreibung des Projektnamens und des Typs, welcher mechanisch (engl. *mechanical*) oder elektrisch/fluidisch (engl. *electrical/fluidical*) sein kann. Das zentrale Element des Austauschformates ist die Komponente (engl. *component*), die Bauteile aus dem MCAD oder Betriebsmittel und Bestandteile aus dem E/FCAD repräsentiert. Eine Komponente kann sich aus weiteren Komponenten zusammensetzen bzw. selbst Teil einer anderen sein (Dekomposition und Komposition). Dadurch können sowohl der hierarchische Aufbau aus der Mechanik als auch die flache Elementestruktur aus der Elektrotechnik und der Fluidik dargestellt werden. Jede Komponente verfügt über einen Namen sowie eine ID. Sie fungiert als Strukturierungsmittel und kann die XML-Elemente *attribute*, *detector*, *mover* oder *symbol* enthalten, die folgenden Funktionen innehaben:

- *attribute*:  
Dieses Element enthält einen Namen und einen Wert. Darin sind zum einen die Attribute aus den CAD-Systemen hinterlegt. Zum anderen werden für das MCAD die Positionen und die Orientierungen von Bauteilen und für das E/FCAD die E/A-Belegung festgehalten.

- *detector*:

Der Gebrauch dieses Elementes ist lediglich auf die Mechanik beschränkt und bezieht sich auf die im CAD-Modell abgebildeten Detektoren und Bindings. Letztere sind von den Detektoren abgeleitet und werden daher unter diesem Tag zusammengefasst. Neben dem Typ des Volumenprimitivs, z. B. ein Zylinder, sind die zur vollständigen Abbildung erforderlichen Geometrieinformationen aufgeführt.

- *mover*:

Das Element *mover* ist ebenfalls auf die Mechanik ausgerichtet und dient zur Übernahme der Bewegungsachsen und -pfade aus dem CAD. Darin festgehalten sind beispielsweise Start- und Endpunkte von Achsen oder Stützpunkte für Verfahrkurven.

- *symbol*:

Die Verwendung dieses Elements ist nur für die Elektrotechnik bzw. die Fluidik vorgesehen. Der Informationsgehalt umfasst unter anderem die Anschlusspunkte und die Verschaltung der Symbole einzelner Betriebsmittel bzw. Komponenten in den Plänen.

Die Abbildungen 6.4 und 6.5 zeigen einen Auszug des XML-Datenformates, die detaillierte Struktur ist im Anhang nachzulesen.

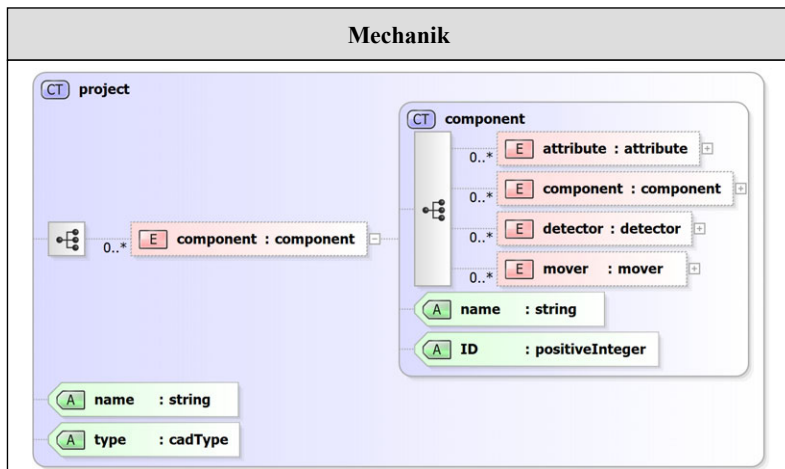


Abbildung 6.4: XML-Schema für den Datenaustausch zwischen Funktionsmodell und MCAD-Systemen



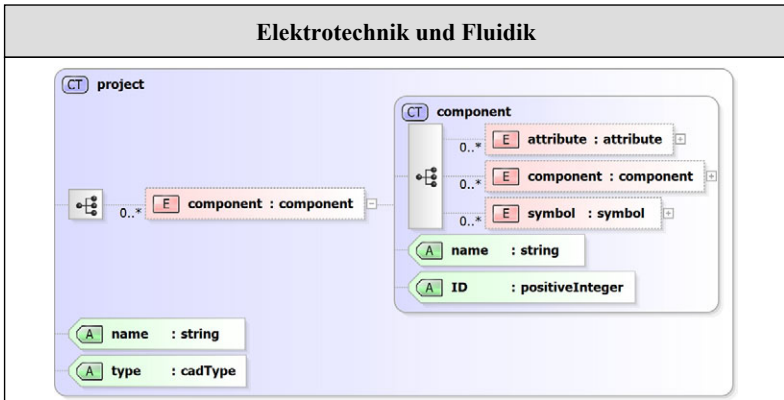


Abbildung 6.5: XML-Schema für den Datenaustausch zwischen Funktionsmodell und E/FCAD-Systemen

### 6.3.5 Konstruktionsrichtlinien

Die Definition von Konstruktionsregeln bzw. -richtlinien dient im Rahmen der vorliegenden Arbeit zur Verfolgung zweier Ziele. Zum einen soll der Automatisierungsgrad der Datenübernahme in das Funktionsmodell erhöht werden. Zum anderen unterstützen die formulierten Vereinbarungen die Konsistenzprüfung, welche Gegenstand von Abschnitt 6.3.7 ist.

Für die Mechanik werden folgende Konstruktionsrichtlinien erlassen:

- Vergabe von eindeutigen Identifikationsnummern (IDs) für jedes Bauteil
- Vergabe einer festen, projektunabhängigen Typnummer für Aktoren und Sensoren
- Einheitliches Koordinatensystem im Funktionsmodell und im MCAD
- Kinematisch korrekte Konstruktion, d. h. Orientierung der Bauteil- und der Baugruppenhierarchie an kinematischen Vorgaben
- Orientierung der Struktur des MCAD-Modells am Funktionsmodell
- Kennzeichnung von Norm- bzw. Standardteilen
- Erstellung und Kennzeichnung der Detektoren und der Bindings

- Erstellung und Kennzeichnung der Bewegungsachsen
- Hinterlegung technischer Daten in den jeweiligen Objektattributen

Die Vergabe einer eindeutigen ID für jedes Bauteil ist zur Identifizierung und zur Erstellung von Querbeziehungen zwischen Funktions- und CAD-Modell dringend erforderlich. Hierfür lassen sich die vom CAD-System automatisch vergebenen Kennzeichnungen nutzen, die von diesem für die modellinterne Referenzierung benötigt werden. Des Weiteren ist jedem Aktor und Sensor eine unverwechselbare und feste Typnummer zu hinterlegen, womit deren technische Ausprägung bestimmt werden kann. Während die ID für jedes CAD-Modell neu erzeugt wird, ist die Typnummer für eine technische Baugruppe projektunabhängig und kann somit gegebenenfalls in einer Bauteilbibliothek hinterlegt werden.

Grundsätzlich gilt es zu beachten, dass die Koordinatensysteme im Funktionsmodell und im CAD gleich ausgerichtet sind. Darüber hinaus wird eine kinematisch korrekte Konstruktion gefordert. Bauteile, die über gemeinsame Bewegungsfreiheitsgrade verfügen, sind in einer Baugruppe abzubilden. Dies unterstützt den automatisierten Export des Geometriemodells, wie er in Abschnitt 6.4.6 noch vorgestellt wird, und verhindert zudem zeitaufwendige manuelle Nacharbeit. Eine generelle Orientierung der Struktur des MCAD-Modells am Funktionsmodell ist sinnvoll, da sich somit auch statische Bestandteile, wie z. B. Maschinengestelle, besser zuordnen lassen.

Gleichfalls wird die Kennzeichnung von Norm- bzw. Standardteilen, wie beispielsweise Schrauben oder Muttern, verlangt, um diese automatisiert entfernen zu können. Derartige Teile sind für die Visualisierung erlässlich und vergrößern die Datenmenge erheblich. Die Kennzeichnung kann einmalig in den Attributen des jeweiligen Objektes erfolgen.

Da Detektoren, Bindings und Bewegungsachsen aus dem CAD übernommen werden sollen, sind diese dort explizit anzulegen, siehe Abbildung 6.6. Detektoren und Bindings werden als eigenständige Elemente im CAD-Modell definiert, die Bewegungsachsen hingegen lassen sich dem zu verfahrenenden Objekt zuordnen. Die erläuterte Herangehensweise hat den Vorteil, dass die Festlegung der Geometrie sowie die Platzierung und die Orientierung im CAD-Werkzeug wesentlich präziser und zudem einfacher vorgenommen werden können. Weiterhin müssen technische Parameter, wie etwa die Übersetzung einer Getriebestufe, durch den Entwickler in den Attributen des Bauteils definiert werden.

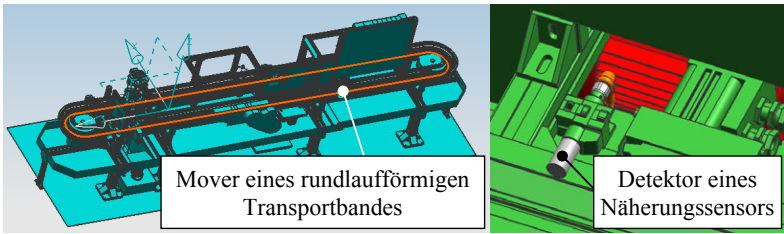


Abbildung 6.6: Anlegen von Achsen, Detektoren und Bindings im CAD

Die Richtlinien für das E/FCAD lassen sich wie folgt zusammenfassen:

- Vergabe von eindeutigen IDs für jedes Betriebsmittel bzw. Bauteil
- Vergabe einer festen, projektunabhängigen Typnummer für Aktoren und Sensoren
- Anlegen von Modellbibliothekselementen
- Hinterlegung von technischen Daten in den Attributen
- Hinterlegung der Steuerungsein- und -ausgänge in den Attributen









Jedes Betriebsmittel bzw. Bauteil muss über eine eindeutige ID verfügen. Hierzu kann, wie es im gleichen Maße für das MCAD möglich ist, die systemeigene ID des E/FCAD-Werkzeuges verwendet werden. Ebenso sind die Typnummern für die Elemente der Steuerketten anzulegen. Da hierfür softwareseitig in der Regel Datenbanken zur Verfügung stehen, ist dies ein einmaliger Vorgang. Jede Typnummer repräsentiert exakt ein Bauteil, für welches ein Simulationsmodell in der Modellbibliothek zu erstellen ist, das im Zuge der automatisierten Detaillierung in das Funktionsmodell eingebunden wird. Technische Daten, die für die Modellerstellung notwendig sind, wie beispielsweise die im Regelgerät eines Frequenzumrichters hinterlegten Beschleunigungsrampen, müssen als Attribute im zugehörigen Objekt abgelegt werden. Selbiges trifft auf Steuerungsein- und -ausgänge zu, wodurch diese eindeutig einer Komponente zugeordnet sind.

### 6.3.6 Querbeziehungen zwischen Funktions- und Konstruktionsebene

Für eine Datenübernahme aus dem CAD in das Funktionsmodell müssen die Zusammenhänge eindeutig und softwaretechnisch auswertbar gestaltet sein. Auf Grund der unterschiedlichen Strukturen der einzelnen Modellkategorien ist eine

geeignete Beschreibungsform erforderlich. In der vorliegenden Arbeit werden dafür IDs verwendet (siehe Konstruktionsrichtlinien in Abschnitt 6.3.5) und ein Modell eingeführt, das die Querbeziehungen verwaltet. Dieses wird im Folgenden als Mapping-Modell bezeichnet.

Jedes Element im Funktionsmodell, sei es eine Komponente, ein Detektor oder ein Port, verfügt über eine eindeutige ID in Form einer natürlichen Zahl, die automatisch vergeben wird. Ebenso werden die von den CAD-Werkzeugen systemintern generierten Kennzeichnungen genutzt. Das Mapping-Modell dient zur Erstellung von Beziehungen zwischen einzelnen Elementen in den jeweiligen Modellkategorien. Die hierarchisch unterschiedlich strukturierten Modelle können auf diese Art und Weise miteinander verbunden werden. Folgende Teile des Funktionsmodells dürfen den Elementen in den CAD-Modellen zugewiesen werden (zugehörige Icons sind mit dargestellt):

- TRM 
- Aktoren , Sensoren   
des TRMs
- Parts   
- Detektoren und Bindings 
- Mover 

Die Erstellung der Querbeziehungen obliegt dem Entwickler, wobei für diesen die IDs im Verborgenen bleiben und eine entsprechende Eingabemaske den Vorgang unterstützt, siehe Abbildung 6.7. Die Verbindungsinformationen dienen dem automatisierten Datenimport in das Funktionsmodell. Weiterhin werden die Aktoren und die Sensoren des TRMs im Mapping-Modell für die Konsistenzprüfung zwischen den CAD-Modellen und dem Funktionsmodell benötigt.

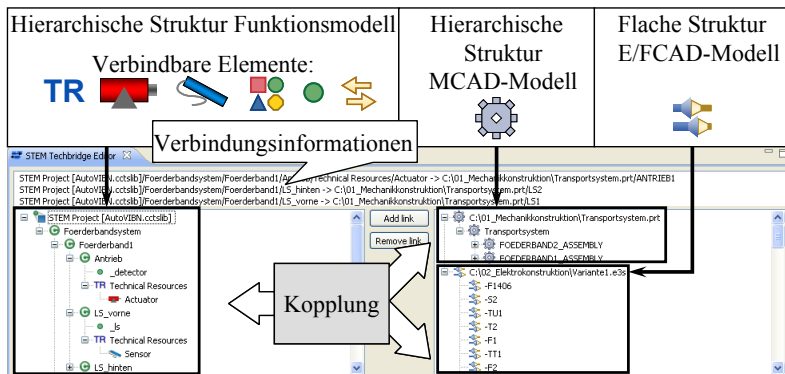


Abbildung 6.7: Modell der Beziehungen zwischen Funktionsmodell und CAD

### 6.3.7 Konsistenz von Funktions- und Konstruktionsmodellen

Der Begriff der Konsistenz wird auch mit der Widerspruchsfreiheit gleichgesetzt (BUCHER 1998). Bezogen auf den vorliegenden Anwendungsfall bedeutet dies, dass die Daten im Funktionsmodell und in den CAD-Systemen denselben Entwicklungsstand beschreiben müssen. Auf Grund dessen werden ausgewählte Eigenschaften verglichen bzw. überprüft. Die Konsistenz lässt sich zum einen zwischen dem MCAD- und dem E/FCAD-Modell testen. Die Schnittstelle zwischen beiden bilden die Aktoren und die Sensoren. Deren Querbeziehungen sind durch das Mapping über das Funktionsmodell abgebildet. Zum anderen kann die Konsistenz zwischen CAD und Funktionsmodell untersucht werden.

Zunächst ist sicherzustellen, dass alle Aktoren und Sensoren in den drei Modellkategorien repräsentiert sind, siehe Abbildung 6.8. Mit Hilfe der Typnummer kann ermittelt werden, ob deren technische Ausprägungen im MCAD und im E/FCAD übereinstimmen. Dieser Schritt ist vor allem für die Abstimmung zwischen den Konstruktionsabteilungen von Bedeutung, da sich deren Einflussbereiche hinsichtlich der Aktoren und der Sensoren schneiden. Hierdurch kann ein Divergieren der Entwicklungsergebnisse minimiert werden. Ebenso ist ein Abgleich der Steuerungsein- und -ausgänge in den TRMs des Funktionsmodells mit denen im E/FCAD-Modell möglich.

Denkbar sind weitere Konsistenztests, die Eigenschaften des Funktionsmodells zu denen der CAD-Modelle in Relation stellen. Beispielsweise kann die Position eines Sensors im Funktionsmodell mit der im CAD unter Berücksichtigung definierter Toleranzen verglichen werden.

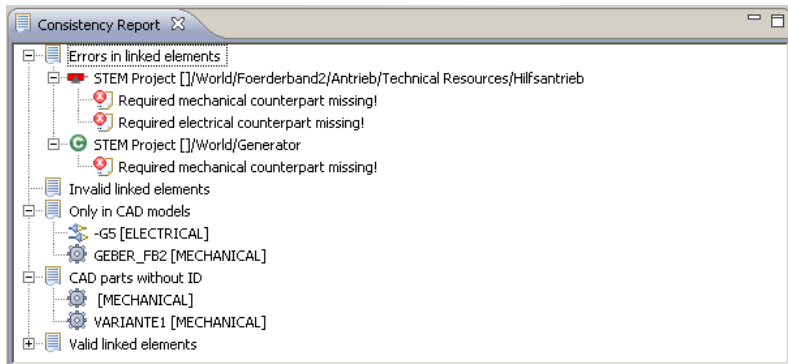


Abbildung 6.8: Konsistenzprüfung der Aktoren und der Sensoren

Das Ziel der hier beschriebenen Konsistenzprüfung ist die Identifikation von Abweichungen zwischen den unterschiedlichen Modellkategorien. Eine automatisierte Beseitigung dieser Ungleichheiten wird jedoch nicht angestrebt. Die erkannten Inkonsistenzen sind von den Entwicklern zu diskutieren und anschließend zu beheben.

## 6.4 Automatisierte Erweiterung des Funktionsmodells mit MCAD-Daten

### 6.4.1 Konzept für die Integration der MCAD-Daten

In diesem Abschnitt wird die Einbindung der MCAD-Daten in das Funktionsmodell der Feinplanungsphase vorgestellt. Das Grundprinzip basiert auf der Verfeinerung einzelner Elemente einer Komponente im Funktionsmodell, wie es Abbildung 6.9 verdeutlicht. Die Zuordnung von Objekten aus dem MCAD zu Bestandteilen des Funktionsmodells ist mit Hilfe des Mapping-Modells vorzunehmen. Integriert werden Parameter für die hybriden Automaten, die Geometrie, die Position und die Orientierung von Parts, Detektoren und Bindings sowie die detaillierten Verläufe der Bewegungsachsen.

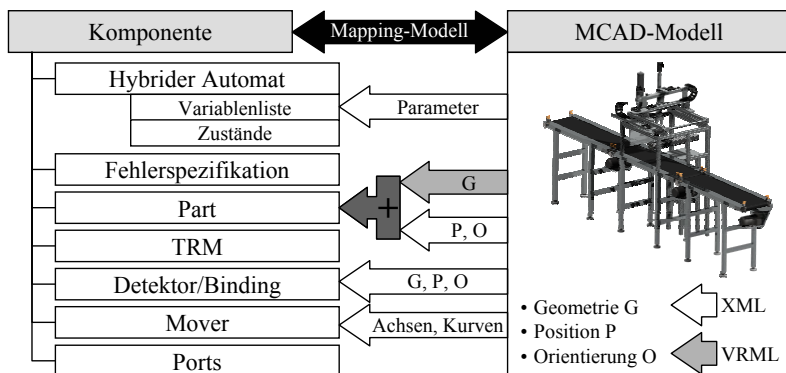


Abbildung 6.9: Automatisierte Integration von MCAD-Daten

Der Datentransfer basiert auf dem vorgestellten XML-Format, lediglich die Geometrie der Parts wird als VRML-Datei übergeben. Weiterhin ist der Zugriff auf das Funktionsmodell geregelt, da es ansonsten zu Informationsverlusten

kommen kann. Überschneidungen ergeben sich im Bereich der Aktorik und der Sensorik. Aus diesem Grund wird ein Ansatz präsentiert, der eine voneinander unabhängige Erweiterung mit MCAD- und E/FCAD-Daten erlaubt.

Das Modell der Feinplanungsphase gliedert Aktoren in eine Steuer- und Stellkomponente sowie in ein Arbeitsglied auf. Die Daten aus dem MCAD-Modell werden ausschließlich in das Arbeitsglied integriert und ersetzen nur die in Abbildung 6.10 dargestellten Modellelemente. Welche Bestandteile des Aktor-modells durch die E/FCAD-Daten erweitert werden, ist in Abschnitt 6.5.1 genauer erläutert.

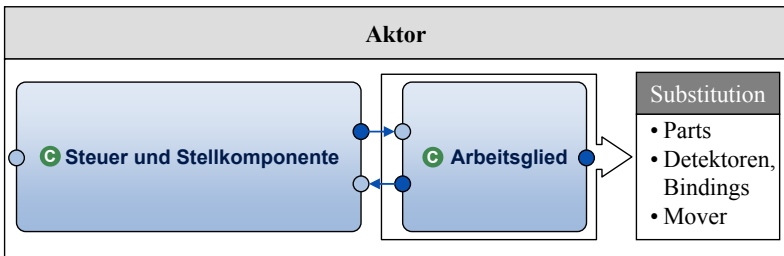


Abbildung 6.10: Erweiterung von Aktoren im Funktionsmodell mit Informationen aus dem MCAD-System

Da Sensoren gemäß der Modellierungsrichtlinie der Feinplanungsphase als eigenständige Komponenten abzubilden sind, ist der Zugriff auf deren einzelne Bestandteile über den gleichen Ansatz wie für die Arbeitsglieder der Aktoren geregelt. Bezüglich des Modellverhaltens der Sensoren sind von Seiten der Mechanik vor allem deren geometrische Erfassungsbereiche von Interesse. Als Beispiel sei der Schaltbereich eines Näherungssensors genannt, welcher mit Hilfe von Detektoren dargestellt wird.

Auf Grund der Tatsache, dass die Informationen zur Detaillierung der Wirkketten ausschließlich aus dem MCAD-Modell bezogen werden, ergeben sich keine Berührungspunkte mit dem E/FCAD. Der Erweiterungsmechanismus für die entsprechenden Komponenten gleicht dem der Aktoren und der Sensoren, außer dass auf die Variablen der Automaten zugegriffen werden darf.

### 6.4.2 Modelltransformation mit Hilfe der MCAD-Daten

Hinsichtlich der Erweiterung der Funktionsmodelle handelt es sich um eine Transformation, wobei nach STAHL & VÖLTER (2005) drei Arten unterschieden werden können:

- Modellmodifikation
- Modelltransformation
- Model-Weaving

Die Modellmodifikation ist dadurch gekennzeichnet, dass das Ausgangsmodell geändert wird. Die Anpassungen basieren auf dem gleichen Metamodell (STAHL & VÖLTER 2005). In Abbildung 6.11 ist dies beispielhaft verdeutlicht, es werden lediglich neue Instanzen im Modell erzeugt.

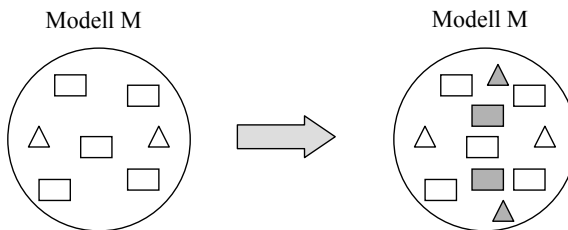


Abbildung 6.11: Prinzip der Modellmodifikation nach STAHL & VÖLTER (2005)

Im Gegensatz dazu wird im Falle der Modelltransformation ein neues, vom Ausgangsmodell unterschiedliches Modell generiert. Das abgeleitete Modell kann dabei über ein anderes Metamodell verfügen. Das ursprüngliche Modell bleibt erhalten (STAHL & VÖLTER 2005). In Abbildung 6.12 ist hierfür ein Beispiel von STAHL & VÖLTER (2005) angegeben, *beide Modelle (M und Q) sind Instanzen unterschiedlicher Metamodelle*.



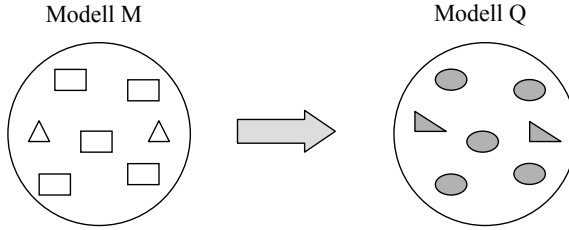


Abbildung 6.12: Prinzip der Modelltransformation nach STAHL & VÖLTER (2005)

Das Model-Weaving beschreibt einen Ansatz, bei dem *verschiedene Modelle [...] nach dem Einlesen und vor der Generierung oder Interpretierung miteinander verlinkt werden* (STAHL & VÖLTER 2005). In Abbildung 6.13 ist dies exemplarisch dargestellt. Gemäß STAHL & VÖLTER (2005) müssen die Modellinstanzen (in Abbildung 6.13 beispielsweise die Modelle M und Q) nicht zwangsläufig über dasselbe Metamodell verfügen.

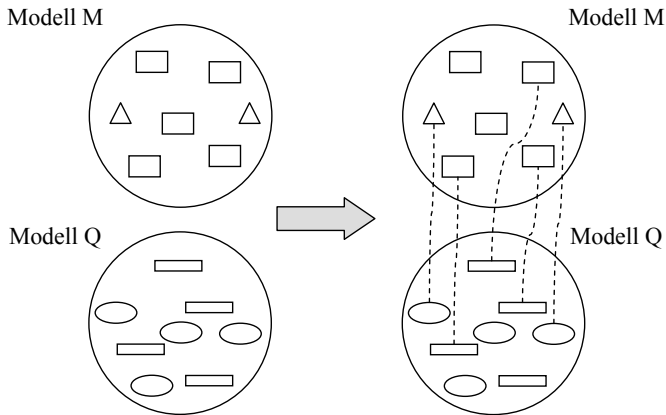


Abbildung 6.13: Prinzip des Model-Weavings nach Stahl & Völter (2005)

Die Transformation des im Rahmen dieser Arbeit verwendeten Funktionsmodells ist in Abbildung 6.14 verdeutlicht. Das Ausgangsgangsmodell und das generierte Funktionsmodell verfügen über das gleiche Metamodell. Die MCAD-Daten werden dazu verwendet, um einzelne Elemente des Funktionsmodells (Variablen der hybriden Automaten, Bewegungsachsen, Detektoren und Bindings, Parts) zu detaillieren. Die Verbindung erfolgt über das bereits in Abschnitt 6.3.6 vorgestellte

Mapping-Modell. Die darin hinterlegten Elementepaare sind über deren IDs eindeutig identifizierbar. Die im MCAD enthaltenen Informationen werden im Zuge der Modelltransformation angepasst und in das erweiterte Funktionsmodell eingebunden.

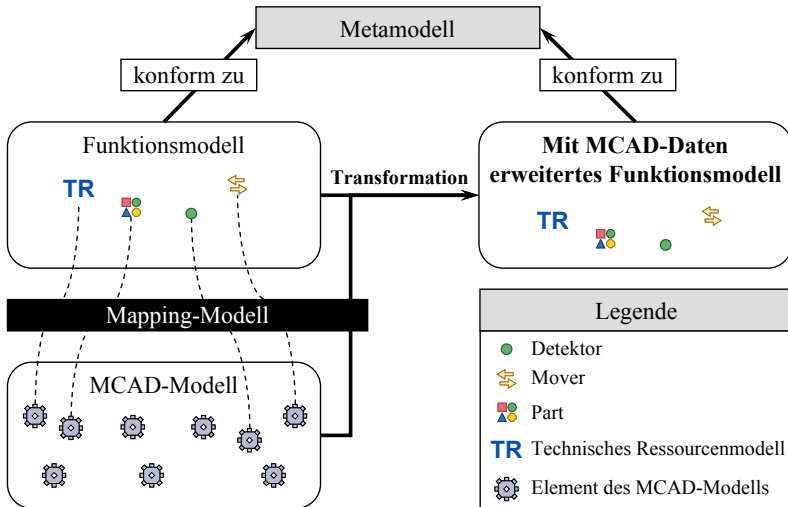


Abbildung 6.14: Transformation des Funktionsmodells durch Integration der MCAD-Daten

Zur Integration der MCAD-Daten sind Transformationsregeln notwendig, nach denen das Funktionsmodell erweitert wird. Diese werden in den nachfolgenden Abschnitten beschrieben. Für die Darstellung werden teilweise Klassen- und Objektdiagramme verwendet. In den Grafiken werden unter anderem auch Ausschnitte des Metamodells der Modellierungstechnik erläutert, die partiell bzw. in ähnlicher Weise bereits vorab in AUTOVIBN (2010) und von HUMMEL (2011) veröffentlicht wurden.

## 6.4.3 Übernahme von Modellparametern

### 6.4.3.1 Beschreibung der Parameteranpassung

Im Funktionsmodell der Feinplanungsphase werden eine Reihe mechanischer Baugruppen projiziert, wie z. B. Getriebe oder Förderbänder. Die Auslegung

dieser Elemente findet im Zuge der mechanischen Konstruktion statt. Daher ist im MCAD-Modell eine Vielzahl von technischen Parametern hinterlegt, die für die Erweiterung des Funktionsmodells von Bedeutung sind. Die verhaltensbestimmenden Variablen des Funktionsmodells sind in den hybriden Automaten abgebildet. Jede Zustandsmaschine verfügt neben den Zuständen und den Transitionen über eine Variablenliste (siehe Metamodell in Abschnitt 5.3.4.1). Auf Letztere kann der Automat für Speicher- oder Berechnungsvorgänge zugreifen. Die aus dem MCAD-Modell entnommenen Daten, z. B. eine Getriebeübersetzung, werden automatisch in die Variablenliste geschrieben, siehe Abbildung 6.15. Dieser Vorgang beschränkt sich auf die Komponenten der Wirkketten und tangiert somit nicht die Integration der E/FCAD-Daten. Eine eindeutige Zuordnung zwischen der Komponente im Funktionsmodell und dem Bauteil im MCAD-Modell ist notwendig. Die Verknüpfung erfolgt mit Hilfe des Mapping-Modells. Dafür ist der Komponente im Funktionsmodell ein TRM (ohne Erweiterung für die Aktoren und die Sensoren) zuzuweisen, um diese als technische Ressource zu kennzeichnen. Die Verbindung des TRMs mit einem Bauteil im MCAD muss im Mapping-Modell vorgenommen werden. Für eine automatische Integration von Parametern eines CAD-Bauteiles sind zwei Randbedingungen einzuhalten. Die Variablen müssen als Attribute im CAD-Modell definiert werden, so dass diese automatisch im XML-Austauschformat gespeichert sind. Des Weiteren gilt eine Namenskonvention: Der Parameter im Funktionsmodell und das zugehörige Attribut im MCAD-Modell müssen über die gleiche Bezeichnung verfügen.

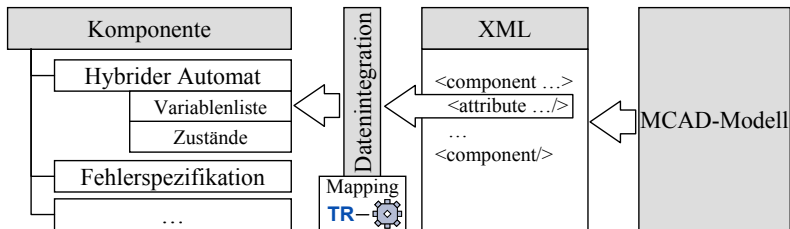


Abbildung 6.15: Integration von technischen Parametern aus dem MCAD-Modell in das Funktionsmodell

Am Beispiel eines Förderbandes soll der Nutzen des Vorgehens verdeutlicht werden. Der Durchmesser und die Drehzahl der Antriebsrolle bestimmen die Geschwindigkeit des Fördergutes auf dem Band. Die vom Winkelschritttegeber erfasste Drehzahl an der gegenüberliegenden Umlenkrolle ist wiederum abhängig

von deren Radius. Werden diese beiden geometrischen Parameter (die beiden Durchmesser) variiert, so ändern sich die tatsächliche und die gemessene Geschwindigkeit des Förderbandes bei gleichbleibender Antriebsdrehzahl des Motors. Ohne genaue Berücksichtigung der geometrischen Verhältnisse kommt es folglich zu Fehlern im Modell.

Das größte Potenzial zur Aufwandseinsparung ist bei parametrisch-assoziativen Konstruktionen gegeben. Diese sind dadurch gekennzeichnet, *dass die Geometrie anstatt mit vielen Einzelmaßen durch Parameter und Formeln beschrieben wird*, eine assoziative Konstruktion enthält zudem einen *aktualisierbaren Bezug* geometrischer Elemente (KLEPZIG & WEISSBACH 2005). Eine Änderung der Parameter führt somit zur Anpassung der Geometrie. Im vorliegenden Fall würde eine Parametervariation darüber hinaus eine automatisierte Angleichung des Funktionsmodells nach sich ziehen.

### 6.4.3.2 Transformationsregeln für Parameter

Zur Einbindung von Parametern aus dem MCAD-Modell ist zunächst eine eindeutige Identifizierung des hybriden Automaten erforderlich, dessen Variablenliste angepasst werden soll. Dieser kann über das Mapping-Modell ermittelt werden, da darin das TRM der zugehörigen Komponente direkt mit dem Element aus dem MCAD verbunden werden kann, siehe Abbildung 6.16. Die Änderungen wirken sich lediglich auf die Variablenwerte des hybriden Automaten aus, die Zustände und die Transitionen bleiben davon unberührt. In einem nächsten Schritt werden die betroffenen Variablen ausfindig gemacht. Namensgleiche Variablen des Automaten und in den Attributen des MCAD-Elementes werden angeglichen. Dabei darf keine Typveränderung vorgenommen werden (Beispiel: ein Double-Wert im MCAD darf nur einem Double-Wert im Funktionsmodell zugeordnet werden), was vorab durch eine Prüfung sicherzustellen ist. Im Anschluss wird der Wert der Variable (initialValue) im Funktionsmodell auf den Wert der gleichen Variable im MCAD-Modell gesetzt.

In Abbildung 6.16 links unten ist hierzu ein einfaches Beispiel in Form eines Getriebes verdeutlicht. Der hybride Automat berechnet die Ausgangsdrehzahl *Nout* mit Hilfe der internen Variable *\_i*, die das Übersetzungsverhältnis widerspiegelt. Hierzu wird zunächst das verbundene MCAD-Element auf die Existenz einer Variablen mit dem Namen *\_i* durchsucht. Ist im MCAD ebenfalls eine Variable mit der gleichen Kennzeichnung definiert, so wird überprüft, ob beide Grö-

Ben vom Typ Double sind. Ist dies der Fall, so kann der Wert für die Übersetzung aus dem MCAD übernommen werden.

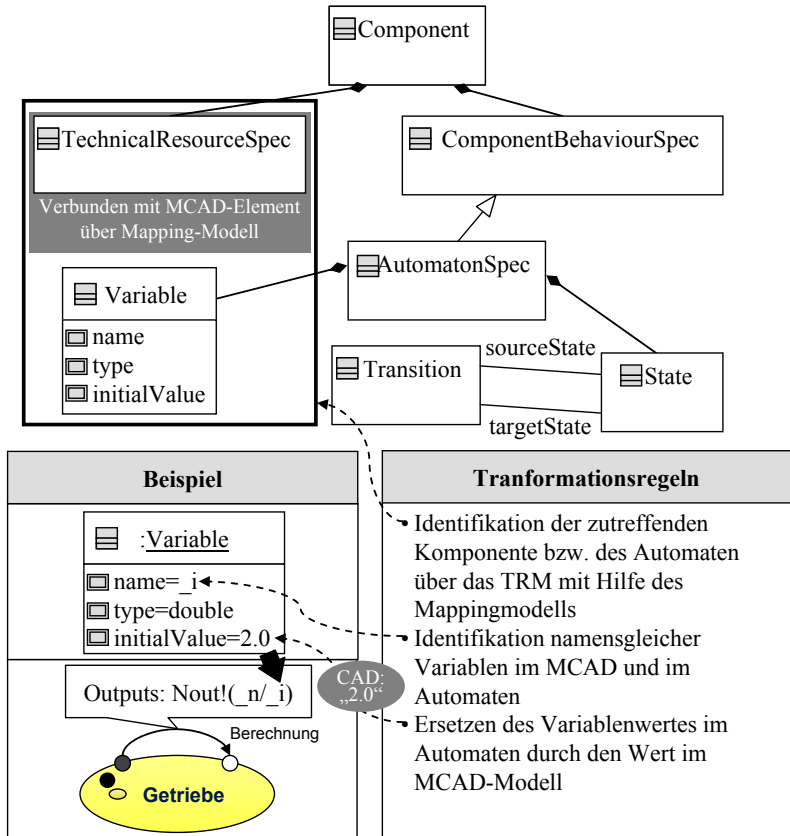


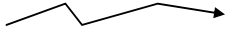
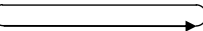

Abbildung 6.16: Transformationsregeln zum Einbinden der Parameter aus dem MCAD-Modell in die Variablenliste der hybriden Automaten – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011)

### 6.4.4 Kinematische Freiheitsgrade

#### 6.4.4.1 Beschreibung der Übernahme von Bewegungsachsen und -pfaden

Die Mover sind optionaler Bestandteil einer Komponente und legen die kinematischen Freiheitsgrade im Raum fest. Mit Hilfe eines MCAD-Werkzeuges lassen sich die Verfahrachsen wesentlich genauer definieren, als dies im Funktionsmodell möglich ist. Daher werden im Rahmen der vorgestellten Methode die Mover des Feinplanungsmodells mit den genauen Werten aus dem MCAD-Modell erweitert, siehe Abbildung 6.17. Die Voraussetzung dafür ist, dass die Verfahrkurven explizit im CAD-Modell angelegt werden (siehe Konstruktionsrichtlinien in Abschnitt 6.3.5). Die Verbindung zur entsprechenden Komponente im MCAD-Modell ist, wie auch bei der Parameterübernahme in Abschnitt 6.4.2, über das Mapping-Modell realisiert. Dazu ist es erforderlich, den Mover der Komponente mit dem Bauteil im MCAD in Relation zu setzen.

Grundsätzlich kann zwischen translatorischen und rotatorischen Achsen unterschieden werden. Die Verfahrbereiche beider Achstypen werden durch einen Start- und einen Endpunkt definiert. Die positive Bewegungs- bzw. Drehrichtung ist durch die Ausrichtung der Achse bestimmt. Komplexere Verfahrbewegungen erfordern interpolierte Bewegungspfade. Im Funktionsmodell werden drei unterschiedliche Interpolationsmechanismen unterstützt:

- Lineare Elemente in Verkettung (Polygonzug) 
- Lineare Elemente in Verbindung mit Kreisbögen 
- Splines 

Die Achstypen müssen entsprechend in den Bauteilen des MCAD-Modells abgebildet werden. Alle zur automatisierten Erweiterung notwendigen Daten sind im XML-Austauschformat gespeichert. Vor der Informationsübernahme werden die Interpolationsarten auf Konsistenz bzw. Übereinstimmung in der jeweiligen Modellkategorie geprüft. Im Anschluss daran werden die Start- und die Endpunkte sowie eventuell vorhandene Stützpunkte in den Movern der Komponenten ersetzt.

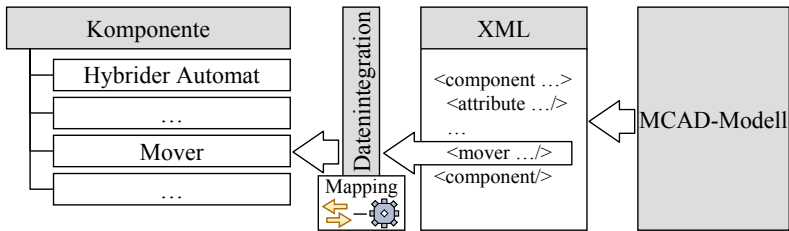


Abbildung 6.17: Integration der Bewegungsachsen aus dem MCAD-Modell in das Funktionsmodell

#### 6.4.4.2 Transformationsregeln für die kinematischen Freiheitsgrade

Wie bereits im vorangegangenen Abschnitt erläutert, sind die Mover des Funktionsmodells über das Mapping-Modell mit dem MCAD verbunden. Ein Mover kann wiederum unterschiedliche Achstypen enthalten. Gemäß dem Metamodell sind Linear- und Rotationsachsen sowie interpolierte Bewegungspfade definiert, siehe Abbildung 6.18 oben links (AUTOVIBN 2010, HUMMEL 2011). Der Achstyp wird während der Modelldetaillierung nicht geändert, da dies eine grundlegende Entscheidung ist, die übergreifend zunächst im Funktionsmodell geklärt werden sollte. Ähnlich wie bei den Variablen der hybriden Automaten wird jedoch geprüft, ob die zu integrierenden Daten den richtigen Achstyp beschreiben (Prüfung ob es sich um eine *LinearAxis*, *RotatoryAxis* oder eine *InterpolatedAxis* handelt). Bei den interpolierten Verfahrenskurven ist zudem sicherzustellen, dass die Interpolationsart von Funktions- und MCAD-Modell übereinstimmen (LINEAR, LINE\_CIRCLE oder SPLINE). Mit Hilfe der Konstruktionsdaten im MCAD werden die einzelnen Achsverläufe im Funktionsmodell an die realen Verhältnisse angepasst.

