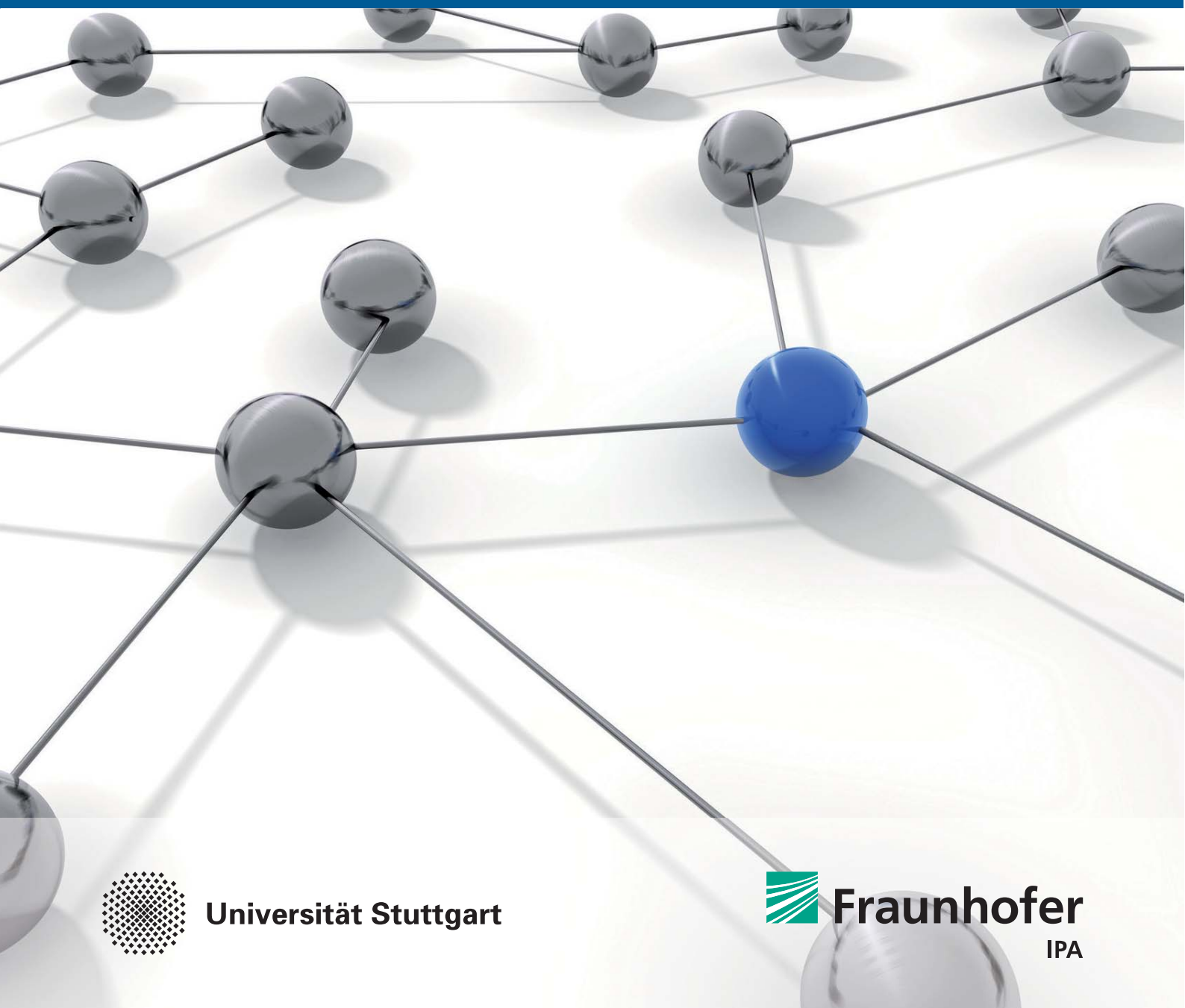


STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG

GERNOT FRANK

Durchgängiges mechatronisches Engineering für Sondermaschinen



Universität Stuttgart



Fraunhofer
IPA

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG BAND 48

Herausgeber:

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl

Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl

Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Gernot Frank

**Durchgängiges mechatronisches Engineering
für Sondermaschinen**

FRAUNHOFER VERLAG

Kontaktadresse:

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 0711 970-00, Telefax 0711 970-1399
info@ipa.fraunhofer.de, www.ipa.fraunhofer.de

STUTTGARTER BEITRÄGE ZUR PRODUKTIONSFORSCHUNG**Herausgeber:**

Univ.-Prof. Dr.-Ing. Thomas Bauernhansl
Univ.-Prof. Dr.-Ing. Dr. h.c. mult. Alexander Verl
Univ.-Prof. a. D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. Engelbert Westkämper

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, Stuttgart
Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) der Universität Stuttgart
Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW)
der Universität Stuttgart

Titelbild: © Cmon - Fotolia

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über www.dnb.de abrufbar.

ISSN: 2195-2892

ISBN (Print): 978-3-8396-0968-2

D 93

Zugl.: Stuttgart, Univ., Diss., 2015

Druck: Mediendienstleistungen des Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart
Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

© by **FRAUNHOFER VERLAG**, 2016

Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 800469, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 0711 970-2500
Telefax 0711 970-2508
E-Mail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Alle Rechte vorbehalten

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften. Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

GELEITWORT DER HERAUSGEBER

Produktionswissenschaftliche Forschungsfragen entstehen in der Regel im Anwendungszusammenhang, die Produktionsforschung ist also weitgehend erfahrungsbasiert. Der wissenschaftliche Anspruch der „Stuttgarter Beiträge zur Produktionsforschung“ liegt unter anderem darin, Dissertation für Dissertation ein übergreifendes ganzheitliches Theoriegebäude der Produktion zu erstellen.

Die Herausgeber dieser Dissertations-Reihe leiten gemeinsam das Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA und jeweils ein Institut der Fakultät für Konstruktions-, Produktions- und Fahrzeugtechnik an der Universität Stuttgart.

Die von ihnen betreuten Dissertationen sind der marktorientierten Nachhaltigkeit verpflichtet, ihr Ansatz ist systemisch und interdisziplinär. Die Autoren bearbeiten anspruchsvolle Forschungsfragen im Spannungsfeld zwischen theoretischen Grundlagen und industrieller Anwendung.

Die „Stuttgarter Beiträge zur Produktionsforschung“ ersetzt die Reihen „IPA-IAO Forschung und Praxis“ (Hrsg. H.J. Warnecke / H.-J. Bullinger / E. Westkämper / D. Spath) bzw. ISW Forschung und Praxis (Hrsg. G. Stute / G. Pritschow / A. Verl). In den vergangenen Jahrzehnten sind darin über 800 Dissertationen erschienen.

Der Strukturwandel in den Industrien unseres Landes muss auch in der Forschung in einen globalen Zusammenhang gestellt werden. Der reine Fokus auf Erkenntnisgewinn ist zu eindimensional. Die „Stuttgarter Beiträge zur Produktionsforschung“ zielen also darauf ab, mittelfristig Lösungen für den Markt anzubieten. Daher konzentrieren sich die Stuttgarter produktionstechnischen Institute auf das Thema ganzheitliche Produktion in den Kernindustrien Deutschlands. Die leitende Forschungsfrage der Arbeiten ist: Wie können wir nachhaltig mit einem hohen Wertschöpfungsanteil in Deutschland für einen globalen Markt produzieren?

Wir wünschen den Autoren, dass ihre „Stuttgarter Beiträge zur Produktionsforschung“ in der breiten Fachwelt als substanziell wahrgenommen werden und so die Produktionsforschung weltweit voranbringen.

Alexander Verl

Thomas Bauernhansl

Engelbert Westkämper

Durchgängiges mechatronisches Engineering für Sondermaschinen

Von der Graduate School of Excellence
advanced Manufacturing Engineering der Universität Stuttgart
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von

Dipl.-Ing. Gernot Frank
aus Crailsheim

Hauptberichter: Univ. Prof. a.D. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult.
Engelbert Westkämper

Mitberichter: Univ. Prof. Dr.-Ing. Peter Klemm

Tag der mündlichen Prüfung: 6. Mai 2015

Institut für Industrielle Fertigung und Fabrikbetrieb der Universität Stuttgart

2015

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als Doktorand der Graduate School of Excellence advanced Manufacturing Engineering (GSaME) der Universität Stuttgart am Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) und einem Unternehmen der Automatisierungstechnik.

Besonderer Dank gilt meinem Doktorvater Prof. Engelbert Westkämper. Die regelmäßigen Diskussionen und die daraus resultierenden Anregungen haben einen wesentlichen Beitrag zu dieser Arbeit geleistet. Zudem möchte ich mich bei ihm für die vertrauensvolle Zusammenarbeit und mein Promotionsumfeld bedanken, welches mir auch außerhalb meines Themenbereiches viele Zusammenhänge beibrachte und Erfahrungen schenkte. Bei Prof. Peter Klemm möchte ich mich für die Übernahme des Mitberichts bedanken.

Ohne die Unterstützung des Unternehmens wäre diese Arbeit nicht möglich gewesen. Daher möchte ich mich in Besonderem Maße bei Dr.-Ing. Wolfgang Schlögl bedanken, welcher mich mit der Aufgabenstellung betraute, stets förderte und mir sämtliche Türen öffnete um das Thema voranzubringen. Allen anderen, die in diesem Umfeld mit mir zusammenarbeiteten und wertvolle Diskussionspartner darstellten, gilt ebenfalls ein herzliches Dankeschön.

Zudem gilt mein Dank dem wissenschaftlichen Umfeld am IFF/IPA bzw. der GSaME, welches mir viele Blicke über den Tellerrand hinaus ermöglichte und mir ein „zweites Zuhause“ bot. Stellvertretend für alle Wegbegleiter möchte ich mich hier bei Jochen Böck, Max Dinkelmann, Holger Haag, Carina Löffler, Jörg Siegert, Andreas Meyke, Hans-Friedrich Jacobi, Caroline Schulz, Oliver Mannuß, Christoph Maier, Jörg Mandel, Kristofer Kolb, Stefan Silcher und Raphael Rahäuser bedanken.

Insbesondere haben meine Familie und Freunde für den notwendigen moralischen Rückhalt und die Bodenhaftung gesorgt. Besonderer Dank gilt meinen Eltern, Hannelore und Friedrich Frank, die mir diesen Weg überhaupt erst ermöglichten. Ich danke Ihnen und meinem Bruder Jochen für die Unterstützung und den familiären Rückhalt, sowie meinem Bruder Henning, der mir auf tragische Weise die wirklich wichtigen Dinge des Lebens beigebracht hat. Meiner Frau Simone möchte ich für ihr Verständnis und ihren Rückhalt in allen Lebenslagen danken.

Kurzinhalt

Immer kürzer werdende Produktlebenszyklen und kundenindividuellere Produkte zwingen Hersteller dazu ihr Produktionssystem immer schneller anzupassen. Eine solche Anpassung bietet erhebliches Einsparpotenzial, wenn damit die Prozesskette optimiert wird. Dies kann häufig nur über individuell angepasste Maschinen erfolgen. Gerade in Hochlohnländern resultiert daraus eine erhöhte Nachfrage nach Sondermaschinen. Die Maschinenhersteller sind damit gezwungen auftragsbezogene Entwicklungen durchzuführen, was einen hohen Anteil von Entwicklungskosten an der Gesamtinvestition mit sich bringt. Unter des Gesichtspunkts eines bevorstehenden Fachkräftemangel ist ersichtlich, dass kurze Auftragsdurchlaufzeiten nur durch ein äußerst effizientes Engineering erreichbar sind. Allerdings ist die Entwicklung solcher Maschinen immer noch geprägt von einer sequentiellen Reihenfolge der Disziplinen Mechanik-, Elektrokonstruktion und Automatisierung. Neben weiteren Defiziten führt eine ungenügende Anforderungsaufnahme für die Entwicklung, eine sehr geringe Wiederverwendung von mechatronischen Modulen bzw. ein sehr hohen Aufwand zur Erstellung dieser Module und voneinander entkoppelte Softwaresysteme zur Unterstützung der disziplinspezifischen Aufgaben zu unnötigen Aufwänden während der Projektarbeit. Deshalb ist das Ziel der vorliegenden Dissertation eine durchgängige softwaregestützte Methode für die integrierte mechatronische Sondermaschinenentwicklung bereitzustellen, welche die Disziplinen parallelisiert und eine Wiederverwendung bereits entwickelter mechatronischer Module mit sehr geringem Aufwand ermöglicht. Dazu wird zunächst analysiert, welche Abhängigkeiten die Disziplinen voneinander haben und wie mechatronische Module definiert werden können. Dies bildet die Basis für die erarbeitete Methode. Darin wird mit der Definition von Anforderungen, unterstützt von einer Sachmerkmaliste begonnen. Anschließend werden die Anforderungen im Rahmen der Systementwurfphase unter Einbeziehung aller Disziplinen auf Komponenten und Module heruntergebrochen. Dadurch entsteht ein mechatronisch abgestimmtes Maschinenkonzept. Dieses bildet die Basis für die nächste Phase parallel in den verschiedenen Disziplinen die Detaillierungen auszuarbeiten. Die abschließende Phase der Systemintegration validiert die Ergebnisse. Zur Unterstützung aller Phasen der beschriebenen Methode werden erhältliche Softwareprodukte bzw. eigens entwickelte Prototypen genutzt und über ein semantisches Netz integriert. Die Methode und deren softwaretechnische Unterstützung werden anhand einer Firmwarelademaschine validiert. Dadurch lassen sich die Anwendbarkeit und der Nutzen für eine Entwicklung von Sondermaschinen nachweisen.

Short Summary

Short Product Lifecycles and customized products force manufacturers to adapt their production systems faster and faster. Such adaptations offer chances of high cost reductions, particularly if thereby the process chain is optimized. This can be achieved through using customized machines. Especially in high-wage countries an increasing demand to specialized production machines can be noticed. Therefore machine builders are forced to engineer machines related to orders, which brings up a high amount of engineering-costs based on the total investment. It is obvious that the perspective of skilled worker shortage needs a highly efficient engineering to get short order throughput times. Certainly, the development of such machines is characterized by engineering sequentially within the disciplines of mechanics, electrics and automation. Beside further deficits insufficient definitions of requirements, low reuse of mechatronic modules respectively high efforts for definition of such modules and disconnected software-tools supporting the discipline specific tasks lead to unnecessary efforts along the project. Based on that, the goal of this thesis is to provide a seamless method for integrated mechatronic engineering of specialized production machines, parallelizing disciplines and reusing already developed mechatronic modules with nominal efforts. Thereto an analysis is done, getting the dependencies the disciplines have to each other and how mechatronic modules can be defined. This is the base for the presented method, starting with the definition of requirements, which is supported via a class list of characteristics. Afterwards, the requirements are broken down to components and modules within the phase of system design. This leads to a mechatronic machine concept. Within the next phase the different disciplines are working on the detailing in parallel. In conclusion a system-integration is done for validating the results. Supporting the different phases, available software-tools or specially built prototypes are used and integrated into a semantic net. The method and their software-assistance is validated based on a firmwareloading-machine. This shows the applicability and the benefits for the engineering of specialized production machines.

Inhaltsverzeichnis

Inhaltsverzeichnis	IX
Abkürzungsverzeichnis	XII
Abbildungsverzeichnis.....	XIII
Tabellenverzeichnis.....	XVIII
1 Einleitung	1
1.1 Ausgangssituation	1
1.2 Problemstellung.....	2
1.3 Zielsetzung und Aufgabenstellung	7
1.4 Vorgehensweise und Aufbau der Arbeit	8
2 Stand der Wissenschaft und Technik	10
2.1 Disziplinspezifische Entwicklungsmethoden und sequenzieller Maschinenentwicklungsprozess	10
2.1.1 Mechanik.....	10
2.1.2 Elektrik.....	13
2.1.3 Automatisierung.....	16
2.1.4 Zusammenfassung.....	18
2.2 Mechatronische System- und Begriffsdefinitionen für SPS-gesteuerte Sondermaschinen. 19	
2.2.1 Mechatronisches System.....	20
2.2.2 Grundgedanken des Systems Engineering.....	21
2.2.3 Mechatronischer Entwicklungsprozess nach VDI-Richtlinie 2206	22
2.3 Ansätze mechatronischer Systementwicklung	24
2.3.1 Problemorientierte Ansätze.....	24
2.3.2 Produktionsprozessorientierte Ansätze.....	27
2.4 Grundlagen und Ansätze der Maschinensimulation.....	29
2.5 Ansätze zur Aufwandsreduktion in mechatronischen Entwicklungsprozessen	32
2.5.1 Wiederverwendung entlang des Entwicklungsprozesses.....	32
2.5.2 Wiederverwendung über Projekte hinweg	34
2.6 Verknüpfung von Softwarewerkzeugen	38
2.6.1 Punkt-zu-Punkt-Schnittstellen und neutrale Datenformate.....	39
2.6.2 Ansätze auf Basis strukturierter Daten.....	40
2.6.3 Ansätze auf Basis semistrukturierter Daten	41

2.6.4	Ansätze semantischer Technologien	41
2.7	Defizite des Standes der Technik und Handlungsbedarf.....	44
2.7.1	Anforderungsdefinition zur Bewertung des Standes der Technik	44
2.7.2	Bewertung der interdisziplinären Entwicklungsmethoden und Ableitung des methodischen Handlungsbedarfs	47
2.7.3	Bewertung der Ansätze zur Aufwandsreduktion	50
2.7.4	Bewertung der Ansätze zur Vernetzung von Softwarewerkzeugen.....	51
2.7.5	Fazit und Handlungsbedarf	51
3	Analyse der disziplinspezifischen Vorgehensweisen und Ableitung eines Methodenkonzeptes	52
3.1	Mechanische Vorgehensweise und Modellbildung	52
3.2	Elektrotechnische Vorgehensweise und Modellbildung	53
3.3	Automatisierungstechnische Vorgehensweise und Modellbildung	56
3.4	Analyse und Gegenüberstellung der Vorgehensweisen und Modelle.....	57
3.4.1	Statischer Vergleich der Modelle.....	58
3.4.2	Betrachtung der Wiederverwendung in mechatronischem Kontext über Projekte hinweg.....	64
3.4.3	Betrachtung des Entwicklungsablaufes	67
3.5	Methodenkonzept	73
4	Methode zur mechatronischen Entwicklung von Sondermaschinen.....	76
4.1	Vorgehensweise zum systematischen Entwickeln von Sondermaschinen.....	76
4.2	Anforderungsdefinition	78
4.3	Systementwurf.....	79
4.3.1	Funktionale Modellierung.....	80
4.3.2	Finden der prinzipiellen Lösung und der mechatronischen Systembeschreibung	86
4.3.3	Wiederverwendung existierender mechatronischer Module.....	91
4.3.4	Mechatronische Beschreibung neuer Lösungen/Module.....	95
4.3.5	Mechatronische Beschreibung des Gesamtsystems.....	100
4.4	Parallele Detaillierung.....	102
4.4.1	Simultane Ausdetaillierung.....	104
4.4.2	Wiederverwendung mechatronischer Komponentenbeschreibungen	106
4.5	Virtuelle Systemintegration.....	107
4.6	Ausschneiden und Ablegen mechatronischer Objekte.....	108
4.7	Zusammenfassung	109
5	Softwaretechnische Unterstützung	111
5.1	Softwareintegration auf Basis semantischer Netze	111

5.2	Architektur mechatronischer Objekte zur Wiederverwendung.....	114
5.3	Ermittlung von Knoten- und Kantentypen im interdisziplinären Bereich	117
5.3.1	Definition der Knoten- und Kantentypen auf Basis der vorgestellten Methode.....	117
5.3.2	Auswahl von Softwarewerkzeugen und Extraktion ihrer Semantik	120
5.3.3	Abgleich der benötigten und bereitgestellten Informationen.....	124
5.4	Architektur der Softwareunterstützung	125
5.5	Umsetzung der Softwareunterstützung	126
5.5.1	Umsetzung der Phasen Anforderungsdefinition und Systementwurf	126
5.5.2	Parallele Detaillierung.....	129
5.5.3	Simulationsunterstützung der Maschinenentwicklung	130
6	Validierung am Beispiel einer Firmwarelademaschine	134
6.1	Fallbeispiel: Entwicklung einer Firmwarelademaschine	134
6.2	Entwicklungsvorgehen	135
6.3	Erkenntnisse der Validierung	141
6.3.1	Erkenntnisse aus methodischer Sicht.....	141
6.3.2	Erkenntnisse bei der Verknüpfung von Softwarewerkzeugen	143
7	Zusammenfassung und Ausblick	145
	Summary.....	149
	Literaturverzeichnis.....	152

Abkürzungsverzeichnis

2-D	2-dimensional
API	Application Programming Interface
CAEX	Computer Aided Engineering Exchange
DSM	Design Structure Matrix
E/A-Liste	Eingangs-/Ausgangsliste
ECAD	electrical Computer Aided Design
ERM	Entity-Relationship-Modell
FBS-Notation	Function Behaviour State-Notation
FMEA	Failure Method and Effects Analysis/Fehlermöglichkeits- und Einfluss- Analyse
HILS	Hardware-in-the-loop-Simulation
HMI	Human-Machine-Interface
I/O-Karten	Input/Output-Karten
IEC	International Electrotechnical Commission
MCAD	mechanical Computer Aided Design
MCD	Mechatronics Concept Designer
MES	Manufacturing Execution System
MO	Mechatronisches Objekt
OWL	Web Ontology Language
PERT	Program Evaluation And Review Technique
PLC	Programmable Logic Controller/speicherprogrammierbare Steuerung
PLMXML	Product Lifecycle Management Extensible Markup Language
POE	Programmorganisationseinheit
R&I	Rohrleitungs- und Instrumentierungsschema
RDF	Resource Description Framework
SFC	Sequential Function Chart
SPS	Speicherprogrammierbare Steuerung
SSF-Framework	System State Flow Framework
SysML	Systems Modeling Language
UML	Unified Modeling Language
VDI	Verein Deutscher Ingenieure
VMTL	Virtual Machine Tool Language
XML	Extensible Markup Language
XSD	XML Schema Definition

Abbildungsverzeichnis

Abbildung 1.1: Vorgehensweise und Aufbau der Arbeit	8
Abbildung 2.1: Grundvorgehen zum Entwickeln von mechanischen Produkten nach <i>VDI-Richtlinie 2221</i> [1993-05].....	11
Abbildung 2.2: Ablauf der mechanischen Konstruktion in Bezug auf die in der <i>VDI-Richt-</i> <i>linie 2221</i> genannten Schritte drei bis sieben im Sondermaschinenbau.....	12
Abbildung 2.3: Kreisläufe der elektrischen und fluidtechnischen Konstruktion [Grätz 2006].....	14
Abbildung 2.4: Ablauf der Elektrokonstruktion	15
Abbildung 2.5: Ablauf der SPS-Programmierung übertragen in Anlehnung an <i>Seitz</i> [2012].....	17
Abbildung 2.6: Derzeitiger Entwicklungsprozess von Sondermaschinen [in Anlehnung an Bleisteiner 2010].....	19
Abbildung 2.7: Grundstruktur mechatronischer Systeme nach <i>VDI-Richtlinie 2206</i> [2004-06]....	20
Abbildung 2.8: Prozess des Systems Engineering nach <i>Leonard</i> [2001]	21
Abbildung 2.9: Vorgehensmodell zur Entwicklung mechatronischer Systeme nach <i>VDI-Richtlinie 2206</i> [2004-06].....	23
Abbildung 2.10: Interdisziplinäre Funktionsmodellierung nach <i>Kallmeyer</i> [1998].....	25
Abbildung 2.11: Funktionshierarchie der <i>VDW-Richtlinie Funktionsbeschreibung</i> [2002-10].....	28
Abbildung 2.12: Aufbau eines Maschinenmodells für die virtuelle Inbetriebnahme nach <i>Kufner</i> [2012]	31
Abbildung 2.13: Möglichkeiten zur Transformation von Modellen entlang des Entwicklungs- prozesses	34
Abbildung 2.14: Prinzip der Modellgenerierung auf Basis von Bibliotheken	35
Abbildung 2.15: Prozess zur Entkopplung des Auftrags von der Entwicklung	36
Abbildung 2.16: Mechatronisches Planungsobjekt nach <i>Hundt</i> [2012]	37
Abbildung 2.17: Wissenstreppe nach <i>North</i> [2011]	39
Abbildung 2.18: Beispiel eines semantischen Netzes nach <i>Reichenberger</i> [2010].....	42
Abbildung 2.19: Semantische Treppe nach <i>Blumauer et al.</i> [2006].....	44
Abbildung 2.20: Anforderungen an eine optimale softwaregestützte Methode zur Entwicklung von Sondermaschinen bezogen auf die Phasen der <i>VDI-Richtlinie 2206</i> [2004-06].....	47

Abbildung 2.21: Vergleich und Eignung derzeitiger Methoden hinsichtlich der Entwicklung von Sondermaschinen	49
Abbildung 3.1: Baugruppen, Teile und Anschlussbeschreibung in einem MCAD-System.....	53
Abbildung 3.2: Aufbau der Maschinenstruktur in einem ECAD-System	54
Abbildung 3.3: Elemente eines SPS-Programms	57
Abbildung 3.4: Modellierungselemente der verschiedenen Disziplinen	58
Abbildung 3.5: Einordnung der disziplinspezifischen Modelle auf verschiedene Abstraktionsniveaus.....	60
Abbildung 3.6: Vergleich der Repräsentationsformen von Anschlüssen in der Mechanik und der Elektrik	60
Abbildung 3.7: Unterschiedliche Bedeutungen der hierarchischen Ordnung in den disziplinspezifischen Modellen	61
Abbildung 3.8: Disziplinspezifische Interpretationen des Begriffs Aktor	62
Abbildung 3.9: Gültigkeitsbereiche disziplinspezifischer Modelle und ihre Äquivalenz im Projektkontext.....	63
Abbildung 3.10: Systemgrenzen der disziplinspezifischen Sichtweisen	65
Abbildung 3.11: Möglichkeiten zur mechatronischen Modularisierung	66
Abbildung 3.12: Ablauf des sequenziellen Entwicklungsprozesses in Abhängigkeit der Modellabstraktion.....	67
Abbildung 3.13: Einfluss des kinematischen Konzeptes auf disziplinspezifische Planungen	68
Abbildung 3.14: Einfluss der Schlauchlänge des mechanischen Modells auf die Verschaltung	71
Abbildung 3.15: Abhängigkeiten der Disziplinen während einer parallelen Detaillierung.....	72
Abbildung 3.16: Modellierung von Komponenten der Stückliste in verschiedenen Disziplinen.....	72
Abbildung 3.17: Verlauf der Modellierungen in den verschiedenen Disziplinen in Abhängigkeit der Abstraktion und der Entwicklungsphasen	75
Abbildung 4.1: Vorgehen zur systematischen Entwicklung von Sondermaschinen in Bezug auf die in Kapitel 3.5 eingeführten Quality Gates	77
Abbildung 4.2: Sachmerkmal-Leitlinie zur Erhebung von Anforderungen für Sondermaschinen	79
Abbildung 4.3: Anforderungen an die Phase des Systementwurfs.....	80
Abbildung 4.4: Funktionale Dekomposition	81
Abbildung 4.5: Vernetzung von Funktionen	82

Abbildung 4.6:	Gantt-Chart zur Beschreibung des Systemverhaltens	85
Abbildung 4.7:	Übersicht der funktionalen Modellierung.....	85
Abbildung 4.8:	Morphologischer Kasten und die Ableitung von Funktionsintegrationen und Mehrfachverwendung.....	86
Abbildung 4.9:	Reduktion von möglichen Lösungen durch Modellierung von Beziehungen	87
Abbildung 4.10:	Iteratives Vorgehen zur funktionalen Modellierung und der Definition von Lösungsprinzipien	88
Abbildung 4.11:	Statische mechatronische Systembeschreibung für das Gesamtsystem Sondermaschine.....	91
Abbildung 4.12:	Lösungsprinzip als Klassifikation verschiedener physikalischer Alternativen	92
Abbildung 4.13:	Mögliche Einstiegspunkte und die daraus resultierende Zeitreduktion bei mechatronischer Wiederverwendung von Komponenten im Rahmen der Sondermaschinenentwicklung.....	93
Abbildung 4.14:	Notwendige Informationen des kinematischen Konzeptes	95
Abbildung 4.15:	Startpunkt für die elektrische Konstruktion.....	97
Abbildung 4.16:	Zeitdauer der Funktion als Anforderung für den Bewegungsablauf im Impulsdiagramm.....	99
Abbildung 4.17:	Mechatronische Beschreibung eines Lösungsprinzips zur parallelen Detaillierung	100
Abbildung 4.18:	Herunterbrechen der integrationsrelevanten Anforderungen auf einzelne Maschinenmodule.....	101
Abbildung 4.19:	Mechatronisches Modell der gesamten Maschine am Ende des Systementwurfes.....	102
Abbildung 4.20:	Anforderungen an die Phase der parallelen disziplinspezifischen Detaillierung	103
Abbildung 4.21:	Aus der Systementwurfphase definierter Startpunkt in den einzelnen disziplinspezifischen Vorgehen	103
Abbildung 4.22:	Entstehung derselben Informationen in unterschiedlichen Kontexten	105
Abbildung 4.23:	Aufwandsreduktion durch Weitergabe von abstrahierten Informationen	106
Abbildung 4.24:	Modellerstellung zur virtuellen Inbetriebnahme durch Nutzung der Schaltpläne und des kinematisierten mechanischen Modells.....	108
Abbildung 4.25:	Phasen und ihre Beschreibungsmittel der vorgestellten Methode.....	110

Abbildung 5.1: Interdisziplinäre und proprietäre Informationen von Softwarewerkzeugen [vgl. Frank, et al. 2011]	111
Abbildung 5.2: Grundarchitektur zur Interoperabilität zweier Applikationen auf Basis eines semantischen Netzes	112
Abbildung 5.3: Interpretationsbereiche von Applikationen auf dem semantischen Netz	113
Abbildung 5.4: Löschen von Knoten in semantischen Netzen und die Auswirkung auf andere Applikationen.....	114
Abbildung 5.5: Architektur des mechatronischen Objektes	115
Abbildung 5.6: Schneiden mechatronischer Objekte auf Basis eines bestehenden Projektes	116
Abbildung 5.7: Vorgehen zur Ermittlung der Knoten- und Kantentypen	117
Abbildung 5.8: Wesentliche Knoten- und Kantentypen der Methodik in Bezug auf die Aufgabengebiete der Disziplinen	119
Abbildung 5.9: Abdeckung der Phasen durch am Markt erhältliche Softwarewerkzeuge	121
Abbildung 5.10: Semantische Extraktion eines „Collision-Sensors“ des Mechatronics Concept Designers zur Integration in das semantische Netz.....	122
Abbildung 5.11: Ausgewählte Werkzeuge und deren Abdeckung des aufgestellten Informationsmodells	123
Abbildung 5.12: Architektur der prototypischen Umsetzung.....	125
Abbildung 5.13: Userinterface zur funktionalen Modellierung.....	127
Abbildung 5.14: Userinterface zur logischen Modellierung	128
Abbildung 5.15: Beispiel zur semantischen Anreicherung bei der Erstellung eines „Collision Sensors“ in Mechatronics Concept Designer	129
Abbildung 5.16: Navigatoren für die parallele Detaillierung mit Ordner für interdisziplinär relevante Informationen.....	129
Abbildung 5.17: Stufen der virtuellen Inbetriebnahme entlang des Entwicklungsprozesses.....	131
Abbildung 5.18: Generierung von Modellen für die virtuelle Inbetriebnahme auf Basis von Schaltplänen.....	133
Abbildung 6.1: Firmwarelademaschine	134
Abbildung 6.2: Funktionale Struktur der Firmwarelademaschine in der Integrationsapplikation	135
Abbildung 6.3: Morphologischer Kasten zur Suche und Definition von Prinziplösungen	136
Abbildung 6.4: Wiederverwendung eines Greifers	137

Abbildung 6.5: Kinematisches Konzept eines Greifers in Mechatronics Concept Designer mit Angabe des Antriebskonzepts	138
Abbildung 6.6: Mechanische Ablaufbeschreibung in Sequence Designer für den Greifer	138
Abbildung 6.7: Löschen eines Sensors in interdisziplinären Umfeld aus dem ECAD-System heraus	139
Abbildung 6.8: Kontextmenü der Navigatoren.....	140
Abbildung 6.9: Schneiden mechatronischer Objekte.....	141
Abbildung 6.10: Erfüllung der aufgestellten Anforderungen durch die in dieser Arbeit entwickelte Methode.....	142
Abbildung 6.11: Integrierte Umgebungen verlangen nach mehr Änderungsmöglichkeiten	144

Tabellenverzeichnis

Tabelle 3.1:	Benötigte Informationen zur Aufgabenerfüllung in den einzelnen Disziplinen....	70
Tabelle 4.1:	Vergleich von Beschreibungssprachen zur Modellierung des Systemverhaltens auf funktionaler Ebene	84
Tabelle 4.2:	Beziehungen von Funktionen zu Lösungsprinzipien unter dem Gesichtspunkt der Modularisierung zur Wiederverwendung	90
Tabelle 4.3:	Mögliche Systemgrenzen zweier Module im Kontext des Maschinenprojekts	94
Tabelle 4.4:	Vergleich von Beschreibungsmittel hinsichtlich des Sollablaufes des kinematischen Lösungsprinzips	98

1 Einleitung

1.1 Ausgangssituation

Basierend auf Megatrends, wie Urbanisierung, demografischem Wandel, Globalisierung und Klimawandel verändern sich die Märkte und deren Trends äußerst rasant. [Eberle 2012] Dies drückt sich in immer kundenindividuelleren Produkten und verkürzten Produktlebenszyklen aus. [Schlögl 2008; Westkämper, et al. 2009] Produzenten sind folglich gezwungen möglichst schnell neue Produkte, zu niedrigen Kosten, in hoher Qualität und in der benötigten Menge zu produzieren, was mit einer schnellstmöglichen Anpassung des Produktionssystems einhergeht. [Wiendahl, et al. 2007] Dabei ermöglichen neue Verfahren und deren Integration in die Prozessketten erhebliche Einsparpotenziale gegenüber den gängigen Fertigungsverfahren. [Bauernhansl 2012; Böck 2013; Röhrig 2002] Eine Optimierung der Prozesskette benötigt somit zwangsläufig neue oder angepasste Ressourcen bzw. Maschinen. Deshalb sind Maschinenhersteller gezwungen individuell auf die entsprechende Kundensituation zugeschnittene Maschinen möglichst schnell zu entwickeln und in Betrieb zu nehmen. Die technische Herausforderung stellt neben der Automatisierung eines (eventuell sogar neuen) Verfahrens, die Integration der Maschine in das äußerst kundenindividuelle Produktionssystem dar. [Frank, et al. 2013a] Diese Integration kann oft nicht mehr durch die reine Anpassung von Standardmaschinen ermöglicht werden. Deshalb nimmt der Anteil an Sondermaschinen gerade in Hochlohnländern zu. [Possel-Dölken 2010; Westkämper 2008] Die Entwicklung solcher Maschinen ist folglich auftragsbezogen und besitzt häufig die Stückzahl eins. Dies schlägt sich in hohen Kosten für die Entwicklung und Konstruktion nieder. [Esswein, et al. 2010]