Da die Mover im MCAD-Modell in die Bauteile integriert sind, können Erstgenannte translatorisch und rotatorisch mit dem jeweiligen CAD-Element verschoben werden. So wird beispielsweise der Mover samt Linearachse eines Förderbandes im Falle einer Neupositionierung mit diesem platziert. Der eben beschriebene Aspekt kann im Funktionsmodell über die Komponenten berücksichtigt werden, da diese gemäß Metamodell von *TransformableElement* erben und somit translatorische und rotatorische Verschiebungen definiert sind, siehe Abbildung 6.18 oben rechts (AUTOVIBN 2010, HUMMEL 2011).

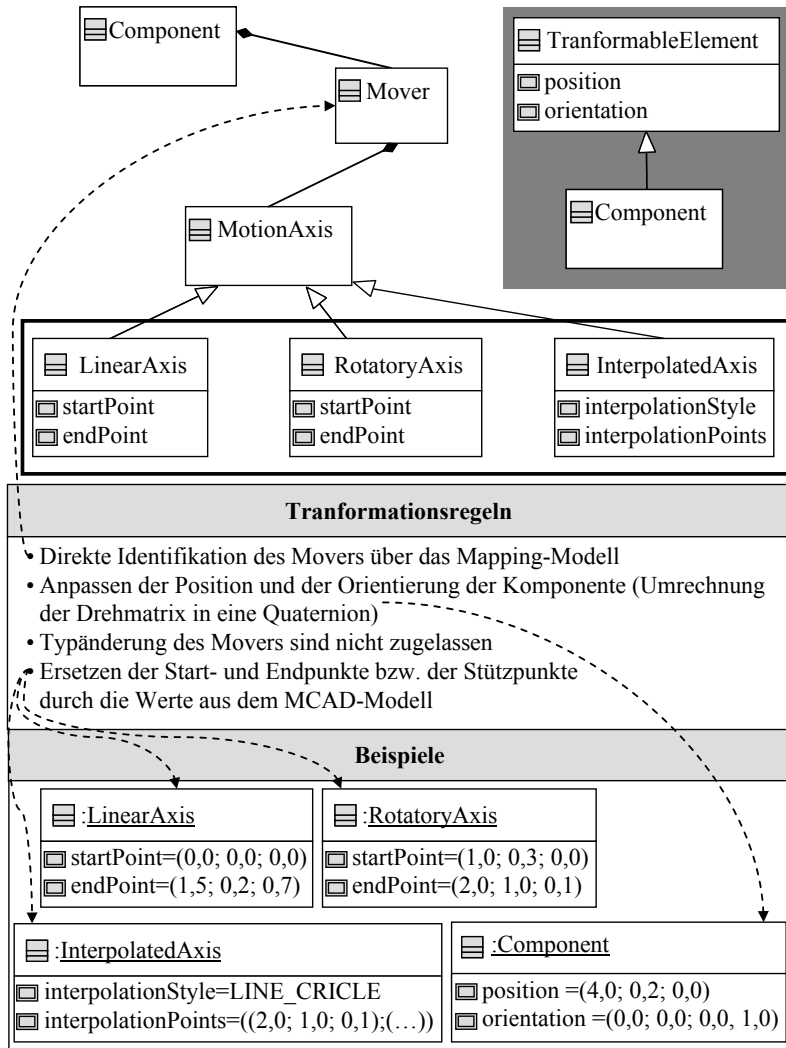


Abbildung 6.18: Transformationsregeln zum Anpassen der Bewegungsachsen und -pfade mit Daten aus dem MCAD-Modell – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011)



Die Daten hierfür werden gleichfalls aus dem MCAD-Modell übernommen. Da im Funktionsmodell Komponentenverschiebungen und -drehungen mittels Quaternionen definiert werden, müssen die Drehmatrizen aus dem MCAD umgerechnet werden. Hinsichtlich der Quaternionen handelt es sich um *verallgemeinerte komplexe Zahlen der Form*  $q = w + ix + jy + kz$  mit den Imaginäranteilen  $i$ ,  $j$ , und  $k$ , die besonders *bei der Hintereinanderausführung von Drehungen* im dreidimensionalen Raum geeignet ist (BRONSTEIN ET AL. 2008).

In Abbildung 6.18 unten werden Beispiele für eine Linear- und für eine Rotationsachse sowie für eine interpolierte Achse dargestellt. Für die beiden Erstgenannten werden die Start- und die Endpunkte (Attribute *startPoint* und *endPoint*) definiert. Die interpolierte Achse, im vorliegenden Anwendungsfall besteht diese aus Kurven und Geraden (Attribut *interpolationStyle=LINE\_CIRCLE*), erfordert die Hinterlegung von Stützpunkten.

### 6.4.5 Detektoren und Bindings

#### 6.4.5.1 Beschreibung der Integration von Detektoren und Bindings

Ebenfalls kann durch die Berücksichtigung der exakten Geometriedaten von Detektoren und Bindings die Aussagekraft des Funktionsmodells gesteigert werden. Der Importmechanismus ist für beide Elemente gleich gestaltet, zumal die Bindings von den Detektoren abgeleitet sind. Für einen automatisierten Informationstransfer müssen diese gemäß den Konstruktionsrichtlinien explizit im MCAD-Modell angelegt und positioniert werden. Die geometrische Modellierung ist anhand von Volumenprimitiven, wie z. B. mit Quadern oder Zylindern, vorzunehmen. Die Kopplung zum MCAD basiert wiederum auf dem Mapping-Modell. Mit Hilfe des XML-Formates werden die Daten ausgetauscht, wobei dies neben der geometrischen Form auch die Position und die Orientierung im Raum umfasst. Im Anschluss werden die Detektoren und die Bindings im Funktionsmodell mit den entsprechenden Informationen erweitert, siehe Abbildung 6.19.

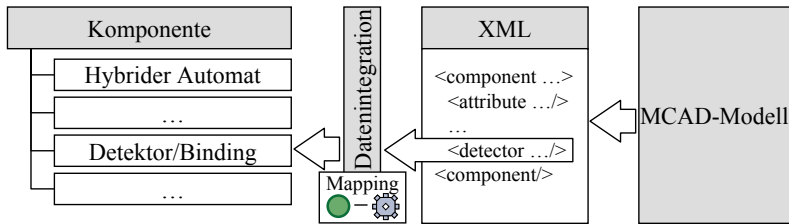


Abbildung 6.19: Integration der Detektoren aus dem MCAD-Modell in das Funktionsmodell

#### 6.4.5.2 Transformationsregeln für Detektoren und Bindings

Detektoren und Bindings erben wie die Komponente von *TransformableElement* (AUTOVIBN 2010, HUMMEL 2011). Dadurch können diese im dreidimensionalen Raum beliebig platziert werden. In Abbildung 6.20 ist das Vorgehen exemplarisch für die Detektoren dargestellt, jedoch in gleicher Weise für die Bindings gültig. Auf die Besonderheiten der Zuletztgenannten wird am Ende dieses Abschnittes noch explizit eingegangen. Detektoren und Bindings können Subgeometries enthalten. Hierfür stehen die Elemente Quader (Box), Zylinder (Cylinder) und Kugel (Sphere) zur Verfügung. Komplizierte Strukturen können prinzipiell über Netze (Mesh) durch den Import von VRML-Daten abgebildet werden. Allerdings kommen die Netze nicht in der Feinplanungsphase zum Einsatz (siehe Modellierungsregeln in Abschnitt 5.5.3), da in frühen Entwicklungsstadien in der Regel noch keine CAD-Daten der zu projektierenden Maschine zur Verfügung stehen.

Die Detektoren und die Bindings werden direkt im Mapping-Modell referenziert und können über dieses eindeutig erkannt werden. Zunächst werden für die Modelltransformation sämtliche Subgeometries werden entfernt, so dass der Detektor bzw. das Binding als leere „Hülle“ für die einzufügenden Geometrieelemente zur Verfügung steht. Im Anschluss werden über die MCAD-Daten die neuen Volumenprimitive erzeugt. Hierfür werden die oben erläuterten Elemente *Würfel*, *Zylinder* und *Kugel* genutzt, für welche die Attribute Höhe (height), Weite (width), Länge (length) und Radius (radius) zu spezifizieren sind. Aus dem MCAD-Modell wird für diese Zwecke vorab der Typ des Volumenprimitives extrahiert und in der Folge ein passendes Element im Funktionsmodell angelegt, bevor abschließend die Parametrierung erfolgen kann. Da die Komponenten, die Detektoren, die Bindings und die Subgeometries von *TransformableElement* er-

ben, müssen wiederum Positionen und Orientierungen definiert werden. Wie schon für die kinematischen Freiheitsgrade erläutert, müssen hierfür die MCAD-Informationen entsprechend umgerechnet werden (z. B. die Drehungen mit Hilfe der Quaternionen).

In den oben dargelegten Ausführungen wurde bereits mehrfach erwähnt, dass die MCAD-Daten nicht als VRML-Modelle in die Detektoren und Bindings rückgeführt werden. Dies resultiert zum einen daher, dass Erkennungs- und Einflussbereiche ausreichend genau durch die Volumenprimitive dargestellt werden können. Der Auslösebereich einer Lichtschranke kann beispielsweise als Zylinder oder dünner Quader modelliert werden, was die realen Verhältnisse befriedigend abbildet. Komplexe Geometrien sind dafür nicht notwendig. Zum anderen sind die oben genannten Grundelemente hinsichtlich der Datenmenge wesentlich kompakter als die VRML-Netze. Letztere würden sich negativ auf die Performanz der Simulation auswirken und diese verlangsamen. Im Beispiel der Lichtschranke würde ansonsten auch der Zylinder oder der Quader in ein datenintensiveres Netz umgewandelt werden. Sollte dennoch eine genauere geometrische Abbildung von Detektoren und Bindings erforderlich sein, so können diese auch manuell importiert werden, ein Automatismus ist im hier geschilderten Ansatz nicht vorgesehen.

Im Zuge der Modelltransformation werden die grundlegenden Eigenschaften der Bindings nicht geändert. Wie in Abschnitt 5.3.5 ausgeführt, existieren unterschiedliche Arten von Bindungen (MoverBindings und PortBindings), zudem ist über das Attribut *Priority* die Rangfolge im Falle eines Mehrfachzugriffes geregelt (siehe auch AUTOVIBN 2010 oder HUMMEL 2011). Die genannten Charakteristika des zu transformierenden Bindings werden nicht modifiziert, die Anpassungen beziehen sich lediglich auf die enthaltenen Subgeometries. Das Verhalten des erweiterten Funktionsmodells ist demnach konform zum Ausgangsmodell, jedoch in der geometrischen Abbildung verfeinert.

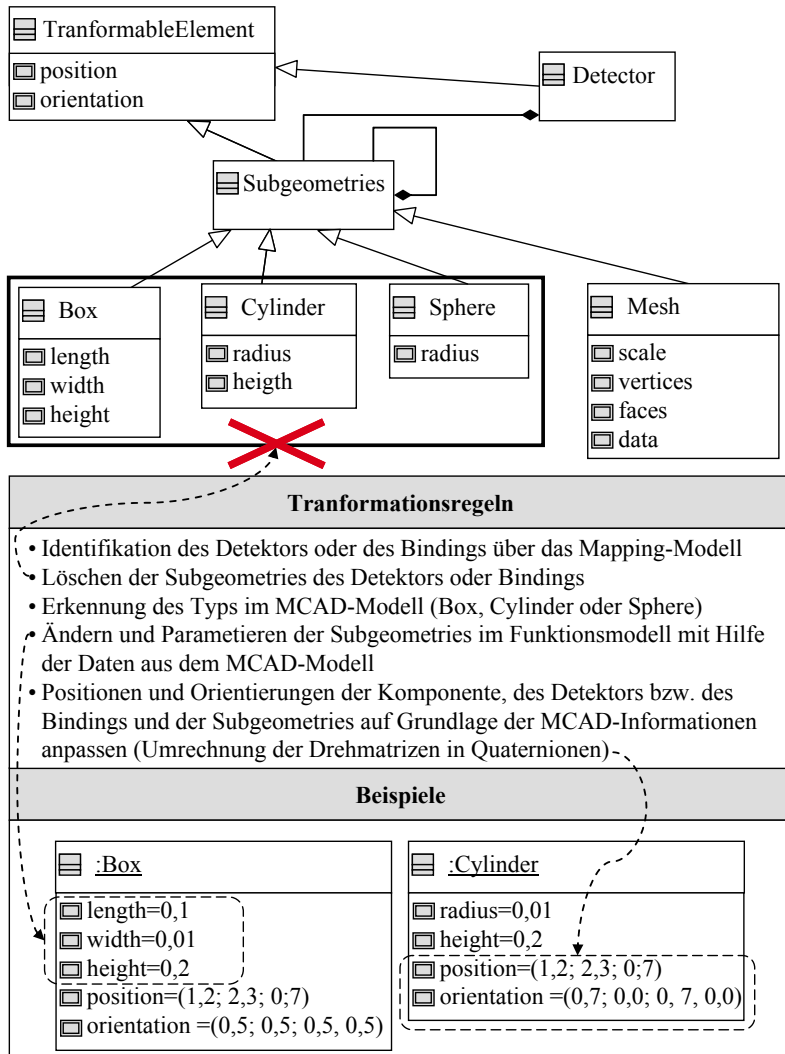


Abbildung 6.20: Transformationsregeln zum Anpassen der Detektoren und der Bindings mit Daten aus dem MCAD-Modell – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010)

## 6.4.6 Visualisierung und Geometriemodell

### 6.4.6.1 Bedeutung und Ansatz

Die Geometrie der Maschine spielt im vorgestellten Ansatz eine wesentliche Rolle, die über eine reine Visualisierungsfunktion hinausgeht. Die räumlichen Ausprägungen sind hier Teil der Verhaltenssimulation. Bereits in der Grob- und der Feinplanungsphase wird daher die äußere Form der Maschine berücksichtigt. Bis dato ist in der Praxis die Erstellung von Geometriemodellen ein zeitaufwendiger Prozess. Aus diesem Grund werden im industriellen Einsatz mitunter zweidimensionale Visualisierungsmodelle verwendet, die durch eine starke Abstraktion den Erstellungsaufwand verringern sollen. Dennoch muss zusätzlich ein logisches Ersatzmodell zur Abfrage von Kollisionen vom Entwickler händisch erstellt werden, das allerdings meist nur Teilaspekte beinhaltet.

Im Gegensatz dazu können dreidimensionale Modelle in Form von VRML-Dateien meist direkt aus MCAD-Systemen exportiert werden. Es handelt sich dabei um polygonale Modelle, auf welche die Grafikhardware von modernen Rechnern ausgelegt ist (CAMPAGNA 1999). Die aus dem CAD exportierten VRML-Modelle erfordern jedoch in der Regel einen erheblichen Vor- und Nachbereitungsaufwand. So müssen z. B. Details wie Bohrungen entfernt und das ganze Modell vereinfacht werden, da ansonsten die Daten einen zu großen Umfang für die Simulation aufweisen. Zudem gehen beim Export häufig die Hierarchie sowie die kinematischen Beziehungen verloren, die nachträglich mit erheblichem Zeitaufwand wiederhergestellt werden müssen. Dies wird zusätzlich dadurch erschwert, dass die Benennungen der Bauteile in vielen Fällen beim Ausleiten in der VRML-Datei geändert werden und somit für den Entwickler nur bedingt oder gar nicht nachvollziehbar sind.

Im Folgenden wird daher ein Ansatz vorgestellt, der die Geometriedaten für die Parts aufwandsarm und teilautomatisiert aus dem MCAD-Modell in das Funktionsmodell transferiert, siehe Abbildung 6.21. Prinzipiell wurde hier ebenso der Nachfolgestandard des VRML-Formates, X3D (Extensible 3D), berücksichtigt. Allerdings wird dieser zurzeit nur von wenigen CAD-Systemen unterstützt, weshalb sich die anschließenden Ausführungen auf die VRML beschränken. Zunächst werden die notwendigen Geometrieinformationen ausgeleitet, wobei das nicht zwingend alle Bauteile im CAD umfasst. Beispielsweise können Gehäuseelemente, welche die Sicht auf die essenziellen Maschinenkomponenten behin-

dem, vernachlässigt werden. Neben dem XML-Austauschformat, das die Positionen und die Orientierungen enthält, wird zur Beschreibung der äußeren Form der VRML-Standard verwendet. Hierfür ist es erforderlich, den Datenumfang der VRML-Dateien zu reduzieren, da ansonsten die Rechenzeit auf Grund der großen Anzahl an Rechenoperationen (abhängig von der Polygonanzahl) beeinträchtigt wird. Abschließend müssen die Geometriemodelle in das Funktionsmodell integriert werden. Aus der XML-Datei werden für jedes Part die Position und die Orientierung übernommen. Darüber hinaus setzt sich ein Part wiederum aus Unterelementen, den sogenannten *Subgeometries*, zusammen. Diese werden durch die Daten aus der VRML-Datei ersetzt. Die einzelnen Schritte des Datenimports werden in den sich anschließenden Abschnitten umfassend beschrieben.

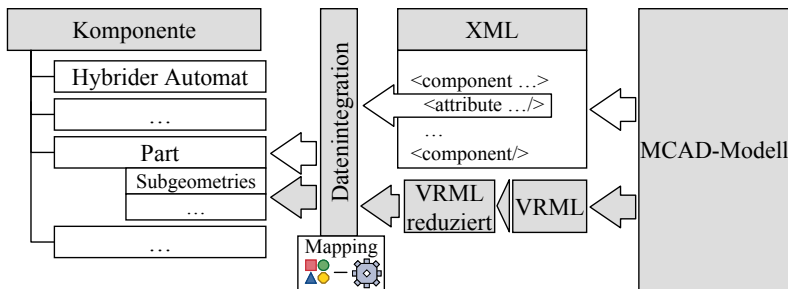


Abbildung 6.21: Integration der Geometrie aus dem MCAD-Modell in die Parts des Funktionsmodells

#### 6.4.6.2 Transformationsregeln für geometriebeschreibende Elemente

Die Geometriebeschreibung der Parts beruht wie bei den Detektoren und den Bindings auf den Subgeometries. Letztere werden zunächst entfernt. Im Gegensatz zu den Detektoren und den Bindings werden jedoch keine neuen Volumenprimitiven eingefügt, sondern detaillierte Netze erzeugt, welche die genauen geometrischen Abmessungen aus dem MCAD repräsentieren, siehe Abbildung 6.22. Neben einer realitätsgetreuen Darstellung der äußeren Form der Maschine können dadurch für bewegte Teile verbesserte Kollisionsbetrachtungen durchgeführt werden.

Das anzupassende Part wird direkt über das Mapping-Modell identifiziert. Nach dem Löschen der Subgeometries werden in einem ersten Schritt die Komponente und das Part positioniert und gedreht. Auch die Parts erben von Transformable-

Element, siehe Abbildung 6.22. Die zur räumlichen Anordnung erforderlichen Informationen werden dazu aus der XML-Datei mit den MCAD-Daten extrahiert und umgerechnet.

In einem zweiten Schritt werden dann die neuen Subgeometries erzeugt und die Geometrieinformationen aus dem MCAD als Netze (Mesh) importiert, siehe Abbildung 6.22. Die Grundlage hierfür sind VRML-Daten, die aus gängigen MCAD-Systemen exportiert werden können.

Das beschriebene Vorgehen auf der Basis von XML- und VRML-Datenquellen wurde deshalb gewählt, da hierdurch auf eine einfache Art und Weise Geometriemodelle (durch VRML-Daten) extrahiert und im Funktionsmodell (durch die XML-Daten) positioniert werden können. Prinzipiell sind auch im VRML-Gesamtmodell der Maschine die Positionen und die Ausrichtungen enthalten. Allerdings ist es nicht in allen MCAD-Systemen möglich, VRML-Teilmodelle (die nur einen Ausschnitt aus dem Gesamtgeometriemodell enthalten, beispielsweise ein einzelnes Förderband eines komplexen Förderbandsystems) zu generieren, welche die Koordinatenverschiebungen und -drehungen in Relation zum Ursprung des Gesamtsystems enthalten. Daher müssten für diese Zwecke aus dem VRML-Gesamtmodell die nicht erforderlichen Teile gelöscht werden, um die gewünschten Positionsangaben zu erhalten, was wiederum deutlich aufwendiger wäre, als der hier vorgeschlagene Weg auf der Grundlage der XML-Daten. Somit ist ein einheitliches Verfahren für die Positionierung und Rotation von Komponenten, Detektoren, Bindings und Parts mit Hilfe des XML-Austauschformates möglich, wobei lediglich für die Parts die Geometriedaten aus dem VRML-Format ergänzt werden.

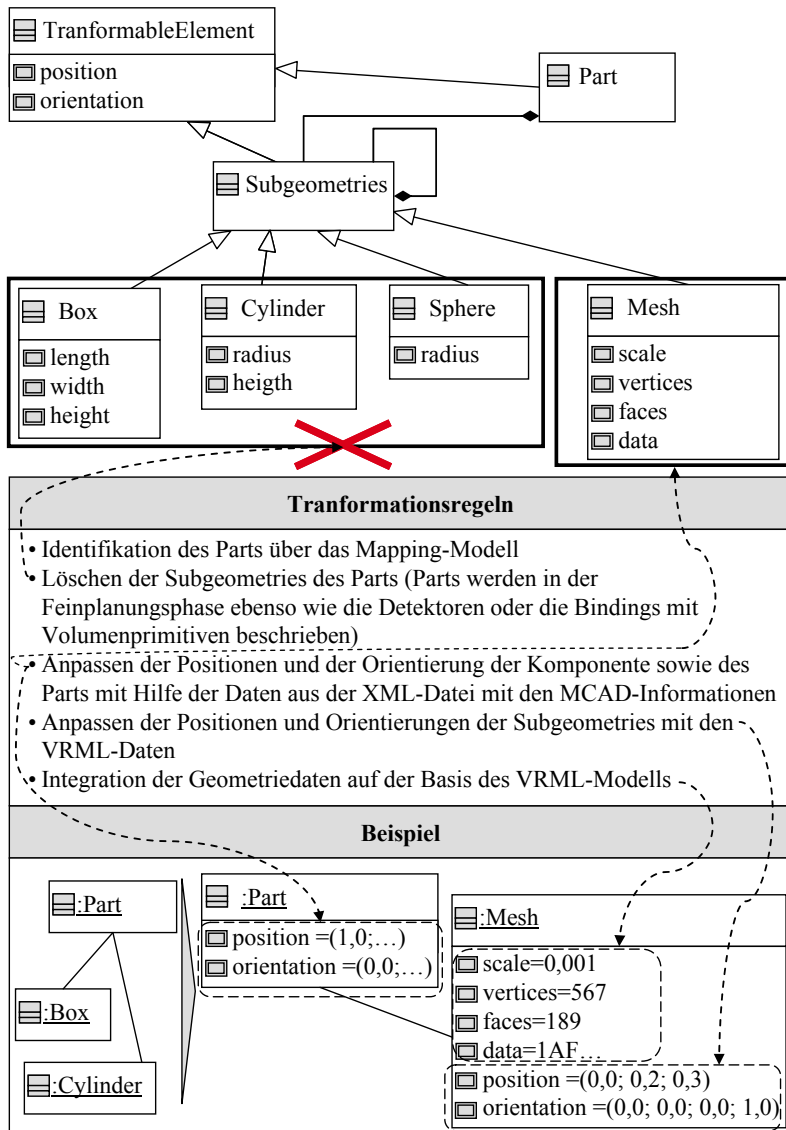


Abbildung 6.22: Transformationsregeln der geometriebeschreibenden Bauteil-daten aus dem MCAD-Modell – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010)



### 6.4.6.3 Ausleitung des 3D-Modells

Eine Grundvoraussetzung für die teilautomatisierte Ausleitung von 3D-Modellen aus dem MCAD-Werkzeug ist eine kinematisch korrekte Konstruktion. Dies bedeutet, dass Bauteile, die sich auf Grund von mechanischen Bindungen gemeinsam im Raum bewegen, zu einer Einheit zusammengefasst werden müssen (siehe Konstruktionsrichtlinien in Abschnitt 6.3.5). Am Beispiel einer Schutztür, die aus mehreren Elementen besteht, sei dies verdeutlicht. Bei Einhaltung der hier genannten Vorgaben kann die Tür als Ganzes in die Simulation eingebunden werden. Wäre dies nicht der Fall, so müsste der Entwickler nachträglich die Einzelteile der Tür verbinden. Sollte dabei auch nur ein Teil unberücksichtigt bleiben, so würden Fehler während der Simulation auftreten. Das nicht integrierte Bauteil verharrt dann im Raum, wohingegen sich die Tür translatorisch fortbewegt. Dies hat nicht nur Einfluss auf die Visualisierung, sondern gleichfalls auf die Kollisionsrechnung. Generell wurde in den Konstruktionsrichtlinien eine Orientierung der mechanischen Konstruktion an der Struktur des Funktionsmodells gefordert. Dadurch lassen sich statische (stationäre) Anteile, wie z. B. Gestellstrukturen, einfacher übernehmen.

Das MCAD-Modell verfügt über einen sehr hohen Detaillierungsgrad und in der Folge über ein entsprechend großes Datenvolumen. Eine direkte Überführung in VRML-Modelle mit anschließender Integration in die Simulation ist meist nicht oder zumindest nur für sehr kleine CAD-Modelle möglich, ohne dass dies eine Beeinträchtigung der Simulationsrechnung nach sich zieht. Aus diesem Grund muss im Allgemeinen die Datenmenge reduziert werden. Im Rahmen der vorliegenden Arbeit werden hierzu zwei Maßnahmen automatisiert vorgenommen bzw. unterstützt. Da nicht alle Elemente im MCAD-Modell simulationsrelevant sind, wird auf deren Ausleitung teilweise verzichtet. Dies umfasst Norm- und Standardbauteile, wie z. B. Schrauben, die automatisiert vor dem Export gelöscht werden. Zusätzlich ist es möglich, sogenannte *Features*, wie beispielsweise Bohrungen oder Fasen, automatisch zu entfernen. Neben der Vernachlässigung von nicht notwendigen Teilen lässt sich die Datengröße durch eine Polygonreduktion der VRML-Dateien verringern. Diese findet nach dem Ausleiten der VRML-Modelle statt und wird deshalb in den sich anschließenden Abschnitten detailliert beschrieben.

Zunächst sind die VRML-Dateien aus dem MCAD-Werkzeug zu exportieren. Hierzu werden die Beziehungen zwischen den Parts und den CAD-Komponenten über das Mapping-Modell ausgewertet. Somit hat der Entwickler die Möglich-

keit, nur den Anteil an Geometriedaten auszuwählen, der für die Simulation erforderlich ist. Die nicht mit Parts verbundenen Bauteile des MCAD-Modells bleiben bei der Ausleitung unberücksichtigt. Jeder VRML-Datei wird zudem die ID des zugehörigen Parts im Funktionsmodell hinterlegt. Dies ist derart umgesetzt, dass die VRML-Datei über die Namenskonvention *<Part-ID>.wrl* kenntlich gemacht ist.

### 6.4.6.4 Polygonreduktion

Das Grundprinzip der Polygonreduktion basiert auf einer Verkleinerung der Anzahl an Polygonen in einem Objekt. Hierfür existiert eine Fülle unterschiedlicher Algorithmen, auf die an dieser Stelle nicht näher eingegangen werden soll. Für weiterführende Informationen sei auf entsprechende Fachliteratur verwiesen, wobei CAMPAGNA (1999) und GIROD ET AL. (2000) einen guten Überblick geben.

Im Rahmen dieser Arbeit kommt die Open-Source-Software *MeshLab* zum Einsatz, die einen externen Zugriff auf die Programmfunktionalität mit Hilfe des *MeshLabserver*s ermöglicht (MESHLAB 2010). Für die Polygonreduktion wurden zwei Algorithmen verwendet: Das *Vertex Clustering*, das auf ROSSIGNAC & BORELL (1993) zurückgeht, und die *Quadratic Error Metrics*, die auf der Theorie von GARLAND & HECKBERT (1997) basiert. Beim Vertex Clustering wird eine sogenannte *Bounding Box* über das Ursprungsnetz gelegt, die dieses in einzelne Zellen unterteilt. Alle Eckpunkte in einer Zelle werden zu einem neuen, gemeinsamen Punkt zusammengefasst, siehe Abbildung 6.23 links. Über die Größe der *Bounding Box* wird der Reduktionsgrad festgelegt. Die Quadratic Error Metrics von GARLAND & HECKBERT (1997) beruhen auf einer iterativen Zusammenfassung von Knotenpaaren, die als *Pair Contraction* bezeichnet wird. Gültige Knotenpaare  $(v_1, v_2)$  sind entweder durch Kanten verbunden oder liegen so nahe beieinander, dass sie die Abstandsbedingung  $\|v_1 - v_2\| < t$  erfüllen, wobei  $t$  ein definierbarer Grenzwert ist, siehe Abbildung 6.23 rechts. Die Auswahl des nächsten Schrittes wird von einer Metrik bestimmt, die auf einer quadratischen  $4 \times 4$ -Matrix aufbaut und zur Fehlerabschätzung eines jeden Knotens im Netz dient.

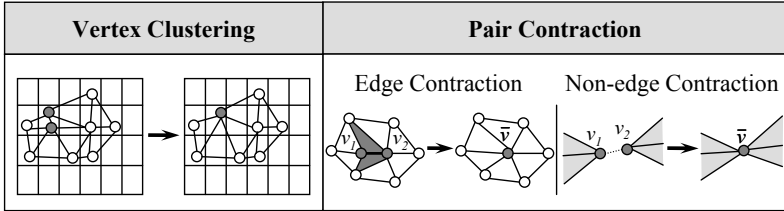


Abbildung 6.23: *Eingesetzte Algorithmen zur Polygonreduktion – Vertex Clustering und Pair Contraction*

Die aus dem MCAD abgeleiteten VRML-Daten werden dem Softwarewerkzeug MeshLab automatisiert zugeführt und darin bearbeitet. Hierzu wird eine Programmierschnittstelle von MeshLab genutzt, mit welcher auf die notwendigen Funktionen zum Verkleinern der VRML-Modelle zugegriffen werden kann. Vorab sind das Reduktionsverfahren sowie die notwendigen Parameter einzustellen. Der Reduzierungsprozess kann bei Bedarf mehrfach vorgenommen werden, bis das gewünschte Ergebnis erreicht ist. Die Modelle sollten nur bis zu dem Grad verkleinert werden, bei welchem noch keine augenscheinlichen Fehler auftreten, wie z. B. offene Körper. Abbildung 6.24 zeigt exemplarisch die Polygonreduktion für das Gehäuse eines Pneumatikzylinders.

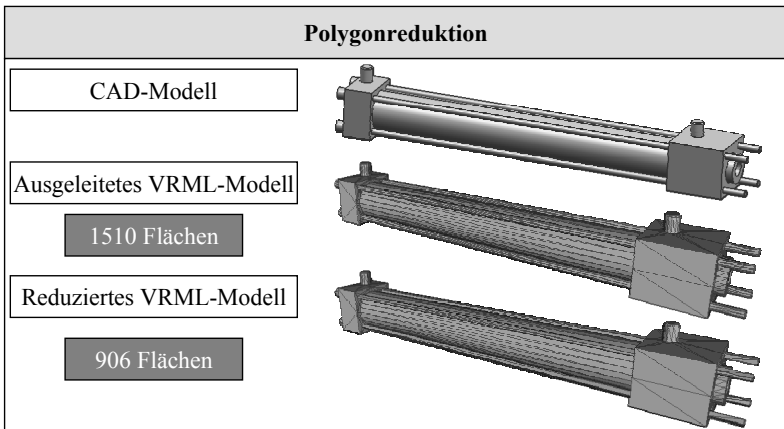


Abbildung 6.24: *Polygonreduktion mit Hilfe des Vertex Clustering am Beispiel des Gehäuses eines Pneumatikzylinders*

### 6.4.6.5 Import der Geometriemodelle

Die reduzierten VRML-Daten sind abschließend in das Funktionsmodell zu importieren, wobei dieser Schritt automatisiert ist. Nachdem ein zufriedenstellender Reduktionsgrad erreicht ist, werden die VRML-Modelle komplett eingelesen und über die ID den entsprechenden Parts zugewiesen. Wie bereits erläutert, verfügt jedes Part über *Subgeometries*, welche die eigentliche Geometriebeschreibung enthalten. Das jeweilige VRML-Modell wird den zugehörigen Parts als Subgeometry beigelegt. Die Positionierung und die Orientierung erfolgen automatisch mit Hilfe der Daten in der XML-Datei, wobei sich dies auf das Part als übergeordnete Einheit der Subgeometries bezieht.

## 6.5 Automatisierte Erweiterung des Funktionsmodells mit E/FCAD-Daten

### 6.5.1 Konzept für die Integration der E/FCAD-Daten

Während sich die Daten aus dem MCAD-Modell hauptsächlich für strukturelle Aspekte verwenden lassen, geben die Informationen aus dem E/FCAD-Modell Aufschluss über das genaue Verhalten der Aktoren und der Sensoren. Die projektierten Betriebsmittel und Bauteile sowie deren Verschaltungen zu Steuerketten können für die Erweiterung des Funktionsmodells herangezogen werden. Zudem müssen die Aktoren und die Sensoren hinsichtlich ihres Kommunikationsverhaltens für die VIBN modifiziert werden. Dies umfasst z. B. Zustandsmeldungen, die von der SPS abgefragt werden und die in dieser Detaillierung nicht im Modell der Feinplanungsphase enthalten sind. Das zugrunde liegende Konzept für die Einbindung der E/FCAD-Daten wird nachfolgend erläutert.

Wie bereits in Abschnitt 6.4.1 vorgestellt, wirken sich die zu integrierenden Daten aus der Mechanik und der Elektrotechnik bzw. der Fluidik auf unterschiedliche Bereiche der Modellelemente einer Komponente aus. In Abbildung 6.25 ist verdeutlicht, auf welche Teile sich die Details aus dem E/FCAD-Modell beziehen. Der Datentransfer ist ebenfalls über das XML-Austauschformat und das Mapping-Modell geregelt. Es werden die Automaten für den Gutablauf und den Fehlerfall durch umfassendere Modelle ersetzt. Ferner werden die Ports samt deren Verbindungen über die Kanäle neu erstellt sowie das TRM an die geänderten Gegebenheiten angepasst.

Mit Hilfe des beschriebenen Ansatzes werden Überschneidungen durch die automatisierte Verfeinerung mit MCAD- und E/FCAD-Daten vermieden. Lediglich bei den hybriden Automaten würden sich Berührungspunkte ergeben, da aus dem MCAD Parameter für selbige importiert werden. Daher werden nur für die Komponenten der Wirkketten, die exklusiv in der Mechanik vorkommen, wie z. B. Getriebestufen, die Parameter für die Zustandsmaschinen aus dem MCAD-Modell entnommen.

Vor der Datenübernahme aus dem CAD ist zunächst die Konsistenz der unterschiedlichen Modelle sicherzustellen. Dabei wird geprüft, ob alle Aktoren und Sensoren vorhanden sind und deren Typen übereinstimmen, siehe auch Abschnitt 6.3.7. Ist dies der Fall, so können die Informationen aus dem E/FCAD für die Detaillierung automatisiert eingebunden werden.

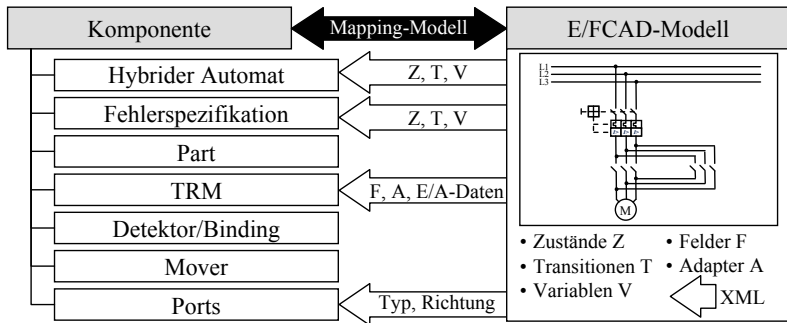


Abbildung 6.25: Automatisierte Integration von E/FCAD-Daten in das Funktionsmodell

Die Grundlage für die Erweiterung der Aktoren bildet das Modell der Feinplanungsphase. In diesem werden die Aktoren in eine Steuer- und Stellkomponente sowie in ein Arbeitsglied aufgeteilt, siehe Abbildung 6.26. Die Komponente des Arbeitsgliedes wird lediglich an den Stellen bzw. an den Bestandteilen modifiziert, die in Abbildung 6.25 dargestellt sind. Des Weiteren ist, anders als bei der Detaillierung mit den Mechanikdaten, ebenso die Steuer- und Stellkomponente in den Prozess mit einbezogen. Darin wird nunmehr die reale Betriebsmittel- bzw. Bauteilstruktur abgebildet. Im Gegensatz zur Modifikation des Arbeitsgliedes müssen für diese Zwecke neue Komponenten hinzugefügt und über Ports und Kanäle verbunden werden. Jede Unterkomponente enthält eine Zustandsmaschine für den Gutablauf, eine Fehlerspezifikation, Ports und ein TRM.

Zur Umsetzung der beschriebenen Methode ist eine Bibliothek mit entsprechenden Modellen erforderlich. Auf diese kann bei der automatisierten Erweiterung zurückgegriffen werden, da sich das Verhalten der Bauteile nur implizit aus dem verwendeten Typ ableiten lässt. Für die Arbeitsglieder sind dazu einzelne Modellelemente, wie z. B. hybride Automaten, zu definieren. Hinsichtlich der Steuer- und Stelleinheiten müssen vollständige Komponenten hinterlegt werden, die ganze Bauteile, wie beispielsweise Wegeventile oder Schütze, beschreiben. Die automatische Integration der Bibliothekselemente in das Funktionsmodell verlangt nach einer Konsistenzprüfung der Schnittstellen zu anderen Modellbestandteilen. Dies umfasst z. B. die Typen der zu verbindenden Ports oder die Mover und die Detektoren, auf welche die hybriden Automaten zugreifen. Hierbei wird sichergestellt, dass die Zustandsmaschine die richtigen Elemente adressiert.

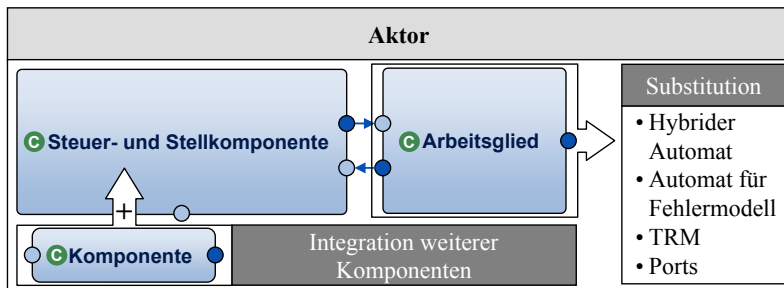


Abbildung 6.26: Erweiterung von Aktoren im Funktionsmodell mit Informationen aus dem E/FCAD-System

Sensoren sind gemäß Modellierungsrichtlinie als eigenständige Komponenten abgebildet. Sie werden auf die gleiche Art und Weise wie die Arbeitsglieder mit den E/FCAD-Daten erweitert. Auch hierfür sind eine Bibliothek sowie eine Konsistenzprüfung vor dem Einfügen der Modellelemente erforderlich. In den folgenden Abschnitten werden die oben erläuterten Mechanismen vertieft betrachtet.

### 6.5.2 Modelltransformation mit Hilfe der E/FCAD-Daten

Zur Integration der E/FCAD-Daten sind wie für die Einbindung MCAD-Informationen Transformationsregeln erforderlich. Das Ausgangsmodell und das generierte Funktionsmodell sind beide konform zum gleichen Metamodell, siehe Abbildung 6.27. Für die Erweiterung mit den E/FCAD-Daten ist zusätzlich eine

Modellbibliothek nötig, die in den Generierungsprozess mit einbezogen wird. Darin enthalten sind neben kompletten Komponenten auch einzelne Modellierungselemente, wie z. B. die hybriden Automaten oder die Ports, siehe Abbildung 6.27 unten rechts. Das ursprüngliche Funktionsmodell ist wie im Falle der MCAD-Daten mit dem E/FCAD-Modell durch das Mapping-Modell verbunden.