Bei der Entwicklung oder dem Engineering einer Maschine ist zu berücksichtigen, dass es sich hierbei um ein mechatronisches System handelt. [Bathelt 2007] Der zugrunde liegende Entwicklungsprozess erfolgt derzeit sequenziell, beginnend in der mechanischen, gefolgt von der elektrischen Konstruktion und einer abschließenden Steuerungsprogrammierung. [Bayrhammer, et al. 2013; Hehenberger, et al. 2010; Obersamer 2013] Aufgrund der hohen Individualität hat die mechanische Konstruktion die Aufgabe auf Basis von Standardkomponenten und Eigenkonstruktionen den Zusammenbau der Maschine zu erstellen. Im Rahmen der elektrischen Konstruktion werden pneumatische, hydraulische und elektrische Standardkomponenten miteinander verschaltet. Dadurch können die Bewegungen und der Prozess gesteuert werden. Dies

wird bei Sondermaschinen üblicherweise durch eine zentrale speicherprogrammierbare Steuerung (SPS) übernommen. Deren Programmierung übernimmt die automatisierungstechnische Disziplin. Gerade der Einsatz von modernen Steuerungen ermöglicht es, mehr Funktionalitäten in die Maschine zu integrieren, was gleichzeitig zu einer höheren Komplexität und einem größeren Softwareanteil führt. [Bathelt 2007; Obersamer 2013; Spiegelberger 2011] Im Gegensatz dazu herrscht im Bereich der Maschinen- und Anlagenhersteller immer noch das Verständnis vor, dass die Softwareentwicklung ein Teil der elektrischen Konstruktion ist. Dies zeigt sich daran, dass zwar Stücklisten über elektrotechnische und mechanische Bauteile vorhanden sind, eine softwaretechnische Architekturbeschreibung aber meist fehlt. [Spiegelberger 2011]

1.2 Problemstellung

Wie beschrieben, kommt den Sondermaschinen eine immer wichtigere Bedeutung im Rahmen optimierter Prozessketten zu. In der Entwicklung und Konstruktion der Maschine entstehen aber hohe Aufwände, was einer Forderung nach immer schnellerer Maschinenbereitstellung entgegensteht. Hinzu kommt, dass Personalaufwände im Bereich des Sondermaschinenbaus oftmals mehr als 50% des Investitionsvolumens einnehmen. [Possel-Dölken 2010] Daraus ist ein erheblicher Fachkräftebedarf abzuleiten. Gleichzeitig ist aber ein Mangel an solchem Personal nicht nur in Deutschland auszumachen. [Bauernhansl 2012] Diese Situation führt zur Erkenntnis, dass die „Effizienz des Engineerings zum kritischen Erfolgsfaktor“ [Westkämper 2012b] auch für den Maschinenbau wird. [Westkämper 2012a]

Ausgehend von einem sequenziellen Entwicklungsprozess eröffnen sich derzeit zwei wesentliche Möglichkeiten Aufwände zu reduzieren. Dies ist entweder eine Entkopplung des Auftragsabwicklungsprozesses von der Entwicklung oder eine Beschleunigung des Prozesses an sich. [Lüder, et al. 2010] Erstere Möglichkeit beinhaltet eine sehr strikte Modularisierung der Maschinen im Rahmen der Entwicklung, was zu einer Verkürzung der Auftragsdurchlaufzeit führt. [Kiefer 2007; Litto 2004; Rodica Maga, et al. 2011; Steden, et al. 2013a] Diese Zeitreduktion wird im Wesentlichen durch die Wiederverwendung von Modulen in unterschiedlichen Maschinenprojekten realisiert. Dabei sind solche Module unter Einbeziehung aller beteiligten Disziplinen, also unter mechatronischem Gesichtspunkt zu definieren. [Litto 2004; Rodica Maga, et al. 2011] Dadurch ist eine weitere Aufwandsreduktion im Rahmen der Auftragsabwicklung möglich, indem Entwicklungsdokumente auf Basis der mechatronischen Modulkonfiguration automatisch erstellt (generiert) werden können. [Litto 2004] Solche Generatoren ermöglichen Aufwandsreduktionen im Bereich der Anlagenplanung von bis zu 93%, unter der Bedingung, dass alle Objekte vorhanden

und keine Anpassungen mehr notwendig sind. [Schlögl 2011] Diesen Einsparungen stehen erhöhte Aufwände zur mechatronischen Modularisierung im Rahmen der Entwicklung entgegen, was sich beispielsweise an der Gewährleistung der Kompatibilität von Subsystemen erkennen lässt. [Hehenberger, et al. 2010] Dieser Ansatz erweist sich spätestens dann als problematisch, wenn kundenindividuelle Anpassungen nicht mehr über Standardmodule abgefangen werden können. [Lindworsky 2008] Dies erfordert Änderungen nach einer Generierung, welche aufwendig sind und nicht unterstützt werden.

Eine Betrachtung des Sondermaschinenbaus mit seiner äußerst kundenorientierten Ausrichtung führt zum Schluss, dass die notwendige Individualität häufig nicht über eine reine Modulkonfiguration erreicht werden kann. Dadurch verteilt sich der erhöhte Modularisierungsaufwand nicht mehr auf viele Wiederverwendungen der Module in verschiedenen Projekten, wodurch der Aufwand den Gesamtnutzen übersteigt. [Possel-Dölken 2010] Potenziale, wie höhere Stückzahlen der Module und eine damit einhergehende Kostenreduktion sind somit nicht erreichbar. Damit liegt der Fokus einer Modularisierung im Rahmen der Sondermaschinenentwicklung auf der Reduktion des Entwicklungsaufwands [Possel-Dölken 2010], welcher im Wesentlichen auf eine Minderung von Wiederholertätigkeiten zurückzuführen ist. Dazu ist allerdings eine Strategie der Standardisierung nötig, welche den Einsatz bestimmter Zulieferkomponenten oder deren Einbau in das Gesamtsystem vorschreibt. [Litto, et al. 2012] Dies geht einher mit der Erkenntnis, dass im Rahmen des Sondermaschinenbaus die Perspektive mit der größten Granularität für eine Modularisierung genutzt wird. Gleichzeitig entspricht dies der Sichtweise der elektrischen Konstruktion mit ihren Standardkomponenten, welche wiederum Zukaufteile repräsentieren. [Possel-Dölken 2010] Neben der Kundenindividualität wird die Wiederverwendung durch das Wissen der beteiligten Konstrukteure über bereits existierende Lösungen beschränkt, was im Vergleich als deutlich geringere Ursache zu werten ist. [Schilke 2010] Weiterhin führen Ansätze der Modularisierung zu dezentralen Steuerungsarchitekturen. [Diedrich, et al. 2011; Weyrich, et al. 2012] Diese besitzen gegenüber einer zentralen Steuerung allerdings die Eigenschaft von erheblichen Kostensteigerungen. [Possel-Dölken 2010]

Die zweite Möglichkeit zur Reduktion von Entwicklungsaufwänden, die der Prozessbeschleunigung, wird gemeinhin als schwierig empfunden. [Kühnle 2010] Wie bereits dargelegt, sind am Entwicklungsprozess verschiedene Disziplinen beteiligt. Jeder Entwickler, geprägt von seiner Disziplin, besitzt dabei eine eigene Sichtweise mit unterschiedlichen Begrifflichkeiten [Göhner, et al. 2013], was eine kooperative Zusammenarbeit und Kommunikation der Beteiligten behindert.

[Alvarez Cabrera, et al. 2010] Zur Unterstützung der Aufgaben im Rahmen der einzelnen Disziplinen sind für jede Disziplin Softwarewerkzeuge verfügbar, beispielsweise ein MCAD-System zur Unterstützung der mechanischen Konstruktion. Daraus ist ersichtlich, dass eine weitere signifikante Verkürzung der einzelnen Phasen und somit des Gesamtprozesses nicht mehr möglich ist. [Schlögl 2008] Um dennoch einer schnelleren Bereitstellung des mechatronischen Systems „Sondermaschine“ nachzukommen, ist eine Parallelisierung der Disziplinen notwendig. [Eberle 2012; Frank, et al. 2011; Schlögl 2008]

Zur Erreichung einer möglichst breiten praktischen Akzeptanz müssen die disziplinspezifischen Methoden und somit auch die eingeführten Softwarewerkzeuge beibehalten werden. Die Planungen der einzelnen Disziplinen sind allerdings voneinander abhängig [Drath, et al. 2011a; Göhner, et al. 2013; Klemm, et al. 2006], was eine Synchronisation dieser und eine Überprüfung aus der Perspektive des mechatronischen Gesamtsystems notwendig macht. Hierzu bietet sich die Methode der Quality Gates an, welche eine solche mittels virtueller Inbetriebnahme durchführt. [Müller, et al. 2009] Dabei wird das Steuerungsprogramm gegen ein virtuelles Verhaltensmodell getestet. Aufgrund der Entkopplung des MCAD-Systems vom elektrischen CAD (ECAD-System) zur Schaltplanerstellung und der fehlenden Verhaltensmodellierung der Maschine in beiden Disziplinen muss dieses vor den Quality Gates nachgeholt werden. Dies ist gleichbedeutend mit einer Erhöhung des Aufwandes aus disziplinspezifischer Sicht.

Zusätzlich gibt es Abhängigkeiten zwischen den Disziplinen, wie beispielsweise maximale Kabellängen bei der Übertragung von Signalen, welche der Elektriker zu berücksichtigen hat. [Schreck 2007] Ermittelt werden diese Längen allerdings im Rahmen der mechanischen Konstruktion. Diese Abhängigkeiten werden durch den Einsatz der virtuellen Inbetriebnahme nicht modelliert und dadurch auch nicht überprüft, was zu einem inkonsistenten mechatronischen Gesamtsystem führen kann. Gleichzeitig veranschaulicht dieses Beispiel, dass bei der derzeitigen Entkopplung der eingesetzten Softwarewerkzeuge Informationen (beispielsweise Kabellänge in MCAD- und ECAD-System) mehrfach eingegeben werden müssen. Um das Rationalisierungspotenzial der Mehrfacheingaben zu heben ist folglich eine Verknüpfung der eingesetzten Entwicklungsumgebungen notwendig. Punkt-zu-Punkt-Schnittstellen und neutrale Austauschformate bieten hierzu eine Möglichkeit. Diese Typen der Datenübertragung sind unidirektionale Datenübertragungen, welche zu einer strikt sequenziellen Abfolge des Entwicklungsprozesses führen. [Drath, et al. 2011b] Gleichzeitig erhöhen sich, eingesetzt in parallelisierten Abläufen, Auf-

wände zur Überprüfung der Modelle in den einzelnen Disziplinen, um wieder eine mechatronische Informationskonsistenz zu erhalten. [Drath, et al. 2011b; Drath, et al. 2011a; Obersamer 2013]

Weiterhin wird in einem sequenziellen Entwicklungsprozess die wesentliche mechatronische Systemarchitektur von der Mechanik definiert. Im Zusammenhang mit dem ständig steigenden Anteil der Steuerungsprogrammierung und der dadurch bereitgestellten Mehrfunktionalität ist offensichtlich, dass somit ein mechatronisches Optimum der Systemarchitektur verhindert wird. Dieses kann nur durch einen interdisziplinären Systementwurf erreicht werden. [Alvarez Cabrera, et al. 2010; Göhner, et al. 2013; Litto 2010] Zur Beschreibung dieser Phase und Kommunikation über die verschiedenen Disziplinen hinweg wird ein interdisziplinär verständliches Mittel benötigt. Dieses Mittel muss nicht nur multidisziplinär beschreibbar sein, sondern auch eine frühe Verifikation der Planungen ermöglichen. [Alvarez Cabrera, et al. 2010; Göhner, et al. 2013] Zur interdisziplinären Kommunikation wird üblicherweise eine funktionale Modellierung genutzt. [vgl. u. a. Bathelt 2007; Gehrke 2005; Kiefer 2007; Kleiner 2003, Kleiner 2013; Lindworsky 2011; Litto 2004] Allerdings sind zwei wesentliche Interpretationen des Begriffs „Funktion“ zu unterscheiden. Die erste klassifiziert Ressourcen oder Prozesse in Fabriken, als Beispiel seien hier „Rollenförderer“ oder „Fördern“ genannt. Diese Interpretation findet vor allem in den Ansätzen der Modularisierung Anwendung. [vgl. u. a. VDW; Kiefer 2007; Litto 2004] Nachteilig ist hier der sofortige Fokus auf einer Wiederverwendung, welche der kundenspezifischen Ausrichtung im Bereich der Sondermaschinen im Wege steht.

Die zweite Interpretation besitzt ihren Ursprung im Bereich der Problemlösung beziehungsweise der Produktentwicklung, wo sie abstrakt zu definieren ist, um möglichst viele Lösungsprinzipien zu finden. [vgl. u. a. Bathelt 2007; Kleiner 2003, Kleiner 2013; Pahl, et al. 2007] Dies unterstützt die kundenindividuelle Ausrichtung des Sondermaschinenbaus, genau jene Maschine zu entwickeln, welche der Kunde tatsächlich benötigt. Dagegen berücksichtigen diese Methoden keine Wiederverwendung und keine Integration der Maschine in ein bestehendes Produktionssystem.

Somit ist derzeit keine Methode zur Beschreibung eines mechatronischen Systementwurfs für Sondermaschinen bekannt, was beinhaltet, dass auch keine ganzheitliche Softwareunterstützung hierfür vorhanden ist. Außerdem wird in beiden Stoßrichtungen keine Integration der zu entwickelnden Maschine in das bestehende Produktionssystem explizit berücksichtigt. Gerade aber eine solche führt zu erheblichen Einsparungen während der Produktion. [Böck 2013] Aus dieser Perspektive sind Anforderungen an die zu entwickelnde Maschine für den Maschinenhersteller aus Sicht der Fabrikplanung nur unzureichend definiert. Das Reallayout, als Ergebnis des

Fabrikplanungsprozesses und Basis der Feinplanung, fokussiert die Anordnung, den Raumbedarf und eventuell noch Versorgungsanschlüsse der benötigten Maschine. [vgl. Grundig 2006; Wiendahl 2005] Aufgrund der mechatronischen Maschinenausprägung ist dadurch zumindest die Automatisierung unterrepräsentiert, was bei der Inbetriebnahme zu erheblichen Verzögerungen aufgrund von Inkompatibilitäten zu anderen Maschinen oder übergeordneten Leitständen führen kann.

Eine weitere Problematik des sequenziellen Entwicklungsprozesses stellen lange Iterationszyklen und eine späte Fehlererkennung dar. [Würslin, et al. 2008] Ein Prototypenbau ist durch den Unikatcharakter von Sondermaschinen zu aufwendig, was viele Unsicherheiten bzgl. wichtiger Anforderungen, wie beispielsweise der zu erreichenden Taktzeit birgt. [Frank, et al. 2013a; Frank, et al. 2013b]

Eine Entwicklungsmethodik für mechatronische Systeme, welche bereits eine Parallelisierung der Disziplinen adressiert, bietet die *VDI-Richtlinie 2206* [2004-06]. Diese ist aber im Hinblick auf die dargestellten Probleme der Sondermaschinen zu abstrakt gehalten. Zusammenfassend lassen sich folgende wesentliche Defizite festhalten:

- Eine Wiederverwendung von mechatronischen Modulen über verschiedene Aufträge bzw. Projekte hinweg im Sondermaschinenbau ist derzeit nur bedingt gegeben, was zu brachliegendem Rationalisierungspotenzial führt.
- Es besteht ein hoher Aufwand zur Erstellung der mechatronischen Module vorab, wobei eine kundenindividuelle Ausprägung, wie sie im Bereich der Sondermaschinen benötigt wird nur unzureichend unterstützt wird.
- Eine parallelisierte Durchführung der disziplinspezifischen Entwicklung ist derzeit nicht möglich,
 - da eine Ausgestaltung der interdisziplinären Systementwurfsphase für alle beteiligten Disziplinen nicht beschrieben und nicht softwareunterstützt ist,
 - die virtuelle Inbetriebnahme allein nicht für interdisziplinäre Konsistenz der Modelle sorgen kann und
 - die Softwarewerkzeuge immer noch voneinander entkoppelt sind.
- Informationen müssen mehrfach in verschiedenen Softwarewerkzeugen eingegeben werden, was zu Inkonsistenzen und Mehraufwänden führt.
- Die Anforderungen an eine neue in das Produktionssystem zu integrierende Maschine sind von der Fabrikplanung nicht ausreichend definiert.

- Eine kooperative Zusammenarbeit und Kommunikation der Beteiligten über Disziplinen hinweg ist nur unzureichend unterstützt.

Die beschriebenen Defizite zeigen zwei wesentliche Handlungsfelder auf. Zunächst muss der Entwicklungsprozess methodisch unterlegt werden, damit die Kundenindividualität der Maschinen und deren Aufbau berücksichtigt werden. Darauf aufbauend ist eine softwaretechnische Unterstützung der Methode notwendig.

1.3 Zielsetzung und Aufgabenstellung

Die im vorigen Kapitel aufgezeigten Defizite veranschaulichen, dass ein Bedarf einer durchgängigen Methode zur Beschleunigung der Entwicklung von Sondermaschinen besteht. Dies führt zu folgender Zielsetzung der Arbeit:

„Es soll eine durchgängige softwaregestützte Methode zur integrierten mechatronischen Sondermaschinenentwicklung von der Anforderungsdefinition bis hin zur Inbetriebnahme erarbeitet werden, welche Disziplinen parallelisiert und eine Wiederverwendung in anderen Projekten auch bei Unikaten mit sehr geringem Aufwand ermöglicht.“

Dadurch sollen Sondermaschinenhersteller in die Lage versetzt werden den Entwicklungs- oder Engineeringaufwand für mechatronisch optimierte und an die Kundenbedürfnisse angepasste Maschinen deutlich zu reduzieren. Dies soll den Unternehmen helfen Kosten einzusparen und somit wettbewerbsfähig zu bleiben. Die Basis soll eine detaillierte Analyse des Informationsbedarfs und der -entstehung in den einzelnen Disziplinen bilden. Darauf aufbauend sollen die Abhängigkeiten dieser zueinander, auch unter dem Gesichtspunkt der zeitlichen Abfolge, betrachtet und in Relation zueinander gesetzt werden. Dadurch soll es möglich sein eine Methode zum parallelen Engineering in den Disziplinen Mechanik, Elektrik und Automatisierung bzw. SPS-Programmierung zu entwickeln. Dies erfordert neben einem planmäßigen in Phasen unterteilten Vorgehen auch die Definition der einzusetzenden Beschreibungsmittel und der Phasenübergänge. Gleichzeitig soll diese Methode das Grundgerüst für eine durchgängige softwaretechnische Unterstützung sein. Alle diese Phasen mit ihren definierten Beschreibungsmitteln sollen softwaretechnisch, wenn möglich mit in der Praxis eingeführten Softwarewerkzeugen, unterstützt werden. Dies erfordert eine Auswahl verschiedener Softwarewerkzeuge zur Unterstützung der einzelnen Phasen. Die Durchgängigkeit entlang des gesamten Prozesses soll sich somit aus der Verknüpfung verschiedener Softwarewerkzeuge ergeben.

1.4 Vorgehensweise und Aufbau der Arbeit

Der Inhalt der Arbeit ist in sieben Kapitel gegliedert. Einführend wurde die Situation produzierender Unternehmen und die sich daraus ergebende Ausgangssituation für Maschinenhersteller als deren Kunden erläutert. Darauf aufbauend wurde die sich ergebende Problematik in der Entwicklung von Sondermaschinen dargestellt, woraus sich die Zielsetzung und Aufgabenstellung im vorigen Kapitel ergibt. Kapitel zwei beginnt mit Grundlagen zu disziplinspezifischen Entwicklungsvorgehen und mechatronischen Systemen bevor problem-relevante aktuelle Ansätze mechatronischer Entwicklungsmethoden, Ansätze zur Aufwandsreduktion und Softwareintegration diskutiert werden.

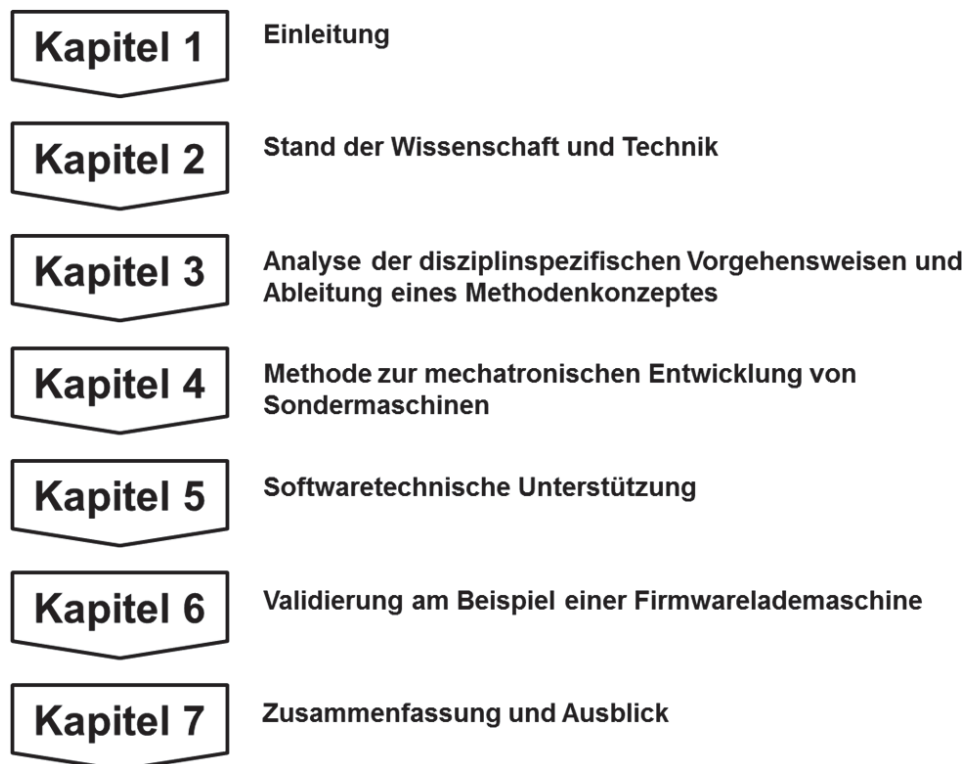


Abbildung 1.1: Vorgehensweise und Aufbau der Arbeit

Beendet wird das Kapitel mit der Ableitung von Anforderungen auf Basis, der in Kapitel eins dargelegten Defizite, der Bewertung des Stands der Technik und der Ableitung des Handlungsbedarfs. In Kapitel drei werden das Entwicklungsvorgehen und die Modellbildung der Mechanik, Elektrik und Automatisierung hinsichtlich Informationsbedarf und -entstehung analysiert und gegenübergestellt. Als Ergebnis wird daraus das Basiskonzept einer auf mechatronische Sondermaschinenentwicklung ausgerichteten Methode vorgestellt. Diese wird in Kapitel vier

zunächst in verschiedene Phasen aufgeteilt, welche im Anschluss detailliert vorgestellt werden. Kapitel fünf zeigt auf, wie verschiedene Softwarewerkzeuge auf Basis von ermittelten Zusammenhängen aus Kapitel drei integriert werden können. Weiterhin wird der Aufbau von wiederverwendbaren mechatronischen Objekten diskutiert und die Architektur des entwickelten Prototyps vorgestellt. Kapitel sechs beschreibt an einem realen Beispiel einer Sondermaschine wie eine mechatronische Entwicklung auf Basis der hier erarbeiteten Methode abläuft. Die Erkenntnisse dieser prototypischen Umsetzung werden hier ebenfalls aufgezeigt und diskutiert. Abschließend wird die Arbeit in Kapitel sieben zusammengefasst und ein Ausblick hinsichtlich des weiteren Forschungsbedarfs gegeben.

2 Stand der Wissenschaft und Technik

Entwicklungsprozesse sind gleichzusetzen mit Problemlösungsprozessen. [Suh 2001] Nach *Suh* [2001] sind dabei fünf Schritte zu durchlaufen: den Kunden verstehen, das Problem definieren, das Konzept durch Synthese erstellen, dieses analysieren und die resultierende Lösung gegen die Kundenanforderungen überprüfen. Diese Basisschritte werden in den folgenden beiden Grundlagenkapiteln hinsichtlich des derzeit vorherrschenden sequenziellen Entwicklungsvorgehen und des mechatronischen Systems „Sondermaschine“ aufgegriffen. Es folgt die Vorstellung derzeitiger Ansätze zur Entwicklung mechatronischer Maschinensysteme und deren Auswirkungen auf die Reduktion von Entwicklungsaufwänden. Danach werden die Möglichkeiten zur Verknüpfung verschiedener Softwarewerkzeuge vorgestellt. Abschließend erfolgt die Definition von Anforderungen an die Methode, deren Softwareunterstützung und die Bewertung der vorgestellten Ansätze. Daraus leitet sich der Handlungsbedarf ab.

2.1 Disziplinspezifische Entwicklungsmethoden und sequenzieller

Maschinenentwicklungsprozess

Wie bereits erläutert, herrscht derzeit ein sequenzieller Entwicklungsprozess im Sondermaschinenbau vor, welcher mit der Konstruktion der Mechanik beginnt, von der elektrischen gefolgt und mit der automatisierungstechnischen Steuerungsprogrammierung beendet wird.

2.1.1 Mechanik

Die im Rahmen einer Sondermaschinenentwicklung zu automatisierenden Produktionsprozesse beinhalten wenigstens das Handling des zu bearbeitenden Werkstückes oder des Ausgangsmaterials. Daraus ist ersichtlich, dass sich die mechanische Disziplin mit der Anordnung und der kinematischen Verknüpfung real existierender Bauteile auseinandersetzt.

Als Basis der mechanischen Konstruktion kann die vom Verein Deutscher Ingenieure (VDI) veröffentlichte *Richtlinie 2221* [1993-05] angesehen werden. Den darin beschriebenen Ablauf zeigt Abbildung 2.1. Dieses Grundvorgehen deckt sich mit den von *Suh* [2001] aufgestellten Aussagen. Eine Detaillierung des Vorgehens hinsichtlich der mechanischen Konstruktion werden durch *Pahl et al.* [2007] gegeben. Dabei wird zunächst die Phase zur Planung und Klärung der Aufgabenstellung eingeleitet. Hierin werden die Anforderungen des Kunden erhoben. *Pahl et al.* [2007] geben hierzu Hauptmerkmalleitlinien aus, um die Erhebung möglichst vollständig zu

erhalten. Nebenbei sind die Anforderungen möglichst quantitativ zu beschreiben. Am Ende steht die sogenannte Anforderungsliste, welche gleichzeitig das Lastenheft darstellt.

Im zweiten Schritt muss zunächst das Hauptproblem abstrahiert werden. Ziel ist es das wesentliche Problem zu erkennen und zu formulieren. [Pahl, et al. 2007] Dazu eignen sich Funktionen. Im Rahmen der mechanischen Entwicklung beschreiben diese zunächst das Verhalten von Produkten oder Teilen davon, oftmals nur das gewünschte oder gewollte. [VDI 2222 Blatt 1] Als Beispiel hierzu sei die Funktion „Benzin fördern“ genannt. Gleichzeitig bildet dies den richtigen Abstraktionsgrad um einen möglichst großen Lösungssuchraum aufzuspannen. Relationen zwischen Funktionen führen auf eine sogenannte Funktionsstruktur oder -gliederung. [DIN 1325 Blatt 1] Die bekannteste Art einer solchen Struktur ist der Funktionsbaum. Im Umfeld des Sondermaschinenbaus wird eine funktionale Modellierung meist nur implizit von den mechanischen Entwicklern durchgeführt.

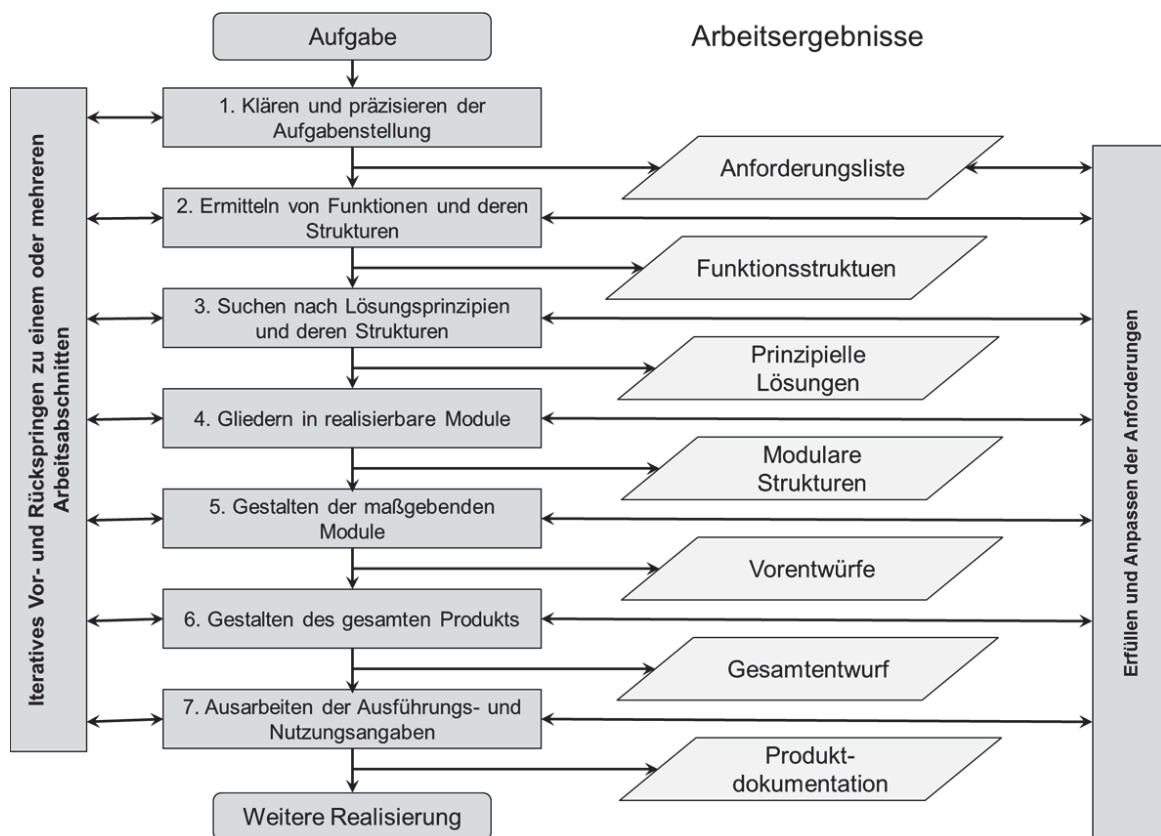


Abbildung 2.1: Grundvorgehen zum Entwickeln von mechanischen Produkten nach VDI-Richtlinie 2221 [1993-05]

Im dritten Schritt werden für die definierten Funktionen Lösungsprinzipien gesucht. Ein Lösungsprinzip beschreibt nach *VDI-Richtlinie 2222* [1997-06] „unscharfe bzw. grobe aber funktionsbestimmende Vorstellungen zur Realisierung durch die Einbeziehung von Effekten“. Dieser Übergang ist aus methodischer Sicht mit Kreativitätsmethoden zu unterstützen. Bezogen auf die Entwicklung von Sondermaschinen sind solche Lösungs- oder Wirkprinzipien häufig mit kinematischen Konzepten gleichzusetzen. Die gefundenen Prinzipien können in einem morphologischen Kasten dokumentiert werden. Gleichzeitig bietet dieser die Möglichkeit Prinzipien zu mehreren Varianten von verschiedenen Gesamtlösungen oder Wirkstrukturen zu kombinieren. Die mechanischen Konstrukteure dokumentieren ihre Wirkprinzipien häufig sehr rudimentär in Form von Skizzen, selten jedoch explizit in einem MCAD-System. In der Phase zur Gliederung in realisierbare Module wird das Gesamtkonzept aufgeteilt. Solche Module stellen entweder kinematisch voneinander entkoppelte, verschiedene Verfahren oder Baugruppen mit einer Bestellnummer dar. Gleichzeitig führt dies auf eine Baugruppenstruktur im MCAD-System.