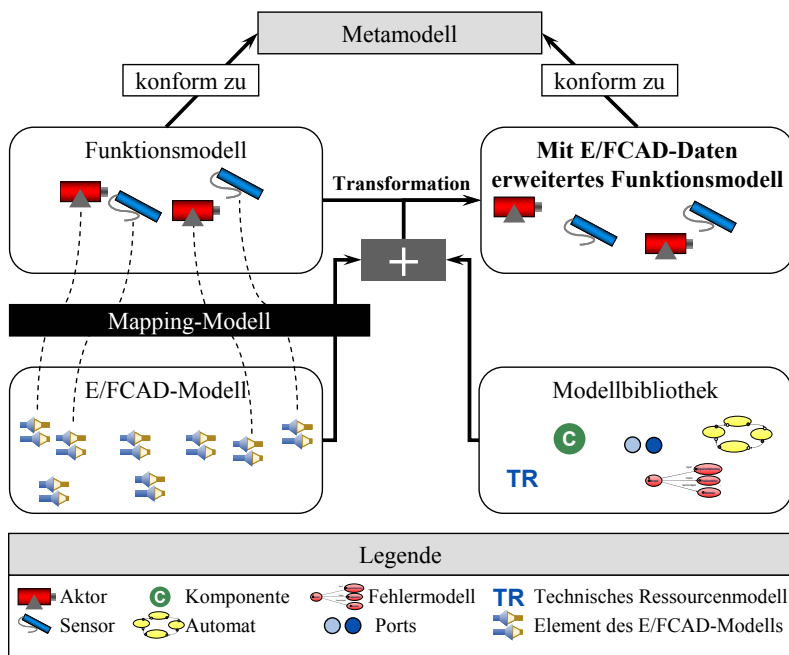


Abbildung 6.27: Transformation des Funktionsmodells durch Integration der E/FCAD-Daten

### 6.5.3 Integration der Aktordaten

#### 6.5.3.1 Erweiterung des Arbeitsglieds

Das in der Feinplanungsphase modellierte Arbeitsglied beschreibt das Verhalten trotz Berücksichtigung erster Wirkstrukturen auf einer abstrakten Ebene. Zwar wird die prinzipielle technische Ausprägung bereits bedacht, jedoch lässt sich mit

Hilfe der Daten aus dem E/FCAD-Modell noch eine erhebliche Detaillierung und höhere Modellgüte erzielen. Am Beispiel eines doppeltwirkenden Pneumatikzylinders soll dies exemplarisch verdeutlicht werden. Ebenso könnte an dieser Stelle ein Exempel aus der Elektrotechnik herangezogen werden. Die Mechanismen für die Datenintegration sind identisch, zumal sich diese in derselben Entwicklungsumgebung und einem gemeinsamen Modell projektieren lassen.

Das Arbeitsglied wurde in der Feinplanungsphase als abstrakter Pneumatikzylinder modelliert, siehe Abbildung 6.28 oben. Die Ansteuerung erfolgt über die zwei booleschen Eingangsports *Aus* und *Ein*, die den Zylinder ein- bzw. ausfahren lassen. Wie der hybride Automat in Abbildung 6.28 oben illustriert, können keine Zwischenpositionen eingenommen werden. Diese Charakteristik spiegelt quasi das Sollverhalten wider, welches als Grundlage für die Ausarbeitung in den Einzeldisziplinen dient.

Der aus den E/FCAD-Daten extrahierte und in das erweiterte Funktionsmodell integrierte Pneumatikzylinder verfügt dagegen über die zwei Eingangsports *P1* und *P2* des Typs *Double*, welche die Druckanschlüsse abbilden (Abbildung 6.28 unten). Im Gegensatz zum Modell der Feinplanungsphase kann der Zylinder Zwischenpositionen einnehmen, was sich aus dem Zusammenwirken mit den Komponenten der Steuerkette ergibt. Das Verhalten des Zylinders ist von der Schaltstellung des angeschlossenen Stellventils abhängig. Da eine Sperrstellung im Feinplanungsmodell nicht vorgesehen ist, muss diese durch die Auswahl entsprechender Wegeventile bei der Konstruktion berücksichtigt werden. Das mit Hilfe von CAD-Daten erweiterte Arbeitsglied ist wesentlich detaillierter und aussagekräftiger als das Modell der Feinplanung. Dies verdeutlicht sich beispielsweise daran, dass eine Druckminderung an den Eingangsports Einfluss auf die Ausfahrgeschwindigkeit des Zylinders hat. Zusätzlich werden weitere Ports angelegt, die für die Kommunikation mit der Steuerung benötigt werden. Im vorliegenden Fall des Zylinders handelt es sich z. B. um eine Diagnosemeldung. Zudem wird das Fehlermodell erzeugt, welches ein Klemmen oder eine Schwergängigkeit des Zylinders simuliert sowie die Diagnosemeldung verfälscht.



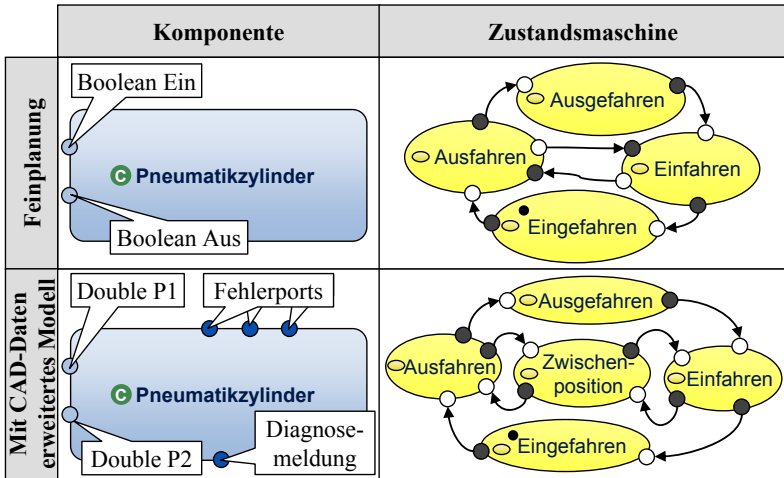


Abbildung 6.28: Automatisierte Erweiterung des Arbeitsgliedes am Beispiel eines doppeltwirkenden Pneumatikzylinders

Nachdem die Unterschiede zwischen dem Arbeitsglied in der Feinplanungsphase und dem mit CAD-Daten erweiterten Modell erläutert wurden, soll nun die automatisierte Datenintegration näher betrachtet werden, siehe Abbildung 6.29. Als Grundlage dienen die Verbindungen zwischen Funktions- und CAD-Modell, wofür wiederum das Mapping-Modell herangezogen wird. Der Actor im Feinplanungsmodell ist mit dem entsprechenden Arbeitsglied im E/FCAD verbunden. Anhand der Typnummer lässt sich die genaue technische Ausprägung ermitteln. Auf Grundlage dessen werden der hybride Automat für den Gutablauf und die Fehlerspezifikation eingefügt bzw. ersetzt. Die zugehörigen Modellelemente werden hierzu aus der Bibliothek entnommen. Ebenso wird mit dem TRM verfahren, welches durch ein Bibliothekselement passend zur Typnummer in das Modell eingefügt wird. Mit Hilfe der im XML-Austauschformat hinterlegten E/A-Daten lässt sich das TRM parametrieren, womit eine Verbindung zwischen Modellports und den Steuerungsein- und -ausgängen geschaffen wird. Die Ports werden gleichfalls ersetzt und angepasst. Abschließend werden die gelöschten Kanäle durch neue Verbindungen mit der Steuer- und Stellkomponente wieder hergestellt. Die Informationen dafür sind in der XML-Struktur abgelegt. Auf die Verbindungserstellung wird im nächsten Abschnitt genauer eingegangen.

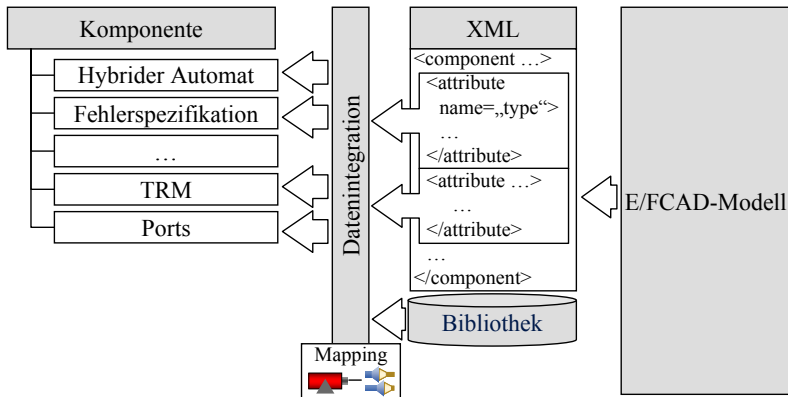


Abbildung 6.29: Automatisierte Erweiterung des Arbeitsgliedes im Funktionsmodell mit Hilfe der E/FCAD-Daten

### 6.5.3.2 Transformationsregeln für die Erweiterung des Arbeitsgliedes

Ein Aktor besteht gemäß der Modellierungsrichtlinien der Feinplanungsphase (siehe Abschnitt 5.5.3) aus einem Arbeitsglied und einer Steuer- und Stellkomponente. Auf die Detaillierung des Arbeitsgliedes mit Hilfe der E/FCAD-Daten soll nachfolgend ausführlicher eingegangen werden. Die Identifikation des anzupassenden Arbeitsgliedes erfolgt über die hinterlegte Verbindung zum E/FCAD im Mapping-Modell.

Das Grundprinzip der Modellverfeinerung besteht darin, dass spezielle Bestandteile einer Komponente durch neue Elemente mit höherem Informationsgehalt ersetzt werden. Vom Austausch betroffen sind die Ports, die hybriden Automaten, die Fehlerautomaten sowie das TRM. Diese Elemente sind in einer Modellbibliothek in Musterkomponenten enthalten, siehe Abbildung 6.30. Zur Detaillierung wird hierauf automatisiert zurückgegriffen. Die Transformationsregeln für die Anpassung der Arbeitsglieder sind in Tabelle 6.2 übergreifend zusammengefasst und werden im Folgenden beschrieben.

Zunächst werden sämtliche Ports des Arbeitsgliedes sowie die verbindenden Kanäle (Channels) gelöscht. Ausgenommen hiervon sind die Ports zur Übertragung von Positionen bzw. zurückgelegten Strecken, von Geschwindigkeiten und von Beschleunigungen, siehe auch Tabelle 6.2. Dies begründet sich darin, dass diese Art von Ports für den Ablauf des Funktionsmodells von großer Relevanz sind.

Beispielsweise sind Winkel oder Winkelgeschwindigkeiten als Eingangsinformationen weiterer Komponenten, die etwa die Kinematik abbilden, für deren interne Berechnungsvorgänge von Bedeutung. Somit bleiben auch die Verbindungen über die Kanäle erhalten. Diese Informationen würden ansonsten beim Löschen verloren gehen, da sie nicht im E/FCAD-Modell enthalten sind und müssten deshalb manuell wieder erstellt werden.

Im Anschluss werden die neuen Ports dem Arbeitsglied zugefügt, dies geschieht entsprechend den Vorgaben der Musterkomponente in der Bibliothek. Ferner müssen der Typ (z. B. Boolean), der Initialisierungswert und das Übertragungsverhalten (ereignisdiskret oder zustandsbasiert) definiert werden. Abschließend werden die Verbindungen der Ports durch die Kanäle wiederhergestellt. Hierzu dient das TRM im Ursprungsmodell, in welchem über die Adapter die Beziehungen von den realen Steuerungsadressen zu den Ports des Funktionsmodells abgebildet sind. Diese Informationen werden genutzt und mit den Steuerungsadressen der zu integrierenden E/FCAD-Daten verglichen. Der Kanal wird folglich nur erstellt, wenn die realen Steuerungsadressen übereinstimmen und der Typ des zu verbindenden Ports zum integrierten Port korrespondiert. Für die VIBN ist die Signalversorgung über die Kanäle außerhalb der Aktorkomponente nicht erforderlich, da in diesem Falle die reale oder die virtuelle Steuerung auf Teile der Portmenge direkt zugreift (siehe detaillierte Erklärung der VIBN-Modellgenerierung in Abschnitt 6.6). Das Funktionsmodell wäre jedoch für einen internen Ablauf nicht mehr simulationsfähig, da die Verbindungen zu den modellierten Komponenten mit der Steuerungslogik fehlen. Auf die Kanäle bzw. deren Erstellung wird speziell noch in Abschnitt 6.5.3.4 eingegangen.

Der hybride Automat wird durch die Variante in der Modellbibliothek ersetzt. Alle Zustände, Transitionen und Variablen werden übernommen. Mit Hilfe der E/CAD-Daten erfolgt zudem die Anpassung der Variablenwerte. Bei der Einbindung des Automaten wird geprüft, ob dieser die Elemente aus dem MCAD, wie die Mover, die Detektoren und die Bindings, richtig referenziert. Gleiches gilt für die Ports, da diese im Automaten abgefragt bzw. beschrieben werden. Des Weiteren wird der Fehlerautomat durch das Bibliothekselement substituiert.

Das TRM wird ebenfalls durch die Bibliotheksvariante ersetzt. Die E/A-Daten werden vom E/FCAD-Modell übernommen. Auf das TRM wird im Detail noch in Abschnitt 6.5.5 eingegangen.

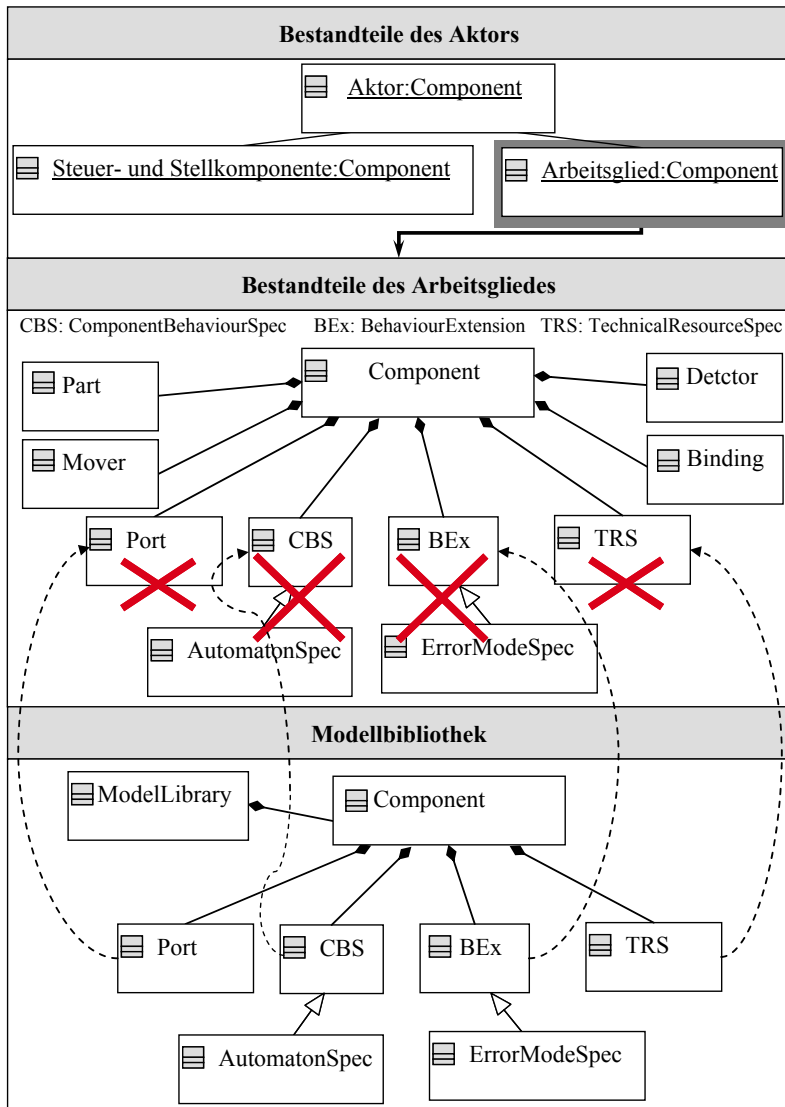


Abbildung 6.30: Transformationsregeln zur Detaillierung des Arbeitsgliedes eines Aktors mit Hilfe der E/FCAD-Daten – Vereinfachte Darstellung des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011)

<b>Identifikation der anzupassenden Komponente</b>
<ul style="list-style-type: none"> <li>Finden der Komponente mit Hilfe des Mappings-Modells.</li> <li>Kennzeichnung als Aktor, Identifikation über ID.</li> </ul>
<b>Ports</b>
<ul style="list-style-type: none"> <li>Löschen aller Ports samt zugehöriger Kanäle (Channels) in der Arbeitskomponente des Funktionsmodells in der Feinplanungsphase.</li> <li>Vom Löschen ausgenommen sind Ports, die Positions-, Geschwindigkeits- und Beschleunigungswerte übertragen (<math>x, \dot{x}, \ddot{x}, \varphi, \dot{\varphi}, \ddot{\varphi}</math>), da diese Werte für die Berechnung von Bewegungsvorgängen in anderen Komponenten liefern. Dadurch bleiben auch die Verbindungen über die Kanäle (Channels) zu diesen Komponenten bestehen.</li> <li>Einfügen der neuen Ports (Eingänge und Ausgänge), die im Bibliothekselement hinterlegt sind. Setzen des Typs (Boolean, Double etc), des Initialisierungswertes (initialValue) und des Übertragungsverhaltens (ereignisdiskret oder zustandsbasiert). Erzeugen neuer Kanäle (Channels).</li> </ul>
<b>Hybride Automaten</b>
<ul style="list-style-type: none"> <li>Löschen des ursprünglichen Automaten mit allen Zuständen und Variablen.</li> <li>Auswahl und Einfügen des passenden Automaten aus der Modellbibliothek mit Hilfe der eindeutigen Typnummer.</li> <li>Parametrieren der Variablenliste mit Hilfe der E/FCAD-Daten.</li> <li>Prüfung, ob Ports, Mover, Detektoren und Bindings richtig referenziert werden.</li> </ul>
<b>Fehlerspezifikation</b>
<ul style="list-style-type: none"> <li>Löschen des ursprünglichen Automaten.</li> <li>Auswahl und Einfügen des passenden Fehlerautomaten aus der Modellbibliothek mit Hilfe der eindeutigen Typnummer.</li> </ul>
<b>Technisches Ressourcenmodell</b>
<ul style="list-style-type: none"> <li>Löschen des ursprünglichen TRM.</li> <li>Auswahl und Einfügen des passenden TRM aus der Modellbibliothek mit Hilfe der eindeutigen Typnummer.</li> <li>Parametrieren der E/A-Werte mit Hilfe der CAD-Daten.</li> </ul>

Tabelle 6.2: *Übersicht der Transformationsregeln für die einzelnen Elemente eines Arbeitsgliedes*

### 6.5.3.3 Erweiterung der Steuer- und Stellkomponente

Die Steuer- und Stellkomponente fasst gemäß Modellierungsrichtlinie der Feinplanungsphase die Steuerkette eines Aktors vom Signalglied bis hin zum Stellglied zusammen. Es wird, wie im Falle des Arbeitsgliedes, das Sollverhalten beschrieben, welches durch geeignete Bauteile während der Konstruktionsphase abzubilden ist. Das im vorangegangenen Abschnitt erläuterte Beispiel des Pneumatikzylinders soll abermals aufgegriffen und für die Steuer- und Stellkomponente ausgeführt werden. Über die booleschen Ausgangsports *Ein* und *Aus* ist das Modell der Feinplanungsphase mit dem Pneumatikzylinder verbunden. Als Modelleingänge der Steuer- und Stellkomponente sind die Eingangsports *Signal1* und *Signal2* angelegt. Mit Hilfe der beiden Eingangssignale wird in der Feinplanungsphase das gewünschte Verhalten auf die Ausgangssignale *Ein* und *Aus* abgebildet. Im vorliegenden Fall sind die Eingangs- mit den Ausgangssignalen über logische Operationen verknüpft, siehe Abbildung 6.31 unten rechts. Durch die entsprechende Schaltung selbiger fährt der Zylinder ein oder aus. In Abbildung 6.31 links ist das zugehörige CAD-Modell dargestellt, welches exakt das in der Feinplanungsphase definierte Sollverhalten widerspiegelt. Der Zylinder fährt nur dann aus, wenn die Eingangssignale *Signal1* und *Signal2* über unterschiedliche boolesche Werte verfügen. Die Steuer- und Stellkomponente im erweiterten Funktionsmodell ist aus den E/FCAD-Daten abgeleitet und wird dabei durch zusätzliche Komponenten, die wesentliche Bauteile repräsentieren, hierarchisch untergliedert, siehe Abbildung 6.31 rechts. So werden beispielsweise ein 5/2-Wegeventil, zwei 4/2-Wegeventile, zwei Zweidruckventile und ein Wechselventil unter der Steuer- und Stellkomponente angelegt und miteinander verschaltet. Wie bereits in Abschnitt 6.5.3.1 verdeutlicht, werden durch die detaillierteren Modelle die Wechselwirkungen zwischen Arbeits-, Steuer-, Stell- und Signalgliedern berücksichtigt, die in dieser Form nicht im Feinplanungsmodell enthalten sind. Eine statische Zwischenposition des Pneumatikzylinders auf Grund einer Sperrstellung des Stellgliedes ist durch das verwendete 5/2-Wegeventil nicht möglich. Es entspricht somit dem im Feinplanungsmodell definierten Sollverhalten.

Im Gegensatz zur automatisierten Erweiterung des Arbeitsgliedes werden im Hinblick auf die Steuer- und Stellkomponente nicht nur einzelne Elemente ausgetauscht, sondern diese strukturell weiter untergliedert. Da die Steuer- und Stelleinheit in der Feinplanungsphase nur stark abstrahiert und lediglich mit einer Komponente dargestellt wird, ist eine direkte Zuordnung zu Objekten im

E/FCAD-Modell nicht möglich. Bisher wurden hierfür im Mapping-Modell Paare mit Eins-zu-eins-Beziehungen gebildet (z. B. TRM und CAD-Element, Part und CAD-Element etc., siehe Abschnitt 6.3.6).

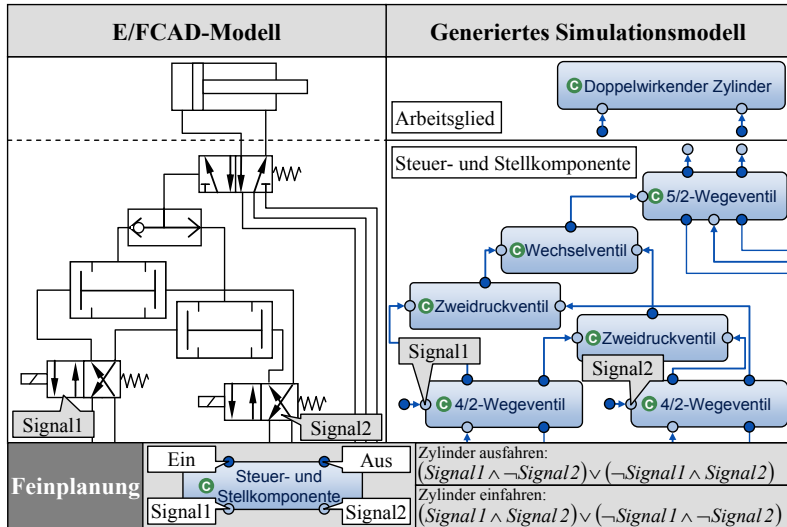


Abbildung 6.31: Automatisierte Erweiterung der Steuer- und Stellkomponente am Beispiel eines doppeltwirkenden Pneumatikzylinders

Daher bildet das Arbeitsglied die Basis zur Erweiterung der Steuer- und Stellkomponente, siehe Abbildung 6.32. Über das Mapping-Modell ist das Arbeitsglied mit dem E/FCAD verbunden. In Letzterem sind alle mit dem Arbeitsglied gekoppelten Baugruppen enthalten und werden somit in der exportierten XML-Datei beschrieben. Beginnend beim Arbeitsglied werden sämtliche Verbindungen zu selbigem ausgewertet sowie die zugehörigen Komponenten in der Steuer- und Stelleinheit angelegt und verschaltet. Die Steuerkette wird so lange durchsucht, bis alle angeschlossenen Bauteile integriert sind. Im oben erläuterten Beispiel des Pneumatikzylinders würde als Verbindungskomponente zunächst das 5/2-Wegeventil erkannt und in die Steuer- und Stellkomponente eingefügt werden. Dieses ist wiederum mit weiteren Elementen verbunden, die sich über den gleichen Ansatz inkludieren lassen.

Die automatische Erweiterung erfordert die Erstellung von Bibliothekselementen, wie z. B. Wegeventile oder Schütze. Im Unterschied zum Arbeitsglied wer-

den vollständige Komponenten eingefügt, die über einen hybriden Automaten für den Gutablauf, die Fehlerspezifikation, Ports und das TRM verfügen. Im gleichen Zug werden die E/A-Daten aus der XML-Datei in das TRM der generierten Komponenten geschrieben.

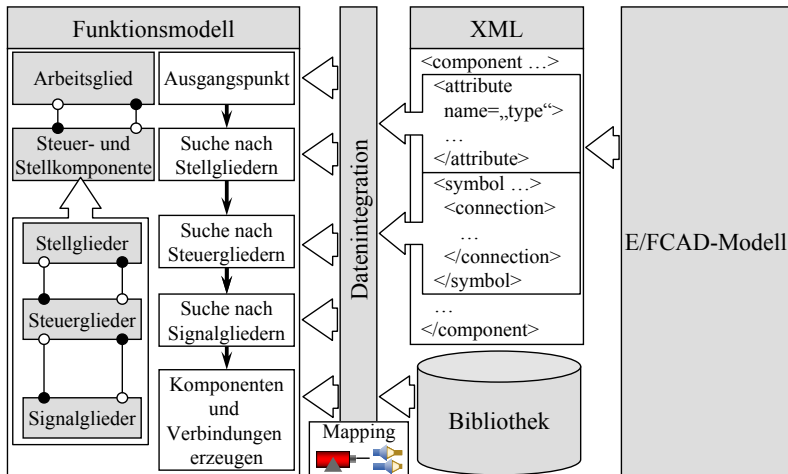


Abbildung 6.32: Automatisierte Erweiterung der Steuer- und Stellkomponente im Funktionsmodell mit Hilfe der E/FCAD-Daten

#### 6.5.3.4 Transformationsregeln für die Erweiterung der Steuer- und Stellkomponente

Im Gegensatz zum Arbeitsglied werden beim Steuer- und Stellglied nicht nur einzelne Elemente einer Komponente, wie z. B. die Automaten, ausgetauscht. Es wird vielmehr die Komponente des Steuer- und Stellgliedes „ausgehöhlt“, indem die verhaltensbeschreibenden Bestandteile, wie der hybride Automat, entfernt und durch weitere Komponenten, die über ein eigenes Verhalten verfügen, ersetzt werden, siehe Abbildung 6.33.

Da die Steuer- und Stellkomponente eine Abstraktion der gesamten Steuerkette des Aktors darstellt, ist eine direkte Zuordnung zu einem Element im E/FCAD über das Mapping-Modell nicht möglich. Die zu modifizierende Komponente wird im Funktionsmodell als Teil des Aktors und über die Verbindung mit dem Arbeitsglied identifiziert.



Im Anschluss werden die Automaten für den Gutablauf und für die Fehlermodellierung gelöscht. Dies umfasst alle Zustände, Transitionen und Variablen. Ebenso wird der Automat zur Fehlerbeschreibung entfernt. Des Weiteren werden die Ports und die Kanäle der Steuer- und Stellkomponente zum Arbeitsglied eliminiert. Die Ports und die Kanäle, die von der übergeordneten Aktorkomponente herführen oder zu dieser gehen, bleiben bestehen, da diese für die Signalversorgung mit der im Funktionsmodell hinterlegten Steuerungslogik benötigt werden. Wie bereits in Abschnitt 6.5.3.2 für die Arbeitsglieder erläutert wurde, sind die Kanäle zu den Komponenten der modellierten Steuerungslogik für einen VIBN-Simulationslauf nicht notwendig, da die Signalversorgung und -verarbeitung die reale oder die virtuelle Steuerung übernimmt. Lediglich für eine interne Simulation des Funktionsmodells sind diese Verbindungen essentiell.

Als nächster Schritt werden die benötigten Daten aus dem E/FCAD-Modell ausgelesen und alle erforderlichen Komponenten der Steuerkette unter der Steuer- und Stellkomponente angelegt sowie die Variablen der zugehörigen hybriden Automaten parametrisiert. Hierzu wird die XML-Austauschdatei beginnend vom Arbeitsglied nach verbundenen Betriebsmitteln durchsucht. Auf diese Art und Weise wird die Steuer- und Stellkomponente hierarchisch weiter untergliedert.

Nachfolgend werden die Kanäle zwischen den Ports der einzelnen Komponenten erstellt, wobei dies in vier unterschiedlichen Phasen realisiert wird. Zuerst müssen neue Ports für die Steuer- und Stellkomponente angelegt werden, über welche das Arbeitsglied anschlossen wird. Dann werden die Verbindungen von den Ports der Steuer- und Stellkomponente nach innen zu den Komponenten der Stellglieder errichtet. Als nächster Schritt werden die Verbindungen zwischen den einzelnen Komponenten der Steuer- und Stellkomponente definiert. Zuletzt werden die Kanäle der neu integrierten Komponenten für die Kommunikation mit der Steuerungslogik zur übergeordneten Steuer- und Stellkomponente erzeugt. Hierfür können, wie bereits für die Arbeitsglieder ausführlicher erläutert, die E/A-Daten der TRMs genutzt werden.

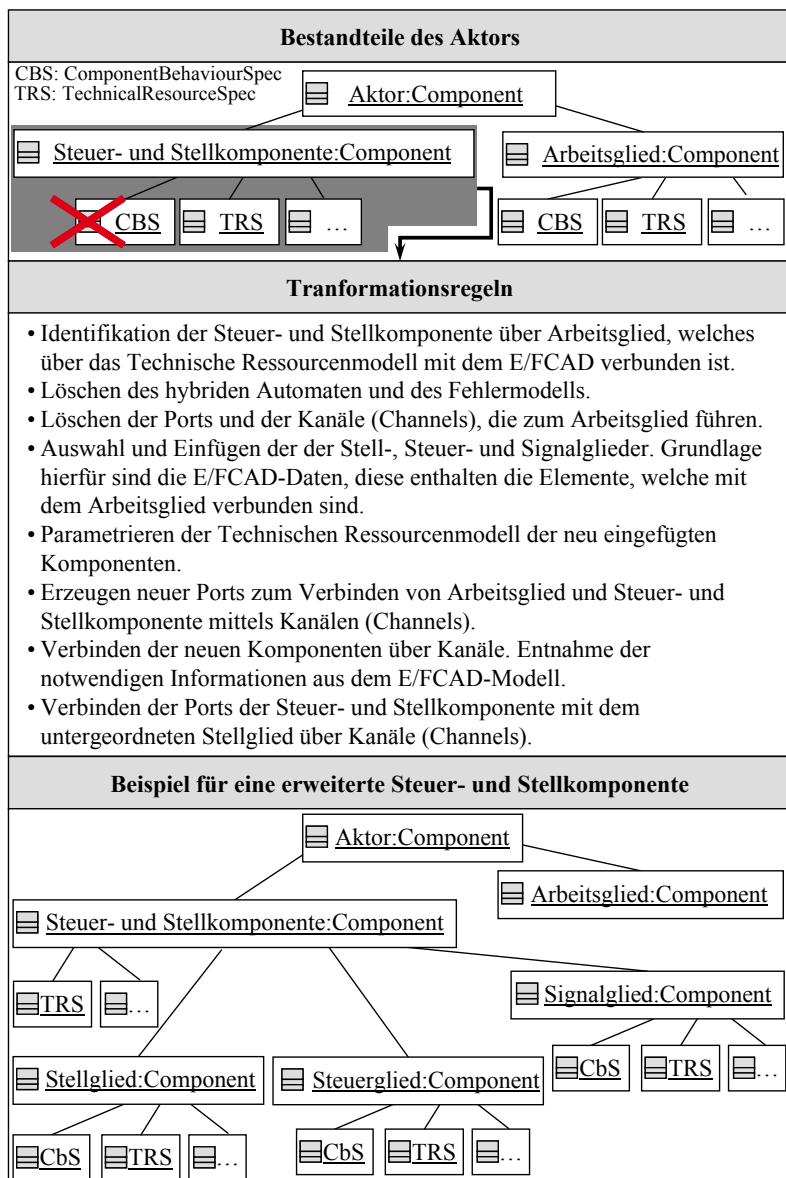


Abbildung 6.33: Transformationsregeln zur Erweiterung der Steuer- und Stellkomponente mit Hilfe der E/FCAD-Daten

Die Erstellung neuer Kanäle ist in Abbildung 6.34 verdeutlicht. In jedem Port sind die ein- und die ausgehenden Kanäle festgehalten (inChannels, outChannels). Beispielsweise kann auch ein InputPort einen ausgehenden Kanal (outChannel) haben, falls die zugehörige Komponente hierarchisch weiter untergliedert ist und die Informationen in das „Innere“ weitergeleitet werden sollen. Die inChannels und die outChannels müssen bei der Erzeugung neuer Kanäle auch in den Ports angepasst werden. Des Weiteren muss jeder Kanal den Quell- und den Zielpport (sourcePort und destinationPort) enthalten.

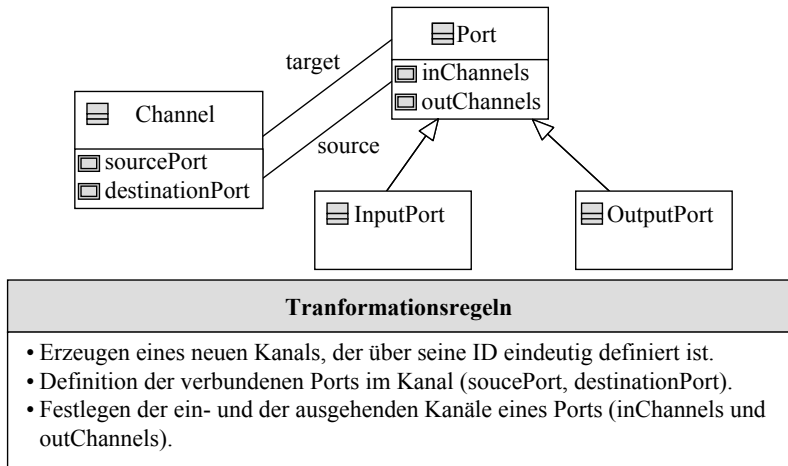


Abbildung 6.34: Erzeugen von Kanälen zum Verbinden von Ports – Ausschnitt des Metamodells in Anlehnung an AUTOVIBN (2010) bzw. HUMMEL (2011)

## 6.5.4 Integration der Sensordaten

### 6.5.4.1 Beschreibung der Anpassungen an Sensormodellen

Der Einfluss geometrischer Aspekte wurde durch die Integration der MCAD-Daten in die Sensorkomponenten bereits berücksichtigt. Auch hinsichtlich des Verhaltens der Sensoren kann mit Hilfe der E/FCAD-Daten eine Detaillierung erzielt werden. Als Beispiel sei ein inkrementeller Drehgeber genannt, der im Funktionsmodell stark abstrahiert zur Geschwindigkeits- bzw. zur Wegmessung herangezogen wird. Das reale Bauteil kann dagegen über erweiterte Funktionen

verfügen, die in der frühen Phase der Feinplanung noch nicht relevant sind. Außerdem ist für die Kommunikation mit der SPS die Abbildung umfangreicher Steuerwörter und Zustandsmeldungen erforderlich.

Die Integration der E/FCAD-Daten ist mit Hilfe der gleichen Herangehensweise umgesetzt, wie dies für die Arbeitsglieder der Aktorik geregelt ist, siehe Abschnitt 6.5.3.1 und Abbildung 6.29. Jeder Sensor wird in der Feinplanungsphase als eigene Komponente definiert. Die Verbindung zum CAD erfolgt im Mapping-Modell, wobei der Sensor direkt mit der Repräsentation im E/FCAD verbunden wird. Geändert werden die Automaten für den Gutablauf und den Fehlerfall sowie die Ports und das TRM. Die Grundlage hierfür bildet wiederum die Bibliothek, in welcher die zugehörigen Modellelemente hinterlegt sind.

### 6.5.4.2 Transformationsregeln für die Sensordatenintegration

Die Erweiterung der Sensormodelle gleicht in den Grundzügen dem des Arbeitsgliedes. Die Komponente wird strukturell nicht weiter untergliedert, sondern an den verhaltensbeschreibenden Bestandteilen angepasst. Die Transformationsregeln sind nachfolgend nochmals kurz zusammengefasst:

- Entfernen der Ports und Kanäle, Ausnahme sind spezielle Ports, wie z. B. für die Geschwindigkeitsübertragen (betreffende Ports wurden bereits in Abschnitt 6.5.3.1 genauer beschrieben)
- Löschen der hybriden Automaten, der Fehlerspezifikation und des TRM
- Einbinden und Parametrieren der Bibliothekselemente (Automaten, Ports, TRM) sowie deren Konsistenzprüfung (z. B. zu den Detektoren)
- Verbinden der Ports über neue Kanäle

### 6.5.5 Anpassung des Technischen Ressourcenmodells

In den vorangegangenen Abschnitten wurde mehrfach erwähnt, dass die E/A-Daten in das TRM der zugehörigen Komponente geschrieben werden. Die Aufgaben des TRMs als Bindeglied zwischen Funktionsmodell und Steuerung wurden in Abschnitt 5.4.4 bereits erörtert. Steuerungsein- und -ausgänge werden zunächst in Felder aufgeteilt und dann mittels der Adapter den Ports im Funktionsmodell zugewiesen. Die Felder und die Adapter sind für jedes technische Bauteil spezifisch und unveränderlich. In den TRMs der Modellbibliothek sind diese hinter-

legt. Im Gegensatz dazu sind die Steuerungsadressen anzupassen, die Daten werden hierzu aus dem CAD übernommen. Die Voraussetzung dafür ist ein erfolgreich absolvierter Konsistenztest, wie er in Abschnitt 6.3.7 behandelt wurde.

Im Folgenden wird die Anpassung des TRMs mit Hilfe der E/FCAD-Daten erläutert. Wie Abbildung 6.35 verdeutlicht, enthalten die Adressen des TRM sogenannte *PacketFields*, die oben auch als Felder bezeichnet wurden (lower und upper schränken den Bitbereich ein). Diese Felder sind von der Anpassung nicht betroffen, da sie projektspezifisch und fest für jeden Aktor oder Sensor definiert sind. Dies gilt in gleicher Weise für die Adapter (IOadapter). Geändert wird die Steuerungsadresse, die sich aus der Anfangsadresse (ioaddress) sowie der Größe des Adressbereichs (Bit, Byte, Word etc.) zusammensetzt.