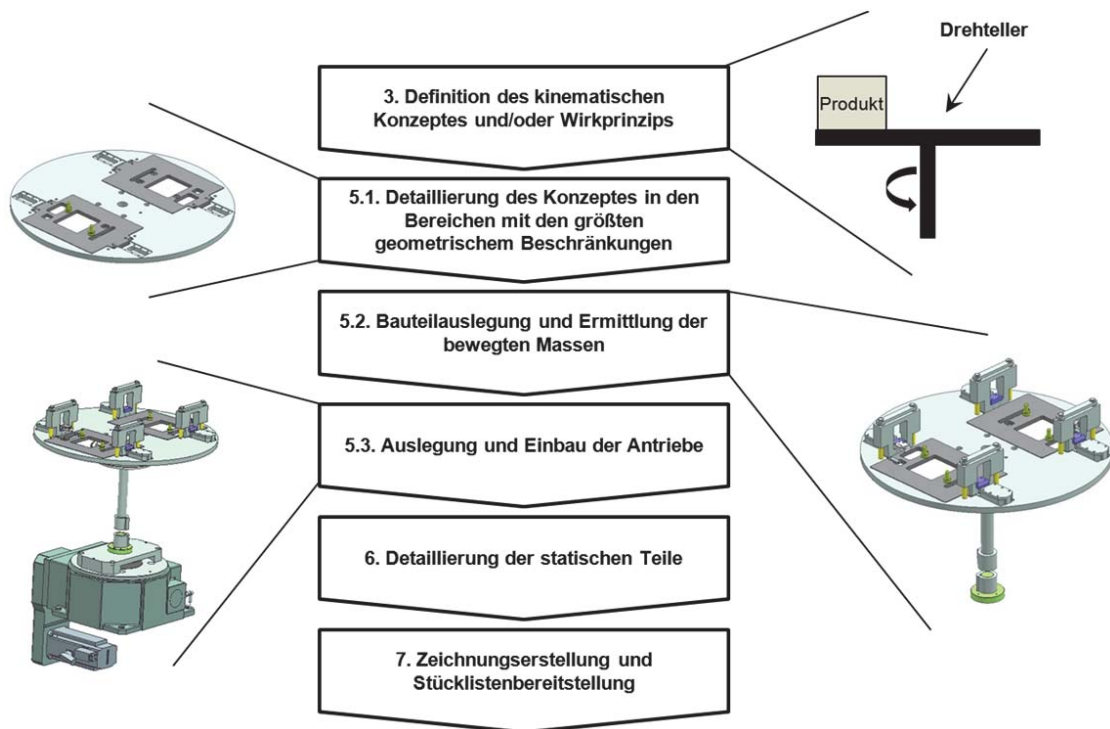


Abbildung 2.2: Ablauf der mechanischen Konstruktion in Bezug auf die in der *VDI-Richtlinie 2221* genannten Schritte drei bis sieben im Sondermaschinenbau

Der Detaillierungsablauf eines Moduls vom Wirkprinzip bis hin zur Gesamtdokumentation zeigt Abbildung 2.2. Den Einstieg bildet das definierte Wirkprinzip und die Modulbildung aus den

Schritten drei und vier der *VDI-Richtlinie 2221*. Die Schritte 5.1, 5.2 und 5.3 sind somit Inhalt der Phase fünf „Gestalten der maßgebenden Module“; entsprechend sind die Phasen sechs und sieben der Abbildungen zu entnehmen. Die Detaillierung der Module wird in jenen Bereichen begonnen, welche die größten kinematischen und geometrischen Beschränkungen besitzen, um die Umsetzbarkeit der Prinziplösung möglichst frühzeitig zu verifizieren. Ist eine solche gegeben, sind die bewegten Teile geometrisch zu detaillieren und weitere notwendige Teile hinzuzufügen. Ziel ist die Bestimmung der bewegten Massen um Kräfte und Momente zu berechnen, welche für die weitere Auslegung der Antriebe und Getriebe notwendig sind.

In Phase sechs werden statische Teile und Baugruppen hinzugefügt, wie beispielsweise das Gehäuse. Gleichzeitig werden die Module zusammen in eine Gesamtbaugruppe integriert. Am Ende sind zum Aufbau Stücklisten, Fertigungszeichnungen und Zusammenbauzeichnungen bereitzustellen. Moderne MCAD-Umgebungen erlauben eine automatische Bereitstellung der Stücklisten. Zeichnungen können mittels Ableitung aus dem 3D-Modell weitgehend ebenfalls sehr aufwandsminimiert erstellt werden. Gleichzeitig ermöglicht dies eine Quasi-Parallelisierung der Zeichnungserstellung zu Schritt fünf und sechs, sobald ein detailliertes 3D-Modell vorliegt.

2.1.2 Elektrik

Kinematische Bewegungen von Sondermaschinen werden beispielsweise durch Motoren, Pneumatik- oder Hydraulikzylinder initiiert. Die Beispiele veranschaulichen, dass nicht nur die Elektrizität, sondern auch Medien, wie Luft oder Öl zur Energiebereitstellung und -transformation genutzt werden können um mechanische Energieformen zu erhalten. Abbildung 2.3 zeigt elektrische und fluidtechnische Kreisläufe nach *Grätz* [2006], die für eine gesteuerte Ausführung von Bewegungen notwendig sind. Die Aufgabe der Elektrokonstruktion besteht darin, die in den Kreisläufen notwendigen Komponenten miteinander zu verschalten. Dies geschieht üblicherweise mit ECAD-Systemen. Da in diesen Programmen ebenfalls die Verschaltung von pneumatischen und hydraulischen Komponenten möglich ist, wird dieses Aufgabenfeld im Rahmen dieser Arbeit ebenfalls im Bereich der Elektrik behandelt. Üblicherweise wird die hydraulische Auslegung, aufgrund der hohen Wechselwirkungen zwischen den mechanischen Bauteilen und der Hydraulik, von speziell ausgebildeten Konstrukteuren oder Mechanikern übernommen. Die pneumatische Verschaltung erfolgt organisatorisch häufig in der Elektroabteilung.

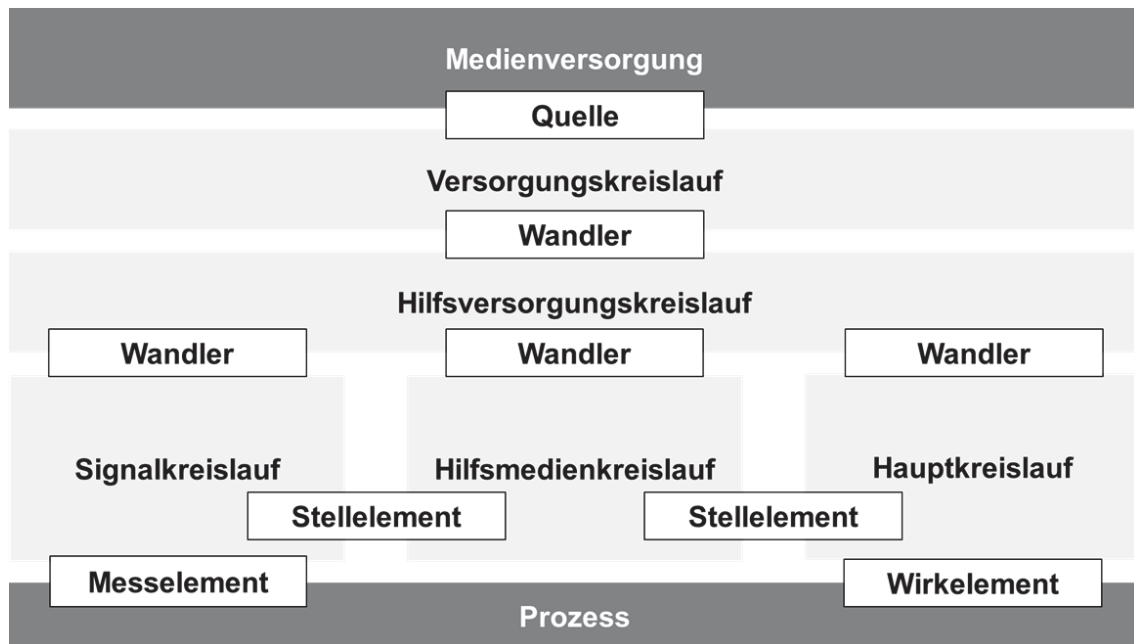


Abbildung 2.3: Kreisläufe der elektrischen und fluidtechnischen Konstruktion [Grätz 2006]

Im sequenziell durchgeführten Maschinenentwicklungsprozess ist die mechanische Konstruktion vorgelagert. Als Eingangsinformationen in die Elektrik sind somit alle Wirkelemente, für die Hydraulik auch Stellelemente, und Messelemente bzw. Sensoren definiert. Diese sind üblicherweise in einer sogenannten Sensor-/Aktorliste festgehalten. [vgl. VDW] Weiterhin ist der Aufbau der Maschine durch die Baugruppenstruktur und der geometrischen Beschreibung in 3D gegeben. Daraus wird der Bewegungsablauf abgeleitet. Entweder explizit in Form von Diagrammen durch den mechanischen Konstrukteur oder auf Basis von mündlicher Erklärung. Diese Informationsbereitstellung bildet den Startpunkt für die Elektrokonstruktion, die ihrerseits beginnt das Projekt zu strukturieren.

Diese Strukturierung ist in *DIN EN 61346 Blatt 1* [1997-01], *DIN EN 61346 Blatt 2* [2000-12] bzw. *DIN EN 81346 Blatt 1* [2010-05], *DIN EN 81346 Blatt 2* [2010-05] genormt. Die Normung sieht insgesamt drei Aspekte, den funktionalen, produkttechnischen und ortsabhängigen Aspekt vor. Dabei sind diese Aspekte als Hierarchien anzusehen, welche miteinander in Relation stehen, die Maschine aber aus unterschiedlichen Gesichtspunkten strukturieren. Derzeit setzt kein ECAD-System die Normen eins zu eins um. Häufig gehen der Funktions- und der Produktaspekt in einer gemeinsamen Hierarchie, einer sogenannten Maschinen- oder Anlagenstruktur auf. [vgl. hierzu Siemens Industry Software, 2010b] Abhängig von den Eingangsinformationen beginnt der Elektrokonstrukteur mit einer Modularisierung der Maschine in einer der beiden bzw. dreien

Hierarchien, häufig zunächst nach funktionalen Gesichtspunkten. Eine Funktion ist hierbei mit einem Prozessschritt, wie „Schneiden“ gleichzusetzen. Darunter werden Module definiert, welche Teilprozesse ausführen und Messelemente und Wirkelemente enthalten. Ein Beispiel für ein solches Modul wäre ein „Greifer“. [vgl. VDW] Dies führt zur Maschinenstruktur, welche in Abbildung 2.4 im ersten Schritt dargestellt ist.

Im nächsten Schritt werden den definierten Modulen, die in der Sensor-/Aktorliste enthaltenen Mess- und Wirkelemente zugeordnet. Dabei ist es zulässig, Messelemente auch unter Wirkelementen zu platzieren, was ausdrückt, dass diese Sensoren beispielsweise für die Positionsdetektion des Wirkelementes verantwortlich sind. Dieses Modell bildet die Ausgangssituation für den Entwurf, der in Abbildung 2.3 dargestellten Kreisläufe.

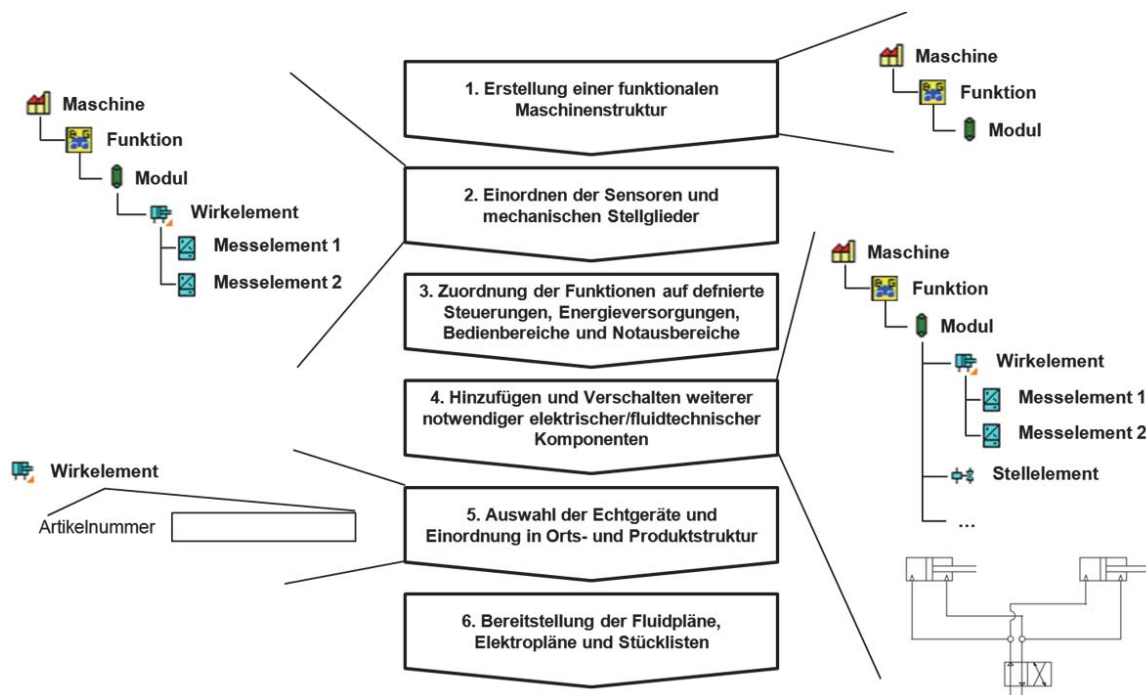


Abbildung 2.4: Ablauf der Elektrokonstruktion

Bei Maschinen mit großem Umfang werden in einem nächsten Schritt den Funktionen und Module entsprechend, eine Steuerung, die Energieversorgung, sowie Bedien- und Notausbereiche zugeordnet. Zur Verschaltung und somit dem Aufbau der Kreisläufe werden weitere Komponenten, wie Schütze, Schalter, Kabel und Schläuche benötigt, welche dann auf Schaltplänen miteinander verbunden werden. Dabei werden bestellbare Komponenten klassifiziert und diesen Klassen Symbole zugewiesen. Diese sind nach *DIN EN 60617 Blatt 2* [1997-08], *DIN EN ISO 1219 Blatt 1*

[2007-12] und *DIN ISO 7000* [2008-12] genormt. Die Auswahl und Verschaltung der Komponententypen kann von der Abfolge der Maschinenbewegungen abhängig sein. So wird beispielsweise bei einem gleichzeitigen Ausfahren zweier pneumatischer Zylinder häufig nur ein Ventil verwendet.

Im weiteren Verlauf werden ausgehend von den Komponententypen entsprechende real bestellbare Komponenten durch die Definition ihrer Bestellnummern ausgewählt. Dies wird auch als Echtgeräteauswahl bezeichnet. Diese Geräte müssen ebenfalls in die verschiedenen Aspekte eingeordnet werden. Als Ergebnis stehen die Schaltpläne der Elektrik, der Hydraulik und Pneumatik zur Verfügung. Stücklisten werden auf Basis der definierten Bestellnummern ebenfalls ausgeleitet. Außerdem ist dadurch der hardwaretechnische Aufbau der SPS (Hardwarekonfiguration) gegeben. Weiterhin ist durch die Verdrahtung der Steuerungsmodule mit den entsprechenden Messelementen und Stellgliedern die Hardwareadresse der Signale bekannt. Diese werden in einer sogenannten Signalliste bereitgestellt.

2.1.3 Automatisierung

Die Steuerung von Sondermaschinen ist üblicherweise mit einer zentralen SPS ausgeführt. Diese setzt sich aus unterschiedlichen Hardwarekomponenten zusammen, einer Komponente für die Stromversorgung, einer Zentralbaugruppe mit Prozessormodul und Einsteckmodulen. Letztere, auch als Input/Output-Karten (I/O-Karten) oder Peripheriebaugruppen bezeichnet, binden je nach Maschinenaufbau Messelemente, Stellelemente oder andere Geräte mit Signalschnittstelle an das Steuerungsprogramm an. [Seitz 2012] Hieraus ist direkt ersichtlich, dass die Hardwarekonfiguration der Steuerung durch die vorzusehenden Stell- und Messelemente bestimmt und durch die Elektrik definiert wird. Die Abarbeitung eines Steuerungsprogramms erfolgt in zyklischer Weise, indem zunächst alle Eingänge eingelesen, dann das Programm auf Basis dieser Eingänge abgearbeitet und danach die Ausgänge gleichzeitig gesetzt werden. [Pritschow 2006]

Die Aufgabe der Automatisierung ist die Bereitstellung des Steuerungsprogramms, welches den Prozess automatisiert ablaufen lässt, diesen wiederholt und die Maschine sicher betreibt. Im Rahmen einer kontinuierlichen Produktion, wie der Prozessindustrie, dient ein sogenanntes Rohrleitungs- und Instrumentierungsschema (R&I-Fließbild) nach *DIN EN 62424* [2010-01] als Basis der Steuerungsprogrammierung. [Seitz 2012] Aufgrund der kontinuierlichen Produktion können hierin gleichzeitig der Ablauf und die benötigten Komponenten beschrieben werden. Im Bereich der diskreten Produktion kann dies so nicht umgesetzt werden. Somit wird neben der beschriebenen Hardwarekonfiguration der Steuerung aus der Elektrik, ein weiteres Dokument die

sogenannte Achsliste benötigt. In ihr ist die Zuordnung von Signalen auf bewegte Achsen bzw. Aktoren definiert. Diese Zuordnung wird ergänzt um die Signalliste, welche eine Zuordnung der symbolischen Namen der Signale auf die entsprechenden Hardwareadressen der I/O-Module enthält. Diese wird auch als Symboltabelle bezeichnet. Dadurch ist der Steuerungsprogrammierer in der Lage die gewünschte Aktion durch das Setzen der entsprechenden Adresse anzusteuern.

Der Sollablauf stellt die ideale Abfolge der automatisierten Teilprozesse der Maschine dar. Dieser liegt, wenn explizit vorhanden, von der Mechanik in Form als Gantt-Charts oder SFC vor. [Bathelt 2007] Dadurch ist bereits der erste Schritt zum Erstellablauf eines SPS-Programms, wie in Abbildung 2.5 dargestellt, beschrieben.

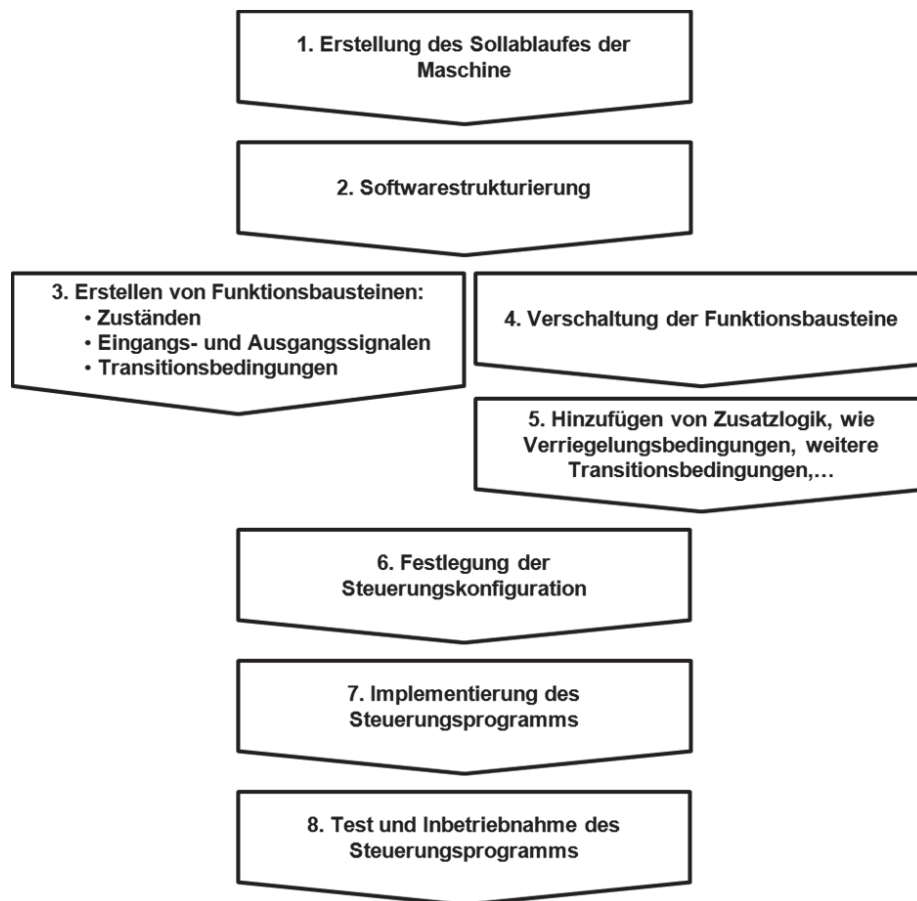


Abbildung 2.5: Ablauf der SPS-Programmierung übertragen in Anlehnung an *Seitz* [2012]

Im Anschluss daran ist es notwendig eine Softwarestrukturierung durchzuführen. Dabei werden alle zu steuernden Elemente, beispielsweise ein Ventil, einem Gerätetyp zugeordnet. Für diese Gerätetypen werden dann, wenn nicht bereits vorhanden, in Schritt drei auf Basis ihrer Zustände

und Signalschnittstellen sogenannte Programmorganisationseinheiten oder Funktionsbausteine entwickelt. [Seitz 2012] Dies geschieht in steuerungsherstellerspezifischen Umgebungen. Die Programmierung der einzelnen Programmorganisationseinheiten (POEs) erfolgt nach den genormten Sprachen der *DIN EN 61131 Blatt 3*. [Seitz 2012] Dies sind die Anweisungsliste, strukturierter Text, Ablaufsprache, was mit SFCs gleichzusetzen ist, Funktionsbausteine und Kontaktplan. [DIN 61131 Blatt 3] Nach *Hundt* [2012] ist es möglich Gantt-Charts in SFCs zu überführen. Die Programmierung des Steuerungsprogramms beginnt mit der Instanziierung der entwickelten Funktionsbausteine für die Gerätetypen.

Durch Verschalten der Funktionsbausteine mittels Übergangs- bzw. Transitionsbedingungen wird zunächst der Sollablauf programmiert. Im Anschluss daran werden Alternativabläufe, Sicherheits-, Unterbrechungs- und Wiedereinstiegsfunktionen berücksichtigt, was durch entsprechende Zusatzlogik bzw. Verriegelungsbedingungen umgesetzt wird. Dabei können zur horizontalen Kommunikation zwischen mehreren Bausteinen globale Instanzvariablen bzw. Datenbausteine benötigt werden. Mess- oder Stellsignale sind aus der Zuordnung der Signalliste auf die Hardwareadressen herauszulesen und als lokale Variable anzugeben. Falls noch nicht vorab geschehen, wird in Schritt sechs auf Basis der Elektroplanungen die Hardwarekonfiguration angegeben. Spätestens in Schritt sieben sind mit Hilfe der Signalliste die Hardwareadressen der Signale zuzuordnen. Weiterhin erfolgt eine Taskkonfiguration. [Seitz 2012] Zum Test des Programms wird dieses auf die SPS geladen und modulweise getestet. Neben dem reinen Steuerungsprogramm kommt der automatisierungstechnischen Disziplin die Erstellung des Human-Machine-Interface (HMI) zu. Dieses ermöglicht dem Bediener, Parameter des Steuerungsprogramms zu ändern oder auszulesen.

2.1.4 Zusammenfassung

Zusammenfassend zeigt Abbildung 2.6 den sequenziellen Entwicklungsprozess von Sondermaschinen auf. Darin ist zu erkennen, dass alle Phasen sehr gut softwareunterstützt sind, so beispielsweise durch ein MCAD-System für die mechanische Konstruktion.

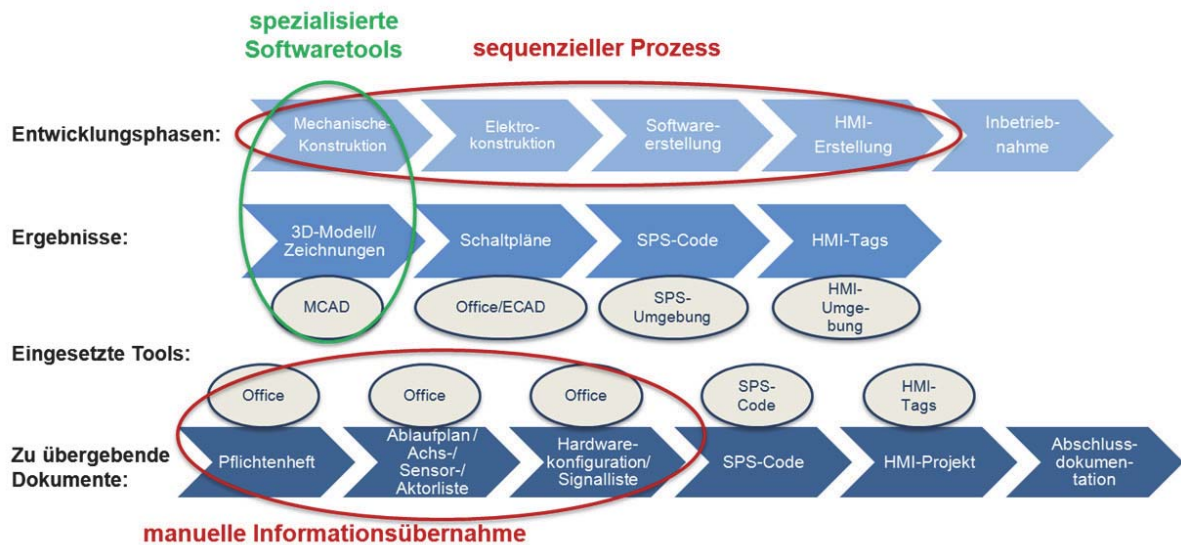


Abbildung 2.6: Derzeitiger Entwicklungsprozess von Sondermaschinen [in Anlehnung an Bleisteiner 2010]

Die Informationsübernahme zwischen den Disziplinen erfolgt allerdings meist auf Basis von Dokumenten, wie einer Sensor-/Aktorliste in einer Office-Datei.

2.2 Mechatronische System- und Begriffsdefinitionen für SPS-gesteuerte Sondermaschinen

Nach *Haberfellner* [2002] kann ein System frei definiert werden und besteht aus Teilen oder Elementen, welche in Beziehung zueinander stehen. Diese Elemente können wiederum als Systeme betrachtet werden und mit anderen in Beziehung stehen. Dabei verfolgen die Systembausteine ein gemeinsames Ziel, welches von den einzelnen nicht erreicht werden kann. [Weilkiens 2006] Die Grenze der Beziehungen eines Systems zu einem anderen wird gemeinhin als Systemgrenze bezeichnet. Sondermaschinen können als mechatronische Systeme aufgefasst werden. Die *VDI-Richtlinie 2206* [2004-06] besitzt das Ziel eine Methode zur Entwicklung mechatronischer Systeme bereitzustellen. Nach *Kleiner* [2013] basiert sie auf den Gedanken des Systems Engineering. Daher wird zunächst das mechatronische System in Bezug auf SPS-gesteuerte Sondermaschinen erläutert. Im Anschluss daran findet sich ein Überblick der Gedanken des Systems Engineering, bevor dessen Adaption auf mechatronische Systeme nach *VDI-Richtlinie 2206* erläutert wird.

2.2.1 Mechatronisches System

Den Grundaufbau eines mechatronischen Systems zeigt Abbildung 2.7. Dieser Aufbau kann auch auf SPS-gesteuerte Maschinen übertragen werden. [Bathelt 2007] Das Grundsystem ist dabei die mechanische Ausprägung der Maschine. Sensoren messen Stellgrößen des Grundsystems und melden diese in Form von Signalen an die Informationsverarbeitung. Diese ist im Falle von Sondermaschinen durch die SPS repräsentiert.

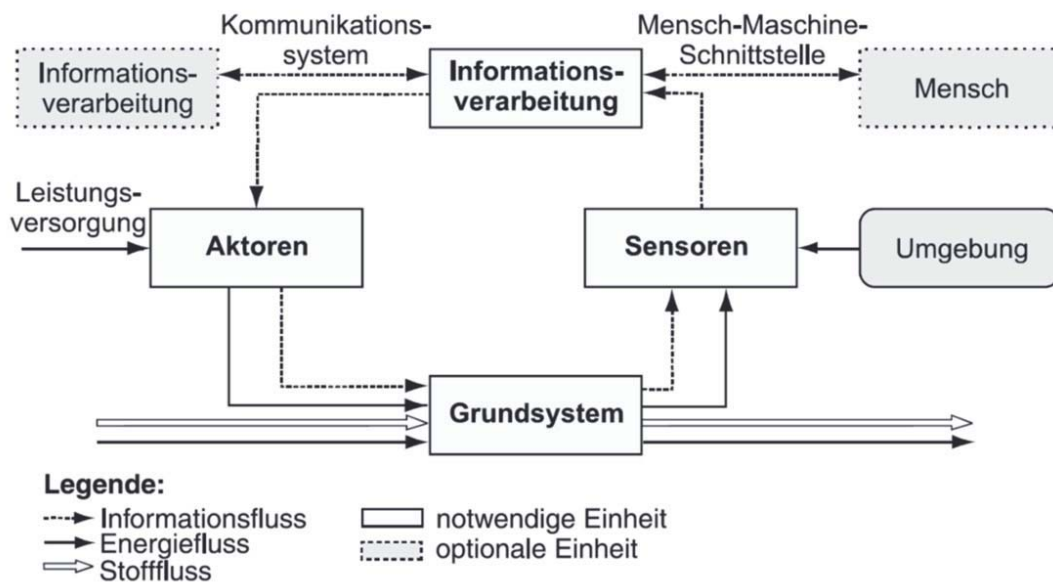


Abbildung 2.7: Grundstruktur mechatronischer Systeme nach VDI-Richtlinie 2206 [2004-06]

Wie bereits in Kapitel 2.1.3 dargelegt, bilden die HMIs die Mensch-Maschine Schnittstelle ab. Über Peripheriemodule kann die Kommunikation zu weiteren Informationsverarbeitungen aufgebaut oder Ausgangssignale für Aktoren gesetzt werden. Die Aktoren wandeln Energie in Abhängigkeit der Steuerungssignale so um, dass diese auf das mechanische Grundsystem wirkt. Dabei werden Aktoren und Sensoren mitsamt der Signalschnittstelle definiert. Im Gegensatz hierzu besitzen die in Kapitel 2.1.2 eingeführten Wirkelemente häufig keine Signalschnittstelle, obwohl sie direkten Einfluss auf das mechanische Grundsystem nehmen. Die Signalschnittstelle wird von den Stellgliedern ohne direkten Einfluss auf das Grundsystem bereitgestellt. Dadurch enthält der Begriff Aktor aus Sicht der VDI-Richtlinie 2206 [2004-06] alle Komponenten, welche notwendig sind um sowohl die Signal-, die Versorgungsschnittstelle als auch die Wirkung auf das Grundsystem bereitzustellen.

physikalischen Architektur mindestens eine Funktion erfüllen. [Leonard 2001] *Pörnbacher* [2011] unterscheidet diese beiden Ebenen als präskriptiv (definierend) und deskriptiv (beschreibend). Rückwirkend, im Rahmen der Designschleife wird überprüft, wie das System die Funktionen umsetzt. [Leonard 2001] Die Systemanalyse evaluiert verschiedene alternative Ansätze, um die technischen Anforderungen zu befriedigen. Softwarewerkzeuge stellen hierbei den Input bereit um Analyseaktivitäten wie Modellierung, Simulation und Experimente zu unterstützen. Lösungsalternativen werden auf Basis der Systemauswirkung ausgewählt. [Leonard 2001] Hierzu haben sich gerade im Bereich der Softwareentwicklung modellbasierte Methoden herausgebildet. Diese setzen anstatt textuellen Beschreibungen der Aufgaben Modelle ein. Die Implementierung wird manuell durchgeführt. [Scheibler 2010] Die Weiterentwicklung dieser Ansätze sind modellgetriebene Entwicklungsmethoden. Dabei verfolgen diese, Software-Applikationen oder Teile davon aus Modellen abzuleiten [Farines, et al. 2011], was zu einem konsistenten Modell über die gesamte Entwicklungs- und Einsatzzeit führt. Im Bereich der Softwareentwicklung hat sich für die Modellierung die Unified Modeling Language (UML) etabliert. Um auch Systeme in dieser Art und Weise zu entwerfen (vgl. Systems Engineering), entwickelte sich die Systems Modeling Language (SysML) auf Basis der UML. [Weilkiens 2006] Hierbei sind verschiedene Diagrammart definiert, wie beispielsweise Aktivitäts- und Sequenzdiagramme, welche in beiden Sprachen Anwendung finden. Allerdings gibt es keine Definition, welche dieser Beschreibungsmittel in welcher Phase und wie zu verwenden ist. Daraus ist abzuleiten, dass der Ansatz des Systems Engineering auf verschiedene Arten von Systemen maßgeschneidert werden muss. [Leonard 2001]

2.2.3 Mechatronischer Entwicklungsprozess nach VDI-Richtlinie 2206

Die *VDI-Richtlinie 2206* [2004-06] fokussiert die Entwicklung von mechatronischen Produkten und besteht im Wesentlichen aus drei Elementen, dem zuvor vorgestellten Systems Engineering, dem in Abbildung 2.9 veranschaulichten Makrovorgehensmodell und Prozessbausteinen für die dort dargestellten Phasen. Zu Beginn ist eine Anforderungsanalyse, wie in den Kapiteln 2.1.1 und 2.2.2 bereits ausgeführt, durchzuführen. Allerdings sind bei Sondermaschinen, die in der Problemstellung aufgezeigten Anforderungen zur Integration der Maschine in das Produktionssystem zu berücksichtigen. Im Rahmen des Systementwurfs wird ein disziplinübergreifendes Lösungskonzept erarbeitet. Hierbei muss berücksichtigt werden, dass das Konzept eines mechatronischen Systems die erreichbaren Parameter definiert. [Hehenberger, et al. 2010] Nach *Böck* [2013] bildet das Produktionsverfahren in einer Prozesskette den begrenzenden Faktor. Aufgrund der Aufgabe solche Verfahren durch Sondermaschinen zu automatisieren, ermöglicht das Maschinenkonzept nur einen

bestimmten Grad der Potenzialausschöpfung des Verfahrens. Zur Definition dieses Konzeptes schlägt die *VDI-Richtlinie 2206* vor, zunächst eine Funktionsstruktur zu erarbeiten. Dabei wird im Wesentlichen nach *VDI-Richtlinie 2221* (vgl. Kapitel 2.1.1) vorgegangen. Somit wird zunächst eine Funktionsstruktur erstellt, welche Funktionen mittels Informations-, Stoff- und Energieflüssen verknüpft. Im Anschluss werden Wirkprinzipien gesucht, die zu einer Wirkstruktur verarbeitet werden. Diese Struktur muss oftmals konkretisiert werden um das System zu beurteilen. Gleichzeitig führt dies auf eine Baustruktur. Eine domänenübergreifende Lösung kann nur durch eine Verfeinerung der Wirkstruktur über die Konkretisierung von prinzipiellen Lösungsvarianten definiert werden. Dies ist mit dem Lösungskonzept gleichzusetzen. Im Rahmen der Modellbildung und -analyse wird der Einsatz von Rechnern vorgeschlagen. Der domänenspezifische Entwurf basiert auf etablierten disziplinspezifischen Entwicklungsmethoden, welche getrennt voneinander erfolgen. Im Rahmen der Systemintegration werden die Planungsfortschritte der domänenspezifischen Ausarbeitungen zum Gesamtsystem integriert.