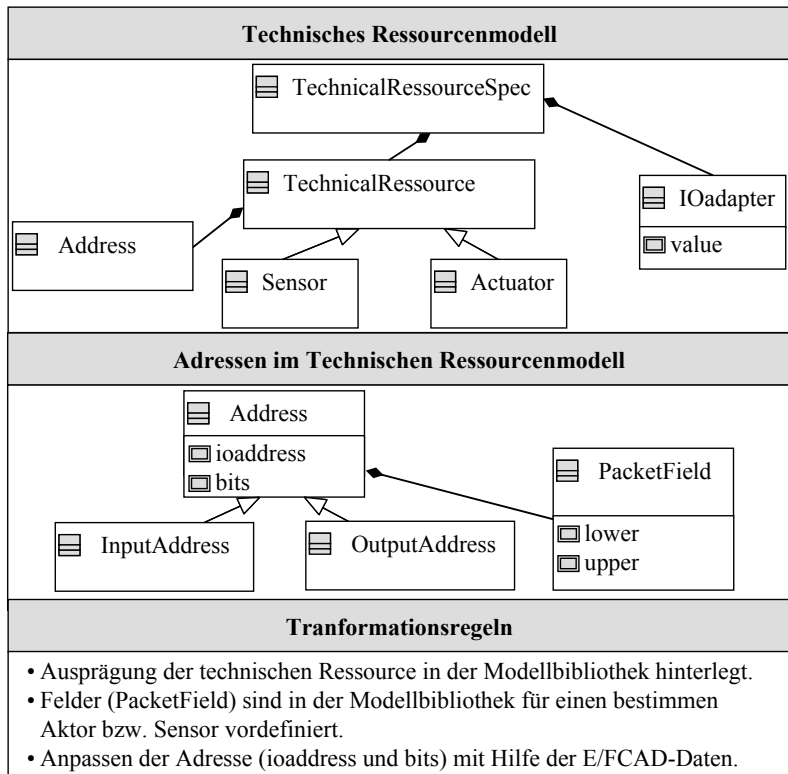


Abbildung 6.35: Transformationsregeln für das Technische Ressourcenmodell – Metamodell nach AUTOVIBN (2010)

## 6.6 Automatische Generierung des VIBN-Modells

### 6.6.1 Ansatz zur Modellgenerierung

Die Grundlage zur Ableitung des VIBN-Modells bildet das Funktionsmodell. Wie in Abschnitt 5.2 ausgeführt, lässt sich bereits die Variante der Feinplanungsphase zur Generierung von VIBN-Modellen heranziehen, solange die notwendigen TRMs mit den Adaptern für die Steuerungsanbindung angelegt wurden. Es ist jedoch ein solches Modell vom Informationsgehalt her im Vergleich zu der durch CAD-Daten erweiterten Version in der Aussagekraft reduziert. Dennoch ist damit ein erster Test von Steuerungscode in frühen Entwicklungsphasen möglich.

Das VIBN-Modell wird als C++-Code ausgeleitet, siehe Abbildung 6.36. Da im Funktionsmodell das Gesamtsystem inklusive Steuerung abgebildet ist, darf bei der Generierung lediglich der Maschinenanteil berücksichtigt werden. Zudem müssen die Kopplungen zur Steuerung erstellt werden. Zu diesem Zwecke wird das TRM in den Prozess mit einbezogen, da es die dafür notwendigen Informationen enthält. Des Weiteren werden die Bedienelemente sowie das Visualisierungsmodell abgeleitet. Das derart erstellte Simulationsmodell lässt sich mit einer realen (HiL) oder einer virtuellen (SiL) Steuerung verbinden.

Die beschriebene Herangehensweise hat den Vorteil, dass durch die Verwendung eines im Rahmen dieser Arbeit neu entwickelten Simulationssystems alle wesentlichen Elemente des Funktionsmodells in das VIBN-Modell überführt werden können. Es entstehen dadurch keine Inkompatibilitäten, wie es bei der Nutzung kommerzieller Simulationslösungen in der Regel der Fall wäre. Diese erlauben zum einen nur einen beschränkten Zugriff von außen. Zum anderen lassen sich die Modellmerkmale der übergeordneten Ebene mit dem im Simulationssystem verfolgten Ansatz nicht zwangsläufig vereinbaren, z. B. wird das Automatenmodell nicht unbedingt unterstützt.

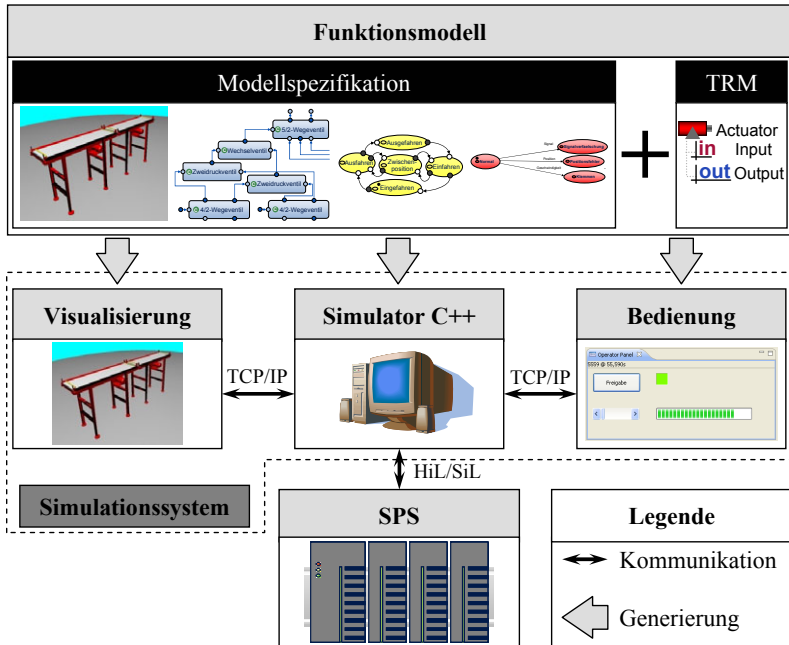


Abbildung 6.36: Generierung des VIBN-Modells als C++-Programm auf der Basis des Funktionsmodells

### 6.6.2 Ausleitung des VIBN-Modells

Aus dem Gesamtfunktionsmodell dürfen für den Steuerungstest nur die Maschinenanteile ausgeleitet werden. Die hier vorgestellte Methode löst die beschriebene Problemstellung in der Weise, dass jede Maschinenkomponente durch ein TRM zu kennzeichnen ist und auch nur in diesem Fall generiert wird, siehe Abbildung 6.37. Die Kommunikationsverbindungen zwischen den derart markierten Komponenten bleiben bestehen. Sofern das TRM zusätzlich Kopplungsinformationen zur Steuerung (Felder, Adapter, E/A-Daten) beinhaltet, werden die zugehörigen Ports darüber mit Nachrichten der realen oder der virtuellen SPS versorgt bzw. versenden Daten an diese.

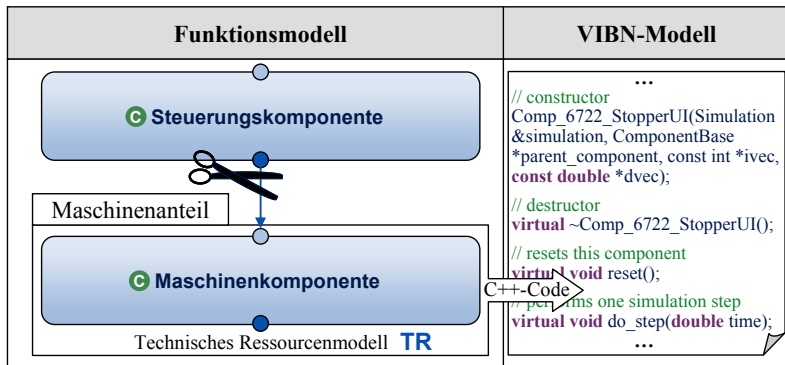


Abbildung 6.37: Berücksichtigung der Maschinenanteile bei der Generierung des VIBN-Modells

Das in der Programmiersprache C++ generierte Modell enthält folgende Bestandteile des Funktionsmodells:

- Komponentenstruktur
- Maschinenverhalten
- Schnittstellen, Kommunikation
- Kollisionserkennung und -handhabung
- Materialquellen und -senken
- Materialfluss
- HiL- oder SiL-Kopplung
- Simulationssteuerung und Visualisierung

Der ausgederivierte Quellcode ist in sogenannte *Jobs* gegliedert, welche die oben erläuterten Aspekte abbilden. Weitere Informationen zur softwaretechnischen Umsetzung sind in Abschnitt 7.2.1.3 zu finden.

### 6.6.3 Steuerungskopplung, Visualisierung und Bedienung

Um eine VIBN durchführen zu können, ist die Integration der realen oder der virtuellen Steuerung erforderlich. Darüber hinaus wird zur Verfolgung von Bewegungsvorgängen eine Visualisierung benötigt. Daher muss der in Abschnitt 6.6.2 erläuterte Simulationskern über entsprechende Kopplungen verfügen. Für die Visualisierung und die Simulationssteuerung wird die Entwicklungsumgebung der Funktionsmodellierung genutzt. Die Anbindung über das Netzwerkprotokoll TCP/IP ermöglicht die getrennte Ausführung von Simulation und Visuali-



sierung auf unterschiedlichen Rechnern. Dies hat den Vorteil, dass der Simulationsrechner von zusätzlichen Aufgaben entlastet wird.

Für die HiL-Kopplung kommt im Rahmen der vorliegenden Dissertation eine Hardware-Simulationskarte zum Einsatz, die über Feldbusanschlüsse (PROFIBUS) verfügt, siehe Abbildung 6.38. Es handelt sich hierbei um ein Produkt der Firma Siemens AG, die sogenannte *SIMBApro-PCI-Karte*, die den aktuellen Stand der Technik darstellt und fast ausschließlich für derartige Anwendungen eingesetzt wird. Auf der Karte werden alle Feldbusteilnehmer mit dem zugehörigen Protokoll simuliert. Über einen PCI-Steckplatz (Peripheral Component Interconnect) ist diese in den Simulationsrechner integriert. Die eigentlichen Nutzdaten, hier die Steuerungsein- und -ausgänge, werden mit Hilfe der generierten Adapter vom Simulationskern ausgelesen bzw. geschrieben.

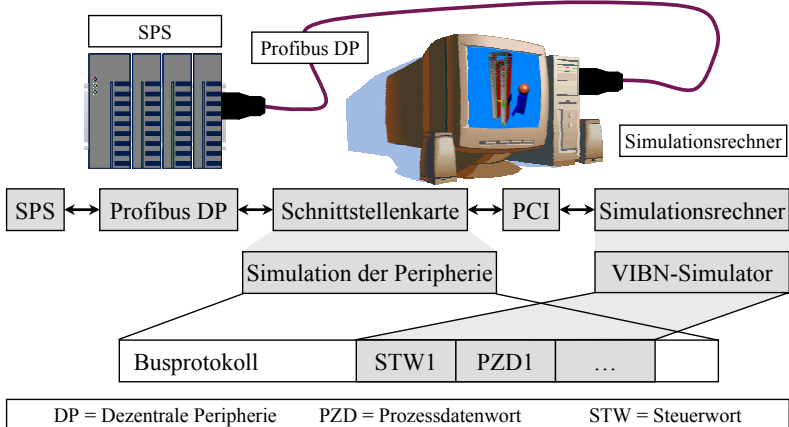


Abbildung 6.38: HiL-Kopplung mit Hilfe einer Hardware-Simulationskarte

Für die SiL-Simulation wird als virtuelle Steuerung eine simulierte CPU der Firma Siemens AG verwendet, welche in Form des Produktes *PLCSIM* kommerziell erhältlich ist. Das Steuerungsprogramm wird auf die virtuelle anstatt auf die reale CPU geladen. Der Simulator greift direkt über eine Softwareschnittstelle, das sogenannte *PROSIM*-Interface, auf die Steuerungsein- und -ausgänge zu. Die Taktung der virtuellen Steuerung ist über den VIBN-Simulator geregelt.

### 6.6.4 Änderungen am Modell

Da das VIBN-Modell aus generiertem C++-Code besteht, ist ein Nachvollziehen der einzelnen Bestandteile nur noch bedingt bzw. mit erheblichem Aufwand möglich. Aus diesem Grund ist bei Änderungen eine Neugenerierung des VIBN-Modells sinnvoll. Modifikationen sind somit zunächst im Funktionsmodell oder in den CAD-Daten umzusetzen, bevor anschließend das VIBN-Modell ein weiteres Mal ausgeleitet werden kann. Diese Vorgehensweise hat den Vorteil, dass Anpassungen im entwicklungsübergreifenden Funktionsmodell vorzunehmen bzw. einzupflegen sind, wodurch sich der Informationsfluss zwischen den Fachabteilungen verbessert.

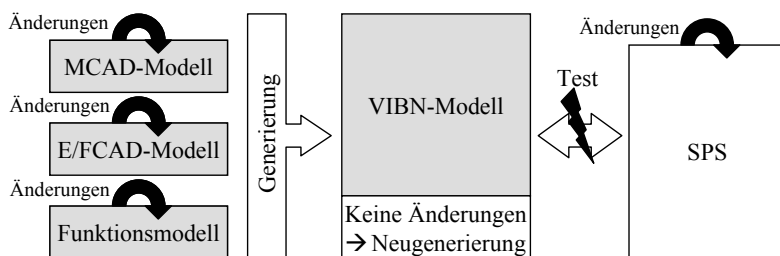


Abbildung 6.39: *Einpfelegen von Änderungen am VIBN-Modell durch Neugenerierung*

## 6.7 Modelltest

### 6.7.1 Testmethoden

Eine VIBN kann nur dann zu verlässlichen Ergebnissen führen, wenn das verwendete Maschinenmodell die Realität dem Modellzweck entsprechend abbildet. In Kapitel 3 wurden bereits Prüfmethoden für Software betrachtet. Die dort beschriebenen Ansätze lassen sich im Prinzip auch auf Simulationssoftware bzw. -modelle anwenden. Im Rahmen der vorliegenden Arbeit werden dynamische Testverfahren vorgeschlagen, da diese im Ingenieurbereich eine große Rolle spielen.

Die Modellierung neuer Komponenten verlangt vom Entwickler, deren korrektes Verhalten sicherzustellen. Daher muss mit Hilfe von Testläufen untersucht wer-

den, ob das betrachtete Modell die gestellten Anforderungen erfüllt. Das Prüfobjekt wird dabei gezielt durch die Testdaten, beispielsweise Systemeingaben, stimuliert und die Reaktionen werden ausgewertet. Die Zielstellung hierbei ist, potenziell vorhandene Fehler im Modell zu identifizieren. Die Testmethode kann unterschiedlich ausgeprägt sein, wobei für den hier präsentierten Ansatz strukturorientierte und funktionsorientierte Verfahren empfohlen werden. In Abbildung 6.40 sind die Unterschiede zwischen den beiden Herangehensweisen dargestellt. Während beim funktionsorientierten Testen (eine Black-Box-Methode) nur die Reaktionen an der Systemgrenze mit Hilfe der Werte an den Ports einer Komponente betrachtet werden, ist beim strukturorientierten Test (eine White-Box-Methode) der innere Aufbau der Komponente mit einbezogen.

Die Spezifikation legt die Sollfunktionalität der Komponente fest. Beim funktionsorientierten Testen werden auf deren Basis die Testfälle abgeleitet und die Vollständigkeit bewertet, siehe Abbildung 6.40 links. Im Gegensatz dazu richten sich beim strukturorientierten Test die Testfälle an der inneren Struktur des Prüflings aus, siehe Abbildung 6.40 rechts. Die Vollständigkeit wird anhand der Abdeckung der Strukturelemente, wie z. B. Anweisungen oder Zweige, beurteilt. Im Hinblick auf das Funktionsmodell wären dies etwa die Anzahl der eingenommenen Zustände oder die ausgeführten Transitionen. Eine Evaluierung der Testergebnisse muss mit Hilfe der Spezifikation vorgenommen werden (LIGGESMEYER 2009).

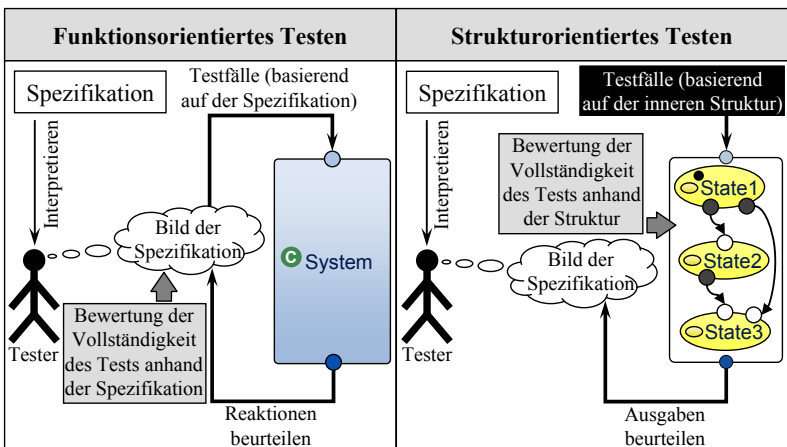


Abbildung 6.40: Funktionsorientiertes und strukturorientiertes Testen von Simulationsmodellen in Anlehnung an LIGGESMEYER (2009)

Darüber hinaus spielen im Unterschied zum klassischen Softwaretest hier die weiteren Modellelemente, wie z. B. die Geometrie in Form von Parts oder die Mover, eine wesentliche Rolle. Diese sind in den Testvorgang zu integrieren, da ansonsten beim strukturorientierten Test nicht alle Pfade des Automaten durchlaufen werden können.

Auf Grund dessen, dass für die Visualisierung und die Darstellung der Simulationsergebnisse sowohl während der Funktionsmodellierung als auch bei der VIBN die gleiche Entwicklungsumgebung genutzt wird, ist ein einfaches Nachvollziehen der Vorgänge möglich. Dies lässt sich beispielsweise anhand der Automatendarstellung verdeutlichen. Obwohl das Modell in schwer verständlichen C++-Code übersetzt wird, können die eingenommenen Zustände und die ausgeführten Transitionen nach wie vor im Editor grafisch verfolgt werden. Somit ist für den Benutzer kein Umdenken hinsichtlich der Modellierungstechnik erforderlich.

### 6.7.2 Test der Simulationsmodelle

Die hier vorgestellte Funktionsmodellierung findet auf unterschiedlichen Abstraktionsniveaus statt (Grobplanung, Feinplanung und erweitertes Funktionsmodell). Eine weitere Abstufung stellt das VIBN-Modell dar, welches zwar vom Verhalten grundsätzlich dem erweiterten Funktionsmodell entspricht, jedoch um die HiL- oder die SiL-Kopplung ergänzt und als C++-Programm übersetzt wird. Ein Test ist auf allen Ebenen notwendig. Da ein vollständiger Test, d. h. die Berücksichtigung aller möglichen Eingabekombinationen, in der Praxis nicht möglich ist, müssen geeignete Testfälle definiert werden. Nach LIGGESMEYER (2009) muss deren Auswahl repräsentativ, fehlerintensiv, redundanzarm und ökonomisch sinnvoll sein.

Im Folgenden soll eine Testmethode für die Modelle von der Grobplanung bis hin zur VIBN beschrieben werden. Hierbei kommen der funktionsorientierte und der strukturorientierte Test zum Einsatz. Da Letzterer sich vor allem bei großen Zustandsräumen wesentlich aufwendiger gestaltet, wird eine Kombination der beiden Verfahren vorgeschlagen. Zudem können mit diesen Ansätzen unterschiedliche Fehlerarten identifiziert werden. So kann beispielsweise eine erfolgreich funktionsorientiert getestete Komponente dennoch strukturelle Fehler aufweisen, wenn bestimmte Pfade des Automaten nicht durchlaufen wurden.

Das Modell der Grobplanungsphase wird auf der Grundlage der Kundenanforderungen erstellt und ist hinsichtlich dieser zu testen. Dabei wird ein sogenannter Bottom-up-Ansatz verfolgt, siehe Abbildung 6.41. Zunächst sind auf der untersten Ebene die Einzelkomponenten zu prüfen. Die vorgestellte Beschreibungstechnik erlaubt die Verwendung von Zustandsmaschinen nur für nichthierarchische Komponenten, siehe Abschnitt 5.3.4.1. Somit spiegeln diese die Basiselemente wider, wie z. B. einen stark abstrahierten Aktor, aus denen sich dann Module oder das Gesamtsystem zusammensetzen lassen. Da die Automaten für solche Komponenten relativ einfach und übersichtlich aufgebaut sind, kann neben einem funktionsorientierten gleichermaßen ein strukturorientierter Test zweckmäßig angewendet werden. Auf Grund der begrenzten Anzahl der Zustände und der Transitionen ist es sinnvoll, alle Pfade im Modell zu durchlaufen und alle Bedingungen in den Transitionen zu prüfen. Der Aufwand hierfür kann durch die häufige Wiederverwendung derartiger Komponenten gerechtfertigt werden. Für den Test von Modulen wird ein funktionsorientierter Ansatz vorgeschlagen. Dies lässt sich dadurch begründen, dass sich Module aus bereits getesteten Komponenten aufbauen lassen, die nur über ihre Schnittstellen (Ports) kommunizieren. Des Weiteren würde sich für einen strukturorientierten Test auf Modulebene eine stark steigende Anzahl von Testfällen ergeben, die aus den Kombinationsmöglichkeiten von Zuständen oder Transitionen resultiert. Für den abschließenden Test des Gesamtsystems empfiehlt sich aus denselben Gründen eine funktionsorientierte Durchführung.

Auf der Ebene der Feinplanung sind die gleichen Testmethoden wie für die Grobplanung vorgesehen, was somit auch den Bottom-up-Ansatz umfasst. Hinsichtlich des Komponententests auf der untersten Stufe sind zusätzlich die Automaten der Fehlerszenarien zu prüfen. Da die Ein- und Ausgabefilter ohnehin nur auf die Ports der Komponente sowie die Variablen der Differenzialgleichungen zugreifen, eignet sich hier besonders der funktionsorientierte Test. Dagegen muss der Fehlerautomat an sich in struktureller Weise getestet werden.

Das mit CAD-Daten erweiterte Funktionsmodell wird durch Geometrieinformationen und um detaillierte Aktor- bzw. Sensorkomponenten ergänzt. Für die Letztgenannten kommt eine Bibliothek mit grundlegenden Komponenten zum Einsatz, die ebenso, wie die Komponenten auf der untersten Ebene der Grob- und der Feinplanung, funktions- und strukturorientiert zu prüfen sind. Dies schließt die Automaten zur Fehlerspezifikation mit ein. Das mit Hilfe von CAD-Daten automatisiert erweiterte Funktionsmodell ist abschließend durch funktionsorientierte Tests abzusichern.

Die als C++-Programm generierten VIBN-Modelle stellen eine Übersetzung der Funktionsmodelle dar und weisen dementsprechend die gleichen Verhaltensmerkmale auf. Dennoch muss das VIBN-Modell geprüft werden, da zur Kommunikation mit der realen oder der virtuellen Steuerung die Adapter der TRMs eine Transformation an den Schnittstellen vornehmen. Hierfür ist ein funktionsorientierter Test ausreichend, es wird lediglich die korrekte Verbindung bzw. Umrechnung der Ein- und Ausgangssignale sichergestellt.

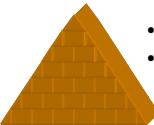
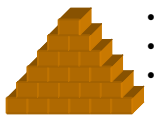
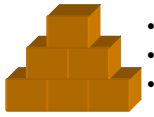
<b>VIBN-Modell</b>	Technisches Ressourcenmodell: Funktionsorientierter Test	
<b>Erweitertes Funktionsmodell</b>		<ul style="list-style-type: none"><li>• Gesamtsystem: Funktionsorientierter Test</li><li>• Bibliothekselemente auf der untersten Komponentenebene: Funktions- und struktororientierte Tests</li></ul> <div>Bottom-Up ↑</div>
<b>Feinplanung</b>		<ul style="list-style-type: none"><li>• Gesamtsystem: Funktionsorientierter Test</li><li>• Modulebene: Funktionsorientierter Test</li><li>• Unterste Komponentenebene: Funktions- und struktororientierte Tests</li></ul> <div>Bottom-Up ↑</div>
<b>Grobplanung</b>		<ul style="list-style-type: none"><li>• Gesamtsystem: Funktionsorientierter Test</li><li>• Modulebene: Funktionsorientierter Test</li><li>• Unterste Komponentenebene: Funktions- und struktororientierte Tests</li></ul> <div>Bottom-Up ↑</div>

Abbildung 6.41: Teststrategie und Vorgehen beim Prüfen der unterschiedlichen Modelle

Um das Gesamtsystem, Module oder Einzelkomponenten zu testen, müssen diese mit entsprechenden Stimuli versorgt werden. Da das Gesamtmodell auch ein Steuerungsmodell enthält, können damit die notwendigen Szenarien implementiert und geprüft werden. Ist das Testobjekt hingegen ein Modul oder eine Einzelkomponente, so muss dafür eigens ein Testrahmen erstellt werden. Dieser besteht aus einer oder mehreren Komponenten, welche über die Ports die Eingangssignale für das Prüfobjekt generieren. Dafür stehen prinzipiell zwei Möglichkeiten zur Verfügung. Zum einen kann dies über ein Bedienpanel und somit manuell durch den Entwickler geschehen, was aber in der Regel eine unzureichende Reproduzierbarkeit mit sich bringt. Zum andern kann für die Komponenten des Testrahmens auf die hybriden Automaten zurückgegriffen werden. Dadurch lässt

sich der Vorgang unter gleichen Bedingungen beliebig oft wiederholen. Zudem ist die zeitliche Folge der Signale exakt einstellbar.

## 6.8 Generierung rudimentärer Konstruktionsmodelle

Der Nutzen eines disziplinenübergreifenden Funktionsmodells ist nicht nur auf die VIBN beschränkt. Ein solches Abbild der Maschine kann zur Unterstützung des Gesamtentwicklungsprozesses herangezogen werden. Insbesondere ist dabei die Generierung weiterer Modelle auf der Konstruktionsebene, wie z. B. Grobmodelle für die CAD-Systeme, zu nennen. Im Rahmen dieser Arbeit soll hierfür kein vollständiger Ansatz präsentiert, jedoch die generelle Umsetzbarkeit kurz aufgezeigt werden.

Die automatisierte Erstellung von Stromlauf- und Fluidplänen war und ist Gegenstand vieler Forschungsarbeiten (VDMA 2004, GRÄTZ 2006, WÜRLIN & WASCHNER 2008), aus denen kommerzielle Produkte, wie das *Eplan Engineering Center* der Firma Eplan oder *Comos ME* der Firma Siemens Industry Software (ehemals Firma innotec), hervorgegangen sind. Unter Zuhilfenahme eines übergeordneten Modells lassen sich die jeweiligen Entwicklungsdokumente ableiten. Diese Rolle kann das hier vorgestellte Funktionsmodell übernehmen. Das TRM und insbesondere die darin hinterlegten technischen Details (TD) bilden die Grundlage zur Generierung von Stromlauf- oder Fluidplänen. Die automatisierte Erstellung von Blättern und das Anlegen von Betriebsmitteln sowie deren Verschaltung können über die Programmierschnittstelle des CAD-Werkzeuges ausgelöst werden. Als Informationsträger wird, wie schon beim Datenimport in das Funktionsmodell, ein XML-Austauschformat verwendet, das über entsprechende Informationen verfügt.

Ebenso können Teile des MCAD-Modells automatisch generiert werden. Im einfachsten Fall wird die im Funktionsmodell definierte Geometrie direkt übertragen, siehe Abbildung 6.42 links unten. Darüber hinaus können auch komplexere Strukturen auf der Basis parametrisch-assoziativer CAD-Modelle erstellt werden. Die Vorzüge dieser Konstruktionsart wurden bereits in Abschnitt 6.4.2 erörtert. Die im Funktionsmodell abgebildeten Daten werden zum Parametrieren der vordefinierten Modelle im CAD verwendet. In Abbildung 6.42 rechts unten ist dies für Teilbereiche eines Förderbandsystems dargestellt. Konstruktionsrichtlinien sind in diesem Zusammenhang von Nutzen; beispielsweise können ab einer bestimmten Bauteilgröße zusätzliche Lagerstellen automatisch angelegt werden.

Des Weiteren kann für das MCAD-Modell eine grobe Bauteilstruktur erzeugt werden, die sich am Aufbau des Funktionsmodells orientiert. Zudem lassen sich Attribute aus der übergeordneten Ebene integrieren. Dies ist zur eindeutigen Identifikation von Bauteilen nutzbar, da hierdurch die IDs aus der Funktionsbeschreibung in den zugehörigen Elementen des MCAD-Modells hinterlegt werden können. Auf diese Art und Weise ist es möglich, Stücke des Mapping-Modells automatisch zu generieren. Die beschriebene Vorgehensweise wurde exemplarisch umgesetzt, wobei der Informationsaustausch wiederum auf einem XML-Format basiert.

Da moderne MCAD-Werkzeuge mitunter über Simulatoren für die Mehrkörper-simulation verfügen, ist eine Unterstützung dafür ebenso denkbar. Es lassen sich beispielsweise die kinematischen Beziehungen des Funktionsmodells für diese Zwecke heranziehen.

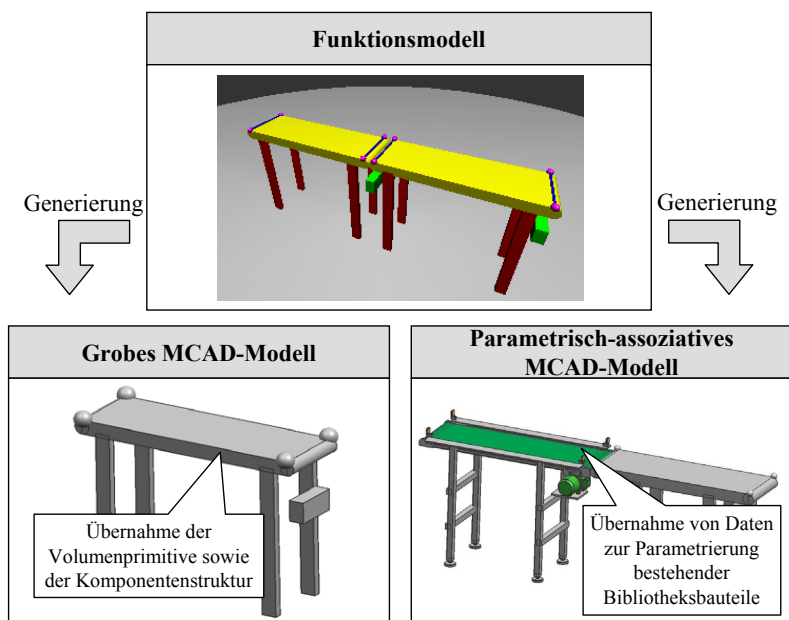


Abbildung 6.42: Generierung eines MCAD-Modells mit Hilfe der Daten aus dem Funktionsmodell



## 6.9 Zusammenfassung

In Kapitel 6 wurde eine Methode zur Integration von CAD-Daten in das Funktionsmodell präsentiert. Als Grundlage dient ein XML-Austauschformat, das die notwendigen Informationen enthält. Dadurch ist eine Unabhängigkeit von den verwendeten CAD-Systemen gegeben. Eine Konsistenzprüfung stellt vorab sicher, dass das Funktions- und die CAD-Modelle den gleichen Entwicklungsstand beschreiben. Des Weiteren wurde gezeigt, wie sich das Funktionsmodell mit Hilfe von CAD-Daten erweitern lässt. Die gewählte Vorgehensweise ermöglicht eine voneinander unabhängige Einbindung der MCAD- und der E/FCAD-Informationen, da diese unterschiedliche Bestandteile des Funktionsmodells detaillieren. Das VIBN-Modell wird auf der Basis des Funktionsmodells in Form eines C++-Programmes abgeleitet. Hierfür wurde ein eigener Simulator entwickelt, der eine vollständige Übernahme der Modellcharakteristika, wie z. B. der Kollisionserkennung oder des Automatenmodells, gestattet. Ferner wurde eine Teststrategie vorgestellt, mit welcher sich die Modelle der jeweiligen Abstraktionsstufen prüfen lassen. Das Kapitel schließt mit einem Ausblick ab, der die Nutzung des Funktionsmodells für die Ableitung weiterer Entwicklungsartefakte auf der Konstruktionsebene, insbesondere CAD-Modelle, umschreibt.

In den Kapiteln 5 und 6 wurde die im Rahmen dieser Dissertation entwickelte Methode ausführlich vorgestellt. Die Bewertung dieser Methode ist Gegenstand des nachfolgenden Kapitels.



## 7 Prototypische Umsetzung und Wirtschaftlichkeitsbetrachtung

### 7.1 Kapitelüberblick

Im siebten Kapitel werden die entwickelten Prototypen für die Funktionsmodellierung und den VIBN-Simulator vorgestellt. Darauf folgend werden zwei industrielle Anwendungsbeispiele präsentiert, die zur Validierung der Methode dienen. Abschließend findet eine Wirtschaftlichkeitsbewertung des Ansatzes statt.

### 7.2 Entwicklungs- und Simulationsumgebung

#### 7.2.1 Funktionsmodellierung

##### 7.2.1.1 Allgemeines

Die in Kapitel 5 vorgestellte Modellierungstechnik wurde für eine prototypische Anwendung in einem Softwarewerkzeug umgesetzt. Die Basis für die Funktionsmodellierung bildet das Werkzeug AutoFOCUS, welches ursprünglich aus dem Bereich eingebetteter Systeme stammt (BROY ET AL. 1999). Im Rahmen des Projektes AUTOVIBN (2010) wurde dieses speziell für den Maschinen- und Anlagenbau zum Softwareprototyp *STEM (Spatio-Temporal Engineering Models)* angepasst und erweitert.

##### 7.2.1.2 Editieren von Modellen

Das Entwicklungswerkzeug STEM setzt sich aus einem Editor und einem Simulator zusammen. Im Editiermodus stehen die in Kapitel 5 vorgestellten Modellierungselemente zur Verfügung, wie z. B. Komponenten oder hybride Automaten. Der Arbeits- bzw. Editierbereich ist zentral angeordnet, siehe Abbildung 7.1. Auf der rechten Seite ist eine Bibliothek mit den Modellbestandteilen installiert. Zur Linken wird die Projektstruktur angezeigt. Die Modellierung ist zum größten Teil grafisch-interaktiv gestaltet, für zusätzliche Details existieren Eingabemasken, wie z. B. für die Differenzialgleichungen der hybriden Automaten.

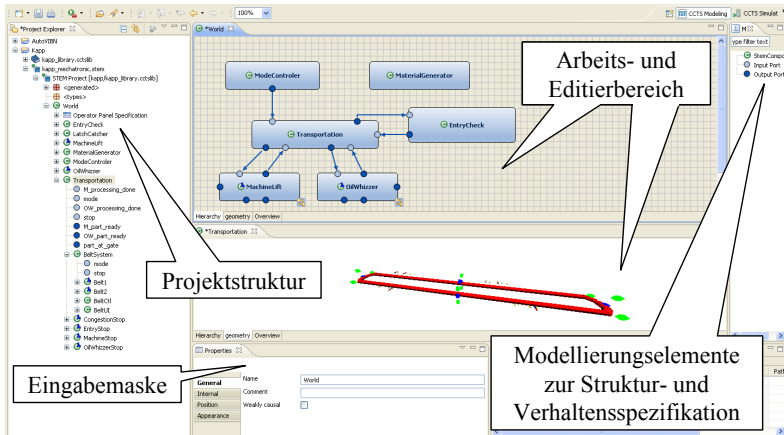


Abbildung 7.1: Editor des Werkzeugprototyps zur Funktionsmodellierung

### 7.2.1.3 Simulation

Zur Analyse und zum Test der modellierten Maschine kann der interne Simulationsmodus des Entwicklungswerkzeuges genutzt werden, siehe Abbildung 7.2.

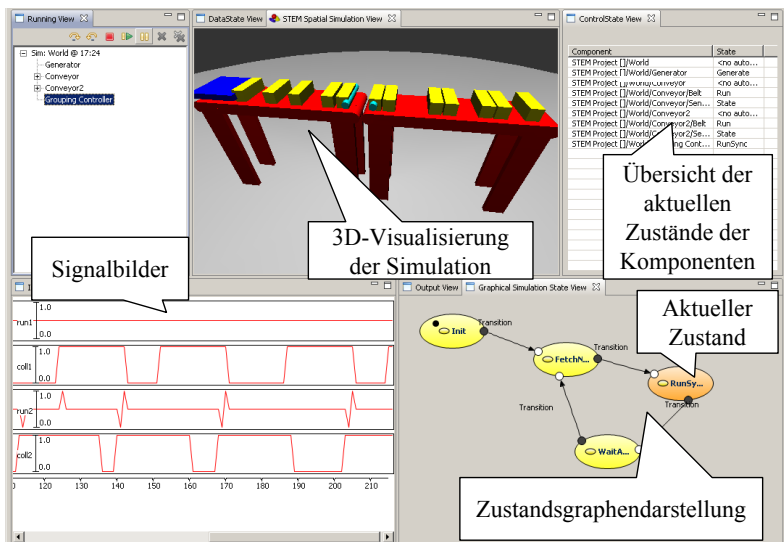


Abbildung 7.2: Simulationsmodus des Funktionsmodells

## 7.2.2 CAD-Datenintegration und VIBN-Simulator

### 7.2.2.1 CAD-Datenexport

Für die vorliegende Arbeit kamen die CAD-Werkzeuge *NX* der Firma Siemens PLM Software GmbH für die Mechanik sowie *E<sup>3</sup>.series* der Firma Zuken für die Elektrotechnik und die Fluidik zum Einsatz. Der Datenaustausch mit *NX* erfolgt über ein externes Schnittstellenprogramm, welches in der Programmiersprache Java verfasst ist. Für den Zugriff auf den Kern des CAD-Werkzeuges stellt *NX* die API *NX Open* bereit. Dadurch sind umfangreiche Manipulationen lesender und schreibender Art am hinterlegten Modell möglich. Die notwendigen Java-Klassen hierfür sind in der Java-Archive-Datei *NXOpen.jar* hinterlegt.

Das ECAD-Werkzeug *E<sup>3</sup>.series* verfügt über eine Schnittstelle auf der Basis des von der Firma Microsoft entwickelten *Component Object Model (COM)*. Als Programmiersprache fand, wie schon für das MCAD, Java Verwendung. Mit Hilfe der COM4J-Technik besteht die Möglichkeit, einen Zugriff mit Java auf das COM-Objekt zu realisieren.

### 7.2.2.2 Datenintegration in das Funktionsmodell

Das Funktionsmodell der Feinplanungsphase, die CAD-Daten und das Mapping-Modell sind die Eingangsinformationen für die automatisierte Erweiterung. Dabei wird zunächst geprüft, ob sich die Anpassungen mit der Modellkonsistenz vereinbaren lassen, siehe Abbildung 7.3 (Funktionsmodell in diesem Zusammenhang als STEM-Modell betitelt). Dies umfasst z. B. in der Mechanik, ob die Bezeichnungen der zu importierenden Parameter übereinstimmen. Hinsichtlich der E/FCAD-Daten wird beispielsweise sichergestellt, dass die Typen der Ports gültig sind oder der in das Arbeitsglied eingefügte hybride Automat die vorhandenen Bewegungsachsen, Parts oder Detektoren richtig adressiert. Neben den CAD-Daten in Form einer XML-Datei sind im Falle der Mechanik die VRML-Modelle der einzubindenden Parts notwendig. Der Mechanismus zur Ausleitung der VRML-Daten ist in einem eigenen Softwareprototyp implementiert, da hierdurch die Unabhängigkeit vom CAD-Werkzeug gewahrt bleibt. Dieser leitet unter Einbeziehung des Mapping-Modells die passenden polygonreduzierten VRML-Modelle aus. Für die Erweiterung mit E/FCAD-Daten sind zusätzlich Bibliothekselemente erforderlich, welche in das Funktionsmodell integriert werden.

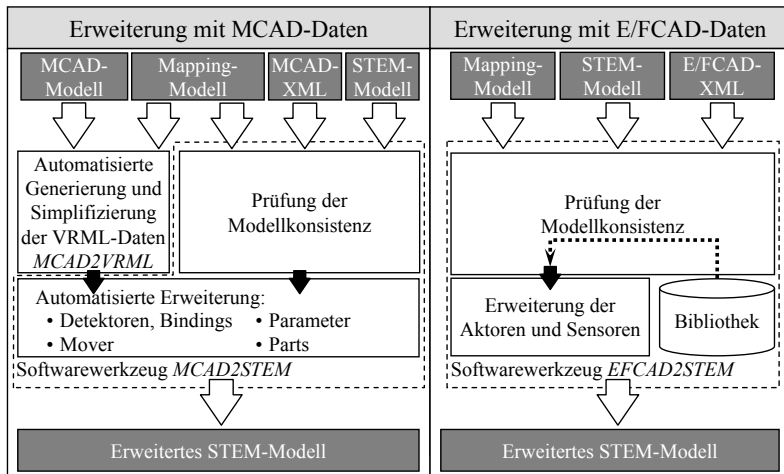


Abbildung 7.3: Prinzipielle Funktionsweise der Softwareprototypen zur Erweiterung des Feinplanungsmodells mit Hilfe von CAD-Daten

### 7.2.2.3 VIBN-Simulator

Das ausgeleitete VIBN-Modell besteht aus C++-Code, der in sogenannte *Jobs* eingeteilt ist, die unterschiedliche Aufgaben innehaben. Im Folgenden werden diese kurz erläutert:

- AGJob: Generierung und Löschen von Materialobjekten
- DelayJob: Sicherstellung der Zykluszeit
- DetectorsJob: Kollisionsberechnung der Detektoren und der Bindings
- MotionJob: Berechnung von Bewegungsvorgängen unter Berücksichtigung von Kollisionen
- SCSCPJob: Netzwerkkommunikation für die Visualisierung und die Bedienung
- SimulationJob: Ausführen der Simulation

Je nach Kopplungsart wird zudem die Steuerungsanbindung erstellt:

- SimbaJob: Kopplung für die HiL-Simulation
- SILPlcSimJob: Kopplung für die SiL-Simulation

Der automatisch geschriebene C++-Code muss vor dem Starten der Simulation kompiliert werden. Der Simulator an sich besteht aus einer Anwendung (*simulation.exe*), die vom Entwickler aufzurufen ist. Zur Visualisierung und zur Bedienung wird der Simulationsmodus des Softwarewerkzeuges STEM verwendet.

## 7.3 Anwendungsbeispiele

### 7.3.1 Allgemeines

Die vorgestellte Methode wurde mit Hilfe von zwei industriellen Anwendungsbeispielen unter praxisgleichen Bedingungen erprobt. Es handelt sich dabei um jeweils einen Ausschnitt eines Produktionssystems der Firma Gebr. Heller Maschinenfabrik GmbH und der Firma Kapp GmbH. Die beiden Anlagen zeichnen sich durch einen umfangreichen Materialfluss aus, der für eine VIBN normalerweise nur sehr zeitintensiv modellierbar ist. Des Weiteren stand zu Versuchsbeginn ein von der Firma Heller bereits eingesetztes VIBN-Modell zur Verfügung, das einen direkten Vergleich zwischen einem konventionellen und dem hier entwickelten Vorgehen erlaubt.

### 7.3.2 Hauptzeitparalleles Rüsten einer Werkzeugmaschine

Das Anwendungsbeispiel der Firma Heller ist Bestandteil der Automation einer Werkzeugmaschine, der als *hauptzeitparalleles Rüsten* bezeichnet wird, siehe Abbildung 7.4. Dessen Aufgabe besteht darin, das Rüsten der Werkzeuge parallel zum eigentlichen Bearbeitungsprozess zu ermöglichen. Hierzu wird ein mäanderförmig verlaufendes Kettenmagazin verwendet, das zum Speichern der Werkzeuge dient. Um einen neuen Auftrag zu rüsten, wählt der Bediener das Werkzeug über die Platznummer an und lässt es zum Ausgabeschacht befördern. Dort kann dieser das alte Werkzeug entnehmen und ein Neues einsetzen. Eine automatische Schutztür trennt den Gefährdungsbereich ab. Über eine Verschiebeeinrichtung wird das Werkzeug zum Bediener verfahren, der dieses dann abgreifen bzw. ein anderes einlegen kann.

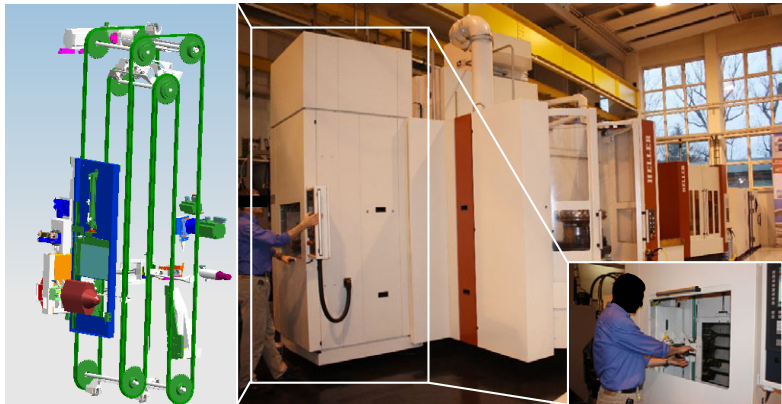


Abbildung 7.4: Hauptzeitparalleles Rüsten einer Werkzeugmaschine der Firma Gebr. Heller Maschinenfabrik GmbH

In Abbildung 7.5 sind die unterschiedlichen Detaillierungsstufen der Modelle von der Grobplanungsphase bis hin zum generierten VIBN-Modell dargestellt. Wie deutlich zu erkennen ist, wird z. B. in der Grobplanungsphase das gesamte Kettenmagazin als abstrakter Werkzeugspeicher abgebildet, da in dieser Phase lediglich die Funktionen von Interesse sind. In der Feinplanungsphase sind dagegen erste technische Details hinterlegt, was sich daran zeigt, dass der ungefähre Verlauf des Kettenmagazins modelliert ist. Das mit CAD-Daten erweiterte Modell bzw. das VIBN-Modell umfasst als letzte Stufe die genauen Ausprägungen.

Grobplanung	Feinplanung	Detailliertes VIBN-Modell

Abbildung 7.5: Grobplanungs-, Feinplanungs- und detailliertes VIBN-Modell des hauptzeitparallelen Rüstens



Abbildung 7.6 zeigt den Versuchsaufbau bei der Firma Heller. Der Simulator wurde über eine SIMBApro-PCI-Karte an die reale Steuerung gekoppelt.

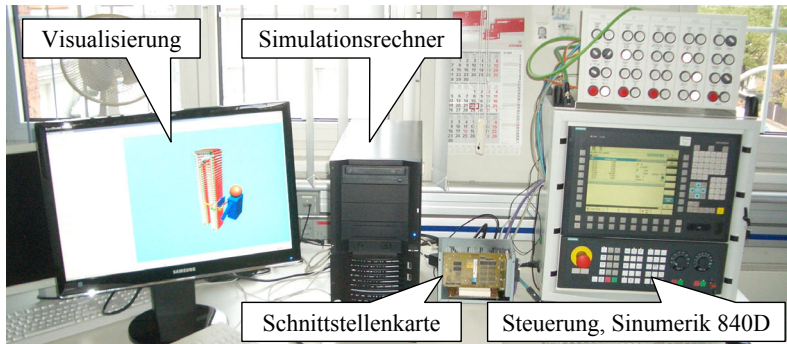


Abbildung 7.6: Versuchsaufbau bei der Firma Heller – HiL-Teststand

Hinsichtlich der Modellierung hat sich vor allem der komplexe Materialfluss der Anlage als Herausforderung dargestellt. Das Magazin setzt sich aus den an der Kette befestigten Köchern, den darin installierten Werkzeughaltern und den Werkzeugen zusammen. Hierdurch ergibt sich ein dreistufiger Materialfluss. Die Werkzeughalter werden samt den Werkzeugen durch die Verschiebeeinrichtung aus den an der Kette befestigten Köchern gefahren. Der Bediener kann wiederum die Werkzeuge aus den Haltern entnehmen. Mit Hilfe der in Kapitel 5 vorgestellten Modellierungstechnik sind alle genannten Aspekte spezifizierbar. Wie Abbildung 7.7 illustriert, kann das Verfahrprofil der Kette durch einen Mover ausgedrückt werden.

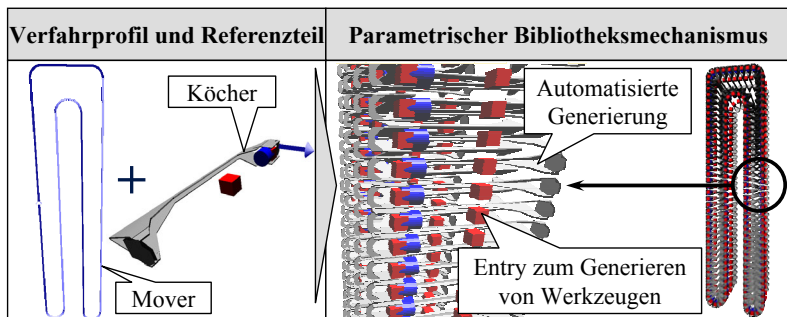


Abbildung 7.7: Modellierung komplexer Verfahrprofile und Materialflüsse

Da das Kettenmagazin aus insgesamt 160 Plätzen besteht, wäre ein manuelles Anlegen aller Bestandteile sehr aufwendig. Daher wurde ein parametrischer Bibliotheksmechanismus implementiert, der lediglich die Definition der Verfahrenskurve und eines einzelnen Referenzteils verlangt. Dieses wird dann entsprechend der geforderten Anzahl und unter Einhaltung weiterer Bedingungen, wie z. B. Abstand oder Ausrichtung entlang der Kette, reproduziert, siehe Abbildung 7.7 rechts.

Wie im vorangegangenen Abschnitt erläutert, konnte ein direkter Vergleich mit einem bereits bestehenden Simulationsmodell der Firma Heller vorgenommen werden. In Abbildung 7.8 sind diese einander gegenübergestellt. Das Modell der Firma Heller wurde mit dem *Sinumerik Machine Simulator* aufgebaut. Wie deutlich erkennbar ist, wird die tatsächliche Geometrie nicht berücksichtigt. Die Visualisierung ist lediglich zweidimensional realisiert. Im hier präsentierten Ansatz ist die räumliche Gestalt dagegen Teil der Funktion. Um den Modellierungsaufwand zu verringern, wurden im Modell der Firma Heller nicht alle Werkzeuge abgebildet. Des Weiteren ist für den Materialfluss der Kette ein logisches Ersatzmodell notwendig, das auf dem Prinzip eines Schieberegisters basiert und nur diskrete Stellen einnimmt.

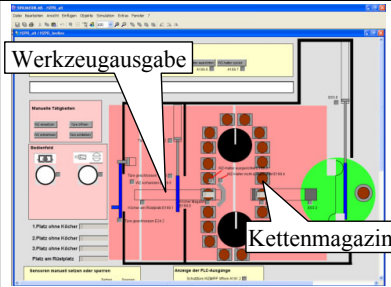
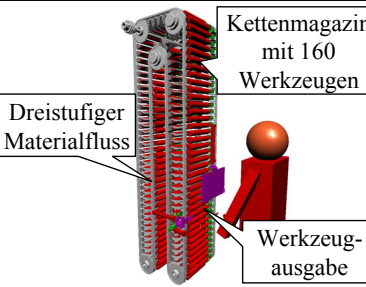
Klassischer Ansatz	Methode mit Funktionsmodell
	
<ul style="list-style-type: none"> <li>• Vereinfachtes 2D-Modell</li> <li>• Keine Kollisionserkennung</li> <li>• Logisches Ersatzmodell für die Abbildung des Materialflusses</li> <li>• Vereinfachter Materialfluss (nicht alle Werkzeuge berücksichtigt)</li> <li>• Keine Abbildung von Störszenarien</li> <li>• Hoher Modellierungsaufwand</li> </ul>	<ul style="list-style-type: none"> <li>• 3D-Modell mit realer Geometrie</li> <li>• Kollisionen berücksichtigt</li> <li>• Materialfluss ergibt sich aus der Struktur des Modells</li> <li>• Detaillierte Abbildung des Materialflusses (Köcher, Werkzeughalter, Werkzeuge)</li> <li>• Reduzierter Modellierungsaufwand durch automatisierte Nutzung von CAD-Daten</li> </ul>

Abbildung 7.8: Vergleich eines konventionellen Simulationsansatzes mit der hier vorgestellten Methode

Trotz des wesentlich komplexeren VIBN-Modells konnte der Modellierungsaufwand durch die hier entwickelte Methode deutlich reduziert werden. Während der Arbeitsumfang für den Modellaufbau von Seiten der Firma Heller mit drei Personenwochen angegeben wurde, beläuft sich dieser bei der Erstellung des Funktionsmodells auf drei Personentage. Dabei sind jedoch die Tätigkeiten zur Anpassung der CAD-Modelle nicht berücksichtigt, da diese bei Einhaltung der Konstruktionsregeln im Entwicklungsprozess entfallen.

### 7.3.3 Be- und Entladesystem einer Schleifmaschine

Bei der Fallstudie der Firma Kapp handelt es sich um ein Be- und Entladesystem einer Schleifmaschine, siehe Abbildung 7.9. Zwei rundlaufförmige Transportbänder befördern Zahnräder, die auf einer Palette installiert sind, zur Bearbeitungsstation. Dort werden diese über eine pneumatische Hubeinrichtung in die Werkzeugmaschine weitergeleitet. Nach dem Schleifen werden die Bauteile auf das Band zurückgelegt und zu einer Ölschleuder verfahren, die die Zahnräder reinigt. Auf den Förderbändern sind mehrere pneumatische Rückhaltesysteme angebracht, die für einen reibungslosen Rundlauf sorgen. Die Anlage erfordert eine Interaktion mit dem Bediener, der Rohteile auf die Paletten legt und die bearbeiteten Zahnräder abgreift.

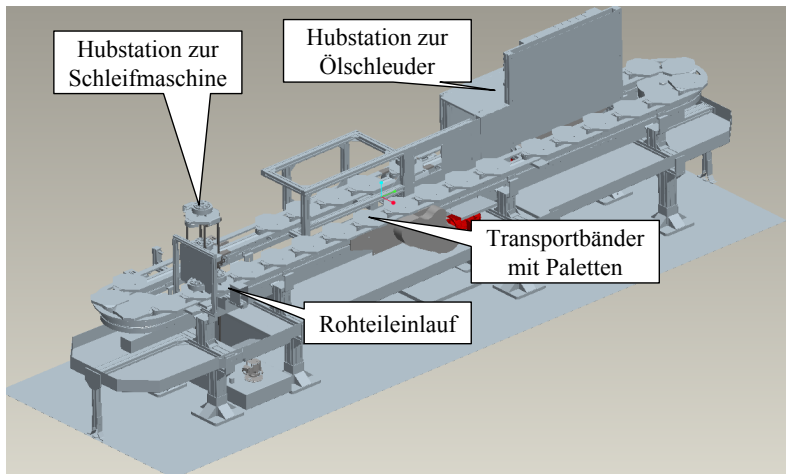


Abbildung 7.9: Be- und Entladesystem einer Schleifmaschine für Zahnräder

In Abbildung 7.10 sind die unterschiedlichen Detaillierungsstufen der Modelle des Anwendungsbeispiels verdeutlicht. In der Grobplanungsphase ist das Gesamtsystem stark abstrahiert dargestellt, was sich sehr deutlich am vereinfachten Materialfluss manifestiert. Das Modell der Feinplanungsphase weist dagegen die grundlegenden Wirkstrukturen auf. Neben der detaillierteren Abbildung der Aktoren und der Sensoren ist auch der Materialtransport durch die Förderbänder samt Führungsschienen genauer spezifiziert.

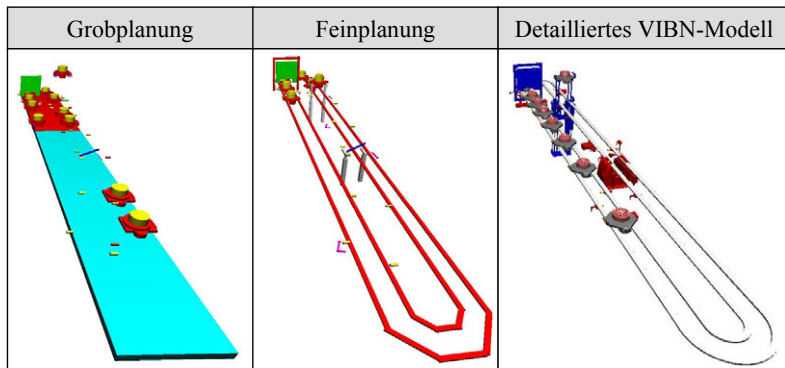
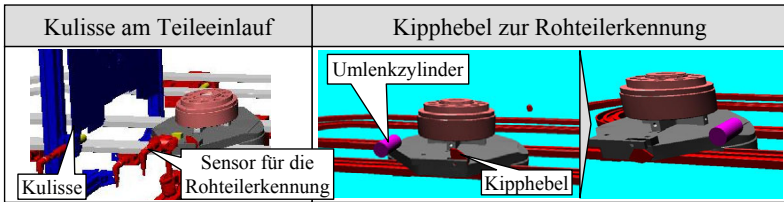


Abbildung 7.10: Grobplanungs-, Feinplanungs- und detailliertes VIBN-Modell des Be- und Entladesystems einer Schleifmaschine

Wie sich unter Nutzung des Geometriemodells Modellierungsarbeit einsparen lässt, zeigt Abbildung 7.11. Mit Hilfe einer Kulisse werden am Einlauf in die Anlage falsche Teile identifiziert. Ein logisches Ersatzmodell ist hierfür nicht erforderlich, da dies über die Kollisionserkennung abgedeckt wird. Des Weiteren verfügen die Paletten des Anwendungsbeispiels über einen mechanischen Kipphebel zur Rohteilerkennung, der über die Kollision mit dem Detektor eines Näherungssensors abgefragt wird. Der Kipphebel wird vom Bediener beim Einbringen von Rohteilen aufgestellt. Die hier präsentierte Modellierungstechnik ermöglicht die Abbildung derartiger Szenarien, indem auch das Stückgut als Komponente beschrieben wird. Dadurch steht der volle Umfang der Spezifikationsmethoden zur Verfügung. Somit kann dem Stückgut ein Verhalten zugewiesen werden, wodurch das Ein- und Ausklappen des Kipphebels modellierbar ist. Die Kommunikation zum Materialobjekt erfolgt mit Hilfe eines PortBindings (siehe Abschnitt 5.3.5), so dass der Vorgang gesteuert werden kann.



*Abbildung 7.11: Kulisse zur Falschteilerkennung und mechanischer Kipphebel zur Rohteilidentifikation*

Die Herausforderungen bei der Modellierung des vorgestellten Be- und Entlade-systems lagen, wie bereits geschildert, hauptsächlich im Bereich des Material-flusses. Mit Hilfe der hier entwickelten Modellierungstechnik konnten im Ver-gleich zu konventionellen VIBN-Ansätzen deutliche Genauigkeitssteigerungen erzielt werden. Dabei ist vor allen Dingen die Modellierung der Materialobjekte zu nennen, die über ein Eigenverhalten verfügen können (siehe Beispiel des me-chanischen Kipphebels). Weiterhin hat sich die automatisierte Erweiterung mit CAD-Daten als Vorteil erwiesen. Im beschriebenen Anwendungsbeispiel trifft dies besonders auf die Platzierung und Ausrichtung der Sensoren zur Materialer-fassung (modelliert durch Detektoren) sowie auf die Definition der Verfahrach-sen zu. Dadurch konnte zum einen der Modellierungsaufwand reduziert werden, da die Anlage über eine Vielzahl derartiger Sensoren verfügt. Zum anderen war eine verbesserte Kollisionsbetrachtung durch die Berücksichtigung der exakten Verfahrswege möglich.

### 7.4 Wirtschaftlichkeitsbetrachtung

#### 7.4.1 Allgemeines

Der wirtschaftliche Erfolg eines Entwicklungsprozesses *misst sich [...] an den Parametern Funktionserfüllung, Kosten- und Termineinhaltung*, die auch als „magisches Dreieck“ aus *Qualität, Zeit und Kosten* bezeichnet werden (EHRENSPIEL ET AL. 1998). Demnach ist der Einsatz einer neuen Methode oder eines neuen Softwareproduktes im Entwicklungsprozess nur dann sinnvoll, wenn sich über die genannten Größen Verbesserungen erzielen lassen. Generell kann der Nutzen einer IT-Unterstützung nur bedingt abgeschätzt werden (SCHNITTLER 2004). Eine monetäre Bewertung ist daher sehr schwer und gegebenenfalls nur im konkreten Anwendungsfall darstellbar. Ihr Ergebnis ist in jedem Fall maßgeblich vom Unternehmen und den hergestellten Maschinen abhängig. Hierbei spielt beispielsweise der Standardisierungsgrad sowohl im Konstruktionsprozess als auch hinsichtlich der Modellbildung eine wesentliche Rolle. Weiterhin haben die bereits existierenden Erfahrungen im Bereich der Simulation Einfluss auf die anfallenden Personalkosten, da durch den Erkenntnisgewinn vorangegangener Projekte die Modellbildung schneller vorstattengehen kann. Aus den genannten Gründen sollen die Auswirkungen der vorgestellten Methode qualitativ erörtert werden. In den folgenden Abschnitten wird dazu die Wirtschaftlichkeit des entwickelten Ansatzes detailliert betrachtet.

#### 7.4.2 Bewertung des wirtschaftlichen Aufwandes

Die Kosten für die Anwendung der beschriebenen Methode setzen sich aus drei Bestandteilen zusammen:

- Kosten für die Hardwareausstattung

Zur Modellbildung und zur Simulation wird ein Rechner mit entsprechendem Zubehör, wie z. B. Tastatur und Bildschirm, benötigt. Des Weiteren ist eine Schnittstellenkarte für die HiL-Simulation erforderlich, über welche die Steuerung an den Simulations-PC gekoppelt werden kann. In der vorliegenden Arbeit wurde eine SIMBApro-PCI-Simulationskarte für den PROFIBUS der Firma Siemens verwendet. Hierfür fallen beispielsweise Kosten in der Größenordnung von ca. 1500

(einfachste, einkanalige Version) bis ca. 4300 Euro (zweikanaliges Spitzenmodell) an. Je nach Anzahl der zu simulierenden Busstränge müssen eventuell mehrere Simulationskarten vorgehalten werden. Darüber hinaus sind die Kosten für die Steuerung zu berücksichtigen, die entweder für die Simulation eigens abgestellt oder zumindest früher beschafft werden muss.

- Kosten für die Software

Da es sich bei den hier entwickelten Softwarewerkzeugen für die Funktionsmodellierung und die VIBN-Simulation um wissenschaftliche Prototypen handelt, sind die dafür anfallenden Kosten in Form von Lizenz- und Wartungsgebühren nur bedingt abschätzbar. Für das VIBN-Simulationswerkzeug können marktübliche Preise vergleichbarer Systeme angenommen werden. Hinsichtlich der SiL-Simulation sind gegebenenfalls die Lizenzkosten für das virtuelle Steuerungssystem einzubeziehen.

- Kosten für das Personal

Ein Großteil der anfallenden Kosten wird durch den Personaleinsatz verursacht. Hierbei sind zunächst die Aufwendungen für die Einführung und die Schulung zu nennen. Weiterhin entstehen Ausgaben für die Erstellung der unterschiedlichen Modelle (Grobplanungs- und Feinplanungsphase). Ferner sind die Pflege der Modellbibliothek sowie der Aufbau des Versuchstandes und die Durchführung der VIBN mit einzukalkulieren. Die Höhe der Personalkosten variiert unternehmensspezifisch und hängt insgesamt von der Anzahl der bearbeiteten Projekte sowie von der Komplexität der zu modellierenden Maschine oder Anlage ab. Sehr große Anlagen können z. B. Inbetriebnahmezeiten im Bereich von Personenjahren aufweisen (STETTER 2006). Infolgedessen gestaltet sich die Modellbildung entsprechend zeitintensiver.

### **7.4.3 Bewertung des wirtschaftlichen Nutzens**

Die im Rahmen dieser Arbeit entwickelte Methode widmet sich zwei Handlungsfeldern im Entwicklungsprozess. Dies ist zum einen die zeitaufwendige und bisher meist zu spät eingesetzte VIBN, zum anderen die Unterstützung der Planungs- und der Konstruktionstätigkeiten durch ein übergreifendes Informationsmodell. Daher soll der wirtschaftliche Mehrwert der Methode nachfolgend im

Hinblick auf die VIBN und auf den Gesamtentwicklungsprozess diskutiert werden.

### **Nutzen der entwickelten Methode im Hinblick auf die VIBN**

Eine Studie des VDW (VDW-BERICHT 1997), die sich der Ermittlung von Zeitanteilen einzelner Arbeitsschritte während der Inbetriebnahme widmete, veranschlagt für die Beseitigung der Softwarefehler zwischen ca. 9 bis 16 Prozent der gesamten Projektdauer. Dies lässt großes Potenzial für die VIBN erwarten. Sowohl in wissenschaftlichen Untersuchungen (ZÄH ET AL. 2006) als auch in der praktischen Anwendung (RÖMBERG 2004, HEINRICH & WORTMANN 2004, STETTER 2006) konnte deren Wirksamkeit nachgewiesen werden. LUTZ (2006) quantifiziert beispielsweise den durch die Simulation erzielten Einspareffekt bei der Einarbeitung eines neuen Werkzeugsatzes für eine Presse auf 70.000 USD, wodurch sich die Ausgaben für die VIBN schnell amortisieren. WÜNSCH (2008) hat sich im Rahmen seiner Forschungstätigkeiten explizit mit der Wirtschaftlichkeit der VIBN beschäftigt und diese prinzipiell belegt. Desgleichen bescheinigt SCHAICH (1996) der VIBN durch die Kosteneinsparung beim Softwaretest eine grundsätzliche Wirtschaftlichkeit, schränkt diese allerdings in Abhängigkeit von den Hardware-, den Software- und den Personalkosten ein.

Auf den Nutzen einer konventionellen VIBN wird an der Stelle nicht mehr eingegangen, da dieser bereits ausführlich in Abschnitt 2.6.6 thematisiert wurde. Im Folgenden sind die wirtschaftlichen und die technologischen Vorteile der hier beschriebenen Methode den klassischen VIBN-Ansätzen gegenübergestellt:

- Beschleunigter Modellaufbau

Wie die Anwendungsbeispiele aus Abschnitt 7.3 verdeutlichen, kann die Modellerstellung zügiger durchgeführt werden. Dies ist zum einen der intuitiveren Modellierungstechnik und zum anderen dem höheren Abstraktionsgrad geschuldet. So können z. B. der Materialfluss oder die Geometrie sehr einfach mit Hilfe der entwickelten Beschreibungsformen abgebildet werden. Gerade die Personalkosten für den Modellaufbau tragen einen erheblichen Teil zu den Gesamtkosten der VIBN bei, zumal dafür hochqualifiziertes Personal mit entsprechenden Tagessätzen zum Einsatz kommt (SPITZWEG 2009). Deswegen wirken sich Reduzierungen in diesem Bereich besonders stark auf die Wirtschaftlichkeit aus.



- Berücksichtigung der Steuerungsfunktionalität

Der vorgestellte Ansatz konzentriert sich nicht ausschließlich auf das Maschinenmodell, sondern bezieht die Steuerungsfunktionalität von Beginn an mit ein. Daher können Maschinenfunktionen bereits frühzeitig, noch vor der eigentlichen VIBN, getestet und optimiert werden. VOGEL-HEUSER & FRIEDRICH (2006) konnten anhand empirischer Versuchsreihen den positiven Einfluss der Modellierung auf die Steuerungssoftwareprogrammierung belegen.

- Verbesserte Modellgüte

Durch die teilautomatisierte Einbindung der CAD-Daten kann die Modellgüte verbessert werden. Das umfasst beispielsweise die Geometrie oder die Achskinematiken. Ebenso kann der Materialfluss detailgetreu abgebildet werden. In dem Zusammenhang ist die Modellierung von Stückgütern zu nennen. Diese erlaubt es, einzelne Materialobjekte mit umfangreichen Eigenschaften zu versehen, welche bisher nur bedingt oder gar nicht modelltechnisch darstellbar waren (siehe Beispiel des Be- und Entladesystems aus Abschnitt 7.3.3). Mit der höheren Modellqualität gehen wirtschaftliche Vorteile einher, da die Steuerungssoftware aussagekräftiger getestet werden kann. Dies wird ferner durch die integrierte Fehlermodellierung unterstützt.

- Frühzeitiger Einsatz der VIBN im Entwicklungsprozess

Die Simulation kann mit Hilfe des erläuterten Ansatzes wesentlich früher in den Entwicklungsprozess eingebunden werden als bei den bisherigen konventionellen Ansätzen. Deshalb steht mehr Zeit für die Prüfung und Optimierung des Steuerungscode zur Verfügung. Auf diese Art und Weise lassen sich die durch Fehler induzierten Folgekosten während der Inbetriebnahme reduzieren.

#### **Nutzen der entwickelten Methode im Hinblick auf den Gesamtentwicklungsprozess**

Bisher wurde in der Nutzenanalyse dieses Teilkapitels lediglich die VIBN an sich betrachtet. Es ist jedoch der Mehrwert der Methode nicht nur darauf beschränkt, sondern er erstreckt sich auf den gesamten Entwicklungsprozess. Nachfolgend wird der identifizierte wirtschaftliche Nutzen zusammengefasst:

- Parallelisierung von Prozessen

Das Funktionsmodell kann als Abstimmungsmittel zwischen den unterschiedlichen Fachabteilungen dienen. Hierdurch können Arbeitsschritte stärker integriert werden, wodurch sich die Gesamtentwicklungszeit verkürzt. VOGEL-HEUSER & FISCHER (2002) beziffern das Einsparpotenzial, das durch eine systematische Anforderungsanalyse und eine darauf basierende Systemspezifikation realisiert werden kann, auf 10 bis 20 Prozent der Durchlaufzeit. Die Verkürzung der Inbetriebnahmezeit wird auf 20 bis 40 Prozent geschätzt. HANSEN & RIEGLER (2002) geben eine Zeitersparnis beim Einsatz von Methoden der Digitalen Fabrik *von bis zu 30 Prozent, eine projektbezogene Kostensenkung von 15 Prozent und eine Erhöhung des Produkt- und Anlagenreifegrads von 5 bis 10 Prozent* an.

- Simulative Unterstützung in der Konzeptfindungsphase

Durch den Einsatz der Simulation bereits zu Beginn des Entwicklungsprozesses können Konzepte noch vor der technischen Umsetzung bewertet werden. KIEFER (2007) sieht in *der frühzeitigen Produktbeeinflussung (eine Möglichkeit), die Produktqualität bei beschleunigten Entwicklungszeiten deutlich (zu) erhöhen bzw. die Phase des Produktionsanlaufs durch optimierte und abgesicherte Planungsprozesse signifikant (zu) beschleunigen*.

- Kürzere Iterationsschleifen

Der hier präsentierte Ansatz erlaubt durch die entwicklungsbegleitende Herangehensweise eine zeitnahe Gegenüberstellung des Maschinenkonzeptes mit den Simulationsergebnissen. Dadurch können Änderungen früher und somit kostengünstiger vollzogen werden.

- Bessere und formale Dokumentation

Das Funktionsmodell kann helfen, Missverständnisse zwischen den beteiligten Disziplinen zu vermeiden. Änderungen können besser abgestimmt und im Modell hinterlegt werden. Zudem ist durch den formalen Charakter eine weitere Auswertung und Analyse möglich. Dies umfasst beispielsweise die Konsistenz des Funktionsmodells zu anderen Entwicklungsmodellen, wie etwa den CAD-Daten.

- Verbesserter Informationsaustausch mit dem Kunden

Informationen lassen sich mit Hilfe des Funktionsmodells eindeutig und exakt formulieren. Hierdurch kann mit dem Kunden bereits in der Angebotsphase eine Klärung der Randbedingungen stattfinden. Dies trägt wiederum dazu bei, den Entwicklungsprozess zielgerichtet auszuführen und somit Folgekosten durch falsch interpretierte Kundenbedürfnisse zu reduzieren.

- Ableitung weiterer Entwicklungsartefakte

Das Funktionsmodell kann zukünftig zur Ableitung weiterer Entwicklungsartefakte dienen, wie z. B. für Teile des Steuerungscode oder für Konstruktionsmodelle. Durch die Automatismen können Arbeitskosten eingespart sowie gleichzeitig die Fehleranzahl reduziert werden.

## **7.5 Zusammenfassung**

In Kapitel 7 wurden die im Rahmen dieser Arbeit entwickelten Softwareprototypen für die Funktionsmodellierung und für die VIBN vorgestellt. Mit Hilfe dieser Werkzeuge wurden ferner zwei industrielle Anwendungsbeispiele umgesetzt und die hier entworfene Methode bewertet. Dabei hat sich gezeigt, dass die Modellierung schneller vonstattengehen kann und sich zudem durch die teilautomatisierte CAD-Datenintegration ein höherer Detaillierungsgrad erreichen lässt. Darüber hinaus konnten durch die neue Modellierungstechnik Einzelheiten abgebildet werden, die bisher bei konventionellen VIBN-Ansätzen nur unzureichend unterstützt wurden, wie z. B. das Eigenverhalten von Materialobjekten (siehe Beispiel der Transportpaletten mit Kipphebel der Firma Kapp) oder komplexe Verfahrenskurven (siehe Beispiel des mäanderförmigen Kettenmagazins der Firma Heller).

Des Weiteren wurde die Wirtschaftlichkeit der Methode bewertet. Der zu leistende Aufwand setzt sich aus den Kosten für die Software, für die Hardwareausstattung und für das Personal zusammen. Demgegenüber steht der erzielbare Nutzen, der sich nicht nur auf die VIBN beschränkt, sondern auch auf den Gesamtentwicklungsprozess auswirkt. In Anlehnung an den in Kapitel 1 beschriebenen Arbeitsablauf (siehe Abbildung 1.4) verdeutlicht Abbildung 7.12 den Nutzen der Methode im Vergleich mit dem herkömmlichen Vorgehen bei einer VIBN. Durch die Verwendung des Funktionsmodells kann der Entwicklungspro-

zess wesentlich früher unterstützt werden. Zusätzlich gestaltet sich die VIBN auf Grund der automatisierten Modellgenerierung wesentlich wirtschaftlicher.

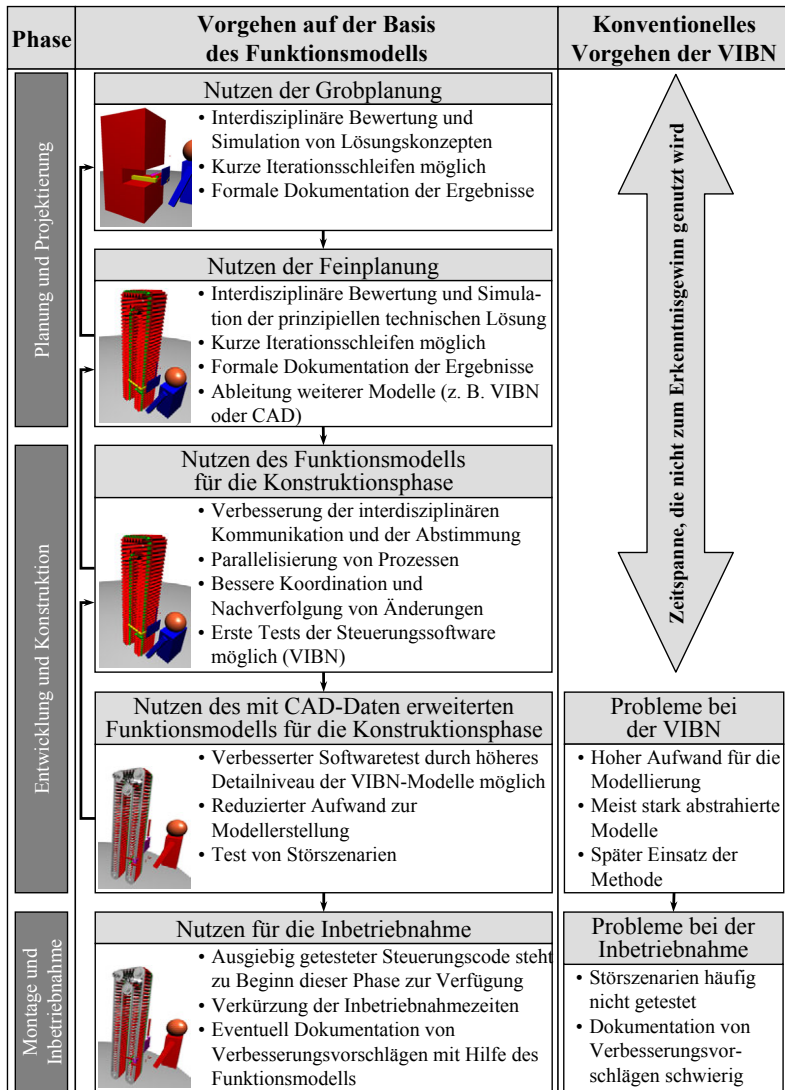


Abbildung 7.12: Nutzen der entwickelten Methode im Vergleich mit einem konventionellen VIBN-Ansatz

## 8 Zusammenfassung und Ausblick

### 8.1 Zusammenfassung

Die zunehmende Integration der unterschiedlichen Entwicklungsdisziplinen Mechanik, Elektrotechnik und Software im Maschinen- und Anlagenbau sorgt dafür, dass die Funktionalität moderner Produktionssysteme nur noch im Zusammenwirken der einzelnen Teilbereiche überprüft werden kann. Vor allem die Software spielt dabei eine bedeutende Rolle, da deren Anteil stetig wächst. Die Flexibilität und die günstigeren Kosten sind entscheidende Gründe dafür, dass verstärkt softwarebasierte Lösungen angestrebt werden. Anpassungen sind hierin wesentlich einfacher vorzunehmen als Modifikationen an der Maschinenstruktur, die in späten Phasen hohe Folgekosten nach sich ziehen.

Da für einen Test der komplexen Steuerungsprogramme normalerweise die reale Maschine notwendig ist, wurden Simulationsansätze entwickelt, die diese durch einen virtuellen Prototyp ersetzen. Die als *Virtuelle Inbetriebnahme (VIBN)* bezeichnete Methode erfordert entsprechende Anlagenmodelle, die bisher nur sehr aufwendig und meist manuell erstellt wurden. Erschwerend kommt hinzu, dass die Modellierung in der Regel erst im Anschluss an die mechanische und die elektrische Konstruktion angegangen wird, da deren Daten als Eingangsinformationen dienen. Aus den genannten Gründen hat sich die VIBN, trotz vieler Vorteile, erst partiell im industriellen Umfeld durchsetzen können.

Die im Rahmen dieser Arbeit entwickelte Methode greift die erläuterten Problemstellungen auf. Im Gegensatz zu konventionellen VIBN-Verfahren wird hier eine entwicklungsbegleitende Vorgehensweise angestrebt. Die Basis der Methode bildet ein sogenanntes Funktionsmodell, das zu Beginn des Entwicklungsprozesses erstellt wird. Dieses enthält disziplinenübergreifende Informationen, die für alle beteiligten Fachbereiche von Bedeutung sind. Dadurch können unterschiedliche Anforderungen der jeweiligen Entwicklungsabteilungen berücksichtigt und optimal aufeinander abgestimmt werden. Weiterhin wird die Suche nach Funktions- und Lösungsprinzipien durch die Simulation unterstützt, die eine zeitnahe Bewertung der Ergebnisse zulässt. Für diese Zwecke ist es erforderlich, sowohl das Steuerungs- als auch das Maschinenverhalten abzubilden. Auf Grund des höheren Abstraktionsgrades lässt sich das Funktionsmodell allerdings wesentlich schneller und einfacher aufbauen als ein VIBN-Modell. Letzteres wird

im Rahmen der hier vorgestellten Methode automatisch aus dem Funktionsmodell generiert. Bereits nach der Planungs- und Projektierungsphase können erste grobe VIBN-Modelle ausgeleitet werden. Mit Hilfe der CAD-Daten wird das Funktionsmodell im Laufe des Entwicklungsprozesses teilautomatisiert um technische Details erweitert. Auf diese Art und Weise können zu späteren Zeitpunkten genauere VIBN-Modelle erstellt werden. Die beschriebene Methode soll eine bessere Synchronisation der einzelnen Entwicklungsschritte ermöglichen. Hierzu kann das Funktionsmodell einen Beitrag leisten, da es auch während der Konstruktionsphase als Kommunikationsmittel und Entscheidungshilfe von Nutzen sein kann. Durch eine stärkere Parallelisierung der disziplinspezifischen Phasen kann die Softwareentwicklung früher mit einbezogen werden. Dies wiederum wirkt sich positiv auf die Qualität der Steuerungssoftware aus und führt zu einer Beschleunigung des Inbetriebnahmeprozesses.

Für die Funktionsmodellierung wurde eine neue Beschreibungstechnik vorgestellt, die schon in frühen Phasen der Maschinenentwicklung die Berücksichtigung zentraler Aspekte, wie z. B. des Materialflusses oder der Geometrie, erlaubt. Steuerungslogik und Maschinenverhalten können mit Hilfe einer gemeinsamen Spezifikationsmethode abgebildet werden. Demzufolge reduziert sich der Einarbeitungsaufwand für den Modellierer, da nicht, wie meist üblich, eine Vielzahl an Modellierungskonstrukten geistig durchdrungen werden muss. Weiterhin wurde ein zugehöriges Vorgehensmodell definiert, das den Einsatz der Modellierungsmethode regelt.