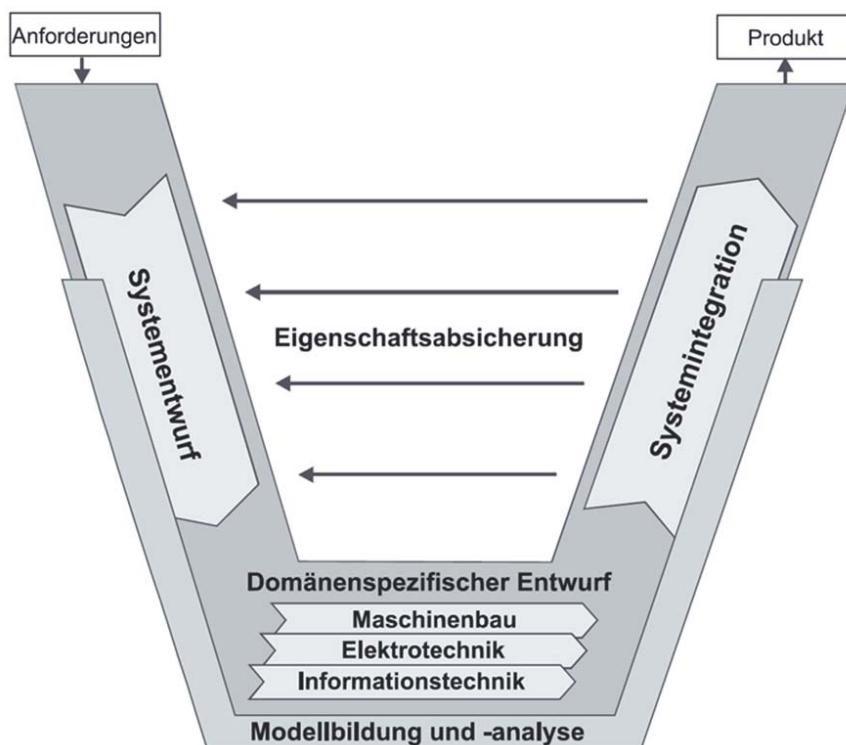


Abbildung 2.9: Vorgehensmodell zur Entwicklung mechatronischer Systeme nach *VDI-Richtlinie 2206* [2004-06]

Im Zuge der Eigenschaftsabsicherung muss das Gesamtsystem gegen die definierten Anforderungen verifiziert bzw. validiert werden. Das V-Modell sagt nichts darüber aus, wie innerhalb, mit partieller Ausnahme im Bereich des Systementwurfs, der verschiedenen Phasen vorgegangen wird und welche Modelle oder Beschreibungsformen hierbei zu berücksichtigen sind.

2.3 Ansätze mechatronischer Systementwicklung

Viele Entwicklungsprozesse durchlaufen sogenannte funktionale Phasen. [Lüder, et al. 2011] Je nach Anwendungsfall des Entwicklungsprozesses sind unterschiedliche Interpretationen des Begriffes „Funktion“ zu erkennen. [Witte 2012] Dies kann auch im Kapitel 2.1, bezogen auf die unterschiedlichen Disziplinen erkannt werden. Allerdings haben all diese das Ziel, eine erste abstrakte Beschreibung des zu entwerfenden Systems (und somit eine disziplinunabhängige Modellierung des Systems) bereitzustellen, um dadurch die Kommunikation zwischen verschiedenen Disziplinen zu fördern. Dabei kommen abstrakte Formulierungen, [vgl. VDI 2222 Blatt 1] im Rahmen von Produktentwicklungsprozessen zum Einsatz. Dies liegt in der Ähnlichkeit zu Problemlösungsprozessen begründet. [Suh 2001] Eine abstrakte Ressourcenbeschreibung findet sich vor allem in der Maschinen- und Anlagenplanung. Dort werden auf Basis von Produktionsprozessen und deren Schritte Ressourcen kombiniert. [Willmann 2011]

2.3.1 Problemorientierte Ansätze

Abstrakte Beschreibungen haben ihren Ursprung in der Problemlösung. Konstruktionsmethoden greifen diese auf, wie Kapitel 2.1.1 darstellt. Dabei ist es das Ziel eine Abstraktion des wesentlichen Problems zu finden und dadurch einen erweiterten Lösungssuchraum aufzuspannen. [VDI 2803 Blatt 1] Außerdem sollen sie helfen das zu entwickelnde Produkt oder System möglichst lösungsneutral zu formulieren. [Campean, et al. 2013] Funktionen werden als Kombination eines Substantives mit einem Verb definiert. [VDI 2221; VDI 2222 Blatt 1] Diese Kombination kann auch als Aktivität interpretiert werden. [Silver 2011] Eine Funktion dieser Begriffsinterpretation gibt an, was das zu entwickelnde Produkt „tun“ soll, ohne jedoch zu sagen, womit dies umgesetzt wird. Gleichzeitig deckt sich dies mit den Grundgedanken des Systems Engineering und des mechanischen Konstruktionsvorgehens (vgl. Kapitel 2.1.1 und 2.2.2.). Die Formulierungen solcher Funktionen können nach *VDI-Richtlinie 2803* [1996-10] eingeteilt werden, wobei sich für eine möglichst neutrale Lösungsbeschreibung die ikonische Ebene am besten eignet. *Kallmeyer* [1998] nutzt eine solche funktionale Beschreibung um die Suche und Modellierung von Lösungsprinzipien vorzubereiten. Dies ist in Abbildung 2.10 dargestellt und gibt im Kern das Grundvorgehen der

Ansätze wieder. Dazu führt er zunächst eine funktionale Dekomposition der aufgestellten Hauptfunktion aus. Die Teilfunktionen werden, wie auch in Kapitel 2.2.3 angedeutet mit den Relationstypen Information, Energie und Stoff miteinander verbunden. Darauf aufbauend werden Lösungsprinzipien zur Erfüllung der Funktionen gesucht. Eine Erfüllungsbeziehung gibt genau dies an. Die gefunden Lösungselemente werden anschließend mit Aspekten angereichert. *Flath* [2002] erweitert den Ansatz um Zustände und Transitionen auf der funktionalen Ebene. *Gehrke* [2005] setzt hier ebenfalls auf, rückt aber den Übergang zwischen funktionaler Ebene und Wirkprinzip in den Fokus. Eine Funktion besteht aus einer Substantiv–Verb–Substantiv Kombination. Dabei sind die verwendbaren Wörter der Funktionen in einer Taxonomie verbunden, welche es erlaubt auf Basis zuvor festgelegter Verbindungen der Funktion zum Lösungsprinzip mehrere Möglichkeiten der Wirkstruktur automatisiert zur Auswahl zu stellen.

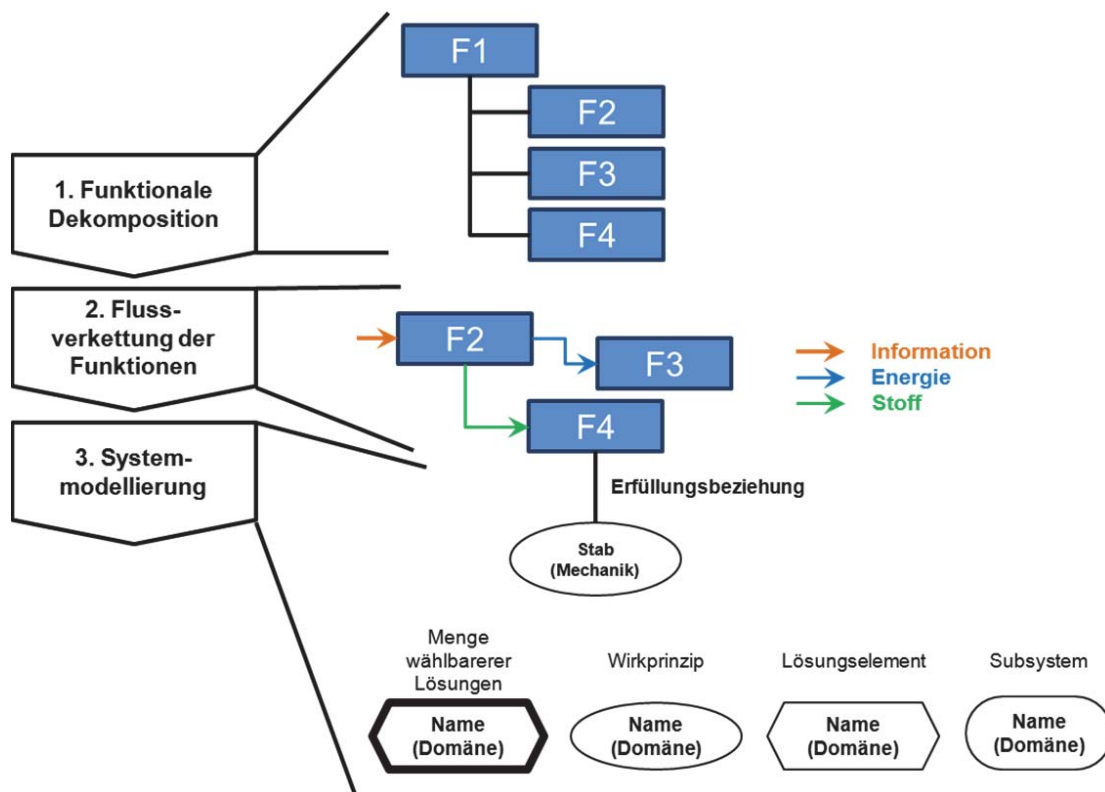


Abbildung 2.10: Interdisziplinäre Funktionsmodellierung nach *Kallmeyer* [1998]

Auch *Kleiner* [2003] rückt den Übergang von Funktion zu Gestalt in den Mittelpunkt. *Bathelt* [2007] beschreibt eine Methodik zur Entwicklung SPS-gesteuerter Maschinen. Den Kern bildet seine erweiterte Funktionsstruktur. Hierbei führt er Transitionsbedingungen in der informations-

technischen Flussverkettung ein. Dadurch ergeben die Funktionen und der zugehörige Informationsfluss ein SFC für die Steuerungsprogrammierung. Dies verknüpft im Bereich des Systementwurfs die mechanische Disziplin mit der Automatisierung und soll dadurch einen parallelen Start der beiden Domänen ermöglichen. [Bathelt 2007]

Kleiner [2013] beschreibt ein Vorgehen auf Basis des Systems Engineering. Darin erfolgt zunächst eine Anforderungsanalyse, dann eine funktionale Beschreibung basierend auf einer Hierarchie und Vernetzung. Die Vernetzung ist auf zwei Typen beschränkt, einen Daten- und einem Steuerungsfluss. Im weiteren Entwicklungsverlauf wird eine sogenannte logische Ebene eingeführt, welche durch ein Modelica-Modell die Simulation des Systemverhaltens ermöglicht. Darauf aufbauend erfolgt eine Modellierung der Komponenten in einem MCAD-System, auf physikalischer Ebene. [Kleiner 2013]

Van Beek et. al. [2010] teilen komplexe mechatronische Systeme in Module auf. Dabei ist es nicht das Ziel diese Module wiederzuverwenden, sondern diese getrennt voneinander parallel zu entwickeln. [vgl. auch Brudniok 2007] Ihrer Ansicht nach besteht die Problematik der Komplexitätsbeherrschung im Rahmen der Entwicklung mechatronischer Produkte darin, dass alle bisherigen Ansätze davon ausgehen die Abhängigkeiten der Module zueinander zu kennen. Davon kann ihrer Meinung nach nicht ausgegangen werden. Zur Lösung wird die Function Behaviour State Notation (FBS-Notation) vorgeschlagen. Darin sind Funktionen ebenfalls wie Aktivitäten definiert. Über eine Dekomposition wird die sequenzielle Zustandsänderung des physikalischen Systems als Behaviour modelliert. Darauf aufbauend werden Komponenten des Systems definiert. Deren Zustände beschreiben dabei die verschiedenen Moden des Systems, welche die Abhängigkeiten zwischen den Modulen definieren. Die Zustandsübergänge werden in einer Design Structure Matrix (DSM) eingetragen, was zur Minimierung der Abhängigkeiten zwischen und zur Maximierung dieser innerhalb der Module führt. [Beek, et al. 2010] *Campean et.al.* [2013] erkennen, dass es gerade bei komplexen mechatronischen Produkten eine besondere Schwierigkeit darstellt, die Interaktion zwischen Sub-Systemen in einer frühen Phase zu bewerten. Üblicherweise geschieht dies mittels Design Structure-Matrizen, welche aber nur eine Zuweisung der Systemstrukturen zu Funktionen darstellen. Die Interaktion der Funktionen bleibt hier außen vor. Zur Lösung wird das System State Flow Framework vorgeschlagen. Darin wird zunächst der Hauptfluss des betrachteten Systems erstellt. Dies geschieht auf Basis von Verb-Substantiv-Funktionen, welche Zustände verbinden. Eine funktionale Dekomposition erfolgt durch die Identifizierung der Zwischenzustände zwischen Out- und Input. Den Funktionen können Lösungen zugeordnet werden. Außerdem ist die

Modellierung des Signalflusses zwischen Funktionen möglich. [Campean, et al. 2013] Zu erkennen ist, dass die Arbeiten der Produktentwicklung den Systementwurf mechatronischer Produkte fokussieren und hierbei die gedankliche Basis des Systems Engineering nutzen. Modularisierung wird verwendet, um Subsysteme parallel zu detaillieren, nicht jedoch um eine Wiederverwendung explizit herbeizuführen.

2.3.2 Produktionsprozessorientierte Ansätze

Produktionsprozessorientierte Ansätze fokussieren sich auf die Umsetzung von Prozessketten im Rahmen der Maschinen- und Anlagenplanung, wobei eine kundenindividuelle Umsetzung häufig nur durch Konfiguration verschiedener Module umgesetzt wird. *Dohmen* [2002] führt hierzu drei Sichten auf das mechatronische System ein. Eine funktionale, eine komponenten- und eine verhaltensbasierte. Funktionen sind hierbei als Verben formuliert. [Dohmen 2002] *Lercher* [2008] greift auf diese Grundgedanken zurück. Dabei setzt er den Fokus auf die Werkzeugmaschine, ausgehend von einem Requirements Engineering. Im Anschluss erfolgt eine Einteilung der Maschine in Haupt- und Nebenfunktionen, wie beispielsweise „Werkstückführung“. Diese sind zwar allgemein gehalten und beschreiben im weitesten Sinne eine Aktivität, sind aber nicht mehr im ikonischen Bereich der *VDI-Richtlinie 2803*. Hierbei werden als Hauptfunktionen alle zum Abarbeiten des Bearbeitungsprozess notwendigen Operationen bezeichnet. Unterstützende Funktionen, wie „Werkzeuggestellung“ sind Nebenfunktionen. Diese werden in einem Aktivitätsdiagramm verfeinert. Parallel dazu erfolgt die Detaillierung auf Basis eines Technologieschemas, welches aus 2D-Skizzen mit Symbolen nach den Normen *DIN 40900*, *DIN ISO 1219* und *DIN 30600* beschrieben ist. [Lercher 2008] Diese beiden Ansätze können als Übertragung der verfahrenstechnischen Vorgehensweise zur Anlagenplanung auf die Maschinenentwicklung angesehen werden. [vgl. *Blass* 1997] Dort sind Prozessschritte ebenfalls als Funktionen definiert. Weiterführend erfolgt die Darstellung eines verfahrenstechnischen Fließbildes, bevor ein R&I-Bild entsteht. [Blass 1997] *Frank et al.* [2013d] unterscheiden darin Anlagen- und Automatisierungsfunktionen. Eine detaillierte Abgrenzung erfolgt allerdings nicht. Lediglich, dass Automatisierungsfunktionen Informationen verarbeiten und dadurch aus steuerungstechnischer Sicht gut standardisierbar sind. [Frank, et al. 2013d] Die *VDW-Richtlinie Funktionsbeschreibung* [2002-10] beschreibt Funktionsobjekte mit gekapselter Funktionalität und definierten Schnittstellen. Jene Schnittstellen bedienen von außen nutzbare Steuerungsfunktionen. Eine Gliederung dieser Funktionsobjekte erfolgt auf Basis einer Hierarchie und ist beispielhaft in *Abbildung 2.11* dargestellt. Darin sind Funktionsgruppen, -untergruppen und -einheiten

unterschieden, wobei letztere standardisierbar und somit wiederverwendbar sind. Gleichzeitig können diese den Automatisierungsfunktionen entsprechen. [VDW] Vergleichbar hierzu sind die bereits im Rahmen der elektrischen Konstruktion eingeführten Normen der *DIN EN 61346 Blatt 2* und der nachfolgenden *DIN EN 81346 Blatt 2*.

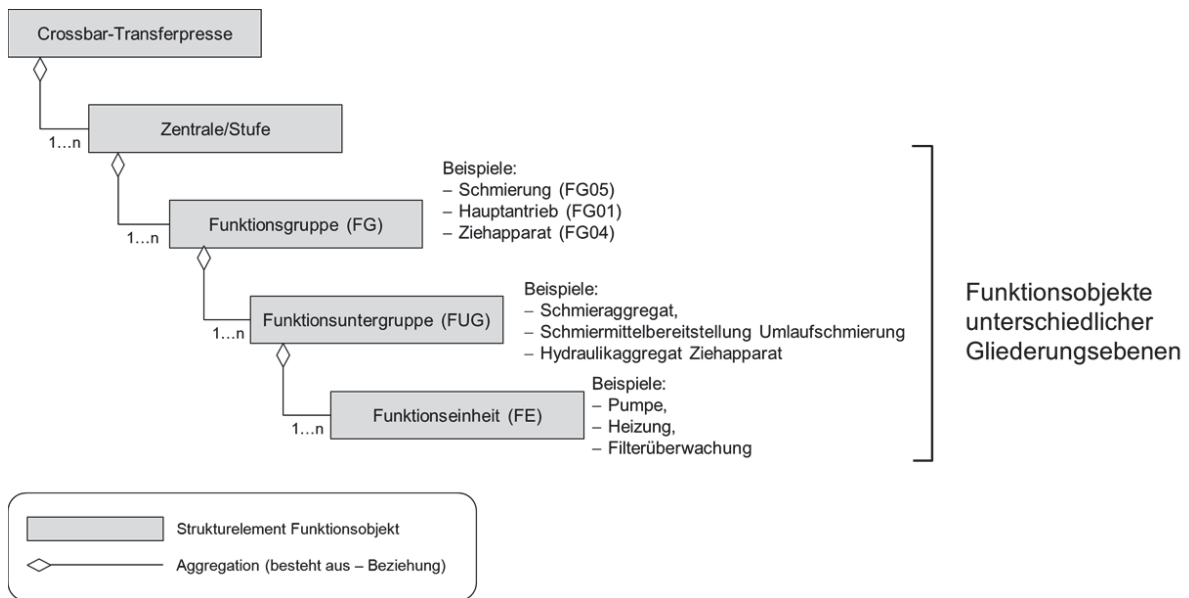


Abbildung 2.11: Funktionshierarchie der *VDW-Richtlinie Funktionsbeschreibung* [2002-10]

Weiterhin nutzen die Forschungsprojekte *Föederal* und *Aquimo* diese *VDW-Richtlinie* als Basis. [vgl. Litto 2004, Litto 2010; Würslin, et al. 2008] Dabei setzt die *Aquimo-Methode* [2010] auf eine Analyse-, Entwurfs- und Installationsphase, bevor eine gleichzeitige Konstruktion in den Disziplinen beginnt. Im Fokus steht allerdings die Phase der Installationsplanung. Darin werden Basiskomponenten eingesetzt, welchen jeweils ein 3D-Modell, ein Schaltplansymbol und ein Verhaltensmodell zugewiesen ist. *Kiefer* [2007] erkennt, dass problemlösungsorientierte Funktionen für die Zellenplanung zu abstrakt sind. Er schlägt ein Vorgehen beginnend bei der Prozesskette in der Fertigungsplanung hin zu einem Anlagenkonzept vor. Diese beinhaltet ein grobes Gesamtlayout. Ein 3D-Zellenlayout bildet darauf aufbauend die Basis um Komponenten eines mechatronischen Ressourcenmodells in der Betriebsmittelkonstruktion auszuwählen. [Kiefer 2007] In *Brecher et al.* [2013] werden Funktionen ebenfalls als Prozessschritt interpretiert um Montagearbeitsplätze zu konfigurieren.

Lindworsky [2011] und *Hensel* [2011] definieren in ihrer funktionalen Maschinenbeschreibung eine Funktion als Komponente, welche wiederum Beschreibungen für Kommunikationsschnittstellen,

Geometrien, Fahrkurven etc. enthält. *Pörnbacher* [2011] entwickelt für die modellgetriebene Entwicklung SPS-gesteuerter Fertigungssysteme eine Domain Specific Language. Dabei setzt er auf drei Abstraktionsebenen einer präskriptiven (vorschreibenden), einer deskriptiven (beschreibenden) Modellierung und den Zielartefakten. Diese Ebenen werden wiederum in das Steuerungsmodell und das Verhaltensmodell des zu steuernden Systems unterteilt. Die erste Ebene wird mittels einer Funktionshierarchie (vgl. VDW) und dem Sequenzdiagramm der UML für die Steuerungssystembeschreibung umgesetzt. In der deskriptiven Phase wird das Verhalten des Steuerungsprogramms mittels Zustands- und Aktivitätsdiagrammen beschrieben. Die Systembeschreibung des zu steuernden Systems wird in Modelica auf beiden Ebenen abgebildet. Auf der deskriptiven Ebene ist zudem eine komponentenorientierte Modellierung notwendig, wozu das Komponentendiagramm der UML genutzt wird. Die Zielartefakte repräsentieren die detaillierte technologische Umsetzung, wie beispielsweise die Hardwareauswahl. [Pörnbacher 2011] Allen Ansätzen gemein ist, dass auch produktionstechnische Ansätze die frühen Phasen der Entwicklung fokussieren. Dazu werden ebenfalls funktionale Strukturen aufgebaut, welche allerdings deutlich konkreter definiert werden, als im Rahmen der Produktentwicklung.

2.4 Grundlagen und Ansätze der Maschinensimulation

Simulationen bilden heutzutage einen wichtigen Baustein um Optimierungen und Überprüfungen von Systemen im Rahmen eines methodischen Vorgehens durchzuführen. Die *VDI-Richtlinie 3633* [VDI 3633] definiert Simulation wie folgt: *„Simulation ist ein Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Im weiteren Sinne wird unter Simulation das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell verstanden.“* [VDI 3633] Im Blatt acht dieser Richtlinie wird die maschinennahe Simulation in Mehrkörper-, Prozess-, 3D-Kinematik-, maschinennahe Materialflusssimulation und Simulation zum Funktionstest der Steuerung unterschieden. [VDI 3633 Blatt 8] Die Mehrkörpersimulation zielt auf die Kräfteermittlung mit gegebener Kinematik und ist somit als Unteraufgabe der mechanischen Konstruktion zu betrachten. Die Prozesssimulation zeigt Zusammenhänge zwischen Einflussparametern auf, welche entsprechend ausgeregelt werden müssen. Aufgrund der Fokussierung dieser Arbeit auf die SPS bleibt dies unberücksichtigt. 3D-Kinematiksimulationen bestehen aus Geometrieelementen und kinematischen Randbedingungen. Sie betrachten somit die Bewegungen geometrischer Teile von Position zu Position. Die maschinennahe Materialflusssimulation kann als Erweiterung der kinematischen Simulation

gesehen werden, indem das zu produzierende Produkt in das Modell der Maschine mit eingebunden wird. Dabei wird beispielsweise das Fördern, Greifen und Umsetzen simuliert. Dazu ist neben der reinen Kinematik auch ein Kollisionsmodell notwendig. Dies ist in verschiedenen industriellen Ansätzen bereits umgesetzt. [vgl. Siemens 2011]

Eine Kopplung solcher Modelle mit der Steuerung fällt bereits unter den Funktionstest von Steuerungen oder die virtuelle Inbetriebnahme. Dabei ist das Ziel am Ende des Entwicklungsprozesses vor der Inbetriebnahme der Maschine das Steuerungsprogramm auf Fehler zu überprüfen. Hierzu sind bereits verschiedenste industrielle Softwarewerkzeuge mit unterschiedlichen Fokussen am Markt. [vgl. Wunsch 2008] *Kövari* [2011] zeigt vier potenzielle Handlungsfelder der virtuellen Inbetriebnahme auf, die Unterstützung der Inbetriebnahme, den Softwaretest, die Präsentation und Akquisition und die Durchführung von Schulungsmaßnahmen. Die Aufwände des ersten Handlungsfeldes können um 75% durch den Einsatz der Simulation reduziert werden. [Zäh, et al. 2006] *Bergert* [2008] detailliert die ersten beiden Handlungsfelder hinsichtlich der Untersuchungen zum Zusammenspiel verschiedener Steuerungen, einer Ablaufs- und Kollisionsanalyse, Simulation von Fehlerszenarien und das Ableiten von genauen Taktzeitaussagen. Dazu muss das mechatronische Modell mit all seinen Komponenten der geplanten Anlage entsprechen. [Bergert, et al. 2008] Nach *Kufner* [2012] besteht bei der Simulation zum Steuerungstest das Modell der Maschine aus einem geometrischen Modell und einem Verhaltensmodell. Dabei ist das Verhaltensmodell mit dem geometrischen Modell gekoppelt. Dies ist in Abbildung 2.12 dargestellt. Der Datenaustausch mit der Steuerung erfolgt über das Verhaltensmodell. Das geometrische Modell dient dabei nur der Visualisierung. [Kufner 2012] Um einen Funktionstest der Steuerung durchzuführen muss ein vollständiges Verhaltensmodell der Maschine vorliegen. Bei Verhaltensmodellen ist zu unterscheiden, ob es sich um gesteuertes oder bewirktes Verhalten handelt. Das gesteuerte repräsentiert das Verhalten des Steuerungsprogramms. Das Verhalten der Maschine, welches durch die Steuerung angestoßen wird, ist als bewirktes Verhalten zu bezeichnen. [Drath 2010] Ein vollständiges Maschinenmodell liegt vor, wenn für die Steuerung kein Unterschied mehr zwischen dem Modell und der realen Anlage besteht. [Bergert, et al. 2008] Dazu müssen alle Signale, welche im Steuerungsprogramm genutzt werden modelliert sein. [Baudisch, et al. 2012, Frank, et al. 2012] Wenn bei einem solchen Test die Steuerungshardware bereits mit integriert ist, wird auch von einer Hardware-in-the-loop-Simulation (HILS) gesprochen.

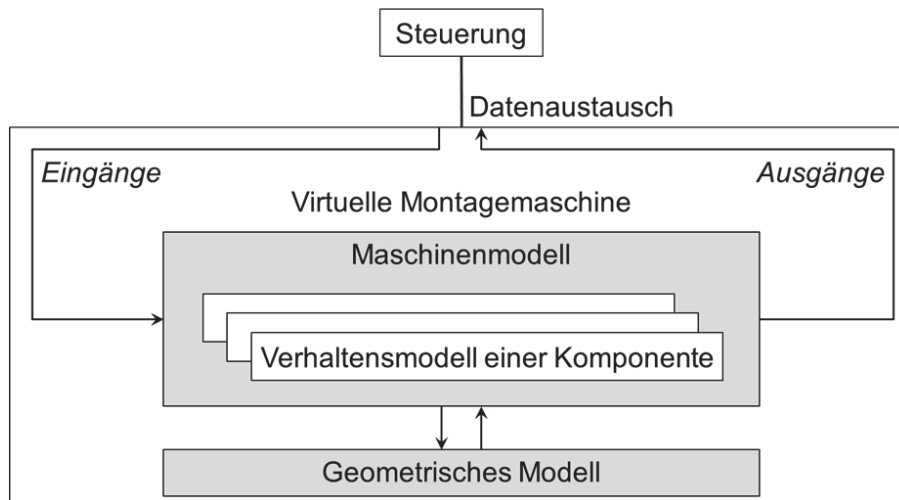


Abbildung 2.12: Aufbau eines Maschinenmodells für die virtuelle Inbetriebnahme nach Kufner [2012]

Nach Nieding [2012] können Simulationen in allen Phasen der Entwicklung eingesetzt werden. Im Speziellen wird ein Einsatz in frühen Phasen zur Entscheidungsunterstützung als Frontloading bezeichnet. Dadurch sollen das Risiko und der Aufwand in späteren Phasen minimiert werden. [Nieding 2012] Für Maschinen ist dies zur Förderung des interdisziplinären Verständnisses zwischen allen Beteiligten bereits in der Phase des Systementwurfs notwendig. [Litto 2010] Eine solche Kommunikationsbrücke wird durch einen Zusammenschluss der Steuerung, Simulation der Aktorik/Sensorik und einer Visualisierung in 3D erreicht. [Betschon, et al. 2008] Eine Erweiterung der virtuellen Inbetriebnahme stellt die physikbasierte Simulation dar. [Stich, et al. 2011] Eine solche Erweiterung besteht aus Massen, Schwerpunkten und Reibungskoeffizienten und hilft gerade auch in frühen Phasen die Auswirkungen der Kinematik besser zu verstehen. [Lacour 2012] Die Industrie steuert hierzu das Softwarewerkzeug Mechatronics Concept Designer bei. [vgl. Siemens PLM 2010] Botaschanjan et al. [2009] schlagen eine Simulation entlang des gesamten Entwicklungsprozesses vor, indem ein in der Systementwurfsphase erstelltes Modell immer weiter detailliert wird. Während einer parallelen Detaillierungsphase der Disziplinen nutzen Müller et. al [2009a] einen auf Quality Gates basierenden Prozess. An diesen Meilensteinen wird eine Überprüfung disziplinspezifischer Planungen in Automatisierung, Elektrik und Mechanik durchgeführt. Die Herstellung mechatronischer Konsistenz erfolgt mittels einer virtuellen Inbetriebnahme. [Müller, et al. 2009]

2.5 Ansätze zur Aufwandsreduktion in mechatronischen Entwicklungsprozessen

Wie in den vorigen Kapiteln erläutert, sind die Methoden der einzelnen Disziplinen jeweils durch Softwarewerkzeuge unterstützt, was eine Modellierung des Systems jeweils aus einer bestimmten Perspektive heraus ermöglicht. Eine Aufwandsreduktion im Kontext dieser Arbeit bedeutet dabei, den manuellen Aufwand zur Erstellung jener Modelle im Rahmen eines Projektes zu minimieren. Dies ist gleichzusetzen mit der Wiederverwendung von existierenden Informationen, wobei Informationen in diesem Zusammenhang nach *North* [2011] definiert werden. Dabei sind zwei Arten zu unterscheiden:

1. Nutzung projektspezifischer Informationen entlang des Entwicklungsprozesses und
2. Wiederverwendung von Informationen über Projekte hinweg.

Ersteres bedeutet, im Wesentlichen existierende Informationen in anderen Werkzeugen zu nutzen und nicht erneut eingeben zu müssen. Eine Wiederverwendung über Projekte hinweg basiert im Wesentlichen auf Baustein-Bibliotheken. Dabei können erstellte Informationen abgelegt und bei Bedarf neu instanziiert werden. Es ist möglich, auf Basis abstrakter Systembeschreibungen detaillierte Ergebnisse und Modelle zu generieren. Ersteres ist unter der Kategorie der Integrationsplattformen zu finden, während die zweite Kategorie von sogenannten Engineeringwerkzeugen repräsentiert ist. [vgl. Klemm, et al. 2006; Pörnbacher 2011]

2.5.1 Wiederverwendung entlang des Entwicklungsprozesses

Im einfachsten Fall ist es möglich in jedem Projekt Routineaufgaben zu automatisieren. *Lüder et al.* [2010] zeigen hierzu eine automatische Adressvergabe im Rahmen der Elektrokonstruktion. Dies erhöht den Informationsgehalt des Gesamtprojektes um die Adresszuordnung der Signale. [Lüder, et al. 2010] Eine softwareübergreifende Ausprägung dieser Kategorie bilden Schnittstellen zum Export und Import von Daten aus einem anderen Werkzeug. Diese setzen auf eine direkte Transformation der Informationen zwischen zwei Werkzeugen. (vgl. Abbildung 2.13) Zur Reduktion von Transformationsaufwänden bieten sich neutrale Datenformate an. Diese besitzen ein Metamodell in Form eines Standards, an welchem sich verschiedene Werkzeuge ankoppeln können. Vorteil ist hier, dass jedes Softwarewerkzeug nur eine Import- und eine Exporttransformation bereitstellen muss. [Drath 2010; Müller, et al. 2010] Die neutralen Datenbeschreibungen fokussieren sich in den meisten Fällen auf eine Disziplin. Bekannte Vertreter dieses Ansatzes sind daher für mechanische Daten STEP [DIN 10303-210], das IGES-Format [US PRO 1996] oder Collada [Barnes, et al. 2008]. Seinen Ursprung in der Elektrik hat eCI@ss. Dieser Standard

beschreibt Produktdaten und deren Klassifizierung mit dem Fokus zum Austausch von Geräteinformationen. [ecl@ss e.V.] Aus dem Bereich der Automatisierung ist als offener Standard PLCopen zu erwähnen, welcher eine neutrale Beschreibung von SPS-Programmen auf Basis von SFCs bereitstellt. [Drath 2010] *Hundt* [2012] beschreibt, wie Gantt-, PERT, Impuls- und Zustandsdiagramme in SFCs transformiert werden können. *Voß* [2012] zeigt einen Ansatz zum Austausch von Simulationsdaten. Einen interdisziplinären Ansatz bildet AutomationML. Dabei wird CAEX als Objektträgerstruktur genutzt, worin jedem Objekt beschreibende Dokumente aus Collada und PLCopenXML zugewiesen werden können. [Drath 2010]

Wie in Kapitel 2.2.2 bereits ausgeführt, versucht der Ansatz zur modellgetriebenen Softwareentwicklung aus einem zusammenhängenden Modell möglichst viele Teile des Gesamtsystems über Generatoren in ausführbaren Programmcode zu transformieren. Dadurch verschiebt sich der Fokus vom reinen Programmcode hin zu einem Modell des zu entwickelnden Projektes, wobei das Modell dann unabhängig vom Zielsystem ist. [Thramboulidis, et al. 2011] Wird das Projektmodell eines Softwarewerkzeuges eins als Basis hierfür genutzt, kann über eine entsprechende Transformation ein modellgetriebener Ansatz in ein Softwarewerkzeug zwei realisiert werden, was gleichzeitig der direkten Transformation in Abbildung 2.13 entspricht. Für die SPS-Programmierung zeigt *Hensel* [2011] einen solchen Ansatz mit einer komponentenorientierten Funktionsmodellierung als Ausgangsmodell. Andere Ansätze zur SPS-Codegenerierung nutzen die UML oder SysML als Ausgangsmodell. Eine erfolgreiche Umsetzung auf Basis von UML ist gegeben, mehr Erfolg verspricht aber SysML. [Thramboulidis, et al. 2011] Eine Spezialisierung der UML bildet die für die Prozessautomatisierung angepasste UML-PA. [Vogel-Heuser, et al. 2005] Außerdem finden sich Ansätze mit regelungstechnischen Modellen, wie Matlab/Simulink [vgl. Bayrak, et al. 2007]. *Pörnbacher* [2011] nutzt seine Domain Specific Language (siehe Kapitel 2.3.2) zur Transformation in SPS-Programme.