Die Integration der CAD-Daten in das Funktionsmodell erfolgt über ein hier definiertes Austauschformat, welches auf der XML (Extensible Markup Language) beruht. Dadurch ist eine Unabhängigkeit vom eingesetzten CAD-Werkzeug gegeben, sofern die benötigten Informationen ausgeleitet werden können. Die einzubindenden Daten werden vorab auf deren Konsistenz geprüft. Das VIBN-Modell wird vollständig aus dem Funktionsmodell in Form eines C++-Programmes abgeleitet. In diesen Prozess werden zusätzliche Informationen eingebunden, um die Schnittstellen des Modells mit den realen Steuerungsein- und -ausgängen koppeln zu können.

Zur Validierung der Methode wurden prototypische Softwarewerkzeuge für die Funktionsmodellierung und den VIBN-Simulator entwickelt. Anhand von zwei industriellen Fallstudien konnte die Wirksamkeit des Vorgehens verifiziert werden. Die generierten Modelle wiesen nicht nur einen wesentlich höheren Detaillierungsgrad auf, sondern konnten deutlich schneller erstellt werden, als dies bei

konventionellen VIBN-Ansätzen möglich ist. In einem direkten Vergleich der Modellierungsaufwände zwischen einem von der Firma Gebr. Heller Maschinenfabrik GmbH erstellten VIBN-Modell und dem Funktionsmodell konnten im Falle der hier entwickelten Methode deutliche Einspareffekte ausgemacht werden. Dies ist durch die neue Modellierungstechnik begründet, mit Hilfe derer vor allem die Materialflussvorgänge deutlich schneller spezifiziert werden können. Darüber hinaus trägt auch die teilautomatisierte Integration von CAD-Daten zur Zeiteinsparung bei (z. B. bei der Erstellung des Geometriemodells oder der Detaillierung des Maschinenverhaltens). Es hat sich ferner gezeigt, dass durch die Modellierungstechnik Eigenschaften (z. B. Materialobjekte mit Eigenverhalten) abgebildet werden können, die bisher nur unzureichend darstellbar waren. Abgesehen davon unterstützen die unterschiedlichen Abstraktionsstufen des Funktionsmodells (Grobplanung, Feinplanung, mit CAD-Daten erweitertes Funktionsmodell) eine entwicklungsbegleitende Vorgehensweise, die bei klassischen VIBN-Ansätzen nicht üblich ist. Wie der Stand der Forschung und der Technik dieser Arbeit verdeutlicht, kann gerade für die frühen Entwicklungsphasen mittels der hier dargelegten Modellierungstechnik ein Mehrwert erzielt werden, da die existierenden Beschreibungsformen wichtige Belange für die Konstruktion von Maschinen und Anlagen (z. B. die Verfahrkurven, die Grobgeometrie oder der Materialfluss) nur teilweise abbilden. Abschließend wurden anhand einer Wirtschaftlichkeitsbetrachtung die Auswirkungen der Methode auf den Entwicklungsprozess bewertet.

## **8.2 Ausblick**

Modelle können den Entwicklungsprozess in vielfältiger Weise unterstützen. Bisher konzentriert sich deren Verwendung sehr stark auf die jeweiligen Entwicklungsdisziplinen. Dort werden vereinzelt schon modellgetriebene Ansätze verwendet, wie z. B. die CAD/CAM-Kopplung, bei welcher auf der Basis des CAD-Modells NC-Code generiert wird. Probleme entstehen hauptsächlich an den Schnittstellen zwischen den Fachabteilungen. Daher kann ein übergreifendes Funktionsmodell von großem Nutzen sein. Neben einer frühzeitigen und verbesserten Abstimmung lassen sich weitere Entwicklungsartefakte zumindest teilweise erzeugen.

In der vorliegenden Arbeit wurde beispielsweise gezeigt, dass mit Hilfe der abstrakten, übergeordneten Modellinstanz detaillierte VIBN-Modelle generiert wer-

den können. Eine Ableitung anderer Modelle auf der Konstruktionsebene ist prinzipiell möglich. Die Geometrie und der hierarchische Aufbau des Funktionsmodells können für die Mechanik genutzt werden. Die projektierten Aktoren und Sensoren lassen sich zur Erstellung von Rumpfstromlaufplänen automatisiert heranziehen. Auch die Verwendung der Steuerungslogik im Funktionsmodell zur Ausleitung von Steuerungscode für eine spezifische Zielplattform ist denkbar. Hierzu könnten beispielsweise generische Fehlerbehandlungsroutinen in den Prozess mit aufgenommen werden.

Ein zusätzlicher Anwendungsbereich besteht im Hinblick auf die Kundenanforderungen. Werden diese ebenso durch eine formale, modellbasierte Notation festgehalten, kann ein Abgleich mit der angestrebten Lösung im Funktionsmodell stattfinden. Die formale Definition der Anforderungen erlaubt zudem eine präzisere und kompaktere Beschreibung. Abgesehen von einer Konsistenzprüfung der Anforderungen können Bezüge zu anderen Entwicklungsartefakten hergestellt werden, die mit einer textuellen Darstellung nicht möglich wären. Daher sind existierende Beschreibungstechniken zur Festlegung von Anforderungen zu untersuchen und für Systeme des Maschinen- und Anlagenbaus zu adaptieren bzw. gegebenenfalls zu erweitern.



## 9 Literaturverzeichnis

ALBERT 1998

Albert, J.: Software-Architektur für virtuelle Maschinen. Dissertation Technische Universität München (1998). München: Utz 1998. (Informationstechnik im Maschinenwesen 26).

ALLA & DAVID 1998

Alla, H.; David, R.: Continuous and Hybrid Petri Nets. Journal of Circuits, Systems & Computers 8 (1998) 1, S. 159-188.

ALT 2009

Alt, O.: Car Multimedia Systeme Modell-basiert testen mit SysML. Dissertation Technische Universität Darmstadt (2008). Wiesbaden: Vieweg 2009. ISBN: 978-3-8348-9567-7.

AUTOVIBN 2010

AutoVIBN: Automatische Generierung von Verhaltensmodellen für die qualitätsorientierte Virtuelle Inbetriebnahme. Technical Report TUM-I1012. München: Technische Universität München 2010. (Abschlussbericht – AiF-Forschungsvorhaben ZN279).

BALZERT 2009

Balzert, H.: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Heidelberg: Spektrum Akademischer Verlag 2009. ISBN: 978-3-8274-1705-3.

BANKS 1998

Banks, J.: Handbook of simulation. New York: Wiley 1998. ISBN: 0-471-13403-1.

BATHELT 2006

Bathelt, J.: Entwicklungsmethodik für SPS-gesteuerte mechatronische Systeme. Dissertation Eidgenössische Technische Hochschule Zürich (2006). Zürich: 2006. (Diss.-Nr. 16879)

### BAUDISCH ET AL. 2007

Baudisch, T.; Zäh, M. F.; Lindworsky, A.: „Entwicklungsumgebung zum Automatisierungsengineering“ – Ein Werkzeug zur integrierten Virtuellen Inbetriebnahme von Produktionsanlagen. In: Gausemeier, J. et al. (Hrsg.): 5. Paderborner Workshop – Entwurf mechatronischer Systeme. Paderborn: Westfalia Druck 2007, S. 399-409. ISBN: 978-3-939350-29-3.

### BAUER 2009

Bauer, G.: Ölhdraulik. 9. Aufl. Wiesbaden: Vieweg 2009. ISBN: 978-3-8351-0247-7.

### BENDER ET AL. 1999

Bender, K. (Hrsg.); Albert, J.; Holzmüller, T.; Jünger, B.; Kaiser, O.; Kriesel, W.; Prinz, O.; Schaich, C.; Schullerer, J.; Tomaszun, J.: Echtzeitsimulation zum Test von Maschinensteuerungen. München: Utz 1999. ISBN: 3-89675-482-3.

### BENDER ET AL. 2002

Bender, K.; Broy, M.; Istvan, P.; Pretschner, A.: Model Based Development of Hybrid Systems: Specification, Simulation, Test Case Generation. In: Engell, S. et al. (Hrsg.): Modelling, analysis, and design of hybrid systems. Berlin: Springer 2002, S. 37-51. ISBN: 3-540-43812-2.

### BENDER 2005

Bender, K.: Embedded Systems. Berlin: Springer 2005. ISBN: 978-3-540-27370-7.

### BOEHM 1988

Boehm, B. W.: A Spiral Model of Software Development and Enhancement. IEEE Computer 21 (1988) 5, S. 61-72.

### BONFÉ ET AL. 2005

Bonfé, M.; Fantuzzi, C.; Secchi, C.: Unified modeling and verification of logic controllers for physical systems. In: Institute of Electrical and Electronic Engineers (IEEE) (Hrsg.): 44th Conference on Decision and Control, and the European Control Conference 2005 (CDC-ECC'05). Sevilla/Spanien, 12.-15. Dezember 2005. S. 8349-8354.

**BOSSEL 1994**

Bossel, H.: Modellbildung und Simulation: Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme. Wiesbaden: Vieweg 1994. ISBN: 3-8334-0984-3.

**BRECHER & KARLBERGER 2009**

Brecher, C.; Karlberger, A.: Modellbasierte Entwicklung von Werkzeugmaschinen. wt Werkstattstechnik online 99 (2009) 5, S. 300-303.

**BRONSTEIN ET AL. 2008**

Bronstein, I.; Semendjajew, K.; Musiol, G.; Muehlig, H.: Taschenbuch der Mathematik. Frankfurt am Main: Harri Deutsch 2008.

**BROY ET AL. 1999**

Broy, M.; Huber, F.; Schätz, B.: AutoFOCUS – Ein Werkzeugprototyp zur Entwicklung eingebetteter Systeme. Informatik Forschung und Entwicklung 14 (1999) 3, S. 121-134.

**BROY & STØLEN 2001**

Broy, M.; Stølen, K.: Specification and Development of Interactive Systems: Focus on Streams, Interfaces and Refinement. New York: Springer 2001. ISBN: 0-387-95073-7.

**BUCHER 1998**

Bucher, T. G.: Einführung in die angewandte Logik. 2. Aufl. Berlin: de Gruyter 1998. ISBN: 3-11-015279-7.

**BURMESTER ET AL. 2007**

Burmester, S.; Giese, H.; Henkler, S.; Hirsch, M.; Tichy, M.; Boiko, V.; Gambuzza, A.; Münch, E.; Vöcking, H.: Tool Support for Developing Advanced Mechatronic Systems: Integrating the Fujaba Real-Time Tool Suite with CAMEL-View. In: Institute of Electrical and Electronics Engineer (IEEE) (Hrsg.): 29th International Conference on Software Engineering 2007 (ICSE). Minneapolis/USA, 19.-27. Mai 2007. Los Alamitos: IEEE Computer Society Press 2007, S. 801-804.

**BURMESTER ET AL. 2005**

Burmester, S.; Giese, H.; Tichy, M.: Model-Driven Development of Reconfigurable Mechatronic Systems with Mechatronic UML. In: Aßmann, U. et al. (Hrsg.): Model Driven Architecture. Berlin: Springer 2005, S. 47-61. (Lecture Notes in Computer Science 3599).

### CAMPAGNA 1999

Campagna, S.: Polygonreduktion zur effizienten Speicherung, Übertragung und Darstellung komplexer polygonaler Modelle. Dissertation Universität Erlangen-Nürnberg (1998). München: Utz 1999. ISBN: 3-89675-480-7.

### CUIPER 2000

Cuiper, R.: Durchgängige rechnerunterstützte Planung und Steuerung von automatisierten Montagevorgängen. Dissertation Technische Universität München (2000). München: Utz 2000. ISBN: 3-89675-783-0. (*iwb*-Forschungsberichte 143).

### DIERSSSEN 2002

Dierßen, S.: Systemkopplung zur komponentenorientierten Simulation digitaler Produkte. Dissertation Eidgenössische Technische Hochschule Zürich (2002). Düsseldorf: VDI-Verlag 2002. ISBN: 3-18-335820-4. (Fortschritts-Berichte VDI, Reihe 20, Nr. 358).

### DIJKSTRA 1970

Dijkstra, E.: Notes on Structured Programming. Eindhoven University of Technology, Department of Mathematics 1970. (TH-Report 70-WSK-03, EWD 249).

### DILL & ALLUR 1994

Dill, D. L.; Alur, R.: A Theory of Timed Automata. Theoretical Computer Science 126 (1994) 2, S. 183-235.

### DIN EN 61131 TEIL 3 2003

DIN EN 61131, Teil 3: Speicherprogrammierbare Steuerungen – Teil3: Programmiersprachen. Berlin: Beuth 2003.

### DIN EN 62264 TEIL 1 2008

DIN EN 62264, Teil 1: Integration von Unternehmensführungs- und Leitsystemen – Teil 1: Modelle und Terminologie. Berlin: Beuth 2008.

### DIN IEC 60050 TEIL 351 2006

DIN IEC 60050, Teil 351: Internationales Elektrotechnisches Wörterbuch – Teil 351: Leittechnik. Berlin: Beuth 2006.

## DOHMEN 2002

Dohmen, W.: Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer Systeme. Dissertation Technische Universität München (2002). München: Utz 2002. ISBN: 3-8316-0214-X. (*iwb*-Forschungsberichte 174).

## DOMINKA 2007

Dominka, S.: Hybride Inbetriebnahme von Produktionsanlagen – von der virtuellen zur realen Inbetriebnahme. Dissertation Technische Universität München (2007). Göttingen: Sierke 2007. ISBN: 978-3-940333-40-7. (Informationstechnik im Maschinenwesen 46).

## EHRENSPIEL ET AL. 1998

Ehrlenspiel, K.; Kiewert, A.; Lindemann, U.: Kostengünstig Entwickeln und Konstruieren. Kostenmanagement bei der integrierten Produktentwicklung. 2. Aufl. Berlin: Springer 1998. ISBN: 3-540-64705-8.

## EHRENSPIEL 2009

Ehrlenspiel, K.: Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit. 4. Aufl. München: Hanser 2009. ISBN: 978-3-446-42013-7.

## ESCHENBURG &amp; JOST 2007

Eschenburg, J.-H.; Jost, J.: Differentialgeometrie und Minimalflächen. 2. Aufl. Berlin: Springer 2007. ISBN: 978-3-540-22227-9.

## EVERSHEIM ET AL. 1990

Eversheim, W.; Koerth, D.; Gentzke, J.: Inbetriebnahme komplexer Maschinen und Anlagen: Strategien und Praxisbeispiele zur Rationalisierung in der Einzel- und Kleinserienproduktion. Düsseldorf: VDI-Verlag 1990.

## EVERSHEIM ET &amp; SOSSENHEIMER 1987

Eversheim, W.; Sossenheimer, K.: Flächenorientierte Termin- und Kapazitätsplanung in der Montage. VDI-Z 129 (1987) 5, S. 82-86.

## FLATH 2002

Flath, M.: Methode zur Konzipierung mechatronischer Systeme. Dissertation Universität Paderborn (2002). Paderborn: Heinz-Nixdorf-Institut 2002. (HNI-Verlagsschriftenreihe 108).

### FRADEN 2004

Fraden, J.: Handbook of Modern Sensors. New York: Springer 2004. ISBN: 0-387-00750-4.

### FREISEISEN ET AL. 2000

Freiseisen, W.; Keber, R.; Medetz, W.; Stelzmüller, D.: Testing PLC programs with Modelica. In: Modelica Association (Hrsg.): Modelica Workshop 2000. Lund/Schweden, 23.-24. Oktober 2000.  
<<http://www.modelica.org/>> – 05.09.2010.

### FREY & BOSSERT 2008

Frey, T.; Bossert, M.: Signal- und Systemtheorie. 2. Aufl. Wiesbaden: Vieweg 2008. ISBN: 978-3-8351-0249-1.

### FRIEDRICH 2009

Friedrich, D.: Anwendbarkeit von Methoden und Werkzeugen des konventionellen Softwareengineering zur Modellierung und Programmierung von Steuerungssystemen. Dissertation Universität Kassel (2009). Kassel: kassel university press GmbH 2009. ISBN: 978-3-89958-730-2.

### FRÜHAUF 1990

Frühau, K.: Die ältesten Fehler sind die teuersten. Technische Rundschau (1990) 28, S. 32-36.

### GARLAND & HECKBERT 1997

Garland, M.; Heckbert, P. S.: Surface simplification using quadric error metrics. Association for Computing Machinery's Special Interest Group on Graphics and Interactive Techniques (ACM SIGGRAPH) (Hrsg.): 24th Annual Conference on Computer Graphics and Interactive Techniques 1997, Los Angeles/USA, 3.-8. August 1997. New York: ACM Press, S. 209-216.

### GEHRKE 2005

Gehrke, M.: Entwurf mechatronischer Systeme auf Basis von Funktionshierarchien und Systemstrukturen. Dissertation Universität Paderborn 2005.  
<<http://ubdok.uni-paderborn.de>> – 05.09.2010.

### GIROD ET AL. 2000

Girod, B.; Greiner, G.; Niemann, H.: Principles of 3D image analysis and synthesis. Boston: Kluwer Academic Publishers 2000. (Kluwer international series in engineering and computer science SECS 556).

## GLAS 1993

Glas, J.: Standardisierter Aufbau anwendungsspezifischer Zellenrechnersoftware. Dissertation Technische Universität München (1993). Berlin: Springer 1993. ISBN: 3-540-56890-5. (*iwb*-Forschungsberichte 61).

## GRÄTZ 2005

Grätz, F. M.: Simultaneous Engineering am mechatronischen System Werkzeugmaschine. Frankfurt am Main: Forschungsvereinigung Werkzeugmaschinen und Fertigungstechnik e.V. (FWF) 2005. (Abschlussbericht – Forschungsvorhaben Nr. 0823).

## GRÄTZ 2006

Grätz, F. M.: Teilautomatische Generierung von Stromlauf- und Fluidplänen für mechatronische Systeme. Dissertation Technische Universität München (2006). München: Utz 2006. ISBN: 3-8316-0643-9. (*iwb*-Forschungsberichte 200).

## GROSS ET AL. 2007

Gross, D.; Hauger, W.; Wriggers, P.: Technische Mechanik. 6. Aufl. Berlin: Springer 2007. ISBN: 3-540-70737-9.

## GÜNTHER &amp; TEN HOMPEL 2010

Günthner, W.; ten Hompel, M.: Internet der Dinge in der Intralogistik. Berlin: Springer 2010. ISBN: 978-3-642-04895-1.

## HAMMEL &amp; STRAUB 1988

Hammel, J.; Straub, A.: CAD-Baustein einer integrierten Lösung. Heidelberg: Dr. Alfred Hüthig 1988.

## HANSEN &amp; RIEGLER 2002

Hanßen, D.; Riegler, T.: Studie Digitale Fabrik – Zentrales Innovationsthema in der Automobilindustrie. Leinfelden: Pressegespräch Digitale Fabrik 2002.

## HAREL 1987

Harel, D.: Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming* 8 (1987) 3, S. 231-274.

## HAROLD &amp; MEANS 2002

Harold, E. R.; Means, W. S.: XML in a Nutshell. 2. Aufl. Beijing: O'Reilly 2002. ISBN: 0-596-00292-0.

### HEDTSTÜCK 2007

Hedtstück, U.: Einführung in die theoretische Informatik. 4. Aufl. München: Oldenbourg 2007. ISBN: 978-3-486-58269-7.

### HEINRICH & WORTMANN 2004

Heinrich, S.; Wortmann, D.: Virtuelle Inbetriebnahme: Entwicklung von Lay-out- und Steuerungskonzepten mit Hilfe von Simulation. In: RKW/AMZ Sachsen: 2. Symposium „Digitale Fabrik“. Leipzig, 24. Juni 2004. Leipzig: 2004.

### HENZINGER 1996

Henzinger, T. A.: The Theory of Hybrid Automata. In: Institute of Electrical and Electronics Engineer (IEEE) (Hrsg.): 11th Annual IEEE Symposium on Logic in Computer Science 1996 (LICS '96). New Brunswick/USA, 27.-30. Juli 1996. Los Alamitos: IEEE Computer Society Press 1996, S. 278-292.

### HEISSE & SCHNELL 2009

Hesse, S.; Schnell, G.: Sensoren für die Prozess- und Fabrikautomation. 4. Aufl. Wiesbaden: Vieweg 2009. ISBN: 978-3-8348-0471-6.

### HOARE 1985

Hoare, C. A. R.: Communicating Sequential Processes. Englewood: Prentice Hall International 1985.

### HOFFMANN 2008

Hoffmann, D. W.: Software-Qualität. Berlin: Springer 2008. ISBN: 978-3-540-76322-2.

### HOFFMANN 2009

Hoffmann, D. W.: Theoretische Informatik. München: Hanser 2009. ISBN: 978-3-446-41511-9.

### HOPCROFT ET AL. 2002

Hopcroft, J. E.; Motwani, R.; Ullman, J. D.: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. 2. Aufl. München: Pearson Studium 2002. ISBN: 3-8273-7020-5.

### HUMMEL 2011

Hummel, B.: Integrated Behavior Modeling of Space-Intensive Mechatronic Systems. Dissertation Technische Universität München (2007). München: 2011.



**ISERMANN 2006**

Isermann, R.: Fahrdynamik-Regelung: Modellbildung, Fahrerassistenzsysteme, Regelung. Wiesbaden: Vieweg 2006. ISBN: 978-3-8348-0109-8.

**ISERMANN 2008**

Isermann, R.: Mechatronische Systeme. 2. Aufl. Berlin: Springer 2008. ISBN: 978-3-540-32336-5.

**ISO 10303-1 1994**

ISO 10303-1: Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles. Genf: ISO 1994.

**KAIN ET AL. 2010**

Kain, S.; Schiller, F, Trenner, T.: Überwachung und Diagnose mit betriebsparallelen Simulationsmethoden. In: Arbeitsgemeinschaft Simulation (ASIM) (Hrsg.): 14. ASIM-Fachtagung – „Simulation in Produktion und Logistik“. Karlsruhe, 6.-8. Oktober 2010. Karlsruhe: ASIM 2010.

**KAISER 2000**

Kaiser, O.: Echtzeitanforderungen zur Simulation technischer Produkte. Dissertation Technische Universität München (1999). München: Utz 2000. ISBN: 3-89675-734-2. (Informationstechnik im Maschinenwesen 28)

**KALLMEYER 1998**

Kallmeyer, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme. Dissertation Universität Paderborn (1998). Paderborn: Heinz-Nixdorf-Institut 1998. ISBN: 3-931466-41-8. (HNI-Verlagsschriftenreihe 42).

**KASTENS & KLEINE 2008**

Kastens, U.; Kleine Büning, H.: Modellierung: Grundlagen und formale Methoden. 2. Aufl. München: Hanser 2008. ISBN: 978-3-446-41537-9.

**KATZKE 2008**

Katzke, U.: Spezifikation und Anwendung einer Modellierungssprache für die Automatisierungstechnik auf Basis der Unified Modeling Language (UML). Dissertation Universität Kassel (2008). Kassel: kassel university press GmbH 2008. ISBN: 978-3-89958-472-1.

### KESTEN & PNUELI 1992

Kesten, Y.; Pnueli, A.: Timed and Hybrid Statecharts and Their Textual Representation. In: Vytopil, J. et al. (Hrsg.): Formal Techniques in Real-Time and Fault-Tolerant Systmes. Berlin: Springer 1992, S. 591-619. (Lecture Notes in Computer Science 571)

### KIEFER 2007

Kiefer, J.: Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau. Dissertation Universität des Saarlandes (2007). Saarbrücken: Lehrstuhl für Fertigungstechnik – Universität des Saarlandes 2007. ISBN: 978-3-930429-72-1. (Schriftenreihe Produktionstechnik 43).

### KLEMM 2007

Klemm, P.: Simulation Aided Engineering. In: Verl, A. et al. (Hrsg.): Statusseminar Simulationstechnik 2007. Stuttgart, 27. Juni 2007. Stuttgart: ISW 2007.

### KLEPZIG & WEISSBACH 2005

Klepzig, W.; Weißbach, L.: 3D-Konstruktion mit CATIA V5. 2. Aufl. München: Hanser 2005. ISBN: 3-446-40397-3.

### KLOCKE & PRITSCHOW 2004

Klocke, F.; Pritschow, G.: Autonome Produktion. Berlin: Springer 2004. ISBN: 3-540-00518-8.

### KOHRING 1993

Kohring, A.: Systematisches Projektieren und Testen von Steuerungssoftware für Werkzeugmaschinen. Dissertation Technische Hochschule Aachen (1993). Aachen: Shaker 1993. ISBN: 3-86111-587-5. (Berichte aus der Produktionstechnik 13).

### KOLLER 1985

Koller, J.: Konstruktionslehre für den Maschinenbau – Grundlagen des methodischen Konstruierens. Berlin: Springer 1985.

### KORN 1996

Korn, G. H.: Informationssysteme als Mittel der Entscheidungsfindung während des Produktentstehungsprozesses. Dissertation Technische Universität Braunschweig (1995). Essen: Vulkan 1996. ISBN: 3-8027-8628-9. (Schriftenreihe des IWF).

KUPPINGER 2006

Kuppinger, S.: Die Schlüssel zur Effizienz – Die Integration von Motion und Logik. Industrie elektrisch-elektronik 51 (2006) 1, S. 22-23.

LANGMANN 2004

Langmann, R.: Taschenbuch der Automatisierung. München: Hanser 2004. ISBN: 3-446-21793-2.

LEONDES 2000

Leondes, C. T.: Mechatronic Systems – Techniques and Applications. Amsterdam: Gordon and Breach 2000. ISBN: 90-5699-679-7.

LERCH 2007

Lerch, R.: Elektrische Messtechnik: Analoge, digitale und computergestützte Verfahren. 4. Aufl. Berlin: Springer 2007. ISBN: 3-540-73610-7.

LERCHER 2008

Lercher, B.: Konzeption und System einer Integrationsplattform zur Entwicklung von Werkzeugmaschinen. Dissertation Technische Universität München (2008). <<http://deposit.d-nb.de/cgi-bin/dokserv?idn=99070999x>> – 03.11.2010

LIGGESMEYER 2009

Liggemeyer, P.: Software-Qualität. 2. Aufl. Heidelberg: Spektrum 2009. ISBN: 978-3-8274-2056-5.

LINDEMANN 2009

Lindemann, U.: Methodische Entwicklung technischer Produkte. 3. Aufl. Berlin: Springer 2009. ISBN: 978-3-642-01422-2.

LINHARES ET AL. 2006

Linhares, M. V.; da Silva, A. J.; Silva de Oliveira, R.: Empirical Evaluation of SysML through the Modeling of an Industrial Automation Unit. In: Institute of Electrical and Electronic Engineers (IEEE) (Hrsg.): 13th IEEE International Conference on Emerging Technologies and Factory Automation 2006 (ETFA 2006). Prag/Tschechien, 20.-22. September 2006. Prag: IEEE 2008, S. 145-152.

### LIPPOLD 2001

Lippold, C.: Eine domänenübergreifende Konzeptionsumgebung für die Entwicklung mechatronischer Systeme. Dissertation Ruhr-Universität Bochum (2000). Aachen: Shaker 2001. ISBN: 3-8265-8707-3.

### LUNZE 2002

Lunze, J.: What Is a Hybrid System? In: Engell, S. et al. (Hrsg.): Modelling, analysis, and design of hybrid systems. Berlin: Springer 2002, S. 3-14. ISBN: 3-540-43812-2.

### LUNZE 2006

Lunze, J.: Ereignisdiskrete Systeme. München: Oldenbourg 2006. ISBN: 978-3-486-58071-6.

### LUTZ 2006

Lutz, R.: Kompetenz durch Simulation – Hilfsmittel bei Entwicklung, Inbetriebnahme und Anwendung komplexer Großpressen. In: Zäh, M. F. et al. (Hrsg.): Virtuelle Inbetriebnahme – Von der Kür zur Pflicht?. Garching, 28. September 2006. München: Utz 2006. ISBN: 978-3-89675-084-6 (*iwb-Seminarberichte* 84).

### MATSUNO ET AL. 2003

Matsuno, H.; Aoshima, H.; Doi, A.; Tanaka, Y.; Matsui, M.; Miyano S.: Biopathways Representation and Simulation on Hybrid Functional Petri Net. *Silico Biology* 3 (2003) 3, S. 389-404.

### MAURMAIER 2008

Maurmaier, M.: Leveraging Model-driven Development for Automation Systems Development. In: Institute of Electrical and Electronic Engineers (IEEE) (Hrsg.): 13th IEEE International Conference on Emerging Technologies and Factory Automation 2008 (ETFA 2008). Hamburg, 15.-18. September 2008. Hamburg: IEEE 2008, S. 733-736.

### MESHLAB 2010

MeshLab: MeshLab. <<http://www.meshlab.net>> – 06.05.2010.

### MEYBERG & VACHENAUER 2001

Meyberg, K.; Vachenaue, P.: Höhere Mathematik 2. 4. Aufl. Berlin: Springer 2001. ISBN: 3-540-41851-2.

MILBERG & EBENER 1994

Milberg, J.; Ebner, K.: Verfügbarkeit von Werkzeugmaschinen. Abschlussbericht der AiF-Forschungsstudie 8649. Frankfurt am Main: VDW 1994.

MIN ET AL. 2002

Min, B. K.; Huang, Z.; Pasek, Z. J.; Yip-Hoi, D.; Husted, F.; Marker, S.: Integration of Real-time Control Solution to a Virtual Manufacturing Environment. International Journal of Advanced Manufacturing Systems, Special Issue on Virtual Manufacturing 1 (2002) 1, S. 67-87.

MIT 2003

MIT Technology Review: 10 Emerging Technologies That Will Change The World. MIT 2003.

NYHUIS 2008

Nyhuis, P.: Beiträge zu einer Theorie der Logistik. Berlin: Springer 2008. ISBN: 9783540756415.

OESTREICHER 1986

Oestreicher, T.: Rechnergestützte Projektierung von Steuerungssystemen. Dissertation Technische Hochschule Karlsruhe (1986). Karlsruhe: Schnelldruck Ernst Grässer 1986.

OESTREICH 2006

Oestereich, B.: Analyse und Design mit UML 2.1: objektorientierte Softwareentwicklung. 8. Aufl. München: Oldenbourg 2006. ISBN: 978-3-486-57926-0.

OSAKRSSON & GLASS 1997

Oskarsson, Ö.; Glass, R. L.: ISO 9000 und Software-Qualität. München: Prentice Hall 1997. ISBN: 3-8272-9538-6.

PAHL & BEITZ 1993

Pahl, G.; Beitz, W.: Konstruktionslehre: Methoden und Anwendung. 3. Aufl. Berlin: Springer 1993. ISBN: 3-540-56194-3.

PAYNTER 1961

Paynter, H. M.: Analysis and Design of Engineering Systems. Cambridge, Massachusetts: MIT Press 1961.

### PETRI 1962

Petri, C. A.: Kommunikation mit Automaten. Dissertation Universität Bonn (1962). Bonn: 1962. (Schriften des Rheinisch-Westfälischen Institutes für instrumentelle Mathematik an der Universität Bonn 2)

### PILONE & PITMAN 2006

Pilone, D.; Pitman, N.; Heymann-Reder, D.: UML 2.0 in a Nutshell. 1. Aufl. Beijing: O'Reilly 2006. ISBN: 978-3-89721-342-5.

### PONN & LINDEMANN 2008

Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte. Berlin: Springer 2008. ISBN: 978-3-540-68562-3.

### PRIESE & WIMMEL 2008

Priese, L.; Wimmel, H.: Petri-Netze. 2. Aufl. Berlin: Springer 2008. ISBN: 978-3-540-76970-5.

### PRITSCHOW 2005

Pritschow, G.; Röck, S.; Rüdele, H.: Echtzeitfähige Simulation von Werkzeugmaschinen. wt Werkstattstechnik online 95 (2005) 5, S. 302-308.

### PRITSCHOW 2006

Pritschow, G.: Einführung in die Steuerungstechnik. München: Hanser 2006. ISBN: 3-446-21422-4.

### REINHART ET AL. 2002

Reinhart, G.; Anton, O.; Lercher, B.: Sichtenorientierte Störungsmodellierung an Werkzeugmaschinen. Industrie-Manager 18 (2002) 1, S. 35-39.

### REINHART ET AL. 2007

Reinhart, G.; Hensel, T.; Lindworsky, A.: Teilautomatisierter Aufbau von Simulationsmodellen – Beschleunigter Modellaufbau am Beispiel der „Virtuellen Inbetriebnahme“. wt Werkstattstechnik online 97 (2007) 9, S. 663-667.

### RÖCK 2007

Röck, S.: Echtzeitsimulation von Produktionsanlagen mit realen Steuerungssystemen. Dissertation Universität Stuttgart (2007). Heimsheim: Jost Jetter 2007. (ISW Forschung und Praxis 168).

RODDECK 2006

Roddeck, W.: Einführung in die Mechatronik. 3. Aufl. Wiesbaden: Teubner 2006. ISBN: 3-8351-0071-8.

ROITZSCH 2005

Roitzsch, E. H. P.: Analytische Softwarequalitätssicherung in Theorie und Praxis. Münster: Verlagshaus Monsenstein und Vannerdat 2005. ISBN: 3-86582-202-9.

RÖMBERG 2004

Römberg, S.: Virtuelle Inbetriebnahme: Innovative Entwicklungen im Bereich der Digitalen Fabrik. In: RKW/AMZ Sachsen: 2. Symposium „Digitale Fabrik“. Leipzig, 24. Juni 2004. Leipzig: 2004.

ROSSIGNAC & BORREL 1993

Rossignac, J.; Borrel, P.: Multi-resolution 3D approximations for rendering complex scenes. In: Falcidieno, B. et al. (Hrsg.): Geometric Modeling in Computer Graphics. Genua/Italien 28. Juni bis 2. Juli 1993. Berlin: Springer 1993, S. 455-465.

ROTH 1982

Roth, K. H.: Konstruieren mit Konstruktionskatalogen. Berlin: Springer 1982.

ROYCE 1970

Royce, W. W.: Managing the Development of large software systems. In: Institute of Electrical and Electronics Engineer (IEEE) (Hrsg.): IEEE WESCON 26. IEEE 1970, S. 1-9.

SCHÄFER & ROLLER 2001

Schäfer, D.; Roller, D.: Electrical Engineering Solutions – ECAE Systeme der dritten Generation. Automatisierungstechnische Praxis 43 (2001) 3, S. 40-44.

SCHAICH 1996

Schaich, C.: Simultaneous Engineering bei der Entwicklung intelligenter Maschinen. In: Reinhart, G. et al. (Hrsg.): Dezentrale Steuerungen in Produktionsanlagen – Plug & Play – Vereinfachung von Entwicklung und Inbetriebnahme. Garching, 8. Mai 1996. München: Utz 1996, S. 21-44. ISBN: 3-931327-20-5. (*iwb*-Seminarberichte 20).

### SCHAICH 2001

Schaich, C.: Informationsmodell zur fachübergreifenden Beschreibung intelligenter Produktionsmaschinen. Dissertation Technische Universität München (2001). München: Utz 2001. (Informationstechnik im Maschinenwesen 31).

### SCHEIFELE 2008

Scheifele, D.: Effizient Planen und Entwickeln mit virtuellen Maschinen. In: Westkämper, E. et al. (Hrsg.): Effiziente Planung und Entwicklung von Automatisierungslösungen: Technologien – Methoden – Anwendungen. Stuttgart, 18. November 2008. Stuttgart: FpF - Verein zur Förderung produktionstechnischer Forschung 2008. (Fraunhofer IPA Workshop F172.)

### SCHMITT 1996

Schmitt, K.: Vorgehensweise bei der Umstellung einer zentralen auf eine dezentrale Steuerungsstruktur. In: Reinhart, G. et al. (Hrsg.): Dezentrale Steuerungen in Produktionsanlagen – Plug & Play – Vereinfachung von Entwicklung und Inbetriebnahme. Garching, 8. Mai 1996. München: Utz 1996, S. 21-44. ISBN: 3-931327-20-5. (*iwb*-Seminarberichte 20).

### SCHNELL & WIEDEMANN 2006

Schnell, G.; Wiedemann, B.: Bussysteme in der Automatisierungs- und Prozesstechnik. 6. Aufl. Wiesbaden: Vieweg 2006. ISBN: 3-8348-0045-7.

### SCHNIEDER 1993

Schnieder, E.: Prozessinformatik. Automatisierung mit Rechensystemen – Einführung mit Petrinetzen. Wiesbaden: Vieweg 1993.

### SCHOB ET AL. 2009

Schob, U.; Blochwitz, T.; Oelsner, O.; Winter, M.: Model Based Virtual Start-up of Automation Systems. In: Casella F. (Hrsg.): 7th International Modelica Conference 2009. Como/Italien, 20.-22. September 2009. Modelica Association 2009, S. 790-796. ISBN: 978-91-7393-513-5.

### SCHNITTLER 2004

Schnittler, V.: IT-Kostenbeurteilung mit Kennzahlen. In: VDMA (Hrsg.): VDMA-Lösung, Ausgabe 4/2004. Frankfurt am Main: VDMA 2004.

### SCHÜTZ & WANNAGAT 2009

Schütz, D.; Wannagat, A.: Domänenspezifische Modellierung für automatisierungstechnische Anlagen mit Hilfe der SysML. *atp Automatisierungstechnische Praxis* 50 (2009) 3, S. 54-62.



SELIC ET AL. 1994

Selic, B.; Gullekson, G.; Ward, P. T.: Real-time object-oriented modeling. 2. Aufl. New York: Wiley 1994. ISBN: 0-471-59917-4.

SPATH & LANDWEHR 2000

Spath, D.; Landwehr, R.: 3-D-Projektierung und Simulation von Ablaufsteuerungen. wt Werkstattstechnik online 90 (2000) 7/8, S. 292-296.

SPILLNER & LINZ 2005

Spillner, A.; Linz, T.: Basiswissen Softwaretest. Heidelberg: dpunkt 2005. ISBN: 3-89864-358-1.

SPITZWEG 2009

Spitzweg, M.: Methode und Konzept für den Einsatz eines physikalischen Modells in der Entwicklung von Produktionsanlagen. Dissertation Technische Universität München (2009). München: Utz 2009. ISBN: 978-3-8316-0931-4. (*iwb*-Forschungsberichte 233).

STACHOWIAK 1973

Stachowiak, H.: Allgemeine Modelltheorie. Berlin: Springer 1973. ISBN: 0-387-81106-0.

STAHL & VÖLTER 2005

Stahl, T.; Völter, M.: Modellgetriebene Softwareentwicklung. 1. Aufl. Heidelberg: dpunkt 2005. ISBN: 978-3-89864-310-8.

STAUNER 2002

Stauner, T.: Systematic Development of Hybrid Systems. Dissertation Technische Universität München (2001). Bonn: Gesellschaft für Informatik 2002. ISBN: 3-88579-406-3.

STAUNER ET AL. 2001

Stauner, T.; Pretschner, A.; Péter, I.: Approaching a Discrete-Continuous UML: Tool Support and Formalization. In: pUML Group et al.: UML'2001 Workshop on Practical UML-Based Rigorous Development Methods – Countering or Integrating the eXtremists. Toronto/Canada, 1. Oktober 2001. Toronto pUML 2001, S. 242-257.

### STETTER 2006

Stetter, R.: Simulationsansätze und Virtuelle Inbetriebnahme. In: Zäh, M. F. et al. (Hrsg.): Virtuelle Inbetriebnahme – Von der Kür zur Pflicht?. Garching, 28. September 2006. München: Utz 2006. ISBN: 978-3-89675-084-6 (*iwb-Seminarberichte* 84).

### THRAMBOULIDIS 2008

Thramboulidis, K.: Challenges in the Development of Mechatronic Systems: The Mechatronic Component. In: Institute of Electrical and Electronic Engineers (IEEE) (Hrsg.): 13th IEEE International Conference on Emerging Technologies and Factory Automation 2008 (ETFA 2008). Hamburg, 15.-18. September 2008. Hamburg: IEEE 2008, S. 624-631.

### TILBURY ET AL. 2010

Tilbury, D. M.; Park, E.; Khargonekar, P. P.: A Modeling and Analysis Methodology for Modular Logic Controllers of Machining Systems using Petri Net Formalism. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 31 (2001) 2, S. 168-188..

### TOMASZUNAS 1998

Tomaszunas, J.: Komponentenbasierte Maschinenmodellierung zur Echtzeit-Simulation für den Steuerungstest. Dissertation Technische Universität München (1998). München: Utz 1998. ISBN: 3-89675-315-0. (Informationstechnik im Maschinenwesen 21).

### VAJNA ET AL. 2009

Vajna, S.; Bley, H.; Hehenberger, P.; Weber, C.; Zeman, K.: CAX für Ingenieure. 2. Aufl. Berlin: Springer 2009. ISBN: 978-3-5403-6039-1.

### VDI-RICHTLINIE 2206 2004

VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Systeme. Berlin: Beuth 2004.

### VDI-RICHTLINIE 2221 1993

VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Berlin: Beuth 1993.

### VDI-RICHTLINIE 2222 BLATT 1 1997

VDI 2222-Richtlinie, Blatt 1: Konstruktionsmethodik: Methodisches Entwickeln von Lösungsprinzipien. Berlin: Beuth 1997.

VDI-RICHTLINIE 3633 1996

VDI-Richtlinie 3633: Simulation von Logistik-, Materialfluss- und Produktionssystemen. Berlin: Beuth 1996.

VDI/VDE-RICHTLINIE 2422 1994

VDI/VDE-Richtlinie 2422: Entwicklungsmethodik für Geräte mit Steuerung durch Mikroelektronik. Berlin: Beuth 1994.

VDI/VDE-RICHTLINIE 3681 2005

VDI-Richtlinie 3681: Einordnung und Bewertung von Beschreibungsmitteln aus der Automatisierungstechnik. Berlin: Beuth 2005.

VDMA 2004

VDMA: Baukastenbasiertes Engineering mit Föderal: Ein Leitfaden für Maschinen- und Anlagenbauer. Frankfurt am Main: VDMA 2004. ISBN: 3-8163-0478-8.

VDMA 2010

VDMA: Maschinenbau in Zahl und Bild 2010. Frankfurt am Main: VDMA 2010.

VDW-Bericht 1997

VDW-Bericht: Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen. Aachen: WZL 1997.

VDW-RICHTLINIE 2001

VDW-Richtlinie: Ergebnisse der Projektgruppe 3 – Richtlinie Funktionsbeschreibung. Frankfurt am Main: VDW 2001.

VOGEL-HEUSER 2003

Vogel-Heuser, B.: Systems Software Engineering. München: Oldenbourg 2003. ISBN: 3-486-27035-4.

VOGEL-HEUSER & FISCHER 2002

Vogel-Heuser, B.; Fischer, K.: UML in der automatisierungstechnischen Anwendung – Stärken und Schwächen. atp Automatisierungstechnische Praxis 44 (2002) 10, S. 63-69.

### VOGEL-HEUSER & FRIEDRICH 2006

Vogel-Heuser, B.; Friedrich, D.: Nutzen von Modellierung für die Qualität und Effizienz der Steuerungsprogrammierung in der Automatisierungstechnik. *atp Automatisierungstechnische Praxis* 48 (2006) 3, S. 54-60.

### VOGEL-HEUSER & KATZKE 2005

Katzke, U.; Vogel-Heuser, B.: UML-PA as an Engineering Model for Distributed Process Automation. In: Horacek, P. (Hrsg.): 16th Triennial World Congress of the International Federation of Automatic Control (IFAC). Prag/Tschechien, 4.-8. Juli 2005. Oxford: Elsevier Science Ltd. 2005.

### VOGEL-HEUSER & WANNAGAT 2009

Vogel-Heuser, B., Wannagat, A.: *Modulares Engineering und Wiederverwendung mit CoDeSys V3*. München: Oldenbourg 2009. ISBN: 978-3-8356-3105-2.

### WEBER 2006

Weber, K. H.: *Inbetriebnahme verfahrenstechnischer Anlagen*. 3. Aufl. Berlin: Springer 2006. ISBN: 978-3-540-34316-5.

### WECK & BRECHER 2006

Weck, M.; Brecher, C.: *Automatisierung von Maschinen und Anlagen*. 6. Aufl. Berlin: Springer 2006. ISBN: 978-3-540-22507-2.

### WEILKIENS 2006

Weilkiens, T.: *Systems engineering mit SysML/UML*. 1. Aufl. Heidelberg: dpunkt 2006. ISBN: 978-3-89864-409-9.

### WELLENREUTHER & ZASTROW 2005

Wellenreuther, G.; Zastrow, D.: *Automatisieren mit SPS*. 3. Aufl. Wiesbaden: Vieweg 2005. ISBN: 3-528-23910-7.

### WÜNSCH 2008

Wünsch, G.: *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. Dissertation Technische Universität München (2007). München: Utz 2008. ISBN: 978-3-8316-07952. (*iwf*-Forschungsberichte 215).

**WÜRSLIN & WASCHNER 2008**

Würslin, R.; Waschner, F.: Mechatronischer Baukasten – Vorgehensmodell und Modellierungsmethodik für ein mechatronisches Entwicklungswerkzeug. In: Kompetenznetzwerk Mechatronik e.V. (Hrsg.): Internationales Forum Mechatronik. Stuttgart, 22.-23. September 2008. Göppingen: Albert Frey 2008, S. 330-343.

**XU 2003**

Xu, L.: Wiederverwendbare Modelle zur Maschinensimulation für den Steuerungstest. Dissertation Technische Universität München (2003). München: Utz 2003. ISBN: 3-8316-0281-6. (Informationstechnik im Maschinenwesen 36).

**ZÄH ET AL. 2004**

Zäh, M. F.; Ehrenstraßer, M.; Pömbacher, C.; Wünsch, G.: Simulation von Maschinenabläufen an virtuellen Werkzeugmaschinen. Frankfurt am Main: FWF 2004.

**ZÄH ET AL. 2005A**

Zäh, M. F.; Möller, N.; Vogel, W.: Symbiosis of Changeable and Virtual Production. 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2005). München, 22.-23. September 2005. München: Utz 2005, S. 3-10. ISBN: 3-8316-0540-8.

**ZÄH ET AL. 2005B**

Zäh, M. F.; Grätz, F. M.; Wünsch, G.: Entwicklung mechatronischer Produktionssysteme – Eine interdisziplinäre Herausforderung. In: Zäh, M. F. et al. (Hrsg.): Seminar Mechatronik – Trends in der interdisziplinären Entwicklung von Werkzeugmaschinen. Garching, 3. März 2005. München: Utz 2005. ISBN: 3-89675-078-X. (*iwb*-Seminarberichte 78)

**ZÄH ET AL. 2006**

Zäh, M. F.; Wünsch, G.; Hensel, T.; Lindworsky, A.: Feldstudie - Virtuelle Inbetriebnahme. *wt Werkstattstechnik online* 96 (2006) 10, S. 767-771.

**ZÄH ET AL. 2008**

Zäh, M. F.; Spitzweg, M.; Lacour, F.-F.: Einsatz eines Physikmodells zur Simulation des Materialflusses einer Produktionsanlage. *it Information Technology* 50 (2008) 3, S. 192-198.

### ZÄH & LINDWORSKY 2008

Zäh, M. F.; Lindworsky, A.: Virtuelle Inbetriebnahme – Nutzenmaximierung durch Automatisierung der Modellerstellung. In: Kompetenznetzwerk Mechatronik e.V. (Hrsg.): Internationales Forum Mechatronik. Stuttgart, 22.-23. September 2008. Göppingen: Albert Frey 2008, S. 410-424.

### ZÄH & PÖRNBACHER 2008

Zäh, M. F.; Pörnbacher, C.: Model-driven development of PLC software for machine tools. Production Engineering 2 (2008) 1, S. 39-46.

### ZEIGLER ET AL. 2000

Zeigler, B. P.; Praehofer, H.; Kim, T. G.: Theory of modeling and simulation. 2. Aufl. San Diego, California: Academic Press 2000. ISBN: 0-12-778455-1.

### ZEUGTRÄGER 1998

Zeugträger, K.: Anlaufmanagement für Großanlagen. Dissertation Universität Hannover (1998). Düsseldorf: VDI-Verlag 1998. (Fortschrittberichte VDI, Reihe 2, Nr. 470).

### ZÖBEL 2008

Zöbel, D.: Echtzeitsysteme. Berlin: Springer 2008. ISBN: 978-3-540-76395-6.

## 10 Anhang

### 10.1 XML-Austauschformat für CAD-Daten

#### 10.1.1 Mechanik

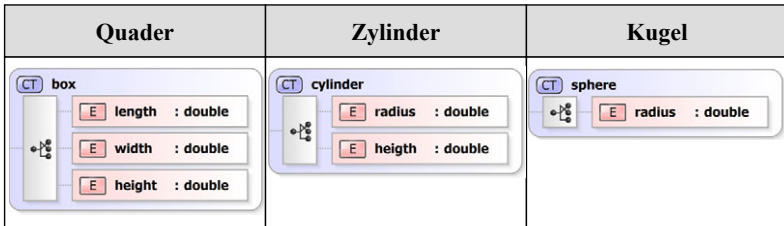


Abbildung 10.1: XML-Schema der Mechanik für Volumenprimitive

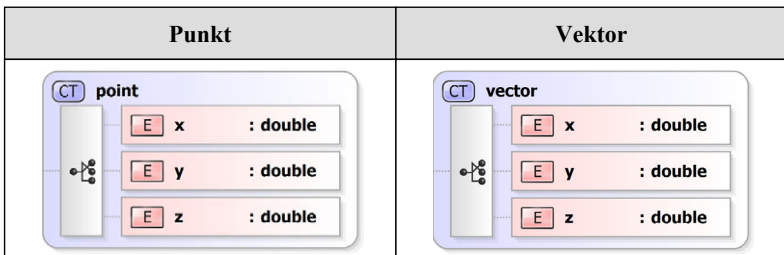


Abbildung 10.2: XML-Schema der Mechanik für Punkte und Vektoren im dreidimensionalen Raum

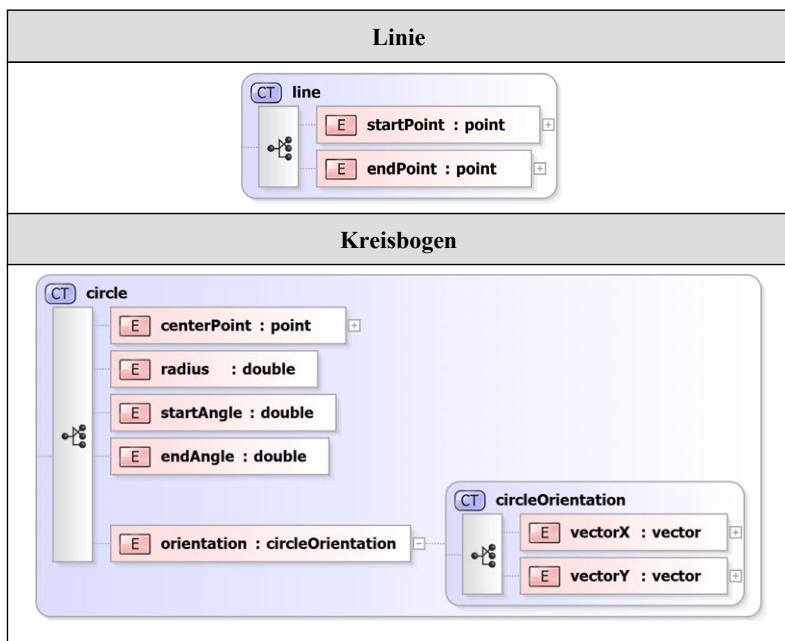


Abbildung 10.3: XML-Schema der Mechanik für Linien und Kreisbögen

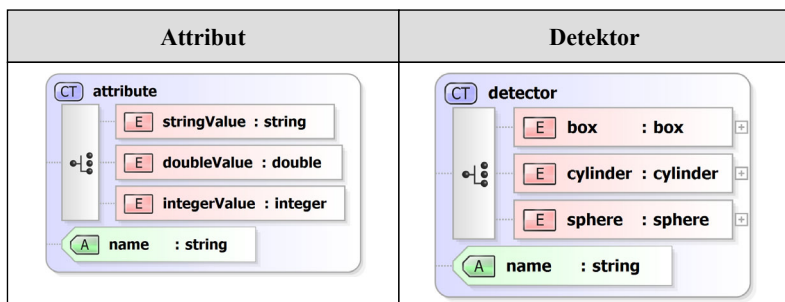


Abbildung 10.4: XML-Schema der Mechanik für Attribute und Detektoren



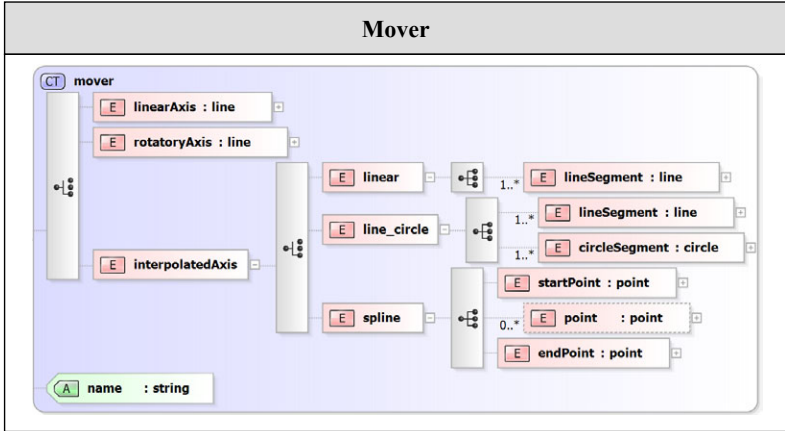


Abbildung 10.5: XML-Schema der Mechanik für Bewegungsachsen und -pfade

### 10.1.2 Elektrotechnik und Fluidik

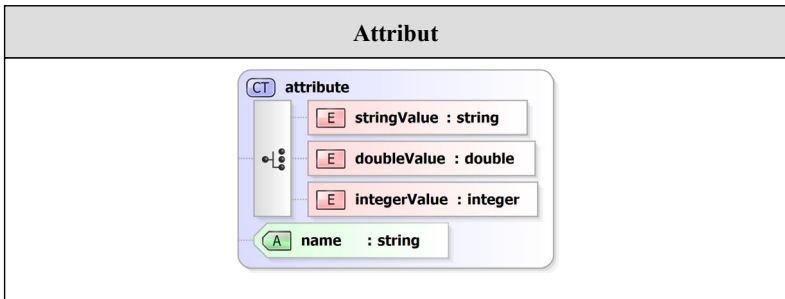


Abbildung 10.6: XML-Schema der Elektrotechnik und der Fluidik für Attribute

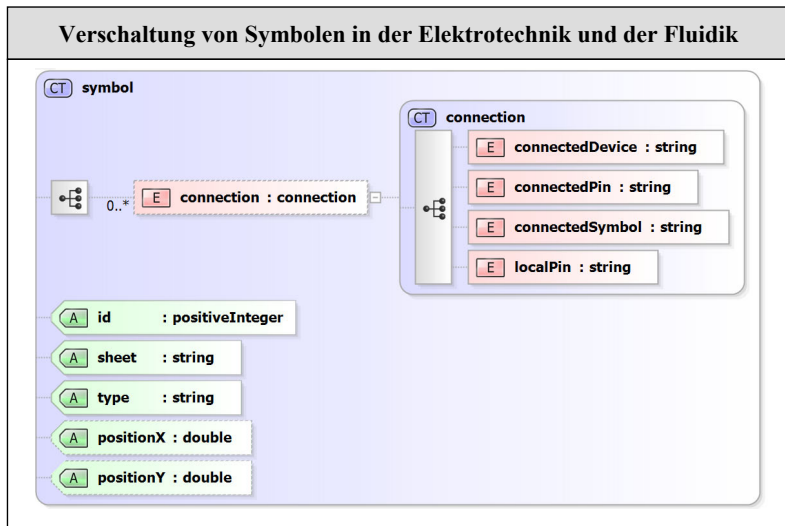


Abbildung 10.7: XML-Schema der Elektrotechnik und der Fluidik für die Verschaltung von Komponenten

## 10.2 Firmen- und Herstellerverzeichnis

### *Dassault Systèmes Headquarters*

10, Rue Marcel Dassault  
78140 Vélizy-Villacoublay  
Frankreich  
<<http://www.3ds.com>>

### *EPLAN Software & Service GmbH & Co. KG*

An der alten Ziegelei 2  
40789 Monheim am Rhein  
Deutschland  
<<http://www.eplan.de>>

### *Gebr. Heller Maschinenfabrik GmbH*

Gebrüder-Heller-Straße 15  
72622 Nürtingen  
Deutschland  
<<http://www.heller.biz>>

### *ISG Industrielle Steuerungstechnik GmbH*

Rosenbergstraße 28  
70174 Stuttgart  
Deutschland  
<<http://www.isg-stuttgart.de>>

*Kapp GmbH*

Callenberger Straße 52  
96450 Coburg  
Deutschland  
<<http://www.kapp-coburg.de>>

*Mewes & Partner GmbH*

Neuendorferstraße 15  
16761 Henningsdorf  
Deutschland  
<<http://www.mewes-partner.de>>

*Microsoft Corporation*

1 Microsoft Way  
Redmond, WA 98052-8300  
USA  
<<http://www.microsoft.com>>

*Siemens AG*

Wittelsbacherplatz 2  
80333 München  
Deutschland  
<<http://www.siemens.com>>

*Siemens Industry Software GmbH & Co. KG*

Eisenwerkstraße 1  
58332 Schwelm  
Deutschland  
<<http://www.innotec.de>>

*Siemens PLM Software GmbH*

Oskar-Messter-Straße 22  
85737 Ismaning  
Deutschland  
<<http://www.plm.automation.siemens.com>>

*The MathWorks Headquarters*

3 Apple Hill Drive  
Natick, MA 01760-2098  
USA  
<<http://www.themathworks.com>>

*Zuken Inc., Shin-Yokohama*

3-1-1 Shin-Yokohama, Kouhoku-ku, Yokohama  
Kanagawa, 222-8505  
Japan  
<<http://www.zuken.com>>



# iwb Forschungsberichte Band 1–121

Herausgeber: Prof. Dr.-Ing. J. Milberg und Prof. Dr.-Ing. G. Reinhart, Institut für Werkzeugmaschinen und Betriebswissenschaften der Technischen Universität München

Band 1–121 erschienen im Springer Verlag, Berlin, Heidelberg und sind im Erscheinungsjahr und den folgenden drei Kalenderjahren erhältlich im Buchhandel oder durch Lange & Springer, Otto-Suhr-Allee 26–28, 10585 Berlin

- 1 *Streitinger, E.*  
Beitrag zur Sicherung der Zuverlässigkeit und Verfügbarkeit moderner Fertigungsmittel  
1986 · 72 Abb. · 167 Seiten · ISBN 3-540-16391-3
- 2 *Fuchsberger, A.*  
Untersuchung der spannenden Bearbeitung von Knochen  
1986 · 90 Abb. · 175 Seiten · ISBN 3-540-16392-1
- 3 *Maier, C.*  
Montageautomatisierung am Beispiel des Schraubens mit Industrierobotern  
1986 · 77 Abb. · 144 Seiten · ISBN 3-540-16393-X
- 4 *Summer, H.*  
Modell zur Berechnung verzweigter Antriebsstrukturen  
1986 · 74 Abb. · 197 Seiten · ISBN 3-540-16394-8
- 5 *Simon, W.*  
Elektrische Vorschubantriebe an NC-Systemen  
1986 · 141 Abb. · 198 Seiten · ISBN 3-540-16693-9
- 6 *Büchs, S.*  
Analytische Untersuchungen zur Technologie der Kugelbearbeitung  
1986 · 74 Abb. · 173 Seiten · ISBN 3-540-16694-7
- 7 *Hunzinger, I.*  
Schneiderodierte Oberflächen  
1986 · 79 Abb. · 162 Seiten · ISBN 3-540-16695-5
- 8 *Pilland, U.*  
Echtzeit-Kollisionsschutz an NC-Drehmaschinen  
1986 · 54 Abb. · 127 Seiten · ISBN 3-540-17274-2
- 9 *Barthelmeß, P.*  
Montagegerechtes Konstruieren durch die Integration von Produkt- und Montageprozeßgestaltung  
1987 · 70 Abb. · 144 Seiten · ISBN 3-540-18120-2
- 10 *Reithofer, N.*  
Nutzungssicherung von flexibel automatisierten Produktionsanlagen  
1987 · 84 Abb. · 176 Seiten · ISBN 3-540-18440-6
- 11 *Diess, H.*  
Rechnerunterstützte Entwicklung flexibler automatisierter Montageprozesse  
1988 · 56 Abb. · 144 Seiten · ISBN 3-540-18799-5
- 12 *Reinhart, G.*  
Flexible Automatisierung der Konstruktion und Fertigung elektrischer Leitungssätze  
1988 · 112 Abb. · 197 Seiten · ISBN 3-540-19003-1
- 13 *Bürstner, H.*  
Investitionsentscheidung in der rechnerintegrierten Produktion  
1988 · 74 Abb. · 190 Seiten · ISBN 3-540-19099-6
- 14 *Groha, A.*  
Universelles Zellenrechnerkonzept für flexible Fertigungssysteme  
1988 · 74 Abb. · 153 Seiten · ISBN 3-540-19182-8
- 15 *Riese, K.*  
Klipsmontage mit Industrierobotern  
1988 · 92 Abb. · 150 Seiten · ISBN 3-540-19183-6
- 16 *Lutz, P.*  
Leitsysteme für rechnerintegrierte Auftragsabwicklung  
1988 · 44 Abb. · 144 Seiten · ISBN 3-540-19260-3
- 17 *Klippel, C.*  
Mobiler Roboter im Materialfluß eines flexiblen Fertigungssystems  
1988 · 86 Abb. · 164 Seiten · ISBN 3-540-50468-0
- 18 *Rascher, R.*  
Experimentelle Untersuchungen zur Technologie der Kugelherstellung  
1989 · 110 Abb. · 200 Seiten · ISBN 3-540-51301-9
- 19 *Heusler, H.-J.*  
Rechnerunterstützte Planung flexibler Montagesysteme  
1989 · 43 Abb. · 154 Seiten · ISBN 3-540-51723-5
- 20 *Kirchknopf, P.*  
Ermittlung modaler Parameter aus Übertragungsfrequenzgängen  
1989 · 57 Abb. · 157 Seiten · ISBN 3-540-51724-3
- 21 *Sauerer, Ch.*  
Beitrag für ein Zerspanprozeßmodell Metallbandsägen  
1990 · 89 Abb. · 166 Seiten · ISBN 3-540-51868-1
- 22 *Karstedt, K.*  
Positionsbestimmung von Objekten in der Montage- und Fertigungsautomatisierung  
1990 · 92 Abb. · 157 Seiten · ISBN 3-540-51879-7
- 23 *Peiker, St.*  
Entwicklung eines integrierten NC-Planungssystems  
1990 · 66 Abb. · 180 Seiten · ISBN 3-540-51880-0
- 24 *Schugmann, R.*  
Nachgiebige Werkzeugaufhängungen für die automatische Montage  
1990 · 71 Abb. · 155 Seiten · ISBN 3-540-52138-0
- 25 *Wirba, P.*  
Simulation als Werkzeug in der Handhabungstechnik  
1990 · 125 Abb. · 178 Seiten · ISBN 3-540-52231-X
- 26 *Eibelshäuser, P.*  
Rechnerunterstützte experimentelle Modalanalyse mittels gestufter Sinusanregung  
1990 · 79 Abb. · 156 Seiten · ISBN 3-540-52451-7
- 27 *Prasch, J.*  
Computerunterstützte Planung von chirurgischen Eingriffen in der Orthopädie  
1990 · 113 Abb. · 164 Seiten · ISBN 3-540-52543-2

- 28 *Teich, K.*  
**Prozeßkommunikation und Rechnerverbund in der Produktion**  
1990 · 52 Abb. · 158 Seiten · ISBN 3-540-52764-8
- 29 *Pfarr, W.*  
**Rechnergestützte und graphische Planung manueller und teilautomatisierter Arbeitsplätze**  
1990 · 59 Abb. · 153 Seiten · ISBN 3-540-52829-6
- 30 *Tauber, A.*  
**Modellbildung kinematischer Strukturen als Komponente der Montageplanung**  
1990 · 93 Abb. · 190 Seiten · ISBN 3-540-52911-X
- 31 *Jäger, A.*  
**Systematische Planung komplexer Produktionssysteme**  
1991 · 75 Abb. · 148 Seiten · ISBN 3-540-53021-5
- 32 *Hartberger, H.*  
**Wissensbasierte Simulation komplexer Produktionssysteme**  
1991 · 58 Abb. · 154 Seiten · ISBN 3-540-53326-5
- 33 *Tuczek, H.*  
**Inspektion von Karosseriepreßteilen auf Risse und Einschnürungen mittels Methoden der Bildverarbeitung**  
1992 · 125 Abb. · 179 Seiten · ISBN 3-540-53965-4
- 34 *Fischbacher, J.*  
**Planungsstrategien zur störungstechnischen Optimierung von Reinraum-Fertigungsgeräten**  
1991 · 60 Abb. · 166 Seiten · ISBN 3-540-54027-X
- 35 *Moser, O.*  
**3D-Echtzeitkollisionsschutz für Drehmaschinen**  
1991 · 66 Abb. · 177 Seiten · ISBN 3-540-54076-8
- 36 *Naber, H.*  
**Aufbau und Einsatz eines mobilen Roboters mit unabhängiger Lokomotions- und Manipulationskomponente**  
1991 · 85 Abb. · 139 Seiten · ISBN 3-540-54216-7
- 37 *Kupec, Th.*  
**Wissensbasiertes Leitsystem zur Steuerung flexibler Fertigungsanlagen**  
1991 · 68 Abb. · 150 Seiten · ISBN 3-540-54260-4
- 38 *Maulhardt, U.*  
**Dynamisches Verhalten von Kreissägen**  
1991 · 109 Abb. · 159 Seiten · ISBN 3-540-54365-1
- 39 *Gatz, R.*  
**Strukturierte Planung flexibel automatisierter Montagesysteme für flächige Bauteile**  
1991 · 86 Abb. · 201 Seiten · ISBN 3-540-54401-1
- 40 *Koepfer, Th.*  
**3D-grafisch-interaktive Arbeitsplanung - ein Ansatz zur Aufhebung der Arbeitsteilung**  
1991 · 74 Abb. · 126 Seiten · ISBN 3-540-54436-4
- 41 *Schmidt, M.*  
**Konzeption und Einsatzplanung flexibel automatisierter Montagesysteme**  
1992 · 108 Abb. · 168 Seiten · ISBN 3-540-55025-9
- 42 *Burger, C.*  
**Produktionsregelung mit entscheidungsunterstützenden Informationssystemen**  
1992 · 94 Abb. · 186 Seiten · ISBN 3-540-55187-5
- 43 *Hoßmann, J.*  
**Methodik zur Planung der automatischen Montage von nicht formstabilen Bauteilen**  
1992 · 73 Abb. · 168 Seiten · ISBN 3-540-5520-0
- 44 *Petry, M.*  
**Systematik zur Entwicklung eines modularen Programmbaukastens für robotergeführte Klebprozesse**  
1992 · 106 Abb. · 139 Seiten · ISBN 3-540-55374-6
- 45 *Schönecker, W.*  
**Integrierte Diagnose in Produktionszellen**  
1992 · 87 Abb. · 159 Seiten · ISBN 3-540-55375-4
- 46 *Bick, W.*  
**Systematische Planung hybrider Montagesysteme unter Berücksichtigung der Ermittlung des optimalen Automatisierungsgrades**  
1992 · 70 Abb. · 156 Seiten · ISBN 3-540-55377-0
- 47 *Gebauer, L.*  
**Prozeßuntersuchungen zur automatisierten Montage von optischen Linsen**  
1992 · 84 Abb. · 150 Seiten · ISBN 3-540-55378-9
- 48 *Schröder, N.*  
**Erstellung eines 3D-Simulationssystems zur Reduzierung von Rüstzeiten bei der NC-Bearbeitung**  
1992 · 103 Abb. · 161 Seiten · ISBN 3-540-55431-9
- 49 *Wiesbacher, J.*  
**Methoden zur rationellen Automatisierung der Montage von Schnellbefestigungselementen**  
1992 · 77 Abb. · 176 Seiten · ISBN 3-540-55512-9
- 50 *Garnich, F.*  
**Laserbearbeitung mit Robotern**  
1992 · 110 Abb. · 184 Seiten · ISBN 3-540-55513-7
- 51 *Eubert, P.*  
**Digitale Zustandsregelung elektrischer Vorschubantriebe**  
1992 · 89 Abb. · 159 Seiten · ISBN 3-540-44441-2
- 52 *Gleas, W.*  
**Rechnerintegrierte Kabelsatzfertigung**  
1992 · 67 Abb. · 140 Seiten · ISBN 3-540-55749-0
- 53 *Helml, H.J.*  
**Ein Verfahren zur On-Line Fehlererkennung und Diagnose**  
1992 · 60 Abb. · 153 Seiten · ISBN 3-540-55750-4
- 54 *Lang, Ch.*  
**Wissensbasierte Unterstützung der Verfügbarkeitsplanung**  
1992 · 75 Abb. · 150 Seiten · ISBN 3-540-55751-2
- 55 *Schuster, G.*  
**Rechnergestütztes Planungssystem für die flexibel automatisierte Montage**  
1992 · 67 Abb. · 135 Seiten · ISBN 3-540-55803-6
- 56 *Bomm, H.*  
**Ein Ziel- und Kennzahlensystem zum Investitioncontrolling komplexer Produktionssysteme**  
1992 · 87 Abb. · 195 Seiten · ISBN 3-540-55964-7
- 57 *Wendt, A.*  
**Qualitätssicherung in flexibel automatisierten Montagesystemen**  
1992 · 74 Abb. · 179 Seiten · ISBN 3-540-56044-0
- 58 *Hansmaier, H.*  
**Rechnergestütztes Verfahren zur Geräuschminderung**  
1993 · 67 Abb. · 156 Seiten · ISBN 3-540-56053-2
- 59 *Dilling, U.*  
**Planung von Fertigungssystemen unterstützt durch Wirtschaftssimulationen**  
1993 · 72 Abb. · 146 Seiten · ISBN 3-540-56307-5



- 60 *Strohmayer, R.*  
Rechnergestützte Auswahl und Konfiguration von  
Zubringeeinrichtungen  
1993 · 80 Abb. · 152 Seiten · ISBN 3-540-56652-X
- 61 *Glas, J.*  
Standardisierter Aufbau anwendungsspezifischer  
Zellenrechnersoftware  
1993 · 80 Abb. · 145 Seiten · ISBN 3-540-56890-5
- 62 *Stetter, R.*  
Rechnergestützte Simulationswerkzeuge zur  
Effizienzsteigerung des Industrieroboteinsatzes  
1994 · 91 Abb. · 146 Seiten · ISBN 3-540-56889-1
- 63 *Dirndorfer, A.*  
Robotersysteme zur förderbandsynchronen Montage  
1993 · 76 Abb. · 144 Seiten · ISBN 3-540-57031-4
- 64 *Wiedemann, M.*  
Simulation des Schwingungsverhaltens spanender  
Werkzeugmaschinen  
1993 · 81 Abb. · 137 Seiten · ISBN 3-540-57177-9
- 65 *Woenckhaus, Ch.*  
Rechnergestütztes System zur automatisierten 3D-  
Layoutoptimierung  
1994 · 81 Abb. · 140 Seiten · ISBN 3-540-57284-8
- 66 *Kummetsteiner, G.*  
3D-Bewegungssimulation als integratives Hilfsmittel zur  
Planung manueller Montagesysteme  
1994 · 62 Abb. · 146 Seiten · ISBN 3-540-57535-9
- 67 *Kugelman, F.*  
Einsatz nachgiebiger Elemente zur wirtschaftlichen  
Automatisierung von Produktionssystemen  
1993 · 76 Abb. · 144 Seiten · ISBN 3-540-57549-9
- 68 *Schwarz, H.*  
Simulationsgestützte CAD/CAM-Kopplung für die 3D-  
Laserbearbeitung mit integrierter Sensorik  
1994 · 96 Abb. · 148 Seiten · ISBN 3-540-57577-4
- 69 *Viethen, U.*  
Systematik zum Prüfen in flexiblen Fertigungssystemen  
1994 · 70 Abb. · 142 Seiten · ISBN 3-540-57794-7
- 70 *Seehuber, M.*  
Automatische Inbetriebnahme  
geschwindigkeitsadaptiver Zustandsregler  
1994 · 72 Abb. · 155 Seiten · ISBN 3-540-57896-X
- 71 *Amann, W.*  
Eine Simulationsumgebung für Planung und Betrieb von  
Produktionssystemen  
1994 · 71 Abb. · 129 Seiten · ISBN 3-540-57924-9
- 72 *Schöpf, M.*  
Rechnergestütztes Projektinformations- und  
Koordinationssystem für das Fertigungsvorfeld  
1997 · 63 Abb. · 130 Seiten · ISBN 3-540-58052-2
- 73 *Welling, A.*  
Effizienter Einsatz bildgebender Sensoren zur  
Flexibilisierung automatisierter Handhabungsvorgänge  
1994 · 66 Abb. · 139 Seiten · ISBN 3-540-580-0
- 74 *Zetlmayer, H.*  
Verfahren zur simulationsgestützten  
Produktionsregelung in der Einzel- und  
Kleinserienproduktion  
1994 · 62 Abb. · 143 Seiten · ISBN 3-540-58134-0
- 75 *Lindl, M.*  
Auftragsleittechnik für Konstruktion und Arbeitsplanung  
1994 · 66 Abb. · 147 Seiten · ISBN 3-540-58221-5
- 76 *Zipper, B.*  
Das integrierte Betriebsmittelwesen - Baustein einer  
flexiblen Fertigung  
1994 · 64 Abb. · 147 Seiten · ISBN 3-540-58222-3
- 77 *Rath, P.*  
Programmierung und Simulation von Zellenabläufen in  
der Arbeitsvorbereitung  
1995 · 51 Abb. · 130 Seiten · ISBN 3-540-58223-1
- 78 *Engel, A.*  
Strömungstechnische Optimierung von  
Produktionssystemen durch Simulation  
1994 · 69 Abb. · 160 Seiten · ISBN 3-540-58258-4
- 79 *Zah, M. F.*  
Dynamisches Prozeßmodell Kreissägen  
1995 · 95 Abb. · 186 Seiten · ISBN 3-540-58624-5
- 80 *Zwenzer, N.*  
Technologisches Prozeßmodell für die  
Kugelschleifbearbeitung  
1995 · 65 Abb. · 150 Seiten · ISBN 3-540-58634-2
- 81 *Romanow, P.*  
Konstruktionsbegleitende Kalkulation von  
Werkzeugmaschinen  
1995 · 66 Abb. · 151 Seiten · ISBN 3-540-58771-3
- 82 *Kahlenberg, R.*  
Integrierte Qualitätssicherung in flexiblen  
Fertigungszellen  
1995 · 71 Abb. · 136 Seiten · ISBN 3-540-58772-1
- 83 *Huber, A.*  
Arbeitsfolgenplanung mehrstufiger Prozesse in der  
Hartbearbeitung  
1995 · 87 Abb. · 152 Seiten · ISBN 3-540-58773-X
- 84 *Birkel, G.*  
Aufwandsminimierter Wissenserwerb für die Diagnose in  
flexiblen Produktionszellen  
1995 · 64 Abb. · 137 Seiten · ISBN 3-540-58869-8
- 85 *Simon, D.*  
Fertigungsregelung durch zielgrößenorientierte Planung  
und logistisches Störungsmanagement  
1995 · 77 Abb. · 132 Seiten · ISBN 3-540-58942-2
- 86 *Nedeljkovic-Groha, V.*  
Systematische Planung anwendungsspezifischer  
Materialflußsteuerungen  
1995 · 94 Abb. · 188 Seiten · ISBN 3-540-58953-8
- 87 *Rockland, M.*  
Flexibilisierung der automatischen Teilbereitstellung in  
Montageanlagen  
1995 · 83 Abb. · 168 Seiten · ISBN 3-540-58989-6
- 88 *Linner, St.*  
Konzept einer integrierten Produktentwicklung  
1995 · 67 Abb. · 168 Seiten · ISBN 3-540-59016-1
- 89 *Eder, Th.*  
Integrierte Planung von Informationssystemen für  
rechnergestützte Produktionssysteme  
1995 · 62 Abb. · 150 Seiten · ISBN 3-540-59084-6
- 90 *Deutsche, U.*  
Prozeßorientierte Organisation der Auftragsentwicklung  
in mittelständischen Unternehmen  
1995 · 80 Abb. · 188 Seiten · ISBN 3-540-59337-3
- 91 *Dieterle, A.*  
Recyclingintegrierte Produktentwicklung  
1995 · 68 Abb. · 146 Seiten · ISBN 3-540-60120-1

- 92 *Hechl, Chr.*  
**Personalorientierte Montageplanung für komplexe und  
variantenreiche Produkte**  
1995 · 73 Abb. · 158 Seiten · ISBN 3-540-60325-5
- 93 *Albertz, F.*  
**Dynamikgerechter Entwurf von Werkzeugmaschinen ·  
Gestellstrukturen**  
1995 · 83 Abb. · 156 Seiten · ISBN 3-540-60608-8
- 94 *Trunzer, W.*  
**Strategien zur On-Line Bahnplanung bei Robotern mit  
3D-Konturfolgesensoren**  
1996 · 101 Abb. · 164 Seiten · ISBN 3-540-60961-X
- 95 *Fichtmüller, N.*  
**Rationalisierung durch flexible, hybride Montagesysteme**  
1996 · 83 Abb. · 145 Seiten · ISBN 3-540-60960-1
- 96 *Trucks, V.*  
**Rechnergestützte Beurteilung von Getriebestrukturen in  
Werkzeugmaschinen**  
1996 · 64 Abb. · 141 Seiten · ISBN 3-540-60599-8
- 97 *Schäffer, G.*  
**Systematische Integration adaptiver  
Produktionssysteme**  
1996 · 71 Abb. · 170 Seiten · ISBN 3-540-60958-X
- 98 *Koch, M. R.*  
**Autonome Fertigungszellen · Gestaltung, Steuerung und  
integrierte Störungsbehandlung**  
1996 · 67 Abb. · 138 Seiten · ISBN 3-540-61104-5
- 99 *Moctezuma de la Barrera, J.L.*  
**Ein durchgängiges System zur computer- und  
rechnergestützten Chirurgie**  
1996 · 99 Abb. · 175 Seiten · ISBN 3-540-61145-2
- 100 *Geuer, A.*  
**Einsatzpotential des Rapid Prototyping in der  
Produktentwicklung**  
1996 · 84 Abb. · 154 Seiten · ISBN 3-540-61495-8
- 101 *Ebner, C.*  
**Ganzheitliches Verfügbarkeits- und Qualitätsmanagement  
unter Verwendung von Felddaten**  
1996 · 67 Abb. · 132 Seiten · ISBN 3-540-61678-0
- 102 *Pischelsrieder, K.*  
**Steuerung autonomer mobiler Roboter in der Produktion**  
1996 · 74 Abb. · 171 Seiten · ISBN 3-540-61714-0
- 103 *Köhler, R.*  
**Disposition und Materialbereitstellung bei komplexen  
variantenreichen Kleinprodukten**  
1997 · 62 Abb. · 177 Seiten · ISBN 3-540-62024-9
- 104 *Feldmann, Ch.*  
**Eine Methode für die integrierte rechnergestützte  
Montageplanung**  
1997 · 71 Abb. · 163 Seiten · ISBN 3-540-62059-1
- 105 *Lehmann, H.*  
**Integrierte Materialfluß- und Layoutplanung durch  
Kopplung von CAD- und Ablaufsimulationssystem**  
1997 · 96 Abb. · 191 Seiten · ISBN 3-540-62202-0
- 106 *Wagner, M.*  
**Steuerungintegrierte Fehlerbehandlung für  
maschinennahe Abläufe**  
1997 · 94 Abb. · 164 Seiten · ISBN 3-540-62656-5
- 107 *Lorenzen, J.*  
**Simulationsgestützte Kostenanalyse in  
produktorientierten Fertigungsstrukturen**  
1997 · 63 Abb. · 129 Seiten · ISBN 3-540-62794-4
- 108 *Krönert, U.*  
**Systematik für die rechnergestützte Ähnlichkeitssuche  
und Standardisierung**  
1997 · 53 Abb. · 127 Seiten · ISBN 3-540-63338-3
- 109 *Pfersdorf, I.*  
**Entwicklung eines systematischen Vorgehens zur  
Organisation des industriellen Service**  
1997 · 74 Abb. · 172 Seiten · ISBN 3-540-63615-3
- 110 *Kuba, R.*  
**Informations- und kommunikationstechnische  
Integration von Menschen in der Produktion**  
1997 · 77 Abb. · 155 Seiten · ISBN 3-540-63642-0
- 111 *Kaiser, J.*  
**Vernetztes Gestalten von Produkt und  
Produktionsprozeß mit Produktmodellen**  
1997 · 67 Abb. · 139 Seiten · ISBN 3-540-63999-3
- 112 *Geyer, M.*  
**Flexibles Planungssystem zur Berücksichtigung  
ergonomischer Aspekte bei der Produkt- und  
Arbeitssystemgestaltung**  
1997 · 85 Abb. · 154 Seiten · ISBN 3-540-64195-5
- 113 *Martin, C.*  
**Produktionsregelung · ein modularer, modellbasierter  
Ansatz**  
1998 · 73 Abb. · 162 Seiten · ISBN 3-540-64401-6
- 114 *Löffler, Th.*  
**Akustische Überwachung automatisierter Fügeprozesse**  
1998 · 85 Abb. · 136 Seiten · ISBN 3-540-64511-X
- 115 *Lindermaier, R.*  
**Qualitätsorientierte Entwicklung von Montagesystemen**  
1998 · 84 Abb. · 164 Seiten · ISBN 3-540-64686-8
- 116 *Koehrer, J.*  
**Prozeßorientierte Teamstrukturen in Betrieben mit  
Großserienfertigung**  
1998 · 75 Abb. · 185 Seiten · ISBN 3-540-65037-7
- 117 *Schuller, R. W.*  
**Leitfaden zum automatisierten Auftrag von  
hochviskosen Dichtmassen**  
1999 · 76 Abb. · 162 Seiten · ISBN 3-540-65320-1
- 118 *Debuschewitz, M.*  
**Integrierte Methodik und Werkzeuge zur  
herstellungorientierten Produktentwicklung**  
1999 · 104 Abb. · 169 Seiten · ISBN 3-540-65350-3
- 119 *Bauer, L.*  
**Strategien zur rechnergestützten Offline-  
Programmierung von 3D-Laseranlagen**  
1999 · 98 Abb. · 145 Seiten · ISBN 3-540-65382-1
- 120 *Pflob, E.*  
**Modellgestützte Arbeitsplanung bei  
Fertigungsmaschinen**  
1999 · 69 Abb. · 154 Seiten · ISBN 3-540-65525-5
- 121 *Spitznagel, J.*  
**Erfahrungsgeleitete Planung von Laseranlagen**  
1999 · 63 Abb. · 156 Seiten · ISBN 3-540-65896-3