Hehenberger et.al. [2010] schlagen am Beispiel für die Entwicklung von Elektromotoren eine mechatronische Kopplungsebene vor, welche die disziplinübergreifenden Parameter enthält. *Arnold et. al.* [2012] realisieren eine solche und zeigen, dass es möglich ist ganze Entwicklungsabläufe zu automatisieren. Hierfür ist allerdings ein sehr detailliertes Verständnis der Entwicklung, inklusive aller Parameter und deren Abhängigkeiten notwendig. Kern bildet ein Graph und ein Aktivitätsdiagramm. Beides ist in UML beschrieben. Darauf aufbauend können verschiedene disziplinspezifische Modelle in entsprechende Softwarewerkzeuge generiert,

analysiert und die relevanten Parameter in weitere Werkzeuge transformiert werden. [Arnold, et al. 2012]

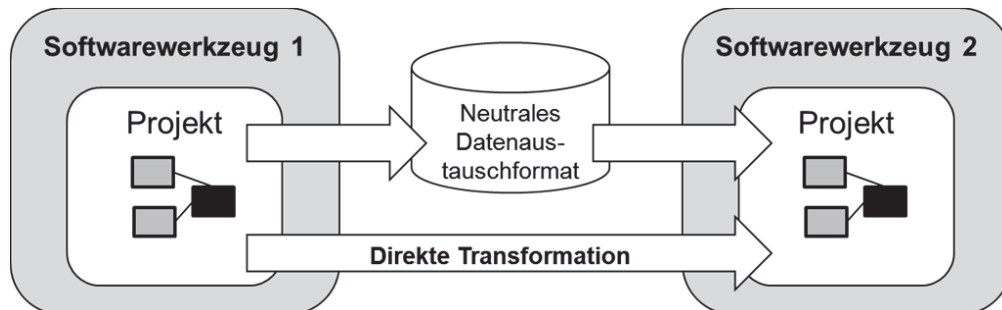


Abbildung 2.13: Möglichkeiten zur Transformation von Modellen entlang des Entwicklungsprozesses

Einen Spezialfall bildet die Übernahme von Informationen mittels einer eins zu eins Zuweisung von Bibliothekselementen beider Softwarewerkzeuge, wodurch ein einfacher Ex- und Import von Projektdaten gewährleistet ist. Die Arbeiten zur Wiederverwendung von Informationen entlang des Entwicklungsprozesses sind entweder dateifokussiert oder repräsentieren modellgetriebene Softwareentwicklung.

2.5.2 Wiederverwendung über Projekte hinweg

Um Aufwände durch Wiederholung von Tätigkeiten zu vermeiden, bietet sich eine Speicherung von bereits erstellten Modellen, Dokumenten etc. in sogenannten Bibliotheken an. Diese können bei Bedarf einfach instanziiert werden. Praktisch alle IT-Werkzeuge bieten diesen Mechanismus. In der Mechanik wird die Geometrie von Bauteilen abgelegt und im Kontext des Einbaus in eine Baugruppe wiederverwendet. Selbiges ist auch für Subbaugruppen in übergeordnete Baugruppen möglich. Um potenzielle Bauteile für eine mögliche Instanziierung zu suchen und zu finden, bietet sich eine Klassifizierung jener an. [Schilke 2010] Im Gegensatz zur Mechanik instanziierten beispielsweise ECAD-Systeme ihre Bibliothekselemente in einen Projektkontext. Eine Aufwandsreduktion wird dadurch erzielt, dass Detailplanungen abgelegt und auf einer aggregierten Ebene Wiederverwendung finden. Es müssen nicht alle Planungen erneut wiederholt werden. Dies ist in Abbildung 2.14 dargestellt.

Grätz [2006] beschreibt ein Vorgehen, welches solche Bibliothekselemente für die Elektro- und Fluidplanung nutzt um teilautomatisch Pläne zu generieren. Dazu setzt er Kreislaufmodelle auf, welche Verbindungen zwischen Quellen und Senken beinhalten. Da beispielsweise Sensoren als

Signalquellen anzusehen sind, können entsprechend der bekannten Verbindung zur Signalsenke auf Basis der in den Bibliothekselementen mitgelieferten Symbole die Schaltpläne generiert werden. [Grätz 2006]

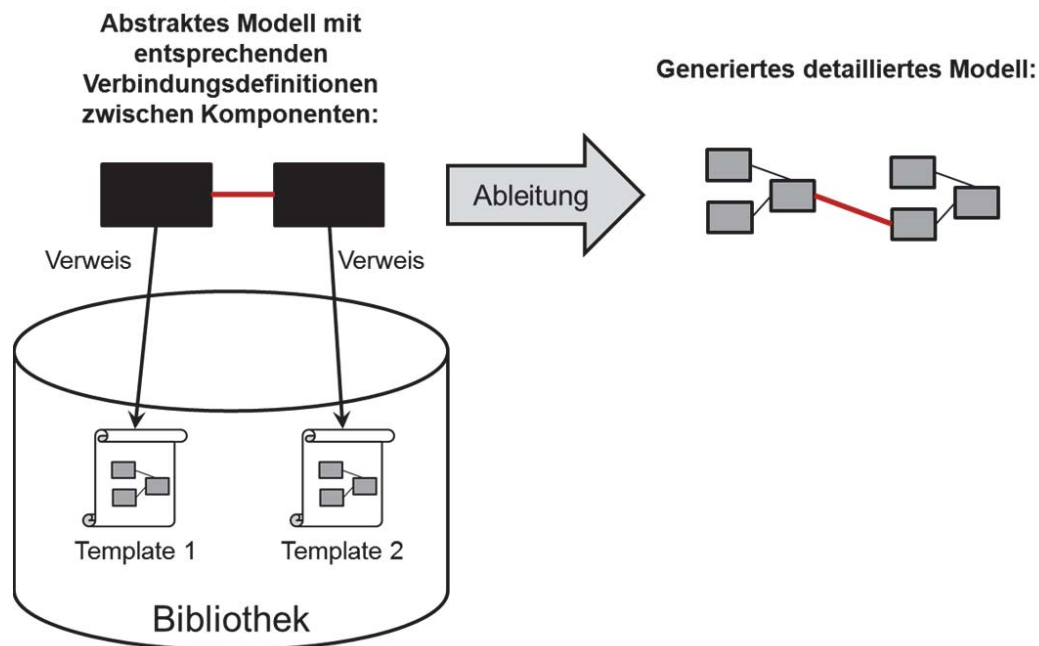


Abbildung 2.14: Prinzip der Modellgenerierung auf Basis von Bibliotheken

Zäh *et. al.* [2009] erkennen, dass die Modellierung der Verhaltensmodelle für die virtuelle Inbetriebnahme sehr aufwändig ist. Um dies zu umgehen, kann eine Funktionsbeschreibung genutzt werden um Bibliothekselemente der Simulationsumgebung zu parametrieren, wodurch ein detailliertes Modell entsteht. [Lindworsky 2008; Reinhart, *et al.* 2007] Kiefer [2009] möchte sich diese Bibliothekselemente von Lieferanten zur Verfügung stellen lassen. Kufner [2012] nutzt als abstraktes Modell die existierenden Engineeringdokumente der Eingangs-/Ausgangsliste und der Buskonfiguration um in der Bibliothek abgelegte Verhaltensmodelle zu einem Modell für die virtuelle Inbetriebnahme zusammenzustellen. Somit ist kein separates abstraktes Modell für die Generierung des Verhaltensmodells mehr notwendig. Steden zeigt, wie man abhängig vom jeweiligen Simulationsziel, verschiedene Bibliothekselemente bewerten kann. Diese können automatisch oder manuell im entsprechenden Softwarewerkzeug genutzt werden. [Steden, *et al.* 2013b, Steden, *et al.* 2013a]

Viele Maschinen- und Anlagenbauunternehmen greifen bereits in der Angebotsphase auf Erfahrungen bereits existierender Lösungen zurück. Dieser Umstand kann zur Verkürzung von

Auftragsdurchlaufzeiten genutzt werden. Dazu wird der Auftragsabwicklungsprozess vom Entwicklungsprozess einzelner Teile und Module entkoppelt. Den dadurch entstehenden Prozess zeigt Abbildung 2.15 auf.

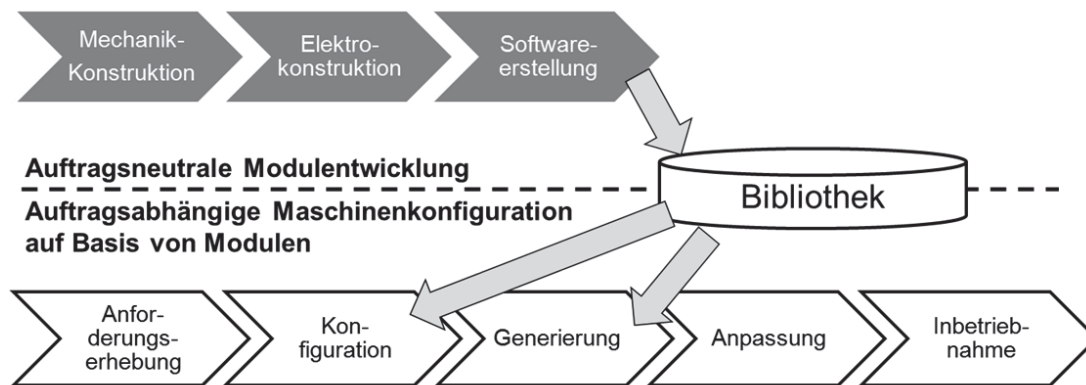


Abbildung 2.15: Prozess zur Entkopplung des Auftrags von der Entwicklung

Die Entwicklung erstellt Module, Ressourcen oder Komponenten, welche in einer Bibliothek abgelegt werden. Dies geschieht auftragsneutral. Im Rahmen der Auftragsabwicklung werden diese Module entsprechend den Kundenanforderungen zusammengestellt. Dieses prinzipielle Vorgehen ist in verschiedenen Ansätzen dokumentiert. [vgl. Doll 2010; Litto 2004; Rodica Maga, et al. 2011; VDI/VDE 3695 Blatt 2] Je nach Produkt unterscheiden *Litto et al.* [2012] drei wesentliche Strategien, Top-Down Konfiguration der Produkte, Bottom-Up konfigurierbare Lösung und die Wiederverwendung von internen Standards. Sondermaschinen sind in diesem Fall als Unikate zu betrachten, was zu einer Strategie hinsichtlich Nutzung vordefinierter Standards führt. Die elektrische Konstruktion kann davon profitieren, da hier nur Standardkomponenten Einsatz finden. Dies geht einher mit der Arbeit von *Possel-Dölken* [2010], der die Speicherung dieser in Patterns vorschlägt. Das wahre Potenzial dieses Ansatzes ergibt sich, wenn mechatronische Aggregationen im Rahmen dieser Strategien erstellt werden können. Dadurch befinden sich in der Bibliothek mechatronische Objekte.

Hundt [2012] bezeichnet solche Objekte als mechatronische Planungsobjekte, welche seiner Auffassung nach informationstechnische Abbilder realer mechatronischer Objekte darstellen. Diese besitzen eine Hülle und spezifische Sichtweisen (vgl. Abbildung 2.16). Ein solches Objekt muss alle Phasen des Lebenszyklus und die notwendigen Eigenschaften unterstützen. [Brecher, et al. 2008] Im Projekt *Föederal* repräsentieren Teilmodelle der Elektrik und SPS-Programmbausteine diese Sichtweisen. [Litto 2004] Industrielle Umsetzungen sind hierzu Eplan bzw. Mind8

Engineering Center [EPLAN 2013] und Automation Designer [Siemens 2010]. *Weyrich et al.* [2011b] sehen diese Objekte als gekapselte Elemente mit mechanischen, hydraulischen, elektrischen und informationstechnischen Elementen an. *Kiefer* [2007] definiert für solche Einheiten Informationen über mechanische (beispielsweise 3D-Geometrien), elektrische, pneumatische und hydraulische, funktionsbeschreibende, technische, betriebswirtschaftliche, organisatorische und steuerungstechnische Informationen. *Lüder et al.* [2010] erweitern diese um topologische Daten und im Speziellen die funktionalen Daten um die Kategorien des Steuerungs- und des Maschinenverhaltens. Die spezifische Hülle besitzt ebenso Informationen, wie den Namen, eine eindeutige ID, etc. *Reuter et al.* [2010] definieren für die Hülle den Inhalt des Namens und eine Klassifikation des Objekts.

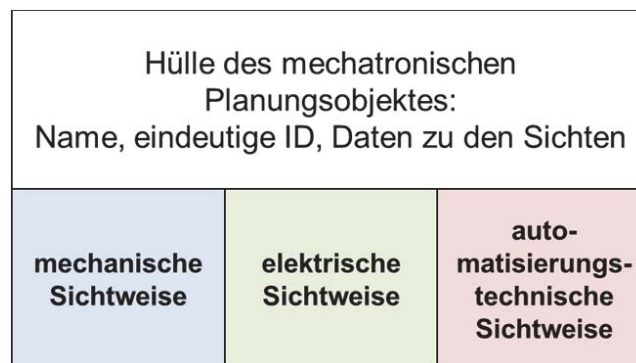


Abbildung 2.16: Mechatronisches Planungsobjekt nach *Hundt* [2012]

Die Auftragsabwicklung erfolgt entsprechend auf Basis eines funktionalen Ansatzes nach Kapitel 2.3.2. Funktionsgruppen und -einheiten definieren, welche mechatronischen Objekte Anwendung finden. Dabei sind diese mit der Klassifikation mechatronischer Objekte gleichzusetzen. Durch Parametrieren der Instanzen ist eine Generierung detaillierter Modelle gleichzeitig in alle Sichtweisen möglich, da die Details in der auftragsneutralen Modulgestaltung bereits festgelegt wurden. [Litto 2004] Die auftragsspezifische Ausprägung der Komponenten hängt somit im Wesentlichen von deren Konfiguration und Parametrierung ab. Besitzen Projekte kundenindividuellen Charakter und kann dieser nicht direkt über die Konfiguration oder Parametrierung abgebildet werden, erfolgt die Anpassung erst nach der Generierung.

Dagegen verläuft die Bildung mechatronischer Objekte im auftragsneutralen Entwicklungsprozess immer noch sequenziell. [Doll 2010] Außerdem ist es notwendig, die dort erstellten Modelle der verschiedenen Disziplinen nachträglich mechatronisch zusammenzustellen. [Litto 2004] Dies ist aufwendig und wird in diesem Zusammenhang auch als Modularisierung bezeichnet.

[Possel-Dölken 2010] Im Gegensatz zur Modularisierung in Kapitel 2.3.1, in dem Module eine parallele Entwicklung verschiedener Teams ermöglichen sollen, haben die mechatronischen Module hier das Ziel alle Disziplinen parallel starten zu lassen. *Possel-Dölken* [2010] zeigt auf, dass bei Sondermaschinen eine mechatronische Modulbildung auf Basis der Elektrik erfolgt, da diese dem größten gemeinsamen Nenner entspricht und Standardkomponenten einsetzt. Dies deckt sich mit den von *Lechler et al.* [2008] aufgestellten Anforderungen an mechatronische Module. Dies sind integrierte Funktionen, schmale und einfache Schnittstellen, ausführliche Dokumentation und hoher Standardisierungsgrad. *Lüder et al.* [2010] und *Weyrich et al.* [2011a] erkennen, dass aber größer gefasste Module eine höhere Zeiteinsparung erreichen, was gleichzeitig zu geringerer Wiederverwendbarkeit führen kann. *Weyrich et al.* [2011b] stellen eine Methode vor, welche Werkzeugmaschinenmodule auf Basis ihrer Funktionalität in drei Kategorien (Basismodule, Serienmodule und kundenspezifische Module) einteilt. *Weyrich et al.* [2011a] beschreiben eine weitere Methode basierend auf dem bekannten axiomatischen Design von *Suh* [2001] zur algorithmenbasierten Modularisierung von Maschinen und Anlagen. Eine Bewertung der Unabhängigkeit zwischen den einzelnen Systembestandteilen bildet hierfür die Basis. Dies wird zunächst aus mechanischer, dann der elektrischen und automatisierungstechnischen Sichtweise durchgeführt, bevor eine Zusammenfassung die Gesamtmatrix ergibt. Darin sind folglich die größtmöglichen unabhängigen mechatronischen Module erkennbar. Zudem sind verbindende Module erkennbar, was nach *Weyrich et al.* [2012] auf eine zentrale Maschinensteuerung zurückzuführen ist. Daraus wird abgeleitet, dass eine Dezentralisierung der Steuerung modulare Ansätze unterstützt. Dies deckt sich mit der Modularisierung aus automatisierungstechnischer Sicht. [Stork, et al. 2011]

Das Projekt *Aquimo* hat zum Ziel den Entwicklungsprozess mechatronischer Module für Maschinen- und Anlagenbauer zu unterstützen. Dies geschieht, indem für die in Kapitel 2.3.2 aufgezeigten Phasen jeweils eigene Bibliotheken erstellt werden. Dadurch ist es möglich Modelle zur Simulation auf unterschiedlichen Abstraktionsebenen zu generieren. Diese Elemente besitzen in der Installationsphase, vor einer parallelen disziplinspezifischen Detaillierung, ein 3D-Modell, ein 2D-Modell eine Verhaltensbeschreibung und eine Parametrierung. [Litto 2010; Würslin, et al. 2008]

2.6 Verknüpfung von Softwarewerkzeugen

Um eine möglichst hohe Akzeptanz bei den Anwendern zu erreichen, müssen disziplinspezifische Softwarewerkzeuge und Methoden genutzt werden. [Bellalouna 2009; Hensel 2011] Im Rückblick auf vorige Kapitel wird deutlich, dass neben den Systemen MCAD, ECAD und SPS-

Programmierumgebung auch eine Simulationsumgebung zur virtuellen Inbetriebnahme zu berücksichtigen ist. Gerade letztere (siehe Kapitel 2.4) benötigt ein 3D-Modell der Maschine, welches üblicherweise im MCAD-System entsteht. Somit sind die Softwarewerkzeuge voneinander abhängig. [Klemm, et al. 2006] Dies führt im Sinne einer Minimierung des Entwicklungsaufwandes auf eine Verknüpfung der verschiedenen Softwarewerkzeuge. Im Weiteren sind die eingesetzten Softwaresysteme auf ihre Aufgabe hin optimiert, was folglich zur Verwendung proprietärer optimierter Datenformate führt. Es entstehen heterogene und verteilte Datenbestände. Nach *Hensel* [2011] ist es für 74% der Maschinen- und Anlagenbauer wichtig Datenkonsistenz entlang des Entwicklungsprozesses zu erzielen. Nach *North* [2011] entsprechen Daten, Zeichen mit einer Syntax und Informationen sind Daten mit Bedeutung bzw. Semantik. Dies ist in Abbildung 2.17 zu sehen.



Abbildung 2.17: Wissenstreppe nach *North* [2011]

Semantik bezeichnet die Lehre der Bedeutung, wobei es sich um die Beziehung zwischen Zeichen und Objekten handelt. [Blumauer, et al. 2006] Wissen ist Information gepaart mit Pragmatik (nach *North* [2011] ist dies mit Erfahrung bzw. Erwartung gleichzusetzen, was der Vernetzung von Information entspricht). [vgl. VDI 5610 Blatt 1] Diese Begriffsdefinitionen bilden die Basis für die Vorstellung der Möglichkeiten zur Verknüpfung verschiedener Softwareapplikationen.

2.6.1 Punkt-zu-Punkt-Schnittstellen und neutrale Datenformate

Werden die eingangs des Kapitels ausgeführten Begriffsdefinitionen genutzt, so ist ersichtlich, dass die Datenformate der einzelnen Applikationen zumindest implizit auch eine Semantik definieren, nämlich jene, welche die Softwarewerkzeuge nutzen um ihre Modelle zu speichern und auszulesen. Bei Punkt-zu-Punkt-Schnittstellen werden die Daten mittels Transformationsregeln überführt. (vgl. hierzu Abbildung 2.13). Die Regeln definieren dabei den semantischen Übergang der Daten zwischen den Systemen und werden vom Entwickler definiert. Bei neutralen Datenformaten ist die Semantik der Daten in Form von Dokumenten und Standards offengelegt. Die Semantik in den

Formaten ist somit implizit definiert. Die relevanten Formate für diese Arbeit sind bereits in Kapitel 2.5.1 genannt.

2.6.2 Ansätze auf Basis strukturierter Daten

Datenbanken definieren die Semantik der Daten implizit in ihrer Struktur. Die Entwicklung dieser verläuft in vier Schritten [siehe Hüsemann, et al. 2000; Kemper, et al. 2011], einer Anforderungsanalyse, einem konzeptionellen Entwurf des Datenbankschemas (oftmals in UML oder als Entity-Relationship-Modell), einem logischen und abschließend dem physikalischen Schema-Design. Die Phase des konzeptionellen Entwurfs ist einerseits noch unabhängig von der gewählten Datenbank, definiert aber die Bedeutung der Daten, also die Informationsstrukturbeschreibung [Kemper, et al. 2011], was einer semantischen Modellierung gleichkommt. Hinsichtlich der geforderten Datenkonsistenz aus der Kapiteleinleitung gewährleisteten Datenbanken dies durch Eindeutigkeit der Daten (Entitätsintegrität), Beschränkung möglicher Werte (Domänenintegrität), Fremdschlüsselbeziehungen (referenzielle Integrität) und zusätzliche Regeln (intrarelationale Integrität). [Kemper, et al. 2011] Hierzu verwendet jedes Datenbanksystem eine eindeutige Struktur, in welcher sich die Daten befinden. Diese kann unterschiedliche Ausprägungen besitzen, so können beispielsweise eine hierarchische, netzartige, relationale oder objektorientierte Struktur zugrunde gelegt werden. Um eine durchgängige Datenkonsistenz zu erreichen, ist es möglich alle Aufgaben entlang des Entwicklungsprozesses auf einer gemeinsamen physischen Datenbasis zu entwickeln. Dies entspräche einem Single-IT-System.

Der Ansatz von Data-Warehouses integriert verschiedene Datenbanken auf physischer Ebene in ein globales Schema. Die Datenbasen der einzelnen Applikationen bleiben erhalten. Dadurch können globale Anfragen bearbeitet werden. Das zentrale Dokumentations- und Steuerungswerkzeug bildet ein Metadatenbanksystem, welches Informationen über die Datenbankobjekte verwaltet. [Kemper, et al. 2010] Metadaten beinhalten somit „Daten über Daten“. [Kemper, et al. 2010]

Soll eine heterogene Systemlandschaft zusammengebracht werden, müssen nach *Hergula* [2003] semantische und strukturelle Heterogenitäten überwunden werden. Wie bereits in der Problemstellung erläutert [vgl. Göhner, et al. 2013] sind die Begrifflichkeiten in den Disziplinen nicht sauber und klar definiert, was einer semantischen Heterogenität entspricht. Da jedes Softwarewerkzeug seine Daten in einer proprietären Weise verwaltet, sind alle in *Hergula* [2003] aufgeführten Fälle der strukturellen Heterogenitäten gegeben. Diese können auch auf logischer Ebene durch föderierte Datenbanksysteme überwunden werden. Dazu wird ein gemeinsames globales Schema genutzt, welches eine Anfrage auf einzelne Datenbanken verteilt. [Helmis 2009]

Hier kann die Arbeit von *Kleiner* [2003] eingeordnet werden. *Bellalouna* [2009] nutzt für seine Integrationsplattform zur Entwicklung mechatronischer Produkte ein föderiertes Datenmodell um über Workflows und Services verschiedene Applikationen miteinander zu verbinden.

2.6.3 Ansätze auf Basis semistrukturierter Daten

Neben sehr strukturierten Datenformen, wie den Datenbanken, haben sich semistrukturierte Formen etabliert. Dazu zählen XML-Formate. Diese bestehen aus Tags und Attributen sowie den Daten selbst. Dabei wird jedes Datum entweder in einen Tag oder ein Attribut eingebettet. Diese Tags oder Attribute entsprechen Metadaten. Die Struktur der Tags und der Attribute wird im Schema-File definiert, dem sogenannten XSD-File. Die Tags wiederum sind alle einem Wurzelknoten des Files untergeordnet. Durch diese Definition ist es möglich mit Parsern die in ihnen enthaltenen Informationen zu extrahieren. Neuere neutrale Austauschformate setzen auf diese Technologie. Beispiele hierzu sind PLCopenXML, AutomationML [Drath 2010] und PLMXML [Siemens PLM 2011]. *Lercher* [2008] zeigt eine Integrationsplattform auf, welche als Kern eine eigens auf XML-basierte Sprache, die sogenannte Virtual Machine Tool Language (VMTL) besitzt. Diese implementiert das von ihm benutzte Metamodell für den Werkzeugmaschinenentwicklungsprozess aus Kapitel 2.3.2. Verschiedene Softwarewerkzeuge werden mittels XML-Adaptern an VMTL angebunden. [Lercher 2008]

2.6.4 Ansätze semantischer Technologien

Die Ansätze der semi- und strukturierten Daten eignen sich zur Integration von verschiedenen IT-Systemen. Dabei ist nach *Reitbauer* [2006] eine Integration von IT-Systemen dadurch charakterisiert, wenn aus zwei Einzelsystemen ein gemeinsames logisches Schema entsteht. Im Gegensatz dazu steht für ihn die Interoperabilität von Systemen, welche es ermöglicht die Interaktion zwischen mehreren autonomen Einheiten zu ermöglichen. In der Praxis wird in beiden Fällen von Integration gesprochen, obwohl häufig Interoperabilität gemeint ist. [Reitbauer 2006] *Drath et al.* [2011b] detaillieren die Definition von Interoperabilität zwischen Softwaresystemen zu: „Interoperabilität zwischen Engineeringwerkzeugen verfolgt das Ziel, Konsistenz zwischen den Daten einer Werkzeugkette computergestützt, systematisch und wiederholt herstellen zu können.“ [Drath, et al. 2011b] Dazu sind aus ihrer Sicht verschiedene Bedingungen zu erfüllen, so muss unter anderem die Syntax der Daten bekannt sein, die Semantik derer definiert, sowie Interoperabilitätsfunktionen bereitgestellt werden.

Epple [2011] schlägt zur Interoperabilität global definierte Merkmalsträger in hierarchischen Vererbungsstrukturen und Merkmalen vor. Diese Merkmalsträger repräsentieren Klassen realer

Gegenstände und können so standardisiert über alle Applikationen hinweg definiert werden. Sind den Applikationen diese Merkmalsträger und deren Merkmale bekannt, können Daten in einem losen Verbund zum Austausch zwischen Systemen genutzt werden. Eine solche Merkmalstandardisierung repräsentiert, wie bereits in Kapitel 2.5.1 eingeführt, *eCl@ss*. Ein instanziiertes Merkmalsträgertyp kann im Rahmen der Informatik als Objekt betrachtet werden. Seine Merkmale sind die zugehörigen Eigenschaften.

Reichenberger [2010] definiert neben den Merkmalen noch weitere Eigenschaften von Objekten, nämlich Relationen zu anderen Objekten. Zur Visualisierung dieser Verbindungen eignen sich gerichtete Graphen. Die Objekte werden durch Knoten, die Relationen durch gerichtete Kanten mit entsprechenden Benennungen repräsentiert. Die Kante und die beiden angrenzenden Knoten bilden dabei ein Tripel. Diese Technik, Informationen als Relationen zwischen Objekten abzulegen bezeichnet *Reichenberger* [2010] als semantische Netze. Ein solches zeigt Abbildung 2.18. Darin kann der Quellknoten einer gerichteten Kante als Subjekt, die Kante als Prädikat und der Zielknoten als Objekt verstanden werden, was einem natürlichsprachlichen Satzbau ähnelt. Ein Kerngedanke dieser Modellierung ist die Objektidentität. [Reichenberger 2010] Diese wird durch Taxonomien nicht unterstützt.

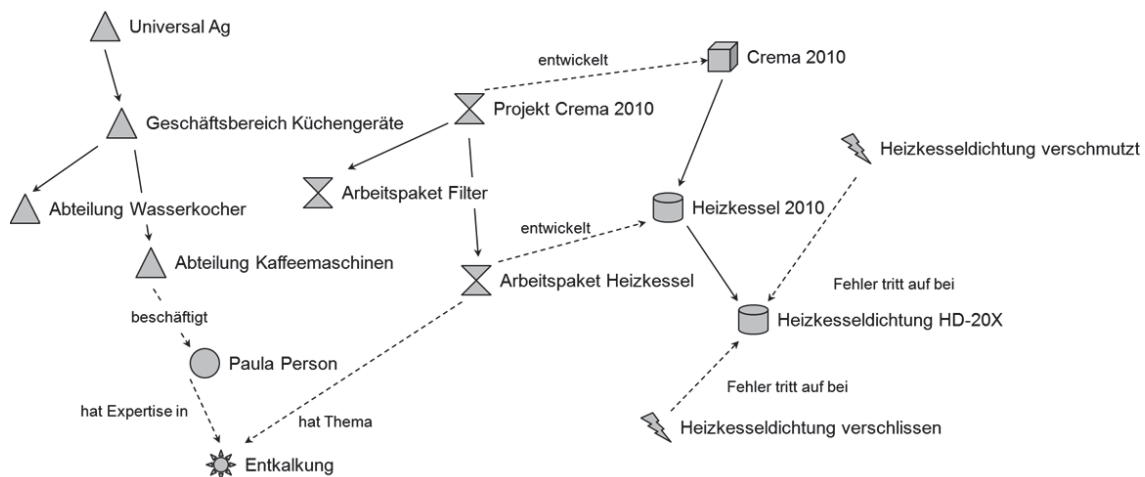


Abbildung 2.18: Beispiel eines semantischen Netzes nach *Reichenberger* [2010]

Weitere Kerngedanken sind die Trennung des Objektes von der Benennung und die Redundanzfreiheit durch Ableitung. [Reichenberger 2010] Ersterer drückt aus, dass die Benennung eines Objektes durch mehrere Namen (Synonyme) erfolgen kann. Letzterer leitet auf Basis von bestehenden Relationen weitere Informationen ab. Werden den Knoten natürlichsprachliche

Dokumente zugeordnet, spricht man auch von Topic Maps. Eine informationstechnische Umsetzung kann beispielsweise im Resource Description Framework erfolgen.

Litto [2004] (vgl. Kapitel 2.5.2) nutzt diese Technologie um interdisziplinäre Parameter in der Bibliothek zu modellieren. *Arnold et al.* [2012] setzen auf UML, modellieren aber ebenfalls gerichtete Graphen. *Bertsche et al.* [2007] definieren ein solches semantisches Netz um eine innovative Produktentwicklung mittels der Vernetzung eines MCAD-Systems, einer FMEA-Software und eines Kostenmodells zu ermöglichen. *Conrad* [2010] untersucht semantische Netze zur Erfassung und Nutzung von Wissen im Rahmen eines Produktentwicklungsprozesses. Dabei soll das Netz helfen den beteiligten Personen Zugang zu vorhandenem Wissen zu geben und das von ihnen generierte Netz Anderen explizit zu machen. Somit ist das semantische Netz als produktentwicklungsprozessbegleitendes Informationssystem anzusehen.

Für *Faatz* [2004] bestehen semantische Netze aus Tokens und Typen. Typen bilden dabei ein spezifisches Konzept oder eine Klasse eines Objektes ab, während Tokens Instanzen eines bestimmten Objektes sind. Taucht beides in einem semantischen Netz als Knoten auf, müssen diese mittels einer „ist ein“ Relation verbunden werden. Im Bereich der semantischen Technologien findet sich neben den semantischen Netzen auch der Begriff der Ontologien. Beiden ist gleich, dass sie die Semantik der Daten explizit vernetzen und definieren. Eine einheitliche Abgrenzung der Begrifflichkeiten existiert nicht. *Faatz* [2004] leitet her, dass aufgrund einer sprachlichen Ungenauigkeit bei der Übersetzung englischer Definitionen ins Deutsche, in Ontologien, im Gegensatz zu semantischen Netzen keine Konzepte, sondern Begriffe existieren. Zu ihnen gibt es wiederum Ober- und Unterbegrifflichkeiten. Eine Ontologie wird somit von Experten eines bestimmten Fachgebietes definiert, was einer Gültigkeitseinschränkung gleichkommt. Nach *Conrad* [2010] kann eine Ontologie benutzt werden um die Typisierung der Knoten und Kanten eines semantischen Netzes zu definieren. Ontologien können mit OWL, einer Erweiterung des RDF-Formates, abgebildet werden. *Schreiber et al.* [2011] zeigen, dass eine Extraktion von Daten in einem mechanischen CAD- und einem ECAD-System in eine Ontologie dazu genutzt werden kann, eine systemübergreifende Navigation zu ermöglichen um Inbetriebnahmezeiten zu verkürzen.

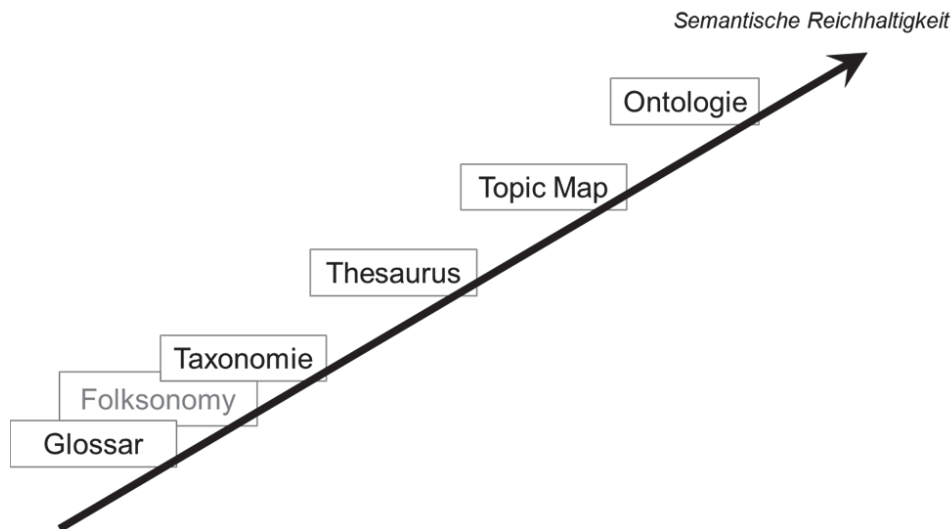


Abbildung 2.19: Semantische Treppe nach *Blumauer et al.* [2006]

Eingeordnet werden können die diskutierten semantischen Technologien auf der in Abbildung 2.19 dargestellten Treppe nach ihrer semantischen Reichhaltigkeit.

2.7 Defizite des Standes der Technik und Handlungsbedarf

Im Rahmen der Problemstellung wird dargestellt, dass die Entwicklung von Sondermaschinen sehr kundenindividuell und immer noch in sequenzieller Weise erfolgt. Die *VDI-Richtlinie 2206* zeigt ein grobes Vorgehensmodell auf, wie eine Parallelisierung durchgeführt werden kann. Allerdings sind im Rahmen der Sondermaschinenentwicklung Randbedingungen gegeben, welche eine Anpassung der dort definierten Phasen notwendig macht. Im Weiteren werden ausgehend von der Problemstellung bei der Sondermaschinenentwicklung die Anforderungen an die zu entwickelnde Methode und deren Softwareunterstützung abgeleitet. Dies dient der Bewertung des Standes der Technik und gleichzeitig zur Ableitung des Handlungsbedarfs.

2.7.1 Anforderungsdefinition zur Bewertung des Standes der Technik

Aus den Defiziten in Kapitel 1.2 ist ersichtlich, dass die Fabrikplanung Anforderungen an eine neue Maschine aus mechatronischer Sicht nicht ausreichend spezifiziert. Daher muss die Methode eine Anforderungserhebung unterstützen, um unnötigen Abstimmungsaufwand zur Integration der Maschine in das Produktionssystem zu vermeiden. Dies wird in Anforderung A1 formuliert.

A1: Die Methode muss die Phase zur Anforderungserhebung zu Beginn eines Sondermaschinenentwicklungsprozesses hinsichtlich der Integration von Maschinen in Produktionssysteme unterstützen.

Aufgrund der hohen Kundenindividualität von Sondermaschinen ist die Charakteristik des Entwicklungsprozesses mit jenem von Produkten vergleichbar. Hierbei ist eine abstrakte Problem-beschreibung notwendig, um den Kreativitätsprozess zu fördern. Daraus erschließt sich die Anforderung A2:

A2: Die Methode muss den Kreativitätsprozess mit einer abstrakten Problembeschreibung unterstützen.

In der Phase des Systementwurfs sind alle Disziplinen gleichermaßen einzubeziehen, um ein mechatronisches Optimum definieren zu können. Dieser Entwurf muss in einer interdisziplinären Maschinenbeschreibung münden, die eine parallele Detaillierung der Mechanik, Elektrik und Automatisierung ermöglicht. Daher lässt sich folgende Anforderung formulieren:

A3: Die Phase des Systementwurfs muss alle Disziplinen gleichermaßen beteiligen, um ein „mechatronisches Optimum“ des Maschinenkonzepts zu definieren, welches anschließend nahtlos genutzt werden kann um eine parallele Ausdetaillierung in allen beteiligten Disziplinen zu ermöglichen.

Im Kapitel 1.2 ist der Fall beschrieben, dass Kabellängen während der Detaillierungsphase in der Mechanik definiert aber vom Elektriker zur Validierung benötigt werden. Dieses Beispiel illustriert, dass im Rahmen bestehender Ansätze bei Vernachlässigung einzelner Disziplinen (beispielsweise Elektrik) die interdisziplinären Modellbeschreibungen für diese Arbeit nicht direkt nutzbar sind. Weiterhin zeigt das genannte Beispiel die Anforderung A4 auf:

A4: Ein Konsistenzabgleich der Modelle während der Detaillierungsphase ist notwendig und nicht allein durch eine virtuelle Inbetriebnahme erreichbar.

Allerdings besitzt die Methode der virtuellen Inbetriebnahme von Maschinen die Fähigkeit, eine bessere Kooperation und Kommunikation zwischen den beteiligten Disziplinen herbeizuführen. Weiterhin kann durch deren Anwendung eine enorme Aufwandsreduktion im Rahmen der Inbetriebnahme erfolgen. Aufgrund des Unikatcharakters von Sondermaschinen wird kein Prototyp erstellt, was die Notwendigkeit einer Absicherung der Steuerungsprogrammierung auf Basis eines virtuellen Modells unterstreicht. Um beide Vorteile optimal auszunutzen wird eine

A5: phasenübergreifende interdisziplinäre Simulation gefordert.

Weiterhin ist im Sondermaschinenbau zu berücksichtigen, dass aufgrund der Kundenindividualität der Maschinen eine Trennung zwischen Auftragsabwicklungs- und Modulentwicklungsprozess keine Vorteile hinsichtlich Aufwands- und Kostenminimierung mit sich bringt. Dennoch muss eine Wiederverwendung von bereits existierenden Modulen und standardisierten Teilen hinsichtlich des

hohen Potenzials zur Aufwandsreduktion umgesetzt werden. Daraus lässt sich folgende Anforderung ableiten:

A6: Es ist eine Wiederverwendung von mechatronischen Modulen vorzusehen, wobei sowohl die Aufwände zur Erstellung als auch deren Nutzung einzubeziehen sind.

Eine breite praktische Akzeptanz der Methode und deren Unterstützung kann nur durch bereits eingeführte Beschreibungsmittel und Werkzeuge erlangt werden. Ausformuliert ist dies in Anforderung A7:

A7: Nutzung eingeführter Softwarewerkzeuge und Beschreibungsmittel.

Die Aufgabe der Entwickler und Konstrukteure besteht im Wesentlichen darin, auf Basis von gegebenen Informationen unter Einbezug ihrer Kreativität und ihres Wissens Entscheidungen zu treffen. Um dies optimal zu unterstützen, müssen die, für ihre Aufgabe relevanten Informationen unter minimierten Aufwänden zur Verfügung stehen und genutzt werden können. Daher ist es erforderlich jede notwendige Information entlang des Prozesses nicht öfter als einmal einzugeben. Dies führt zur Anforderung:

A8: Eine Wiederverwendung der Informationen über den gesamten Entwicklungsprozess ist notwendig, unabhängig davon wann und wo diese Information entsteht und explizit gemacht wird.

Zusammenfassend zeigt Abbildung 2.20 die angeführten Anforderungen und deren Einordnung am Vorgehensmodell der *VDI-Richtlinie 2206*.

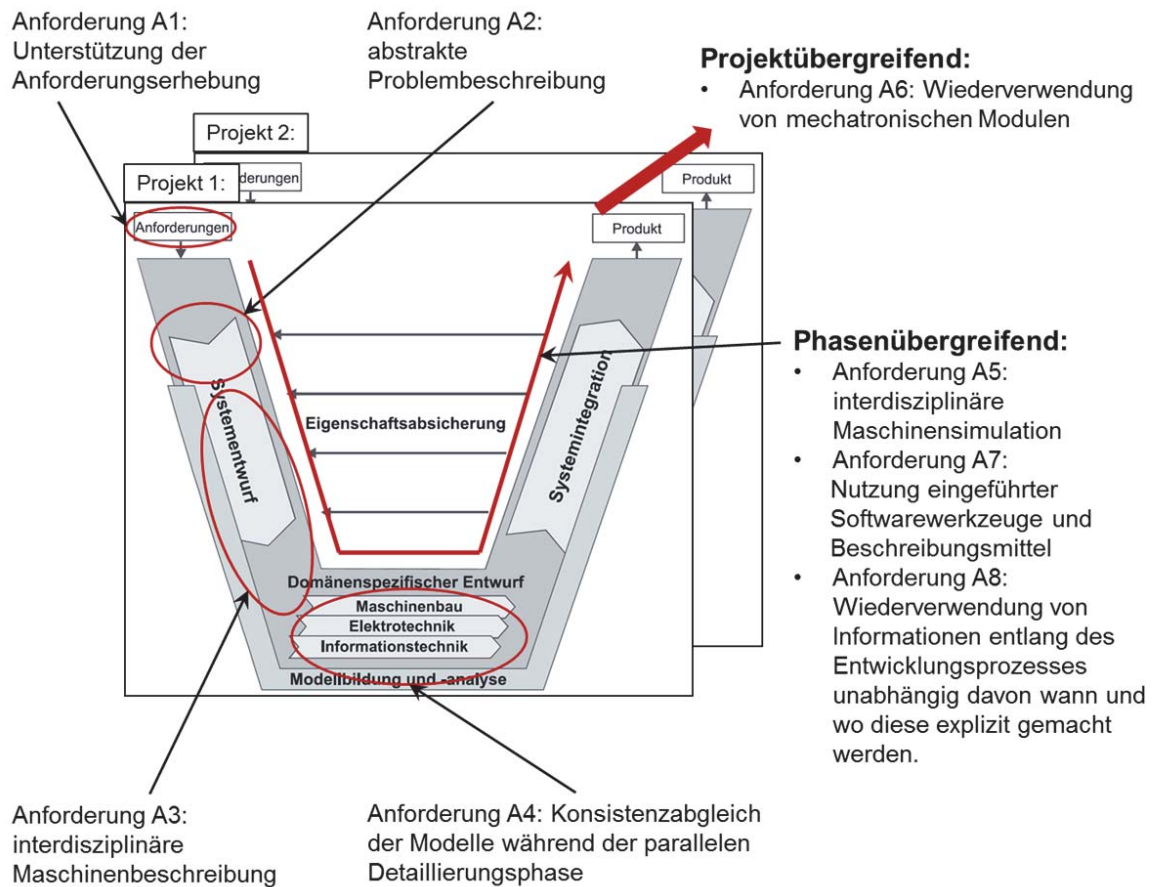


Abbildung 2.20: Anforderungen an eine optimale softwaregestützte Methode zur Entwicklung von Sondermaschinen bezogen auf die Phasen der *VDI-Richtlinie 2206* [2004-06]

Im Weiteren werden diese Anforderungen herangezogen um die aufgezeigten Ansätze zu bewerten und die Handlungsfelder abzuleiten.

2.7.2 Bewertung der interdisziplinären Entwicklungsmethoden und Ableitung des methodischen Handlungsbedarfs

Die in den vorigen Kapiteln dargestellten Methoden zur Entwicklung von mechatronischen Produkten, sowie Maschinen und Anlagen sind in Abbildung 2.21 wiederzufinden. Diese werden auf Basis der Anforderungen A1-A7 bewertet. Anforderung A8 wird nicht heran gezogen, da diese keinen Einfluss auf die Methodik besitzt.

Lediglich *Pahl et al.* [2007] betrachten, wie die Phase der Anforderungserhebung durchzuführen ist. Dazu wird eine Sachmerkmaleitlinie genannt, welche verschiedene Kategorien für die Erhebung von Anforderungen vorgibt. Allerdings spiegelt sich die mechatronische Ausgangssituation zur

Integration von Maschinen in bestehende oder geplante Produktionssysteme in diesen genannten Sachmerkmalen nicht explizit wieder.

Die Methoden der problemorientierten Produktentwicklung beginnen im Anschluss an die Anforderungserhebung mit einer abstrakten funktionalen Modellierung. Diese bildet gleichzeitig eine interdisziplinäre Systembeschreibung, welche aber häufig auf zu regelnde Systeme ausgerichtet, zu abstrakt ist oder nicht alle Disziplinen gleichermaßen berücksichtigt. *Kallmeyer* [1998] und *Flath* [2002] fügen in der Wirkstrukturmodellierung zwar einen Hinweis auf die Domänen hinzu. Die weitere Detaillierung wird ebenso wenig, wie eine simulative Unterstützung angesprochen. *Bathelt* [2007] betrachtet eine parallele Ausdetaillierung, allerdings nur in Bezug auf die Disziplinen der Mechanik und Automatisierung. Eine Wiederverwendung wird in keiner der problemlösungsorientierten Methoden explizit berücksichtigt. Es erfolgt eine Modularisierung, allerdings mit dem Ziel die Module möglichst getrennt voneinander weiterzuentwickeln. Dazu sind die Abhängigkeiten zwischen den Modulen zu minimieren.

Im Gegensatz hierzu wird im Rahmen der prozessorientierten Methoden eine Wiederverwendung bereits existierender Module fokussiert. Ansätze wie *Föederal* [2004] und *Kiefer* [2007] setzen auf mechatronische Komponenten in Bibliotheken. Diese Komponenten werden über einen separaten Entwicklungsprozess, entweder durch einen Lieferanten oder das eigene Unternehmen, erstellt. Dadurch erfolgt eine organisatorische Trennung zwischen Modulentwicklung und Auftragsabwicklung. Gleichzeitig hat dies zur Folge, dass keine interdisziplinäre Beschreibung der Maschine oder Anlage während der Auftragsabwicklung vorliegen muss, wenn die Komponenten entsprechend gestaltet sind. Aufgrund der Ausrichtung auf Produktionsprozesse, wird häufig der Prozessschritt als abstrakte Funktion angenommen, was nur zum Teil einer Problembeschreibung entspricht. Allen anderen produktionsprozessorientierten Methoden ohne getrennte Entwicklungs- und Auftragsabwicklung ist gemein, dass wenigstens eine Disziplin unberücksichtigt bleibt. Der Ansatz von *Lercher* [2008] besitzt im Rahmen der parallelen Detaillierung die größte Überdeckung, es werden lediglich die Elektrik und Fluidtechnik vernachlässigt.

		Anforderungsdefinition	Systementwurf		Detaillierung		Simulationsunterstützung entlang des gesamten Prozesses	Wiederverwendung von mechatronischen Modulen	Nutzung eingeführter Werkzeuge und Beschreibungsmittel	
		Integration in Produktionssystem	abstrakte Problembeschreibung	Interdisziplinäre Maschinenbeschreibung und Detaillierung in allen Disziplinen		Konsistenzabgleich				
Problemorientierte Methoden	Pahl/Beitz	◐	●	○	○	○	○	○	●	
	Kallmeyer	○	●	○	○	○	○	○	◐	
	Flath	○	●	○	○	○	○	○	◐	
	Gehrke	○	●	○	○	○	○	○	◐	
	Kleiner 2003	○	○	○	○	○	○	○	●	
	Bathelt	○	●	◐	○	○	○	○	●	
	Kleiner 2013	○	●	○	○	○	◐	○	●	
	van Beeck	○	●	○	○	○	○	○	○	
	Campean	○	●	○	○	○	○	○	○	
Prozessorientierte Methoden	Einheitliche Auftragsabwicklung	Dohmen	○	◐	◐	○	○	○	●	○
		Lercher	○	◐	◐	○	○	○	○	◐
		VDW-Richtlinie	○	◐	◐	○	○	○	◐	●
		Pömbacher	○	◐	○	○	○	○	○	◐
		Hensel	○	○	◐	○	○	●	○	○
		Lindworsky	○	○	◐	○	○	●	◐	○
		Aquimo	○	○	◐	○	○	◐	◐	◐
Getrennte Prozesse	Föderal	○	◐	◐	○	○	○	●	●	
	Kiefer	○	○	◐	○	○	◐	●	●	

○ Keine Berücksichtigung ◐ Teilweise berücksichtigt ● Vollständig berücksichtigt

Abbildung 2.21: Vergleich und Eignung derzeitiger Methoden hinsichtlich der Entwicklung von Sondermaschinen

Aquimo besitzt das Ziel, Module für eine mechatronische Bibliothek zu entwerfen. [Litto 2010] Wird ein solches Modul als Maschine betrachtet, kann der Ansatz auch als eigenständige Methode angesehen und bewertet werden. Darin wird eine parallele Detaillierungsphase der beteiligten Disziplinen, basierend auf einem gemeinsamen Systementwurf aufgezeigt. Eine planmäßige Vorgehensweise dieser Phase bleibt der Ansatz allerdings schuldig. Daraus lässt sich ableiten, dass eine Analyse der disziplinspezifischen Vorgehensweisen notwendig ist, welche als Ergebnis eine interdisziplinäre Maschinenbeschreibung am Ende des Systementwurfs liefert. Diese muss eine

parallele Ausdetaillierung der verschiedenen Disziplinen ermöglichen und bildet die Basis, um auch deren Ablauf zu beschreiben. Ein Konsistenzabgleich während der parallelen disziplinspezifischen Detaillierung wird in keiner der genannten Methoden berücksichtigt. Um der Anforderung zur Simulationsunterstützung für Sondermaschinen nachzukommen, eignet sich die Methode der virtuellen Inbetriebnahme. *Aquimo* versucht diese in allen Phasen einzusetzen, wofür je Phase eine separate Bibliothek aufgebaut werden muss. Dieser Erstellaufwand und die mangelnde Einbindung eines MCAD-Systems führen zu Abzügen in der Bewertung.

Einige Methoden entwickeln neue Beschreibungsformen, um modellgetriebene Ansätze auch im Bereich des Maschinen- und Anlagenbaus umzusetzen. *Hensel* [2011] und *Lindworsky* [2011] setzen hierzu auf dieselbe Modellierungssprache. Diese ermöglicht einen durchgängigen Einsatz der Simulation und der Steuerungsprogrammgenerierung. Andere Ansätze erweitern bestehende Beschreibungssprachen oder kombinieren bestehende mit neuen, um bestimmte Bereiche abzudecken. So beispielsweise *Aquimo* zur Beschreibung von Bewegungspfaden. In der Forschung ist SysML sehr populär um modellgetriebene Entwicklungen umzusetzen. Allerdings zeigt *Albers* [2013], dass diese gerade in der Praxis noch nicht angekommen ist. Allen Ansätzen, basierend auf neu entwickelten oder nicht eingeführten Modellierungssprachen, widersprechen allerdings Anforderung A7.

2.7.3 Bewertung der Ansätze zur Aufwandsreduktion

Eine Minimierung der Entwicklungsaufwände wird erreicht, wenn insbesondere Anforderung A8 erfüllt wird. Dabei steht der Austausch interdisziplinär relevanter Informationen im Vordergrund. Dies kann von disziplinspezifischen Datenformaten nicht geleistet werden. AutomationML als Einziges mit interdisziplinärem Ansatz berücksichtigt derzeit keine Modellierung der Elektrik. Ansätze, wie der von *Arnold et al.* [2012], welche eine äußerst genaue Spezifikation des Entwicklungsablaufes und der Modellabhängigkeiten benötigen, sind als konträr zu Anforderung A8 zu bewerten. Modellgetriebene Entwicklungen auf Basis von SysML und UML widersprechen Anforderung A7. Erfüllt wird diese durch die Nutzung bereits existierender Dokumente und Informationen als abstraktes Modell, wie dies von *Kufner* [2012] gezeigt wurde. Angesichts der Struktur von Sondermaschinen ist ersichtlich, dass ein vorheriger Aufbau einer Bibliothek oder gar die Bereitstellung von Daten durch Lieferanten nur bei Zukaufteilen der Elektrik sinnvoll erscheint, wie es bereits in *Litto et al.* [2012] und *Possel-Dölken* [2010] vorgestellt ist. Für aggregierte Elemente, wie beispielsweise einen Greifer muss ein aufwandsminimierter Ansatz zur Erstellung solch mechatronisch orientierter Einheiten gefunden werden.

2.7.4 Bewertung der Ansätze zur Vernetzung von Softwarewerkzeugen

Hinsichtlich der Verknüpfung von Softwarewerkzeugen sind zwei wesentliche Anforderungen zu beachten. Einerseits die Nutzung bestehender Werkzeuge mit ihren proprietären Datenformaten (A7) und andererseits die Notwendigkeit, jede benötigte Information exakt einmal einzugeben (A 8). Ersteres schließt eine gemeinsame Datenbasis aus, was einer Einschränkung der Ansätze auf föderierte Datenbanken oder semantischen Technologien gleichkommt. Die Kernidee der semantischen Technologien von Redundanzfreiheit durch Ableitung kommt Anforderung A8 gleich. Daher ist eine Umsetzung zur Vernetzung der einzusetzenden Softwarewerkzeugen mit semantischen Technologien anzustreben.

2.7.5 Fazit und Handlungsbedarf

Es ist derzeit keine Arbeit bekannt, welche es dem Sondermaschinenbau ermöglicht, eine parallele disziplinspezifische Entwicklung seiner Maschinen unter minimierten Aufwänden zu betreiben. Die disziplinspezifischen Vorgehen sind in der Literatur nur unzulänglich beschrieben. Dieser Umstand drückt sich unter anderem in der Tatsache aus, dass gerade im Bereich der Fabrikplanung die Phase der Feinplanung nicht oder nur sehr oberflächlich dargelegt ist.

Im Bereich der Forschungsarbeiten können zwei Gruppen von Arbeiten identifiziert werden. Die erste Kategorie setzt standardisierte Bibliotheken ein und bildet kundenindividuelle Lösungen durch Varianten der Maschinen ab. Dabei wird kein abstrakter Problemlösungsprozess benötigt. Dem stehen die Ansätze der Produktentwicklung gegenüber, welche versuchen hoch innovative Produkte mit ausgeprägter Problemlösung zu entwickeln, die einem Kunden viele Vorteile bieten können. Diese Ansätze berücksichtigen allerdings keine Wiederverwendung und deren enormes Potenzial der Aufwandsminimierung. Deshalb sind zunächst die disziplinspezifischen Vorgehensweisen, deren Abhängigkeiten und die Problematik der Wiederverwendung zu analysieren. Davon ist ein Zielprozess für die Entwicklung von Sondermaschinen abzuleiten. Die Detaillierung des Prozesses findet auf Basis der Analyse statt. Die Analyse definiert zudem letzte Randbedingungen für die Auswahl und Kopplung der Softwarewerkzeuge zur Unterstützung der einzelnen Phasen. Die Verknüpfung der Softwarewerkzeuge ist auf Basis semantischer Technologien umzusetzen.

3 Analyse der disziplinspezifischen Vorgehensweisen und Ableitung eines Methodenkonzeptes

Die Basis für die mechatronische Entwicklungsmethode bildet eine Analyse der disziplinspezifischen Entwicklungsvorgehensweisen und Modellierungen. Dazu werden zunächst die einzelnen Modelle und deren Entstehungen betrachtet. Im Anschluss erfolgt ein Vergleich der Modelle, um Abhängigkeiten zwischen den Disziplinen zu identifizieren. Insbesondere die Betrachtung der Modellerstellung über den zeitlichen Verlauf führt auf das Methodenkonzept.

3.1 Mechanische Vorgehensweise und Modellbildung

In Kapitel 2.1.1 ist bereits das Entwicklungsvorgehen im Rahmen der mechanischen Konstruktion aufgezeigt. Zur Unterstützung dieses Vorgehens werden MCAD-Systeme eingesetzt. Zu Beginn eines Entwicklungsprozesses werden Anforderungen üblicherweise in Word-Dokumenten gesammelt. Mechanische Konstrukteure verzichten in der Praxis meist auf eine explizite funktionale Modellierung. Sie gehen durch eine implizite Abstraktion des Problems sofort zur Definition von prinzipiellen Lösungen über. Eine Dokumentation verschiedener Prinzipien erfolgt häufig nicht oder nur auf Basis von Skizzen. Erst nach Definition eines Gesamtkonzeptes, wird dieses im MCAD-System modelliert. Dazu stehen dem mechanischen Konstrukteur Mittel zur Beschreibung von Geometrien zur Verfügung. Die Geometriebeschreibungen werden in sogenannten Parts oder Bauteilen gespeichert. Ein Zusammenschluss mehrerer Parts zu einer Baugruppe wird auch als Assembly bezeichnet. Dies ist in Abbildung 3.1 dargestellt. Jedes der geometrischen Teile kann als physische Komponente betrachtet werden. Physische Komponenten zeichnen sich durch eine mögliche Zuordnung einer Bestellnummer oder Fertigungszeichnung aus. Eine Baugruppe selbst enthält üblicherweise keine geometrische Beschreibung an sich, sondern all jene Bauteile, welche mitgeliefert oder montiert werden müssen. Daher kann jedes Element einer Baugruppenstruktur im MCAD-System als „bestellbar“ angesehen werden, was eine Repräsentanz in einer Stückliste nach sich ziehen kann. Ein Maschinenprojekt wird durch einen Baugruppenknoten repräsentiert. Diesem Elternknoten werden alle mechanischen Unterbaugruppen und Teile der Maschine zugeordnet. Dabei speichert das Datenobjekt der Maschinenbaugruppe nur die Instanzen der nächsten Unterebene. In Abbildung 3.1 ist zu erkennen, dass elektrische, pneumatische oder hydraulische Anschlüsse geometrisch in Form von Vektoren, Bohrungen oder Gewinden dargestellt sind. Kabel, Schläuche oder Rohre sind wiederum einzelne Bauteile oder Baugruppen. Gleiches gilt für Motoren

und Sensoren. Motoren werden im mechanischen Sprachgebrauch auch als Aktoren bezeichnet. Dies liegt darin begründet, dass die Mechanik häufig nicht alle Teile detailliert darstellt. So ist es üblich beispielsweise Schaltschränke oder Ventilinseln als geometrische „Black-Boxen“ zu betrachten, welche entsprechenden Raum einnehmen. Eine Detaillierung ihres Innenlebens erfolgt nicht. Dadurch wird aus mechanischer Sicht eine Wirkung auf das mechanische Grundsystem, ausgehend von einem Signal, durch ein Wirkelement erzeugt, da Stellelemente oftmals nicht dargestellt werden.

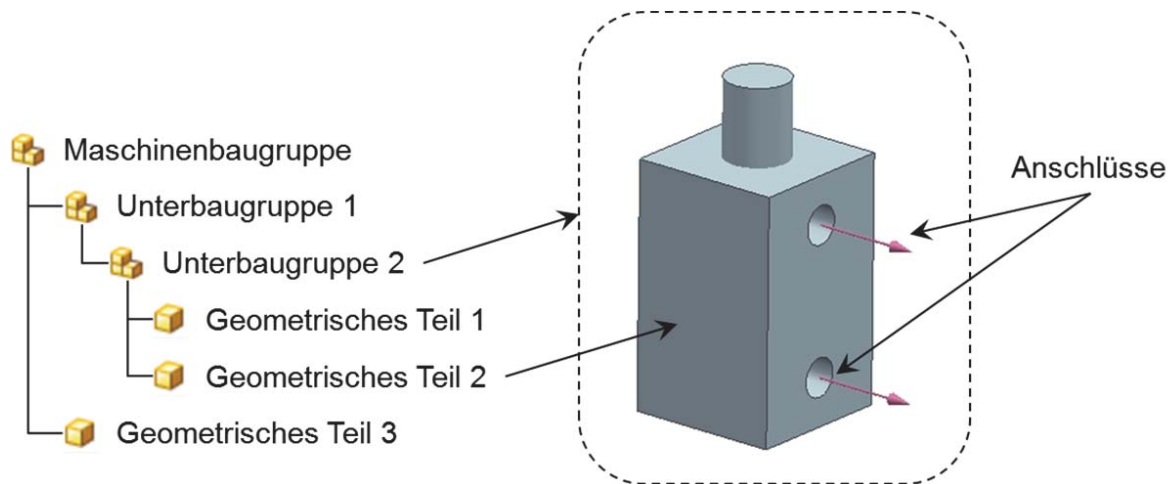


Abbildung 3.1: Baugruppen, Teile und Anschlussbeschreibung in einem MCAD-System

Ebenfalls schwierig ist die Beschreibung der kinematischen Zusammenhänge zwischen Bauteilen. Diese können in herkömmlichen MCAD-Systemen nicht explizit definiert werden, sind aber durch die geometrische Beschreibung und die physikalischen Randbedingungen implizit gegeben. Die mechanischen Stellglieder (Wirkelemente) und Sensoren werden in einem sequenziellen Entwicklungsprozess manuell aus der Baugruppenstruktur herausgesucht und in die Sensor-/Aktorliste übertragen. Eine explizite Beschreibung des Sollverhaltens der Maschine, also deren Bewegungsabfolge, erfolgt entweder nicht explizit oder auf Basis von Gantt-Charts in Dokumentenform. (vgl. Kapitel 2.1.2)

3.2 Elektrotechnische Vorgehensweise und Modellbildung

Die Sensor-/Aktorliste bildet eines der Hauptdokumente für die weitere elektrotechnische Umsetzung. Diese liegt entweder bereits in Form eines Office-Dokumentes aus der mechanischen Konstruktion vor oder der Elektrokonstrukteur erstellt diese manuell auf Basis der mechanischen Baugruppenstruktur. Weiterhin bildet in einem sequenziellen Entwicklungsprozess das mecha-

nische Modell die Basis um durch Erklärungen den elektrischen Konstrukteuren, den Aufbau und die Bewegungsabfolge in der Maschine zu verdeutlichen. Dabei können explizite Beschreibungen und das 3D-Modell sehr hilfreich sein. Gleichzeitig führt dies zu einer intellektuellen Modelltransformation von der mechanischen Modellbeschreibung hin zu einer elektrotechnischen. Als wichtiger Wissenserwerb der Elektrokonstruktion ist hierbei der Zusammenhang zwischen Maschinenbewegungen und einzelnen Wirk- bzw. Messelemente anzusehen. Dies bildet gleichzeitig die Strukturierungsgrundlage aus elektrischer Sicht, welche in einer Maschinenstruktur (siehe Abbildung 3.2) mit verschiedenen Modulen mündet.

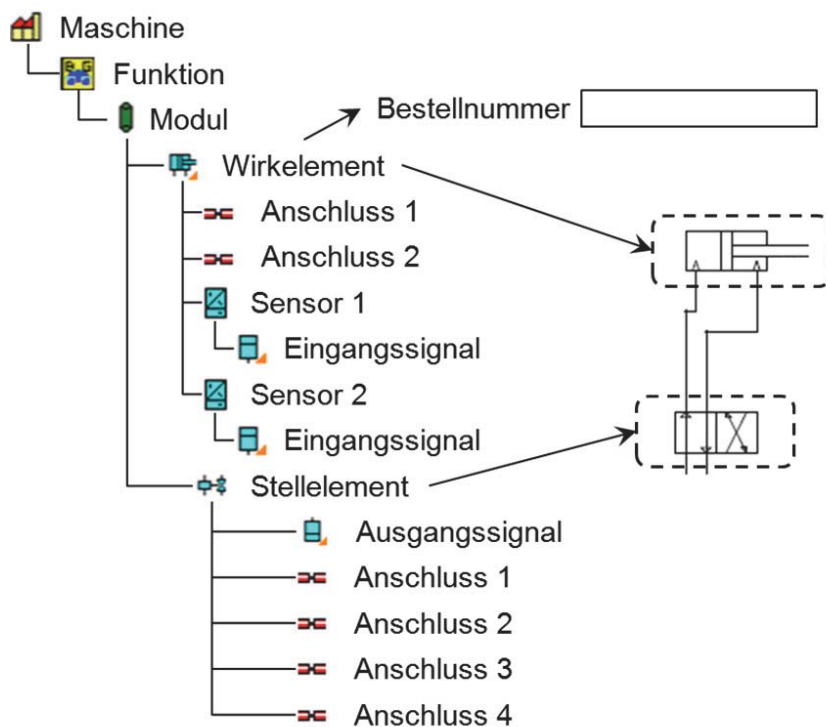


Abbildung 3.2: Aufbau der Maschinenstruktur in einem ECAD-System

Zusätzlich zur Maschinenstruktur ist eine Beschreibung des Wirkzusammenhangs innerhalb der Maschine notwendig, um beispielsweise Abhängigkeiten der Verschaltung in Folge paralleler Bewegungsausführung besser verstehen und berücksichtigen zu können. Da Elektrokonstrukteure keine Ausbildung in der Bedienung von MCAD-Systemen besitzen und ihre organisatorische Rolle im Rahmen eines Entwicklungsprojektes einen Zugriff auf diese Daten häufig nicht zulässt, werden diese Informationen in Form eines Technologieschemas in der Elektroplanungsumgebung manuell aufgebaut und vorgehalten. Dieses Schema ist in der Anlagenplanung mit einem Layout zu

vergleichen und zeigt schematisch (2-D Darstellung) die Module, deren Anordnung und den Materialfluss zwischen ihnen auf.

Zusammen mit der Maschinenstruktur, der Sensor-/Aktorliste und dem Wissen über die Bewegungsabläufe bildet dies die Basis für die weitere elektrotechnische Konstruktion. Die Elemente der Sensor-/Aktorliste werden entsprechenden Modulen der Maschinenstruktur durch Instanziierung aus der elektrotechnischen Bibliothek zugeordnet. Diese Instanzen besitzen bereits Symboldefinitionen für unterschiedliche Schaltplanausprägungen, eine große Anzahl verschiedener Attribute und entsprechende Signal- und Anschlusstypen in Form von Kindobjekten. Bibliotheksobjekte, welche keine Funktionen oder Module (sogenannte Gruppierungsobjekte) repräsentieren, werden in zwei weitere Kategorien unterschieden: klassifizierte bzw. typisierte Objekte und Echtgeräte. Die erste Kategorie besitzt bei Instanziierung viele verschiedene Attribute, welche noch nicht mit Werten belegt sind. Ein solches Attribut bildet beispielsweise die Bestellnummer. Im weiteren Projektfortschritt werden diese Attribute auf Basis von getroffenen Entscheidungen oder neuen Informationen mit Werten belegt, was je nach ECAD-Ausprägung zur Einschränkung der Auswahl von Echtgeräten führen kann. Echtgeräte bilden die zweite Kategorie. Sie können direkt bestellt werden, wodurch alle Eigenschaften eindeutig beschrieben sind. Sensoren und Wirkelemente der Sensor-/Aktorliste sind in einem sequenziellen Entwicklungsprozess bereits vollständig definiert und werden somit durch Instanziierung der Echtgeräte unter dem entsprechenden Modul modelliert.

Das Technologieschema bildet die Grundlage um Bereichsplanungen bei großen Maschinen, für verschiedene HMI's, Steuerungen und Notausbereiche vorzunehmen. Dabei erfolgt eine Zuordnung der Module auf bestimmte Bereiche. Diese definiert, welche Steuerung für die einzelnen Module zuständig ist, welche Bedienpanels Zugriff auf einen Bereich besitzen und welcher Bereich im Notfall abgeschaltet werden muss. Aufgrund der Komponentenzuordnung zu den einzelnen Modulen und deren Bereichszuweisung ist auch eine entsprechende Komponentenzuweisung definiert.

Danach erfolgt die Verschaltung der Komponenten, wobei dies zunächst aus Versorgungs- dann aus Signalsicht umgesetzt wird. Dazu werden in Abhängigkeit des Bewegungsablaufes (vgl. Kapitel 2.1.2), die Wirk- bzw. Messelemente mit Stellelementen zum Hauptkreislauf verschalten. Die Verschaltung wird auf den Plänen durch Striche zwischen Anschlüssen der Symbole visualisiert. Verschiedene Strichtypen repräsentieren dabei Kabel, Schläuche oder Rohre, welche wiederum als typisierte Objekte in der Maschinenstruktur einzupflegen sind. Wurden im

Rahmen der mechanischen Konstruktion noch keine Stellelemente definiert, so werden diese zunächst als klassifizierte Objekte in die Maschinenstruktur eingefügt. Eine konkrete Auswahl des Echtgerätes für das entsprechende Stellelement wird häufig noch durch vorgelagerte Geräte, beispielsweise der Signalverschaltung oder Versorgung beeinflusst. Außerdem sind Schlauch-, Rohr- und Kabellängen mit der Mechanik abzustimmen, da diese, wie bereits aufgezeigt, eine gewisse Grenzlänge nicht überschreiten dürfen. Im weiteren Verlauf werden der Hilfsmedien-, Hilfsversorgungs- und der Versorgungskreislauf erstellt, indem benötigte Elemente aus der Bibliothek zunächst als klassifizierte Objekte instanziiert und mit anderen verschalten werden. Gleichzeitig sind noch Betriebsmittelkennzeichen für die einzelnen Geräte zu vergeben. Im Anschluss daran erfolgt die Auswahl der Echtgeräte. Diese müssen zur Ableitung der Stücklisten auch in die Produktstruktur eingeordnet werden. Die Erstellung des Signalkreislaufes erfolgt analog. Gleichzeitig ergibt sich dadurch die Hardwarekonfiguration der SPS.