# Seminarberichte iw b

herausgegeben von Prof. Dr.-Ing. Gunther Reinhart und Prof. Dr.-Ing. Michael Zäh,  
Institut für Werkzeugmaschinen und Betriebswissenschaften  
der Technischen Universität München

Seminarberichte iw b sind erhältlich im Buchhandel oder beim  
Herbert Utz Verlag, München, Fax 089-277791-01, info@utz.de

- 1 Innovative Montagesysteme · Anlagengestaltung, -bewertung und -überwachung  
115 Seiten · ISBN 3-931327-01-9
- 2 Integriertes Produktmodell · Von der Idee zum fertigen Produkt  
82 Seiten · ISBN 3-931327-02-7
- 3 Konstruktion von Werkzeugmaschinen · Berechnung, Simulation und Optimierung  
110 Seiten · ISBN 3-931327-03-5
- 4 Simulation · Einsatzmöglichkeiten und Erfahrungsberichte  
134 Seiten · ISBN 3-931327-04-3
- 5 Optimierung der Kooperation in der Produktentwicklung  
95 Seiten · ISBN 3-931327-05-1
- 6 Materialbearbeitung mit Laser · von der Planung zur Anwendung  
86 Seiten · ISBN 3-931327-76-0
- 7 Dynamisches Verhalten von Werkzeugmaschinen  
80 Seiten · ISBN 3-931327-77-9
- 8 Qualitätsmanagement · der Weg ist das Ziel  
130 Seiten · ISBN 3-931327-78-7
- 9 Installationstechnik an Werkzeugmaschinen · Analysen und Konzepte  
120 Seiten · ISBN 3-931327-79-5
- 10 3D-Simulation · Schneller, sicherer und kostengünstiger zum Ziel  
90 Seiten · ISBN 3-931327-10-8
- 11 Unternehmensorganisation · Schlüssel für eine effiziente Produktion  
110 Seiten · ISBN 3-931327-11-6
- 12 Autonome Produktionssysteme  
100 Seiten · ISBN 3-931327-12-4
- 13 Planung von Montageanlagen  
130 Seiten · ISBN 3-931327-13-2
- 14 Nicht erschienen – wird nicht erscheinen
- 15 Flexible Fluide Kleb/Dichtstoffe · Dosierung und Prozeßgestaltung  
80 Seiten · ISBN 3-931327-15-9
- 16 Time to Market · Von der Idee zum Produktionsstart  
80 Seiten · ISBN 3-931327-16-7
- 17 Industriekeramik in Forschung und Praxis · Probleme, Analysen und Lösungen  
80 Seiten · ISBN 3-931327-17-5
- 18 Das Unternehmen im Internet · Chancen für produzierende Unternehmen  
165 Seiten · ISBN 3-931327-18-3
- 19 Leittechnik und Informationslogistik · mehr Transparenz in der Fertigung  
85 Seiten · ISBN 3-931327-19-1
- 20 Dezentrale Steuerungen in Produktionsanlagen · Plug & Play · Vereinfachung von Entwicklung und Inbetriebnahme  
105 Seiten · ISBN 3-931327-20-5
- 21 Rapid Prototyping · Rapid Tooling · Schnell zu funktionalen Prototypen  
95 Seiten · ISBN 3-931327-21-3
- 22 Mikrotechnik für die Produktion · Greifbare Produkte und Anwendungspotentiale  
95 Seiten · ISBN 3-931327-22-1
- 24 EDM Engineering Data Management  
195 Seiten · ISBN 3-931327-24-8
- 25 Rationelle Nutzung der Simulationstechnik · Entwicklungstrends und Praxisbeispiele  
152 Seiten · ISBN 3-931327-25-6
- 26 Alternative Dichtungssysteme · Konzepte zur Dichtungsmontage und zum Dichtmittelauftrag  
110 Seiten · ISBN 3-931327-26-4
- 27 Rapid Prototyping · Mit neuen Technologien schnell vom Entwurf zum Serienprodukt  
111 Seiten · ISBN 3-931327-27-2
- 28 Rapid Tooling · Mit neuen Technologien schnell vom Entwurf zum Serienprodukt  
154 Seiten · ISBN 3-931327-28-0
- 29 Installationstechnik an Werkzeugmaschinen · Abschlußseminar  
156 Seiten · ISBN 3-931327-29-9
- 30 Nicht erschienen – wird nicht erscheinen
- 31 Engineering Data Management (EDM) · Erfahrungsberichte und Trends  
183 Seiten · ISBN 3-931327-31-0
- 32 Nicht erschienen – wird nicht erscheinen
- 33 3D-CAD · Mehr als nur eine dritte Dimension  
181 Seiten · ISBN 3-931327-33-7
- 34 Laser in der Produktion · Technologische Randbedingungen für den wirtschaftlichen Einsatz  
102 Seiten · ISBN 3-931327-34-5
- 35 Ablaufsimulation · Anlagen effizient und sicher planen und betreiben  
129 Seiten · ISBN 3-931327-35-3
- 36 Moderne Methoden zur Montageplanung · Schlüssel für eine effiziente Produktion  
124 Seiten · ISBN 3-931327-36-1
- 37 Wettbewerbsfaktor Verfügbarkeit · Produktivitätsteigerung durch technische und organisatorische Ansätze  
95 Seiten · ISBN 3-931327-37-X
- 38 Rapid Prototyping · Effizienter Einsatz von Modellen in der Produktentwicklung  
128 Seiten · ISBN 3-931327-38-8
- 39 Rapid Tooling · Neue Strategien für den Werkzeug- und Formenbau  
130 Seiten · ISBN 3-931327-39-6
- 40 Erfolgreich kooperieren in der produzierenden Industrie · Flexibler und schneller mit modernen Kooperationen  
160 Seiten · ISBN 3-931327-40-X
- 41 Innovative Entwicklung von Produktionsmaschinen  
146 Seiten · ISBN 3-89675-041-0
- 42 Stückzahlflexible Montagesysteme  
139 Seiten · ISBN 3-89675-042-9
- 43 Produktivität und Verfügbarkeit · ...durch Kooperation steigern  
120 Seiten · ISBN 3-89675-043-7
- 44 Automatisierte Mikromontage · Handhaben und Positionieren von Mikrobautteilen  
125 Seiten · ISBN 3-89675-044-5
- 45 Produzieren in Netzwerken · Lösungsansätze, Methoden, Praxisbeispiele  
173 Seiten · ISBN 3-89675-045-3
- 46 Virtuelle Produktion · Ablaufsimulation  
108 Seiten · ISBN 3-89675-046-1

- 47 Virtuelle Produktion · Prozeß- und Produktsimulation  
131 Seiten · ISBN 3-89675-047-X
- 48 Sicherheitstechnik an Werkzeugmaschinen  
106 Seiten · ISBN 3-89675-048-8
- 49 Rapid Prototyping · Methoden für die reaktionsfähige Produktentwicklung  
150 Seiten · ISBN 3-89675-049-6
- 50 Rapid Manufacturing · Methoden für die reaktionsfähige Produktion  
121 Seiten · ISBN 3-89675-050-X
- 51 Flexibles Kleben und Dichten · Produkt- & Prozeßgestaltung, Mischverbindungen, Qualitätskontrolle  
137 Seiten · ISBN 3-89675-051-8
- 52 Rapid Manufacturing · Schnelle Herstellung von Klein- und Prototypenserien  
124 Seiten · ISBN 3-89675-052-6
- 53 Mischverbindungen · Werkstoffauswahl, Verfahrensauswahl, Umsetzung  
107 Seiten · ISBN 3-89675-054-2
- 54 Virtuelle Produktion · Integrierte Prozess- und Produktsimulation  
133 Seiten · ISBN 3-89675-054-2
- 55 e-Business in der Produktion · Organisationskonzepte, IT-Lösungen, Praxisbeispiele  
150 Seiten · ISBN 3-89675-055-0
- 56 Virtuelle Produktion – Ablaufsimulation als planungsbegleitendes Werkzeug  
150 Seiten · ISBN 3-89675-056-9
- 57 Virtuelle Produktion – Datenintegration und Benutzerschnittstellen  
150 Seiten · ISBN 3-89675-057-7
- 58 Rapid Manufacturing · Schnelle Herstellung qualitativ hochwertiger Bauteile oder Kleinserien  
169 Seiten · ISBN 3-89675-058-7
- 59 Automatisierte Mikromontage · Werkzeuge und Fügetechnologien für die Mikrosystemtechnik  
114 Seiten · ISBN 3-89675-059-3
- 60 Mechatronische Produktionssysteme · Genauigkeit gezielt entwickeln  
131 Seiten · ISBN 3-89675-060-7
- 61 Nicht erschienen – wird nicht erscheinen
- 62 Rapid Technologien · Anspruch – Realität – Technologien  
100 Seiten · ISBN 3-89675-062-3
- 63 Fabrikplanung 2002 · Visionen – Umsetzung – Werkzeuge  
124 Seiten · ISBN 3-89675-063-1
- 64 Mischverbindungen · Einsatz und Innovationspotenzial  
143 Seiten · ISBN 3-89675-064-X
- 65 Fabrikplanung 2003 – Basis für Wachstum · Erfahrungen Werkzeuge Visionen  
136 Seiten · ISBN 3-89675-065-8
- 66 Mit Rapid Technologien zum Aufschwung · Neue Rapid Technologien und Verfahren, Neue Qualitäten, Neue Möglichkeiten, Neue Anwendungsfelder  
185 Seiten · ISBN 3-89675-066-6
- 67 Mechatronische Produktionssysteme · Die Virtuelle Werkzeugmaschine: Mechatronisches Entwicklungsvorgehen, Integrierte Modellbildung, Applikationsfelder  
148 Seiten · ISBN 3-89675-067-4
- 68 Virtuelle Produktion · Nutzenpotenziale im Lebenszyklus der Fabrik  
139 Seiten · ISBN 3-89675-068-2
- 69 Kooperationsmanagement in der Produktion · Visionen und Methoden zur Kooperation – Geschäftsmodelle und Rechtsformen für die Kooperation – Kooperation entlang der Wertschöpfungskette  
134 Seiten · ISBN 3-89675-069-0
- 70 Mechatronik · Strukturdynamik von Werkzeugmaschinen  
161 Seiten · ISBN 3-89675-070-4
- 71 Klebtechnik · Zerstörungsfreie Qualitätssicherung beim flexibel automatisierten Kleben und Dichten  
ISBN 3-89675-071-2 · vergriffen
- 72 Fabrikplanung 2004 · Erfolgsfaktor im Wettbewerb · Erfahrungen – Werkzeuge – Visionen  
ISBN 3-89675-072-0 · vergriffen
- 73 Rapid Manufacturing Vom Prototyp zur Produktion · Erwartungen – Erfahrungen – Entwicklungen  
179 Seiten · ISBN 3-89675-073-9
- 74 Virtuelle Produktionssystemplanung · Virtuelle Inbetriebnahme und Digitale Fabrik  
133 Seiten · ISBN 3-89675-074-7
- 75 Nicht erschienen – wird nicht erscheinen
- 76 Berührungslose Handhabung · Vom Wafer zur Glaslinse, von der Kapselfür aseptischen Ampulle  
95 Seiten · ISBN 3-89675-076-3
- 77 ERP-Systeme · Einführung in die betriebliche Praxis · Erfahrungen, Best Practices, Visionen  
153 Seiten · ISBN 3-89675-077-7
- 78 Mechatronik · Trends in der interdisziplinären Entwicklung von Werkzeugmaschinen  
155 Seiten · ISBN 3-89675-078-X
- 79 Produktionsmanagement  
267 Seiten · ISBN 3-89675-079-8
- 80 Rapid Manufacturing · Fertigungsverfahren für alle Ansprüche  
154 Seiten · ISBN 3-89675-080-1
- 81 Rapid Manufacturing · Heutige Trends – Zukünftige Anwendungsfelder  
172 Seiten · ISBN 3-89675-081-X
- 82 Produktionsmanagement · Herausforderung Variantenmanagement  
100 Seiten · ISBN 3-89675-082-8
- 83 Mechatronik · Optimierungspotenzial der Werkzeugmaschine nutzen  
160 Seiten · ISBN 3-89675-083-6
- 84 Virtuelle Inbetriebnahme · Von der Kür zur Pflicht?  
104 Seiten · ISBN 978-3-89675-084-6
- 85 3D-Erfahrungsforum · Innovation im Werkzeug- und Formenbau  
375 Seiten · ISBN 978-3-89675-085-3
- 86 Rapid Manufacturing · Erfolgreich produzieren durch innovative Fertigung  
162 Seiten · ISBN 978-3-89675-086-0
- 87 Produktionsmanagement · Schlank im Mittelstand  
102 Seiten · ISBN 978-3-89675-087-7
- 88 Mechatronik · Vorsprung durch Simulation  
134 Seiten · ISBN 978-3-89675-088-4
- 89 RFID in der Produktion · Wertschöpfung effizient gestalten  
122 Seiten · ISBN 978-3-89675-089-1

# Forschungsberichte iwb

herausgegeben von Prof. Dr.-Ing. Gunther Reinhart und Prof. Dr.-Ing. Michael Zäh,  
Institut für Werkzeugmaschinen und Betriebswissenschaften  
der Technischen Universität München

Forschungsberichte iwb ab Band 122 sind erhältlich im Buchhandel oder beim  
Herbert Utz Verlag, München, Fax 089-277791-01, info@utz.de

- 122 Schneider, Burghard  
**Prozesskettenorientierte Bereitstellung nicht formstabiler Bauteile**  
1999 · 183 Seiten · 98 Abb. · 14 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-559-5
- 123 Goldstein, Bernd  
**Modellgestützte Geschäftsprozeßgestaltung in der Produktentwicklung**  
1999 · 170 Seiten · 65 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-546-3
- 124 Moßmer, Helmut E.  
**Methode zur simulationsbasierten Regelung zeitvarianter Produktionssysteme**  
1999 · 164 Seiten · 67 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-585-4
- 125 Gräser, Ralf-Gunter  
**Ein Verfahren zur Kompensation temperaturinduzierter Verformungen an Industrierobotern**  
1999 · 167 Seiten · 63 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-603-6
- 126 Trossin, Hans-Jürgen  
**Nutzung der Ähnlichkeitstheorie zur Modellbildung in der Produktionstechnik**  
1999 · 162 Seiten · 75 Abb. · 11 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-614-1
- 127 Kugelman, Doris  
**Aufgabenorientierte Offline-Programmierung von Industrierobotern**  
1999 · 168 Seiten · 68 Abb. · 2 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-615-X
- 128 Diesch, Rolf  
**Steigerung der organisatorischen Verfügbarkeit von Fertigungszellen**  
1999 · 160 Seiten · 69 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-618-4
- 129 Lulay, Werner E.  
**Hybrid-hierarchische Simulationsmodelle zur Koordination teilautonomer Produktionsstrukturen**  
1999 · 182 Seiten · 51 Abb. · 14 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-620-6
- 130 Murr, Otto  
**Adaptive Planung und Steuerung von integrierten Entwicklungs- und Planungsprozessen**  
1999 · 178 Seiten · 85 Abb. · 3 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-636-2
- 131 Macht, Michael  
**Ein Vorgehensmodell für den Einsatz von Rapid Prototyping**  
1999 · 170 Seiten · 87 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-638-9
- 132 Mehler, Bruno H.  
**Aufbau virtueller Fabriken aus dezentralen Partnerverbünden**  
1999 · 152 Seiten · 44 Abb. · 27 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-645-1
- 133 Heitmann, Knut  
**Sichere Prognosen für die Produktionsptimierung mittels stochastischer Modelle**  
1999 · 146 Seiten · 60 Abb. · 13 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-675-3
- 134 Blessing, Stefan  
**Gestaltung der Materialflußsteuerung in dynamischen Produktionsstrukturen**  
1999 · 160 Seiten · 67 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-690-7
- 135 Abay, Can  
**Numerische Optimierung multivariater mehrstufiger Prozesse am Beispiel der Hartbearbeitung von Industriekeramik**  
2000 · 159 Seiten · 46 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-697-4

- 136 Brandner, Stefan  
**Integriertes Produktdaten- und Prozeßmanagement in virtuellen Fabriken**  
2000 · 172 Seiten · 61 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-715-6
- 137 Hirschberg, Arnd G.  
**Verbindung der Produkt- und Funktionsorientierung in der Fertigung**  
2000 · 165 Seiten · 49 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-729-6
- 138 Reek, Alexandra  
**Strategien zur Fokuspositionierung beim Laserstrahlschweißen**  
2000 · 193 Seiten · 103 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-730-X
- 139 Sabbah, Khalid-Alexander  
**Methodische Entwicklung störungstoleranter Steuerungen**  
2000 · 148 Seiten · 75 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-739-3
- 140 Schliffenbacher, Klaus U.  
**Konfiguration virtueller Wertschöpfungsketten in dynamischen, heterarchischen Kompetenznetzwerken**  
2000 · 187 Seiten · 70 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-754-7
- 141 Sprengel, Andreas  
**Integrierte Kostenkalkulationsverfahren für die Werkzeugmaschinenentwicklung**  
2000 · 144 Seiten · 55 Abb. · 6 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-757-1
- 142 Gallasch, Andreas  
**Informationstechnische Architektur zur Unterstützung des Wandels in der Produktion**  
2000 · 150 Seiten · 69 Abb. · 6 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-781-4
- 143 Cuiper, Ralf  
**Durchgängige rechnergestützte Planung und Steuerung von automatisierten Montagevorgängen**  
2000 · 168 Seiten · 75 Abb. · 3 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-783-0
- 144 Schneider, Christian  
**Strukturmechanische Berechnungen in der Werkzeugmaschinenkonstruktion**  
2000 · 180 Seiten · 66 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-789-X
- 145 Jonas, Christian  
**Konzept einer durchgängigen, rechnergestützten Planung von Montageanlagen**  
2000 · 183 Seiten · 82 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-870-5
- 146 Willnecker, Ulrich  
**Gestaltung und Planung leistungsorientierter manueller Fließmontagen**  
2001 · 175 Seiten · 67 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-891-8
- 147 Lehner, Christof  
**Beschreibung des Nd:Yag-Laserstrahlschweißprozesses von Magnesiumdruckguss**  
2001 · 205 Seiten · 94 Abb. · 24 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0004-X
- 148 Rick, Frank  
**Simulationsgestützte Gestaltung von Produkt und Prozess am Beispiel Laserstrahlschweißen**  
2001 · 145 Seiten · 57 Abb. · 2 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0008-2
- 149 Höhn, Michael  
**Sensorgeführte Montage hybrider Mikrosysteme**  
2001 · 171 Seiten · 74 Abb. · 7 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0012-0
- 150 Bohl, Jörn  
**Wissensmanagement im Klein- und mittelständischen Unternehmen der Einzel- und Kleinserienfertigung**  
2001 · 179 Seiten · 88 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0020-1
- 151 Bürgel, Robert  
**Prozessanalyse an spanenden Werkzeugmaschinen mit digital geregelten Antrieben**  
2001 · 185 Seiten · 60 Abb. · 10 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0021-X
- 152 Stephan Dürrschmidt  
**Planung und Betrieb wandlungsfähiger Logistiksysteme in der variantenreichen Serienproduktion**  
2001 · 914 Seiten · 61 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0023-6
- 153 Bernhard Eich  
**Methode zur prozesskettenorientierten Planung der Teilebereitstellung**  
2001 · 132 Seiten · 48 Abb. · 6 Tabellen · 20,5 x 14,5 cm · ISBN 3-8316-0028-7

- 154 Wolfgang Rudorfer  
**Eine Methode zur Qualifizierung von produzierenden Unternehmen für Kompetenznetzwerke**  
 2001 · 207 Seiten · 89 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0037-6
- 155 Hans Meier  
**Verteilte kooperative Steuerung maschinennaher Abläufe**  
 2001 · 162 Seiten · 85 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0044-9
- 156 Gerhard Nowak  
**Informationstechnische Integration des industriellen Service in das Unternehmen**  
 2001 · 203 Seiten · 95 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0055-4
- 157 Martin Werner  
**Simulationsgestützte Reorganisation von Produktions- und Logistikprozessen**  
 2001 · 191 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0058-9
- 158 Bernhard Lenz  
**Finite Elemente-Modellierung des Laserstrahlschweißens für den Einsatz in der Fertigungsplanung**  
 2001 · 150 Seiten · 47 Abb. · 5 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0094-5
- 159 Stefan Grunwald  
**Methode zur Anwendung der flexiblen integrierten Produktentwicklung und Montageplanung**  
 2002 · 206 Seiten · 80 Abb. · 25 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0095-3
- 160 Josef Gartner  
**Qualitätssicherung bei der automatisierten Applikation hochviskoser Dichtungen**  
 2002 · 165 Seiten · 74 Abb. · 21 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0096-1
- 161 Wolfgang Zeller  
**Gesamtheitliches Sicherheitskonzept für die Antriebs- und Steuerungstechnik bei Werkzeugmaschinen**  
 2002 · 192 Seiten · 54 Abb. · 15 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0100-3
- 162 Michael Loferer  
**Rechnergestützte Gestaltung von Montagesystemen**  
 2002 · 178 Seiten · 80 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0118-6
- 163 Jörg Fahrer  
**Ganzheitliche Optimierung des indirekten Metall-Lasersinterprozesses**  
 2002 · 176 Seiten · 69 Abb. · 13 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0124-0
- 164 Jürgen Höppner  
**Verfahren zur berührungslosen Handhabung mittels leistungsstarker Schallwandler**  
 2002 · 132 Seiten · 24 Abb. · 3 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0125-9
- 165 Hubert Götte  
**Entwicklung eines Assistenzrobotersystems für die Knieendoprothetik**  
 2002 · 258 Seiten · 123 Abb. · 5 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0126-7
- 166 Martin Weissenberger  
**Optimierung der Bewegungsdynamik von Werkzeugmaschinen im rechnergestützten Entwicklungsprozess**  
 2002 · 210 Seiten · 86 Abb. · 2 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0138-0
- 167 Dirk Jacob  
**Verfahren zur Positionierung unterseitenstrukturierter Bauelemente in der Mikrosystemtechnik**  
 2002 · 200 Seiten · 82 Abb. · 24 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0142-9
- 168 Ulrich Roßgoderer  
**System zur effizienten Layout- und Prozessplanung von hybriden Montageanlagen**  
 2002 · 175 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0154-2
- 169 Robert Klingel  
**Anziehverfahren für hochfeste Schraubenverbindungen auf Basis akustischer Emissionen**  
 2002 · 164 Seiten · 89 Abb. · 27 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0174-7
- 170 Paul Jens Peter Ross  
**Bestimmung des wirtschaftlichen Automatisierungsgrades von Montageprozessen in der frühen Phase der Montageplanung**  
 2002 · 144 Seiten · 38 Abb. · 38 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0191-7
- 171 Stefan von Praun  
**Toleranzanalyse nachgiebiger Baugruppen im Produktentstehungsprozess**  
 2002 · 250 Seiten · 62 Abb. · 7 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0202-6

- 172 Florian von der Hagen  
**Gestaltung kurzfristiger und unternehmensübergreifender Engineering-Kooperationen**  
 2002 · 220 Seiten · 104 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0208-5
- 173 Oliver Kramer  
**Methode zur Optimierung der Wertschöpfungskette mittelständischer Betriebe**  
 2002 · 212 Seiten · 84 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0211-5
- 174 Winfried Dohmen  
**Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer Systeme**  
 2002 · 200 Seiten · 67 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0214-X
- 175 Oliver Anton  
**Ein Beitrag zur Entwicklung telepräsender Montagesysteme**  
 2002 · 158 Seiten · 85 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0215-8
- 176 Welf Broser  
**Methode zur Definition und Bewertung von Anwendungsfeldern für Kompetenznetzwerke**  
 2002 · 224 Seiten · 122 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0217-4
- 177 Frank Breiting  
**Ein ganzheitliches Konzept zum Einsatz des indirekten Metall-Lasersinterns für das Druckgießen**  
 2003 · 156 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0227-1
- 178 Johann von Pieverling  
**Ein Vorgehensmodell zur Auswahl von Konturfertigungsverfahren für das Rapid Tooling**  
 2003 · 163 Seiten · 88 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0230-1
- 179 Thomas Baudisch  
**Simulationsumgebung zur Auslegung der Bewegungsdynamik des mechatronischen Systems Werkzeugmaschine**  
 2003 · 190 Seiten · 87 Abb. · 8 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0249-2
- 180 Heinrich Schieferstein  
**Experimentelle Analyse des menschlichen Kausystems**  
 2003 · 132 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0251-4
- 181 Joachim Berlek  
**Methodik zur strukturierten Auswahl von Auftragsabwicklungssystemen**  
 2003 · 244 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0258-1
- 182 Christian Meierlohr  
**Konzept zur rechnergestützten Integration von Produktions- und Gebäudeplanung in der Fabrikgestaltung**  
 2003 · 181 Seiten · 84 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0292-1
- 183 Volker Weber  
**Dynamisches Kostenmanagement in kompetenzzentrierten Unternehmensnetzwerken**  
 2004 · 210 Seiten · 64 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0330-8
- 184 Thomas Bongardt  
**Methode zur Kompensation betriebsabhängiger Einflüsse auf die Absolutgenauigkeit von Industrierobotern**  
 2004 · 170 Seiten · 40 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0332-4
- 185 Tim Angerer  
**Effizienzsteigerung in der automatisierten Montage durch aktive Nutzung mechatronischer Produktkomponenten**  
 2004 · 180 Seiten · 67 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0336-7
- 186 Alexander Krüger  
**Planung und Kapazitätsabstimmung stückzahlflexibler Montagesysteme**  
 2004 · 197 Seiten · 83 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0371-5
- 187 Matthias Meindl  
**Beitrag zur Entwicklung generativer Fertigungsverfahren für das Rapid Manufacturing**  
 2005 · 222 Seiten · 97 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0465-7
- 188 Thomas Fusch  
**Betriebsbegleitende Prozessplanung in der Montage mit Hilfe der Virtuellen Produktion am Beispiel der Automobilindustrie**  
 2005 · 190 Seiten · 99 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0467-3



- 189 Thomas Mosandl  
**Qualitätssteigerung bei automatisiertem Klebstoffauftrag durch den Einsatz optischer Konturfolgesysteme**  
 2005 · 182 Seiten · 58 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0471-1
- 190 Christian Patron  
**Konzept für den Einsatz von Augmented Reality in der Montageplanung**  
 2005 · 150 Seiten · 61 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0474-6
- 191 Robert Cisek  
**Planung und Bewertung von Rekonfigurationsprozessen in Produktionssystemen**  
 2005 · 200 Seiten · 64 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0475-4
- 192 Florian Auer  
**Methode zur Simulation des Laserstrahlschweißens unter Berücksichtigung der Ergebnisse vorangegangener Umformsimulationen**  
 2005 · 160 Seiten · 65 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0485-1
- 193 Carsten Selke  
**Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung**  
 2005 · 137 Seiten · 53 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0495-9
- 194 Markus Seefried  
**Simulation des Prozessschrittes der Wärmebehandlung beim Indirekten-Metall-Lasersintern**  
 2005 · 216 Seiten · 82 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0503-3
- 195 Wolfgang Wagner  
**Fabrikplanung für die standortübergreifende Kostensenkung bei marktnaher Produktion**  
 2006 · 208 Seiten · 43 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0586-6
- 196 Christopher Ulrich  
**Erhöhung des Nutzungsgrades von Laserstrahlquellen durch Mehrfach-Anwendungen**  
 2006 · 178 Seiten · 74 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0590-4
- 197 Johann Hartl  
**Prozessgaseinfluss beim Schweißen mit Hochleistungsdiodenlasern**  
 2006 · 140 Seiten · 55 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0611-0
- 198 Bernd Hartmann  
**Die Bestimmung des Personalbedarfs für den Materialfluss in Abhängigkeit von Produktionsfläche und -menge**  
 2006 · 208 Seiten · 105 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0615-3
- 199 Michael Schilp  
**Auslegung und Gestaltung von Werkzeugen zum berührungslosen Greifen kleiner Bauteile in der Mikromontage**  
 2006 · 130 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0631-5
- 200 Florian Manfred Grätz  
**Teilautomatische Generierung von Stromlauf- und Fluidplänen für mechatronische Systeme**  
 2006 · 192 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0643-9
- 201 Dieter Eireiner  
**Prozessmodelle zur statischen Auslegung von Anlagen für das Friction Stir Welding**  
 2006 · 214 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0650-1
- 202 Gerhard Volkwein  
**Konzept zur effizienten Bereitstellung von Steuerungsfunktionalität für die NC-Simulation**  
 2007 · 192 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0668-9
- 203 Sven Roeren  
**Komplexitätsvariable Einflussgrößen für die bauteilbezogene Struktursimulation thermischer Fertigungsprozesse**  
 2007 · 224 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0680-1
- 204 Henning Rudolf  
**Wissensbasierte Montageplanung in der Digitalen Fabrik am Beispiel der Automobilindustrie**  
 2007 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0697-9
- 205 Stella Clarke-Griebsch  
**Overcoming the Network Problem in Telepresence Systems with Prediction and Inertia**  
 2007 · 150 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0701-3
- 206 Michael Ehrenstraßer  
**Sensoreinsatz in der telepräsenten Mikromontage**  
 2008 · 160 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0743-3

- 207 Rainer Schack  
**Methodik zur bewertungsorientierten Skalierung der Digitalen Fabrik**  
2008 · 248 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0748-8
- 208 Wolfgang Sudhoff  
**Methodik zur Bewertung standortübergreifender Mobilität in der Produktion**  
2008 · 276 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0749-5
- 209 Stefan Müller  
**Methodik für die entwicklungs- und planungsbegleitende Generierung und Bewertung von Produktionsalternativen**  
2008 · 240 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0750-1
- 210 Ulrich Kohler  
**Methodik zur kontinuierlichen und kostenorientierten Planung produktionstechnischer Systeme**  
2008 · 232 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0753-2
- 211 Klaus Schlickenrieder  
**Methodik zur Prozessoptimierung beim automatisierten elastischen Kleben großflächiger Bauteile**  
2008 · 204 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0776-1
- 212 Niklas Möller  
**Bestimmung der Wirtschaftlichkeit wandlungsfähiger Produktionssysteme**  
2008 · 260 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0778-5
- 213 Daniel Siedl  
**Simulation des dynamischen Verhaltens von Werkzeugmaschinen während Verfahrbewegungen**  
2008 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0779-2
- 214 Dirk Ansoerge  
**Auftragsabwicklung in heterogenen Produktionsstrukturen mit spezifischen Planungsfreiräumen**  
2008 · 146 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0785-3
- 215 Georg Wunsch  
**Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme**  
2008 · 224 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0795-2
- 216 Thomas Oertli  
**Strukturmechanische Berechnung und Regelungssimulation von Werkzeugmaschinen mit elektromechanischen Vorschubantrieben**  
2008 · 194 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0798-3
- 217 Bernd Petzold  
**Entwicklung eines Operatorarbeitsplatzes für die telepräsenste Mikromontage**  
2008 · 234 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0805-8
- 218 Loucas Papadakis  
**Simulation of the Structural Effects of Welded Frame Assemblies in Manufacturing Process Chains**  
2008 · 260 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0813-3
- 219 Mathias Mörtl  
**Ressourcenplanung in der variantenreichen Fertigung**  
2008 · 210 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0820-1
- 220 Sebastian Weig  
**Konzept eines integrierten Risikomanagements für die Ablauf- und Strukturgestaltung in Fabrikplanungsprojekten**  
2008 · 232 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0823-2
- 221 Tobias Hornfeck  
**Laserstrahlbiegen komplexer Aluminiumstrukturen für Anwendungen in der Luftfahrtindustrie**  
2008 · 150 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0826-3
- 222 Hans Egermeier  
**Entwicklung eines Virtual-Reality-Systems für die Montagesimulation mit kraftrückkoppelnden Handschuhen**  
2008 · 210 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0833-1
- 223 Matthäus Sigl  
**Ein Beitrag zur Entwicklung des Elektronenstrahlsinterns**  
2008 · 185 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0841-6

- 224 Mark Harfensteller  
**Eine Methodik zur Entwicklung und Herstellung von Radiumtargets**  
 2009 · 196 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0849-8
- 225 Jochen Werner  
**Methode zur roboterbasierten förderbandsynchronen Fließmontage am Beispiel der Automobilindustrie**  
 2009 · 210 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0857-7
- 226 Florian Hagemann  
**Ein formflexibles Werkzeug für das Rapid Tooling beim Spritzgießen**  
 2009 · 226 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0861-4
- 227 Haitham Rashidy  
**Knowledge-based quality control in manufacturing processes with application to the automotive industry**  
 2009 · 212 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0862-1
- 228 Wolfgang Vogl  
**Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern**  
 2009 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0869-0
- 229 Sonja Schedl  
**Integration von Anforderungsmanagement in den mechatronischen Entwicklungsprozess**  
 2009 · 160 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0874-4
- 230 Andreas Trautmann  
**Bifocal Hybrid Laser Welding – A Technology for Welding of Aluminium and Zinc-Coated Steels**  
 2009 · 268 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0876-8
- 231 Patrick Neise  
**Managing Quality and Delivery Reliability of Suppliers by Using Incentives and Simulation Models**  
 2009 · 224 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0878-2
- 232 Christian Habicht  
**Einsatz und Auslegung zeitenfensterbasierter Planungssysteme in überbetrieblichen Wertschöpfungsketten**  
 2009 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0891-1
- 233 Michael Spitzweg  
**Methode und Konzept für den Einsatz eines physikalischen Modells in der Entwicklung von Produktionsanlagen**  
 2009 · 180 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0931-4
- 234 Ulrich Munzert  
**Bahnplanungsalgorithmen für das robotergestützte Remote-Laserstrahlschweißen**  
 2010 · 176 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0948-2
- 235 Georg Völlner  
**Rührreischweißen mit Schwerlast-Industrierobotern**  
 2010 · 232 Seiten · 20,5 x 14,5 cm · 978-3-8316-0955-0
- 236 Nils Müller  
**Modell für die Beherrschung und Reduktion von Nachfrageschwankungen**  
 2010 · 270 Seiten · 20,5 x 14,5 cm · 978-3-8316-0992-5
- 237 Franz Decker  
**Unternehmensspezifische Strukturierung der Produktion als permanente Aufgabe**  
 2010 · 180 Seiten · 20,5 x 14,5 cm · 978-3-8316-0996-3
- 238 Christian Lau  
**Methodik für eine selbstoptimierende Produktionssteuerung**  
 2010 · 200 Seiten · 20,5 x 14,5 cm · 978-3-8316-4012-6
- 239 Christoph Rimpau  
**Wissensbasierte Risikobewertung in der Angebotskalkulation für hochgradig individualisierte Produkte**  
 2010 · 200 Seiten · 20,5 x 14,5 cm · 978-3-8316-4015-7
- 240 Michael Loy  
**Modulare Vibrationswendelförderer zur flexiblen Teilezuführung**  
 2010 · 169 Seiten · 20,5 x 14,5 cm · 978-3-8316-4027-0
- 241 Andreas Eursch  
**Konzept eines immersiven Assistenzsystems mit Augmented Reality zur Unterstützung manueller Aktivitäten in radioaktiven Produktionsumgebungen**  
 2010 · 205 Seiten · 20,5 x 14,5 cm · 978-3-8316-4029-4

- 242 Florian Schwarz  
**Simulation der Wechselwirkungen zwischen Prozess und Struktur bei der Drehbearbeitung**  
 2010 · 256 Seiten · 20,5 x 14,5 cm · 978-3-8316-4030-0
- 243 Martin Georg Prasch  
**Integration leistungsgewandelter Mitarbeiter in die variantenreiche Serienmontage**  
 2010 · 261 Seiten · 20,5 x 14,5 cm · 978-3-8316-4033-1
- 244 Johannes Schilp  
**Adaptive Montagesysteme für hybride Mikrosysteme unter Einsatz von Telepräsenz**  
 2011 · 160 Seiten · 20,5 x 14,5 cm · 978-3-8316-4063-8
- 245 Stefan Lutzmann  
**Beitrag zur Prozessbeherrschung des Elektronenstrahlschmelzens**  
 2011 · 222 Seiten · 20,5 x 14,5 cm · 978-3-8316-4070-6
- 246 Gregor Branner  
**Modellierung transienter Effekte in der Struktursimulation von Schichtbauverfahren**  
 2011 · 230 Seiten · 20,5 x 14,5 cm · 978-3-8316-4071-3
- 247 Josef Ludwig Zimmermann  
**Eine Methodik zur Gestaltung berührungslos arbeitender Handhabungssysteme**  
 2011 · 184 Seiten · 20,5 x 14,5 cm · 978-3-8316-4091-1
- 248 Clemens Pörnbacher  
**Modellgetriebene Entwicklung der Steuerungssoftware automatisierter Fertigungssysteme**  
 2011 · 280 Seiten · 20,5 x 14,5 cm · 978-3-8316-4108-6
- 249 Alexander Lindworsky  
**Teilautomatische Generierung von Simulationsmodellen für den entwicklungsbegleitenden Steuerungstest**  
 2011 · 300 Seiten · 20,5 x 14,5 cm · 978-3-8316-4125-3
- 250 Michael Mauderer  
**Ein Beitrag zur Planung und Entwicklung von rekonfigurierbaren mechatronischen Systemen – am Beispiel von starren Fertigungssystemen**  
 2011 · 150 Seiten · 20,5 x 14,5 cm · 978-3-8316-4126-0
- 251 Roland Mork  
**Qualitätsbewertung und -regelung für die Fertigung von Karosserieteilen in Presswerken auf Basis Neuronaler Netze**  
 2011 · 220 Seiten · 20,5 x 14,5 cm · 978-3-8316-4127-7
- 252 Florian Reichl  
**Methode zum Management der Kooperation von Fabrik- und Technologieplanung**  
 2011 · 220 Seiten · 20,5 x 14,5 cm · 978-3-8316-4128-4
- 253 Paul Gebhard  
**Dynamisches Verhalten von Werkzeugmaschinen bei Anwendung für das Rührreibschweißen**  
 2011 · 220 Seiten · 20,5 x 14,5 cm · 978-3-8316-4129-1