3.3 Automatisierungstechnische Vorgehensweise und Modellbildung

Den Ausgangspunkt für die Programmierung der SPS bildet der angestrebte Ablauf in der Maschine. Liegt eine Dokumentation dessen aus der Mechanik vor, ist dies, wie in Kapitel 2.1.3 aufgezeigt, in Form eines Gantt-Charts oder SFCs gegeben. Als weitere Eingangsinformationen werden die Signalliste und die Hardwarekonfiguration der Steuerung benötigt. Bevor die Programmierung der Steuerung erfolgen kann, muss der Automatisierer ein Verständnis für den Sollablauf der Maschine und dem Zusammenhang zwischen den Signalen und deren Wirkung auf das konstruierte System, aufgebaut haben. Dieser Zusammenhang kann in einer Achsliste dokumentiert sein. Dabei ist zu beachten, dass nur ca. 50% aller, im SPS-Programm verarbeiteten Signale direkten Einfluss auf die Mechanik besitzen. Die andere Hälfte bezieht sich auf den Status eines elektrischen Gerätes, wie den Betriebszustand eines Stellgliedes oder ähnliches. Somit muss der Programmierer, die übrigen Signale ebenfalls den entsprechenden Zuständen der Maschine bzw. Geräte zuordnen. Auf Basis dieses Verständnisses, strukturiert die Automatisierung ihren zu entwickelnden Steuerungscode. Hierzu nutzt der Entwickler POEs (im Weiteren auch Funktionsblöcke genannt). Diese enthalten Ansteuerungsprogramme für Module oder Geräte. Dies ist in Abbildung 3.3 auf Basis eines SFCs dargestellt. Zu erkennen ist, dass SPS-Programme die Werte von Eingangssignalen (auch internen Variablen oder Funktionen) verarbeiten, um in Abhängigkeit davon Zustandsänderungen des Systems durch Setzen von Ausgangssignalen zu bewirken. Die Werte, welche die Signale annehmen können, sind vom Signaltyp abhängig. Signale werden mittels Transitions- oder Übergangsbedingungen verschalten. Führen die Werte und die

definierten Bedingungen zu einer wahren Aussage, erfolgt der Übergang von einem Zustand zum Nächsten. In diesem wird das entsprechende Ausgangssignal gesetzt und somit eine Wirkung auf das mechanische Grundsystem über das Stell- und Wirkelement erzeugt. Neben Übergangsbedingungen ist die Definition von Verriegelungsbedingungen möglich, welche bei Eintreten einer „unwahren“ Aussage den Programmablauf stoppen oder Sicherheitsfunktionen anstoßen. Die Kommunikation zwischen POEs erfolgt über globale Variablen, welche wiederum miteinander über Transitions- oder Verriegelungsbedingungen verschalten werden können. Im Steuerungsprogramm wird mit aussagekräftigen Variablennamen gearbeitet. Dadurch ist direkt der Bezug zum mechatronischen System Maschine und dessen Zuständen erkennbar. Dies wird auch als symbolischer Signalname bezeichnet. Aufgrund der äußerst modularen Bauform sind die Ein- und Ausgänge der SPS von deren hardwaretechnischen Konfiguration abhängig. Deshalb müssen die symbolischen Signale des Programms den korrekten Ein- und Ausgängen zugeordnet werden. Dazu wird die entsprechende Modulkonfiguration, welche im Rahmen der Elektrokonstruktion definiert worden ist, benötigt.

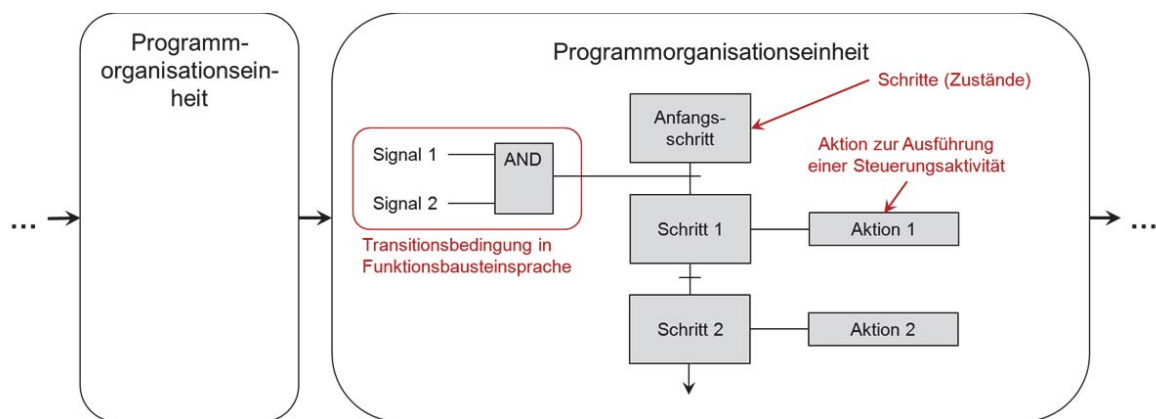


Abbildung 3.3: Elemente eines SPS-Programms

Die Symboltabelle gibt an, welche Steuerungssignale oder -variablen auf welchen Adressen eingelesen bzw. ausgegeben werden.

3.4 Analyse und Gegenüberstellung der Vorgehensweisen und Modelle

In Abbildung 3.4 werden die Modellierungselemente der verschiedenen Disziplinen aus den Kapiteln 3.1, 3.2 und 3.3 zusammenfassend dargestellt. Offensichtlich besitzen bestimmte Informationen Relevanz in mehr als einer Disziplin und damit auch in mehr als einer Software-

unterstützung. Ein Beispiel hierfür sind Anschlüsse von Komponenten, welche sowohl in der Elektrik als auch der Mechanik berücksichtigt werden müssen.

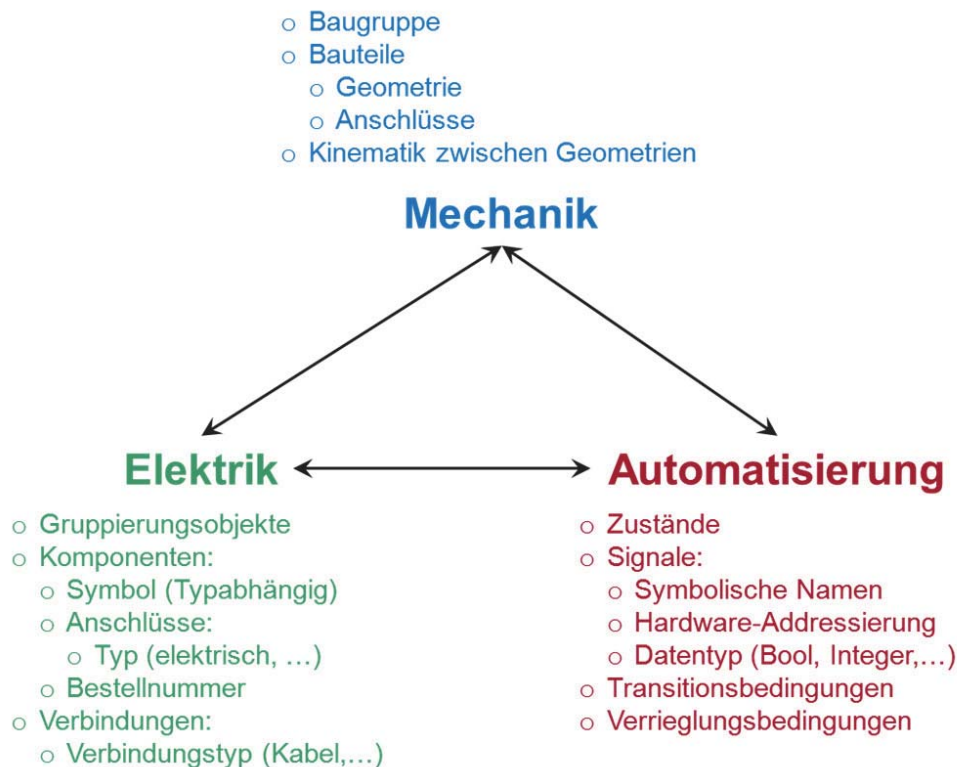


Abbildung 3.4: Modellierungselemente der verschiedenen Disziplinen

Andere Informationen hingegen sind nur disziplinspezifisch relevant, wie die Geometrie eines Bauteils in der Mechanik. Diese Informationen können allerdings Kontextinformationen zur Förderung des Verständnisses in anderen Disziplinen darstellen, werden dort aber nicht für die unmittelbare Entwicklungstätigkeit benötigt.

3.4.1 Statischer Vergleich der Modelle

Im Rahmen der statischen Betrachtung wird untersucht in welcher Relation, die in den Kapiteln 3.1, 3.2 und 3.3 aufgezeigten Modellierungen zueinander stehen, ohne die zeitliche Entstehung der Informationen zu berücksichtigen. Die mechanische Konstruktion erstellt im Wesentlichen drei Modelle, eine funktionale Struktur, ein Wirk- oder Lösungsprinzip und den mechanischen Zusammenbau. Allerdings liegen derzeit die ersten beiden Modelle häufig nicht explizit vor. Das kinematische Maschinenkonzept kann als Teil des Wirk- oder Lösungsprinzips betrachtet werden. Ein Vergleich der Modelle zeigt, dass die Abstraktion derer in der genannten Reihenfolge abnimmt.

Eine genauere Betrachtung ergibt, dass der mechanische Zusammenbau mit seinen geometrischen Beschreibungen Informationen enthält, welche nicht weiter verfeinert werden. Selbst wenn nur eine grobe Geometrie angegeben ist, wäre prinzipiell eine Ableitung der Fertigungszeichnung möglich. Dies entspricht der höchsten Detaillierungsstufe, welche im Rahmen einer Sondermaschinenentwicklung sinnvoll ist. Außerdem kann das Bauteil oder der Baugruppenknoten eine Bestellnummer repräsentieren, was das Modell vollständig definiert. Beide Varianten ermöglichen dadurch eine physikalische Bereitstellung des Bauteils bzw. der Baugruppe, entweder durch Herstellung oder Beschaffung.

Das Wirk- oder Lösungsprinzip verdeutlicht wodurch eine Funktion umgesetzt werden soll. Eine Skizze dieses Prinzips kann in Analogie zu *Kleiner* [2013] als logische Ebene interpretiert werden. Übergeordnet derer findet sich die abstrakte Funktionsbeschreibung wieder. Die Einordnung der mechanischen Modelle auf die genannten Ebenen führt auf Abbildung 3.5.

Die elektrotechnische Konstruktion durchläuft ebenfalls mindestens zwei Ebenen. Die Interpretation des Funktionsbegriffs in der Elektrik, als Prozessschritt, führt auf ein niedrigeres Abstraktionsniveau als bei der funktionalen Problembeschreibung der Mechanik. Diese Begriffsinterpretation liegt folglich an der Grenze zwischen funktionaler und logischer Beschreibung. Module können wie klassifizierte Bibliotheksobjekte als Abstraktion der physikalischen Elemente angesehen werden, daher deren Einordnung auf der logischen Ebene. Durch Angabe einer Bestellnummer erfolgt der Übergang zur physikalischen Beschreibung, da hierdurch ein Echtgerät vollständig spezifiziert ist und bereitgestellt werden kann. Entsprechend ist die Produktstruktur der Elektrik zu werten.

Die Automatisierung erhält üblicherweise als Eingangsinformationen Gantt-Charts oder SFC. Mit Gantt-Charts lassen sich keine ablauffähigen SPS-Programme beschreiben, da Signale und Transitionsbedingungen nicht berücksichtigt sind. Selbiges gilt für andere Beschreibungssprachen mit dieser Charakteristik. Ein Einsatz von SFCs mit Signal- und Bedingungsdefinitionen entspricht deshalb der physikalischen Ebene, ebenso wie die anderen Sprachen der *DIN EN 61131 Blatt 3* [2003-12]. Gutabläufe sind in diesem Kontext als logische Beschreibung zu werten, da häufig Signale und Bedingungen fehlen. Diese Zusammenhänge münden in folgender Aussage:

- Die Modelle der verschiedenen Disziplinen liegen auf unterschiedlichen Abstraktionsebenen.

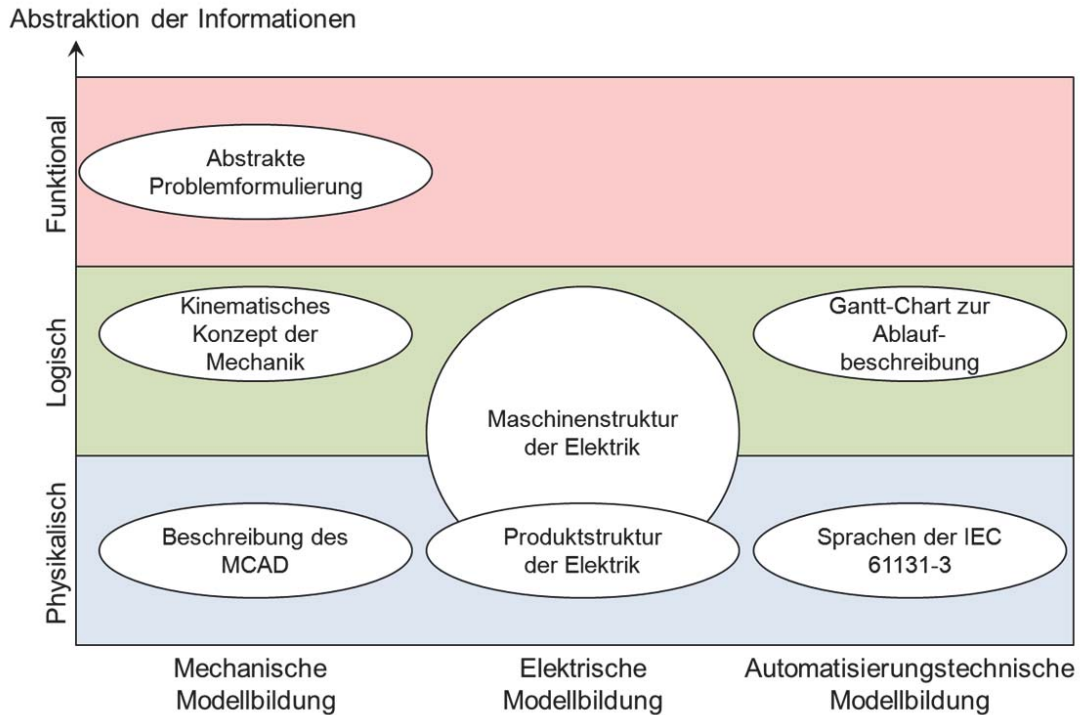


Abbildung 3.5: Einordnung der disziplinspezifischen Modelle auf verschiedene Abstraktionsniveaus

Wie bereits eingangs des Kapitels dargestellt, sind verschiedene Informationen interdisziplinär relevant. Dies betrifft im Besonderen Mess- und Wirkelemente, welche sowohl in der Mechanik als auch der Elektrik modelliert werden. Ein Vergleich beider Modelle eines Wirkelementes zeigt Abbildung 3.6.

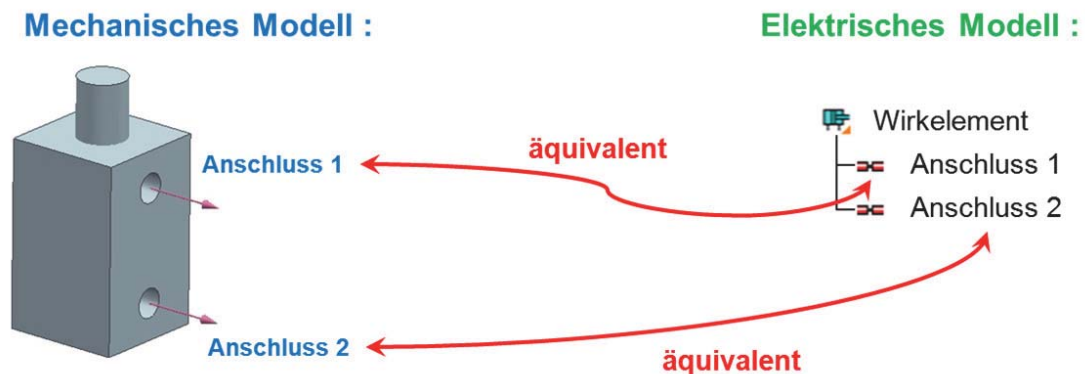


Abbildung 3.6: Vergleich der Repräsentationsformen von Anschlüssen in der Mechanik und der Elektrik

Darin ist zu erkennen, dass auch die Anschlüsse in beiden Disziplinen bzw. Sichtweisen modelliert sind. Jeder Anschluss der elektrischen Sichtweise besitzt in der Mechanik ein äquivalentes Pendant. Allerdings sind Anschlüsse im Rahmen der Elektrik explizit modelliert, während die Mechanik diese nur implizit über Vektoren, Bohrungen oder Gewinde darstellt. Dies führt zu folgender Feststellung:

- Manche interdisziplinär relevanten Informationen sind nur implizit in Modellen vorhanden. Eine explizite Modellierung ist in den herkömmlichen Softwaresystemen nicht möglich.

Neben Mess- und Wirkelementen ist auf den ersten Blick auch eine Zuordnung von Modulen zu Baugruppen möglich, wie sie in Abbildung 3.7 dargestellt ist. Ein Vergleich der hierarchischen Positionen fördert allerdings Unterschiede zu Tage. Die Mechanik ordnet im Rahmen der Baugruppenstruktur die Objekte nach dem Gesichtspunkt der Bestellnummer bzw. Montage. Die Elektrik strukturiert hingegen nach einer wirkungstechnischen Sichtweise. Als Beispiel sei hier die Position der Sensoren unterhalb des Wirkelementes in der Elektrik betrachtet. Diese geben an, welche Elemente für die Positionserkennung zuständig sind. In der Mechanik sind diese parallel dazu angeordnet, da das Wirkelement ohne Sensoren geliefert wird. Analoges gilt für das Stellelement.

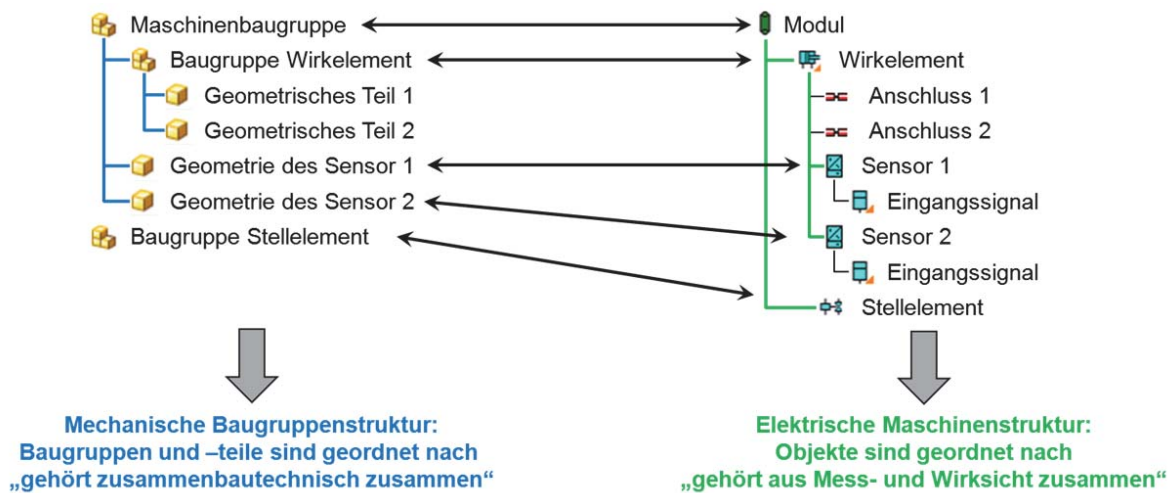


Abbildung 3.7: Unterschiedliche Bedeutungen der hierarchischen Ordnung in den disziplinspezifischen Modellen

Daraus kann abgeleitet werden, dass den aufgebauten Hierarchien unterschiedliche Strukturierungsprinzipien zugrunde liegen, was in folgender Aussage mündet:

- Die hierarchischen Projektstrukturen der Disziplinen sind semantisch unterschiedlich.

Die Automatisierungstechnik besitzt keine Möglichkeit Komponenten zu modellieren. Um dennoch das Steuerungsprogramm zu strukturieren werden POEs erstellt. Solche Funktionsblöcke steuern aus Sicht des SPS-Programmierers beispielsweise Aktoren. Somit enthält ein Aktor im Rahmen der Automatisierung nur die Verbindung zwischen der im Programmcode abgebildeten Signalschnittstelle und den durch dessen Wirkung hervorgerufenen Zustände, dargestellt durch die rote Linie zwischen Ausgangssignal und Wirkelement. Im Gegensatz dazu bezeichnet ein Aktor in der Mechanik nur das Wirkelement. Die Elektrik betrachtet die gesamte Elementkette zwischen Signalschnittstelle und Wirkelement. Diese unterschiedlichen Begriffsauffassungen sind in Abbildung 3.8 dargestellt. In dieser Arbeit ist ein Aktor im Sinne der Automatisierung definiert. Es wird festgehalten, dass

- gleiche Begriffe in den Sichtweisen unterschiedliche Semantik besitzen.

Die Verbindung zwischen Ausgangssignal und Wirkelement, in Form der roten gestrichelten Linie, wird durch das Erstellen des Schaltplanes in der Elektrik detailliert. Somit kann diese Verbindung auch als Aufgabenstellung der Elektrik angesehen werden, indem der Elektrokonstrukteur verschiedene Objekte und Verbindungen zwischen diesen plant. Dies führt auf folgende Aussage:

- Ein Objekt/Relation/Attribut in einer Sichtweise kann durch mehrere Objekte und Relationen in einer anderen Sichtweise repräsentiert werden.

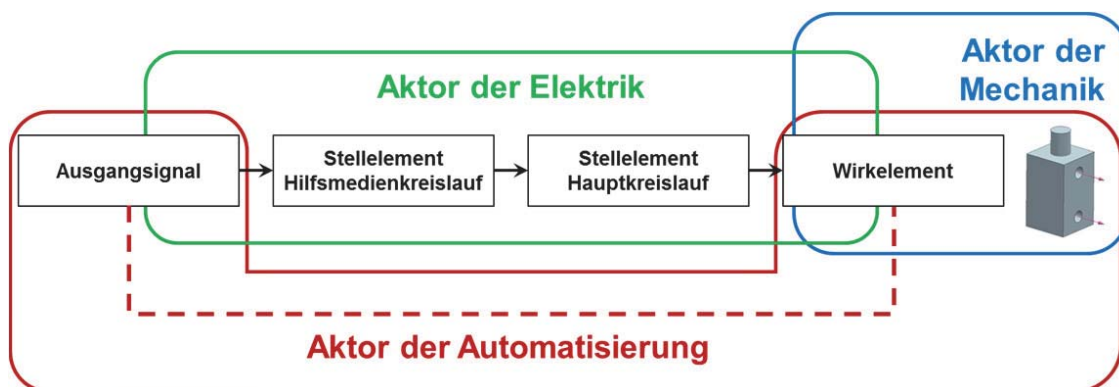


Abbildung 3.8: Disziplinspezifische Interpretationen des Begriffs Aktor

Die Vorgehensweise, hinsichtlich einer guten Softwarestrukturierung, POEs für Komponententypen auf Basis derer Signalschnittstellen und Zustände zu modellieren führt dazu, dass Komponenten neben der elektrischen und mechanischen auch eine automatisierungstechnische Repräsentanz besitzen. Ein Vergleich einer POE, des zugehörigen elektrischen und mechanischen Modells,

welche im Projektkontext als äquivalent betrachtet werden können, zeigt Abbildung 3.9. Die Gültigkeit der POE ist dabei von ihrem zugrundeliegenden Strukturierungsprinzip abhängig.

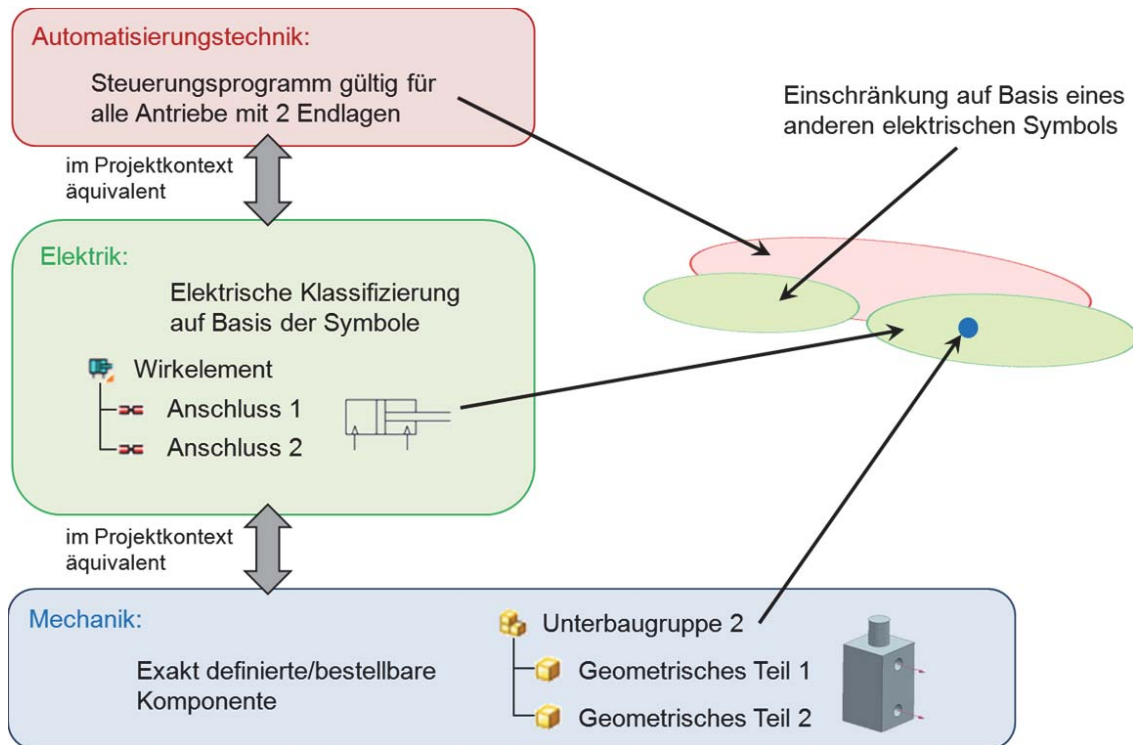


Abbildung 3.9: Gültigkeitsbereiche disziplinspezifischer Modelle und ihre Äquivalenz im Projektkontext

Als Beispiel sei ein Ventil, welches einen Differentialzylinder ansteuert genannt. Die Automatisierung würde hierfür einen Funktionsbaustein auf Basis der Signalschnittstelle des Ventils und den beiden Endlagen des Zylinders entwerfen. Wird dasselbe Ventil mit einem Gleichlaufzylinder verbunden, ist der Funktionsbaustein immer noch gültig, da auch dieses Wirkelement nur zwei Endlagen besitzt und dieselbe Signalschnittstelle des Stellelements Verwendung findet. Dies führt dazu, dass

- Änderungen in Sichtweisen, auch wenn diese auf physikalischer Ebene geschehen und Einfluss auf interdisziplinäre Informationen besitzen, im Projektkontext keine Auswirkungen auf andere disziplinspezifische Modelle besitzen müssen. Die Geltungsbereiche von disziplinspezifischen Modellen sind unterschiedlich, was in den zugrunde liegenden Strukturierungsprinzipien begründet ist.

Eine Umkehrung des Beispiels ist ebenfalls denkbar. Dieselbe Ventil-Zylinder-Kombination kann natürlich mit unterschiedlichen Funktionsbausteinen betrieben werden. In Abhängigkeit des mechanischen Grundsystems können sich beispielsweise die Abläufe innerhalb des Funktionsbausteines unterscheiden. Daraus ist ersichtlich, dass

- eine einzige Klassifizierungs- und Vererbungshierarchie von Komponenten, beginnend beim automatisierungstechnischen, über das elektrische hin zum mechanischen Modell, aufgrund des Wechsels der zugrunde liegenden Strukturierungsprinzipien nicht allgemeingültig möglich ist.

Zusammenfassend ergibt die Analyse, dass die Modelle auf unterschiedlichen Abstraktionsebenen zu finden sind. Weiterhin besitzen diese, unterschiedliche semantische Modellierungen, die mehr oder minder explizit sind und verschiedene Geltungsbereiche, selbst auf einer gemeinsamen Abstraktionsebene.

3.4.2 Betrachtung der Wiederverwendung in mechatronischem Kontext über Projekte hinweg

Eine Wiederverwendung von Planungen über Projekte hinweg führt zur Reduktion von Entwicklungsaufwänden. Hierbei müssen Module gefunden werden, welche in verschiedenen Projekten genutzt werden können. Je größer diese gefasst werden, desto höher ist die Zeit- und Aufwandsersparnis (vgl. Kapitel 2.5.2.). Prinzipiell bietet sich bei Sondermaschinen die Wiederverwendung von Modulen an, welche bestimmte Verfahren automatisieren oder kinematische Konzepte umsetzen. Der Begriff Modul interpretiert üblicherweise eine abgeschlossene Kapselung und Bereitstellung von Bausteinen. Die in Kapitel 2 eingeführten Ansätze der Modularisierung versuchen mechatronische Komponenten zu identifizieren und zu definieren, welche solche Kapselungen aus allen beteiligten mechatronisch relevanten Sichtweisen bilden. Dabei betrachtet die Automatisierung Module in Form von Funktionsblöcken bis zur Signalschnittstelle. Weiterhin besitzt ein Steuerungsprogramm die Aufgabe mehrere Wirkungsweisen der Maschine so miteinander zu verknüpfen, dass der Gesamtablauf gewährleistet ist. Die Elektrik verbindet die Signalschnittstellen der Steuerung mit den in der Mechanik definierten Wirkelementen. Somit sind hier alle Signale und deren Verdrahtung bis hin zum Wirkelement bekannt. In der Mechanik werden Module als geometrisch zusammenhängende Baugruppen definiert. Dies ist entsprechend in Abbildung 3.10 visualisiert. Eine Modularisierung aus automatisierungstechnischer und elektrischer Sicht geht mit dem Mapping einer POE auf ein elektrisches Modul einher, da beide nach dem

Prinzip der Wirkungsweise strukturiert sind. Gleichzeitig ist zu erkennen, dass dadurch die Signalschnittstelle und das Wirkelement die Systemgrenze bilden. Eine strikte Forderung nach einer mechanischen „Kapselung“ würde dazu führen, dass das Stellelement ebenfalls der aufgezeigten Maschinenbaugruppe zugeordnet werden muss. Gerade im Bereich der Pneumatik werden Ventilinseln zentral zusammengefasst, was einer solchen mechanischen Modulbildung widerspricht. Ein analoger Vergleich der Mechanik und Elektrik hinsichtlich der Steuerung führt auf dasselbe Ergebnis. Das Hinzunehmen des Stellelementes in die mechanische Baugruppe führt dazu, dass diese im Raum nicht positionierte Bauteile enthält. Dies würde der üblichen Interpretation des Begriffes Modul als Kapselung in der Mechanik entgegenstehen.

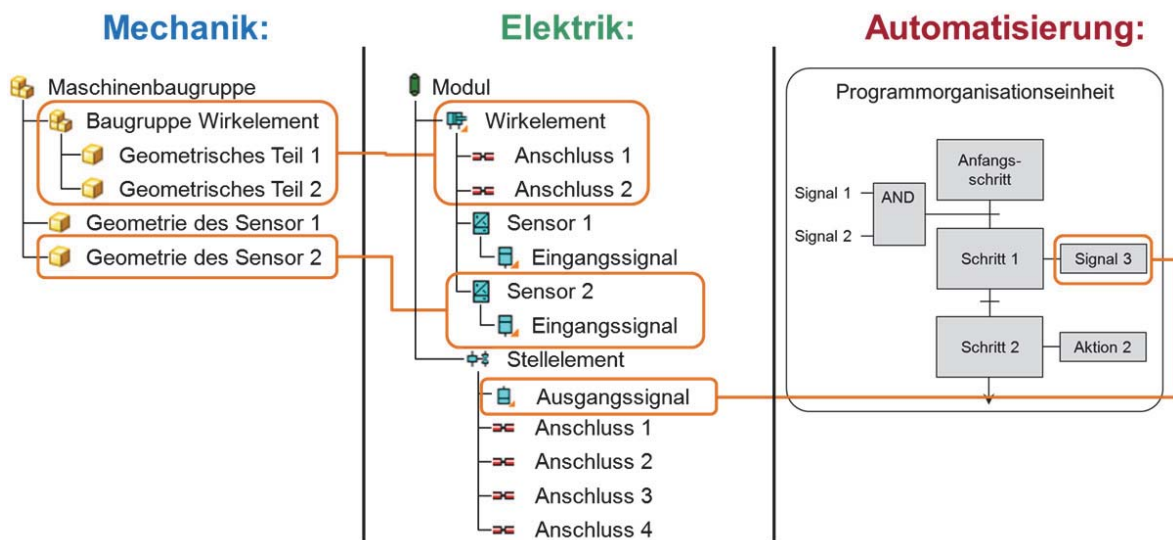


Abbildung 3.10: Systemgrenzen der disziplinspezifischen Sichtweisen

Ursächlich sind die unterschiedlichen Systemgrenzen der disziplinspezifischen Sichtweisen und deren Strukturierungsprinzipien. Modularisierung auf Basis der Automatisierung führt auf große kinematisch zusammenhängende Gruppierungen mit definierten Zuständen und Signalschnittstellen. Dies verkörpert gleichzeitig das Entwicklungsziel der Steuerungsprogrammierung. Der Inhalt eines solchen Moduls ist in Abbildung 3.11 grün umrandet. Ein Versuch der Kapselung zwischen Mechanik und Elektrik führt am Beispiel aus Abbildung 3.10 zur Zusammenführung der Maschinenbaugruppe und des elektrischen Moduls, allerdings ohne Stellelement. Dadurch würde sich in der Automatisierung bei gleicher Systemgrenze ein unvollständiges Steuerungsprogramm ergeben, da spezifische Zustände des Stellelementes (vgl. Kapitel 3.3) und dessen Signalschnittstelle nicht berücksichtigt sind. Derselbe Effekt des unvollständigen Programms entsteht

beim Versuch einer mechatronischen Zusammenfassung auf Komponentenebene (vgl. Wirkelement in Abbildung 3.10 und Modul 3 in Abbildung 3.11). Ursächlich hierfür ist, dass je nach Aufgabenstellung mit derselben Kinematik, desselben Wirk- und Stellelements unterschiedliche Abfolgen der Zustände realisiert werden können. Wird die Verschaltung der Elektrik als Basis für eine Modularisierung genutzt, sind alle Verhaltenszustände des Stellelements und die notwendigen Ausgangssignale aus Sicht der Automatisierung definiert. Ein vollständiges Programm kann allerdings nur in Zusammenhang mit der zu bewegenden Kinematik und der Automatisierungsaufgabe erfolgen (vgl. Modul 2 in Abbildung 3.11).

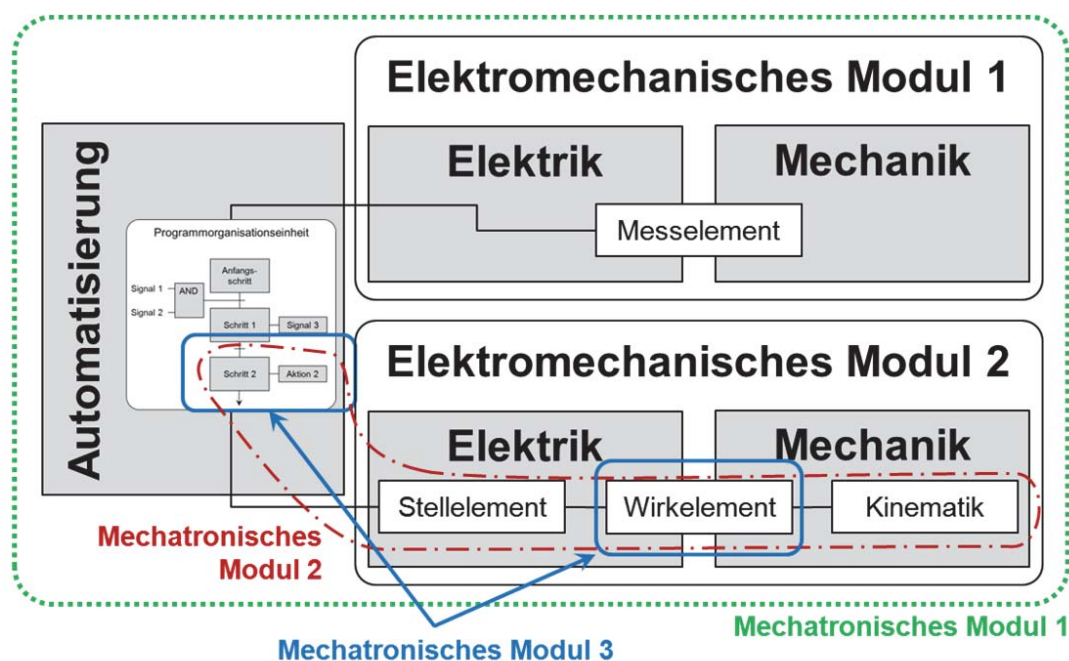


Abbildung 3.11: Möglichkeiten zur mechatronischen Modularisierung

Um dennoch eine Modularisierung im Sondermaschinenbau durchzuführen sind prinzipiell zwei Lösungsmöglichkeiten ableitbar. Die erste Möglichkeit führt zu einer Dezentralisierung der Steuerung auf Modul- bzw. Komponentenebene um auch geometrisch und elektrisch abgeschlossene Einheiten zu erreichen. Die andere Lösung sieht die Abkehr einer Kapselung in mechatronischem Kontext vor. Dadurch müsste es ermöglicht werden, nicht zusammenhängende Modelle der einzelnen Disziplinen im Rahmen der Entwicklung als Module zu definieren. Ursächlich hierfür sind die unterschiedlichen Systemgrenzen und Modellierungsprinzipien der Disziplinen. Bei Messelementen ist diese Problematik oft nicht direkt gegeben, da die Systemgrenzen der mechanischen Bauform und der Signalschnittstelle übereinander liegen.

3.4.3 Betrachtung des Entwicklungsablaufes

Ausgehend von den Ergebnissen des statischen Vergleichs und deren Einordnung der Modelle auf unterschiedlichen Abstraktionsebenen wird im Folgenden untersucht, wann Informationen entstehen und welchen weiteren Einfluss diese besitzen. Die Betrachtung des derzeitigen sequenziellen Entwicklungsprozesses zeigt, dass die beginnende Mechanik sehr schnell auf der physikalischen Ebene ankommt. Ein qualitativer Verlauf dessen ist in Abbildung 3.12 dargestellt.

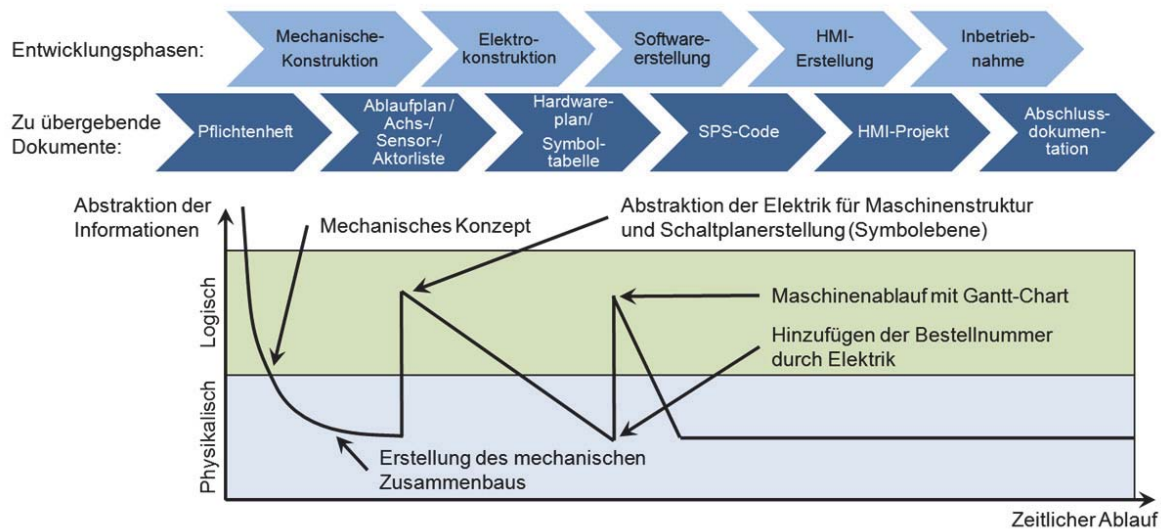


Abbildung 3.12: Ablauf des sequenziellen Entwicklungsprozesses in Abhängigkeit der Modellabstraktion

Da gerade die funktionale Phase häufig nicht explizit umgesetzt wird, fällt die Modellierung sehr schnell auf die logische Ebene ab. Werden mechanische Konzepte (beispielsweise kinematische Maschinendefinition) bereits in einem MCAD-System dargestellt, ist dies, wie bereits in Kapitel 3.4.1 verdeutlicht, als Übergang zwischen logischer und physikalischer Ebene zu betrachten. Gleichzeitig entscheidet sich hier ob bereits existierende Lösungen wiederverwendet werden können oder nicht. Wie bereits eingangs von Kapitel 3.4.2 geschildert, eignen sich Module von kinematischen Konzepten oder Verfahren am besten um erneut genutzt zu werden. Die mechanische Detaillierung im MCAD-System erfolgt vollständig auf physikalischer Ebene. Der Übergang zwischen mechanischer und elektrischer Phase ist geprägt von impliziten Informationen auf Basis des mechanischen Modells, unstrukturierten Daten und erzwingt dadurch eine intellektuelle Modelltransformation des Elektrokonstruktors. Das Ergebnis bildet die Projekt-

strukturierung im ECAD-System, was einer erneuten Abstraktion des Gesamtsystems gleichkommt. Im weiteren Verlauf der Elektrokonstruktion erfolgt eine Detaillierung, bis alle Komponenten wieder vollständig beschrieben und somit auf der physikalischen Ebene definiert sind. Auch der nächste Übergang zwischen der Elektrik und der Automatisierung weist dieselben Charakteristika auf als zuvor. Dadurch erfolgt wiederum eine Abstraktion des Systems und eine erneute Ausgestaltung zu einem lauffähigen Steuerungscode.

Somit ist ausgehend von einer Anforderungsdefinition, wie bereits in Kapitel 2.7.1 definiert, eine abstrakte Problembeschreibung zu berücksichtigen. Eine Ebene tiefer erfolgt die Definition von Lösungsprinzipien. Die Leistungsfähigkeit der entwickelten Maschine und das mechatronische Optimum hängen entscheidend von der Wahl dieses Prinzips ab (vgl. Kapitel 2.2.3 und 2.3.2). Im Sondermaschinenbau sind kinematische Konzepte gleichzusetzen mit Lösungsprinzipien. Den Einfluss eines solchen Konzeptes auf die beteiligten Disziplinen veranschaulicht Abbildung 3.13.

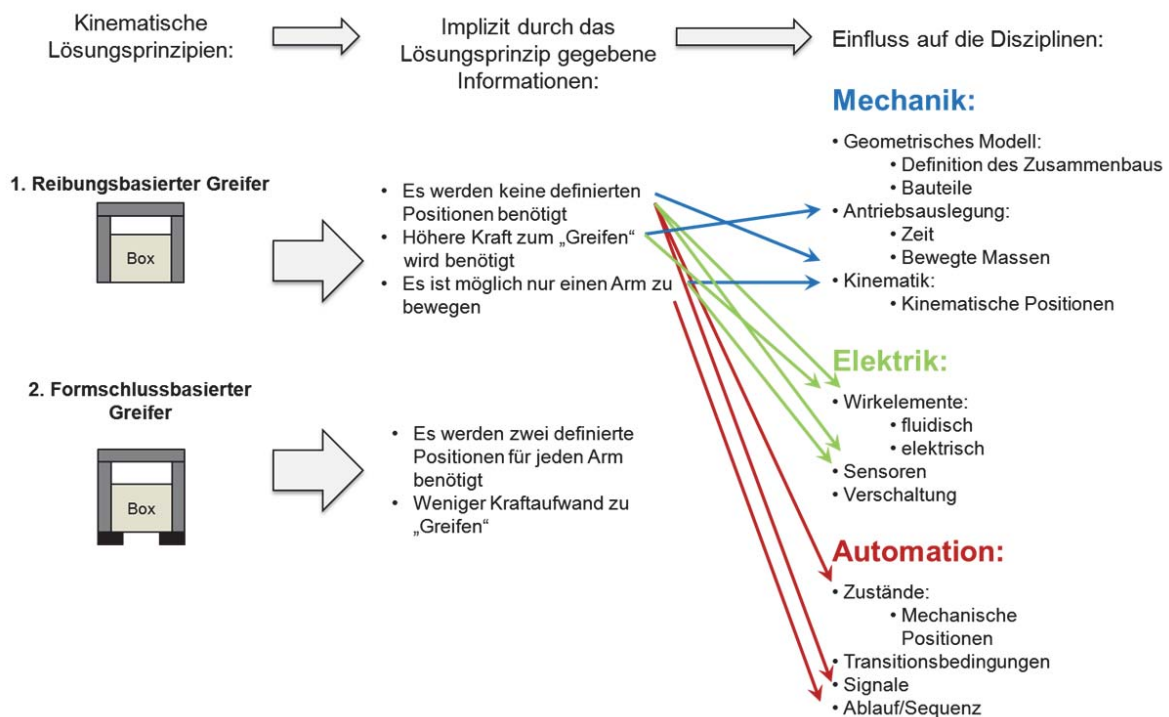


Abbildung 3.13: Einfluss des kinematischen Konzeptes auf disziplinspezifische Planungen

Diese kinematischen Lösungsprinzipien beinhalten bereits implizit Informationen, welche direkte Auswirkungen auf alle Disziplinen besitzen. So wird beispielsweise, je nach Wirkprinzip des Greifers, eine unterschiedliche Anzahl definiert anzufahrender Positionen benötigt. Diese benötigen

wiederum eine sensorische Überprüfung aus steuerungstechnischer Sicht und müssen somit auch in der elektrischen Verschaltung berücksichtigt sein. Es ist erkennbar, wie wichtig der Einbezug aller Disziplinen bereits in dieser Phase ist, um ein mechatronisches Optimum zu erreichen.

Ausgehend von einer gemeinsamen Phase des Systementwurfs, wie sie auch die *VDI-Richtlinie 2206* vorsieht, sollen die Disziplinen in der Lage sein parallel weiterzuarbeiten. Hierzu muss eine Beschreibung vorliegen, welche genau dies ermöglicht. Die Aufgabe des Elektrokonstruktors ist es, die Sensoren und mechanischen Stellglieder so zu verschalten, dass der Automatisierer die Eingänge auswerten und Ausgänge setzen bzw. rücksetzen kann, wodurch eine Wirkung auf das mechanische System erzielt wird. Weiterhin müssen Wirkelemente an Versorgungsquellen angeschlossen werden. Aus elektrischer Sicht sind folglich definierte Sensoren und Wirkelemente, sowie die zur Verfügung stehenden Energiequellen bereitzustellen. Die Mechanik baut auf das kinematische Konzept auf. Um allerdings ein Antriebskonzept hierfür zu definieren, wie es die Definition der mechanischen Stellglieder tut, müssen die bewegten Massen abschätzbar sein. Zur Auslegung der Antriebe im nächsten Schritt ist es erforderlich die Wege zwischen den anzufahrenden Positionen sowie die maximalen Bewegungszeiten zu kennen. Außerdem sind als Kontextinformation die gleichzeitig auszuführenden Bewegungen wichtig, da diese Auswirkungen auf die Verschaltung der Komponenten besitzen. Die Automatisierung kann mit der Programmierung beginnen, wenn der Sollablauf der Bewegungen feststeht, also die Reihenfolge der anzufahrenden kinematischen Positionen. Weiterhin sind Übergangsbedingungen interessant und wie die Signalschnittstelle der Sensoren und Aktoren definiert ist. Zusammenfassend zeigt Tabelle 3.1 die benötigten Informationen der einzelnen Disziplinen um ihrer Aufgabe im Rahmen einer parallelen Ausdetaillierung nachzukommen.

Disziplin	Benötigte Informationen
Mechanik	<ul style="list-style-type: none"> ○ Grobe Geometriebeschreibungen: <ul style="list-style-type: none"> • Erste Zusammenbauten • Geometrische Teile ○ Kinematisches Konzept mit Positionen ○ Mechanische Stellglieder/Wirkelemente: <ul style="list-style-type: none"> • Typ des Wirkelements • Maximale Bewegungszeiten • Bewegte Massen
Elektrik	<ul style="list-style-type: none"> ○ Mechanische Stellglieder/Wirkelemente ○ Messelemente ○ Kontextinformationen: <ul style="list-style-type: none"> • Abläufe • Signalschnittstellen
Automatisierung	<ul style="list-style-type: none"> ○ Zustände der Maschine <ul style="list-style-type: none"> • Kinematische Positionen • Andere Zustände • Abfolge der Zustände ○ Signale der Zustände ○ Übergangsbedingungen ○ Kontextinformationen: <ul style="list-style-type: none"> • Aktoren und Sensoren

Tabelle 3.1: Benötigte Informationen zur Aufgabenerfüllung in den einzelnen Disziplinen

Im Rahmen einer simultanen Entwicklung in den verschiedenen Disziplinen sind weitere Abhängigkeiten derer voneinander zu beachten. In Kapitel 1.2 ist bereits erläutert, dass Kabellängen bestimmte Grenzlängen nicht überschreiten dürfen. Die Überprüfung dessen obliegt der Elektrik. Das Modell zur Ermittlung der Längen ist allerdings in Besitz der Mechanik. Ist die Länge beispielsweise zu groß und eine Verkürzung im Rahmen der mechanischen Konstruktion nicht möglich, müssen Komponenten zur Verstärkung berücksichtigt werden. Der entsprechende Ablauf dieses Zusammenhangs ist in Abbildung 3.14 dargestellt.

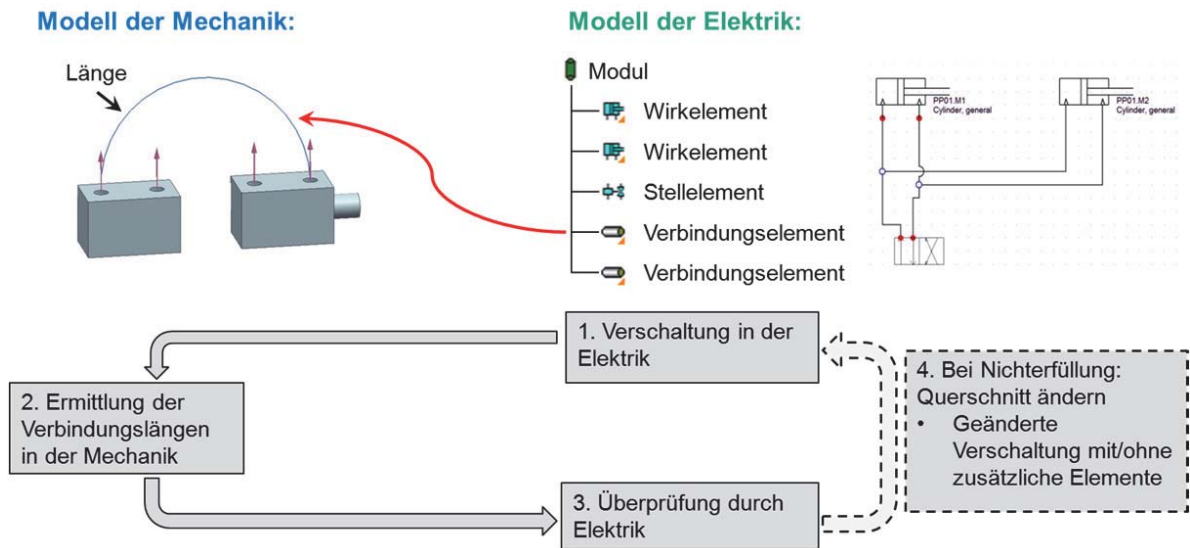


Abbildung 3.14: Einfluss der Schlauchlänge des mechanischen Modells auf die Verschaltung

Es ist ersichtlich, dass

- einzelne Attribute einer Disziplin Auswirkungen auf die Planungen einer anderen auch während einer Ausdetaillierungsphase besitzen.

Diese Änderungen an interdisziplinär relevanten Informationen müssen möglichst schnell zwischen den Domänen kommuniziert und bereitgestellt werden. Auf Basis der disziplinspezifischen Modelle werden diese Informationen als Abhängigkeiten in Abbildung 3.15 dargestellt. Benötigt beispielsweise die Automatisierung ein Signal über eine Position, weil diese noch nicht sensorisch erfasst ist, kann dies als „Signalanforderung“ in der Elektrik verstanden werden. Erfüllt wird diese durch Instanziierung eines Sensors zur Positionsdetektion, was gleichzeitig dem Hinzufügen einer Komponente entspricht. Dies wiederum ist in der Mechanik zu berücksichtigen indem der entsprechende Sensor eingebaut wird. Daraus ergibt sich, dass Informationen nicht nur zwischen zwei Disziplinen fließen und der Vorgang abgeschlossen ist, sondern dass diese durch Reaktion darauf weitere Sichtweisen tangieren können. Dieses Beispiel eines hinzuzufügenden Signals zeigt aber auch auf, dass der mechanische Konstrukteur eine Information zur Detektion eines Zustandes erhalten muss. Somit ist der Informationsfluss in beide Richtungen im Rahmen unterschiedlicher Kontexte gegeben. Schlussendlich ist dieses Informationsdreieck immer in Balance zu halten, so dass keine offenen Forderungen bestehen.

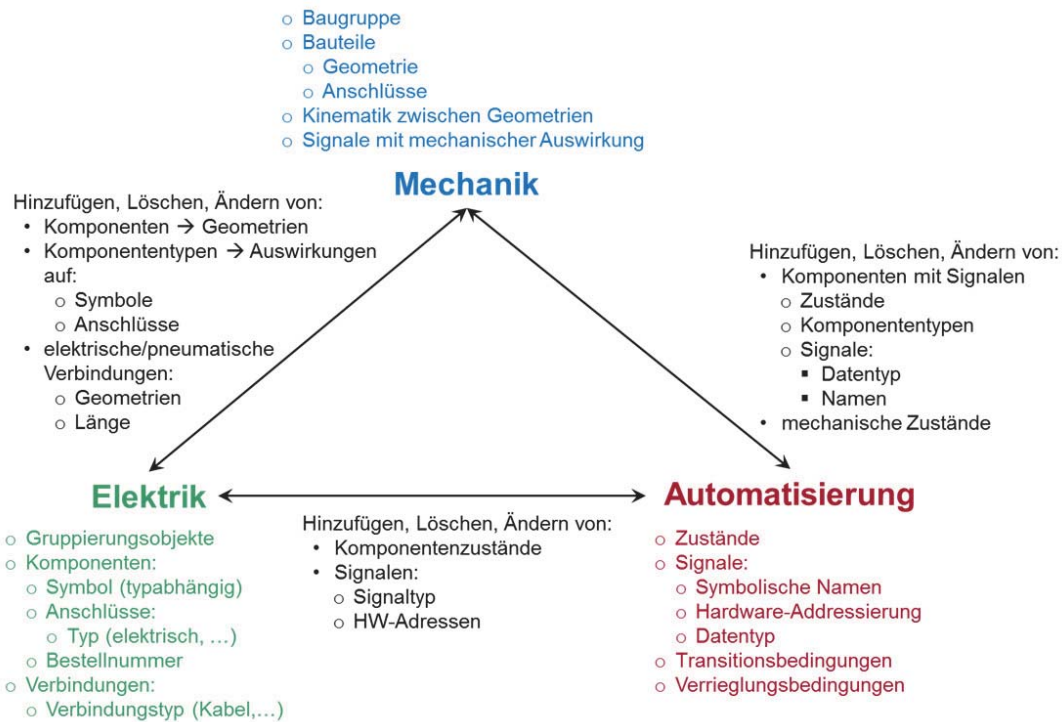


Abbildung 3.15: Abhängigkeiten der Disziplinen während einer parallelen Detaillierung

Aus diesen interdisziplinär relevanten Informationen lässt sich ableiten, dass sowohl das MCAD- als auch das ECAD-System Komponenten für die Bereitstellung der Bestellstücklisten modellieren. Zumindest Wirk- und Messelemente werden in beiden Systemen berücksichtigt. Wie bereits in Kapitel 3.1 beschrieben, werden häufig nicht alle Details in der mechanischen Beschreibung umgesetzt. Dadurch müssen Komponenten bestellt werden, welche keine Repräsentanz in der Mechanik besitzen. Diesen Zusammenhang zeigt Abbildung 3.16 auf.

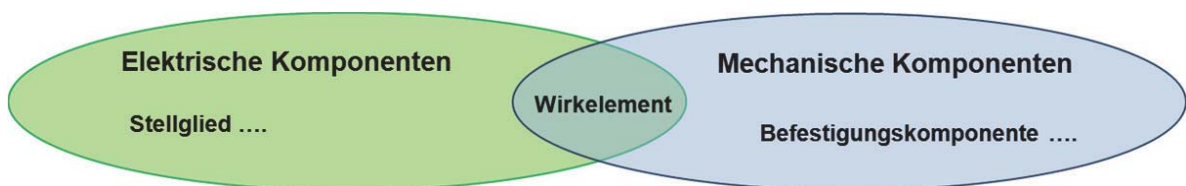


Abbildung 3.16: Modellierung von Komponenten der Stückliste in verschiedenen Disziplinen

Um alle Komponenten beim Aufbau und der Inbetriebnahme bereitliegend zu haben, ist eine Zusammenführung dieser aus beiden Sichten notwendig. Dadurch muss verhindert werden, dass Komponenten doppelt oder überhaupt nicht bereitgestellt sind. Neben dieser mechatronisch bereinigten Stückliste müssen am Ende des Entwicklungsprozesses Zeichnungen oder geometrische

Beschreibungen von Bauteilen zur Ableitung von NC-Programmen und ihrer Fertigung vorhanden sein. Schaltpläne zur Verkabelung, -schlauchung und -rohrung, sowie das Steuerungsprogramm werden ebenfalls benötigt.

3.5 Methodenkonzept

Die derzeitige Situation im Rahmen einer sequenziellen Maschinenentwicklung zeigt auf, dass jede Disziplin zum Systemverständnis ihre konkreten Planungen zur Detaillierung auf der eingeführten logischen Ebene beginnt (vgl. Abbildung 3.12). Gleichzeitig wird hier das Maschinenkonzept definiert, was als Ausgangspunkt für die weitere Ausdetaillierung in den Disziplinen zu nutzen ist. Deshalb kann eine parallele Entwicklung unter minimierten Aufwänden nur dann stattfinden, wenn die Disziplinen die verschiedenen Abstraktionsebenen möglichst im Gleichschritt durchlaufen. Dadurch werden die notwendigen Abstraktionen der Elektrik und Automatisierung bereits gleichzeitig zum kinematischen Konzept in der Phase des Systementwurfs eingeführt. Basierend auf den Phasen der *VDI-Richtlinie 2206* ergibt sich daraus der Zielverlauf aus Abbildung 3.17.

Die Entwicklung beginnt mit der Anforderungsdefinition für die Maschine. Am Ende steht das abgestimmte Lastenheft. Ausgehend davon wird bereits in der Phase des Systementwurfs eine abstrakte Problembeschreibung auf Basis funktionaler Strukturen vorgenommen. Den Übergang zur Lösungs- und Wirkprinzipdefinition markiert Quality Gate 2. Gleichzeitig muss hier ein Wechsel von einer abstrakten funktionalen Beschreibung zu einer logischen Modellierung des Wirkprinzips erfolgen. Die gefundenen Funktionen sind ebenfalls mit dem Kunden abzustimmen und möglichst lösungsneutral zu gestalten. Ziel ist, alle Funktionen zu identifizieren, welche der Kunde wirklich benötigt.

Da sich bereits in der Phase der Lösungssuche entscheidet (vgl. Kapitel 3.4.2), ob eine Wiederverwendung stattfinden kann, muss dies entsprechend unterstützt werden. Die Beschreibung des Lösungsprinzips ist bereits in einer mechatronischer Weise durchzuführen, worin jede Disziplin gleichberechtigt ist. Weiterhin ist im Hinblick auf die gestellten Anforderungen eine unterstützende Simulation vorzusehen. Den Übergang vom Systementwurf zur parallelen Ausgestaltung markieren die in Tabelle 3.1 definierten Informationen.

Während der parallelen Entwicklung sind die Abhängigkeiten der Disziplinen aus Abbildung 3.15 zu beachten. Außerdem ist zu berücksichtigen, dass beispielsweise Bauteile bereits in dieser Phase in übergeordnete Baugruppen integriert werden müssen, was im Rahmen der *VDI-Richtlinie 2206* auch unter Systemintegration verstanden wird. Gleiches gilt für die Elektrik und SPS-

Programmierung. Daraus folgt, dass bei Quality Gate 4 bereits alle notwendigen Komponenten zum Aufbau und das Steuerungsprogramm feststehen.

Dadurch reduziert sich die Phase der Systemintegration zunächst auf die virtuelle Inbetriebnahme, während Bauteile gefertigt, bestellt und montiert werden. Dies ermöglicht fehlerminimiert Teile der Maschine nach und nach in Betrieb zu nehmen. Im Idealfall stellt Quality Gate 5 den Übergang zur Inbetriebnahme beim Kunden vor Ort unter optimalen Bedingungen dar.

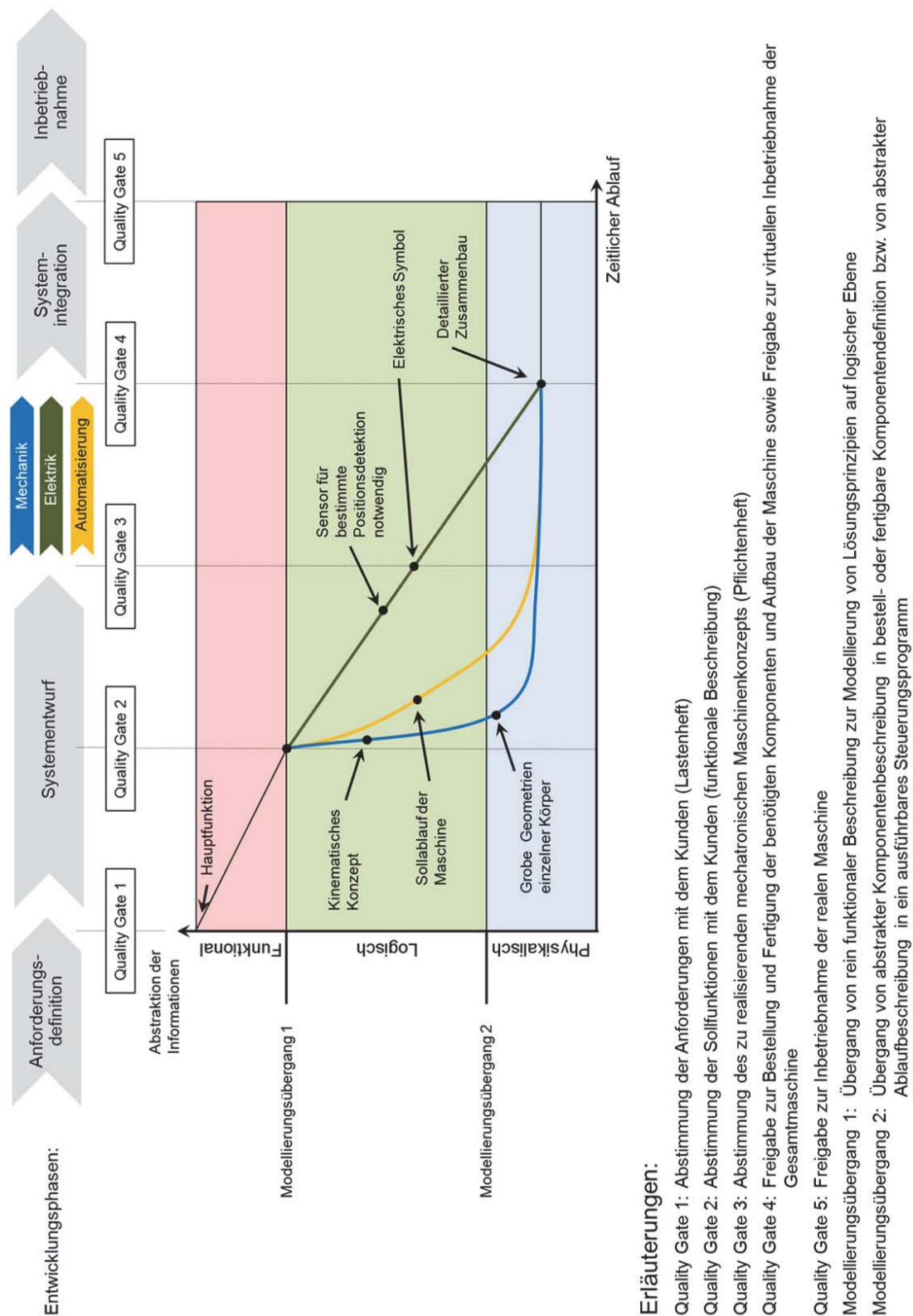


Abbildung 3.17: Verlauf der Modellierungen in den verschiedenen Disziplinen in Abhängigkeit der Abstraktion und der Entwicklungsphasen

4 Methode zur mechatronischen Entwicklung von Sondermaschinen

Aufbauend auf der Analyse der disziplinspezifischen Vorgehensweisen und dem daraus abgeleiteten Methodenkonzept, wird im Weiteren die Methode zur mechatronischen Entwicklung von Sondermaschinen vorgestellt.

4.1 Vorgehensweise zum systematischen Entwickeln von Sondermaschinen

Basis des Zielprozesses bilden die Phasen der *VDI-Richtlinie 2206* Anforderungsdefinition, Systementwurf, parallele Detaillierung und Systemintegration (vgl. Abbildung 4.1). Ausgehend von den definierten Anforderungen ist eine abstrakte Problemformulierung erforderlich, welche die Funktionen der zu entwickelnden Maschinen wiedergibt. Diese Funktionsdefinition ist nach den Gedanken des Systems Engineering bzw. *Pahl et al.* [2007] auszuführen, da hierdurch zunächst die nötige Abstraktion ermöglicht wird. Die Suche nach Lösungsprinzipien für die Funktionen entscheidet bereits über eine mögliche Wiederverwendung von bereits existierenden Modulen oder einer notwendigen neuen Lösung. Wird in dieser Phase eine Entscheidung für die Ausführung eines Rollenförderers getroffen, ist eine mögliche Nutzung eines bereits existierenden Förderbandes ausgeschlossen. Es ist ersichtlich, dass Lösungsprinzipien verständlich benannt werden müssen. Eine Zuordnung mehrerer Lösungen zu dieser Benennung kommt der geforderten Klassifizierung von existierenden Lösungen gleich. Die Einordnung der Prinzipien erfolgt auf der logischen Ebene. Falls eine Wiederverwendung möglich sein sollte, ist eine Generierung der benötigten disziplinspezifischen Modelle vorzusehen. Neue Module sind im Gesamtkontext der Maschine entsprechend so zu definieren, dass diese parallel in allen Disziplinen und zusammen mit den generierten Modellen ausdetailliert werden können. Spätestens nach der Detaillierung werden die Bestandteile neuer Module definiert und abgelegt. Dies muss in einer Weise geschehen, dass jede Disziplin die Systemgrenze ihrer Objekte selbst definieren kann. Nur dadurch ist es möglich größere Module zu definieren, welche nicht auf standardisierten Komponenten der elektrotechnischen Sichtweise oder Modulen mit eigener Steuerung basieren. Es sind folglich variable Systemgrenzen umzusetzen. Während der parallelen Entwicklungsphase in den verschiedenen Disziplinen sind die bereits im vorigen Kapitel ermittelten Abhängigkeiten zu berücksichtigen. Die Phase der Systemintegration dient dazu das Gesamtsystem zu überprüfen. Dazu wird eine virtuelle Inbetriebnahme umgesetzt. Weitere Tätigkeiten entfallen in dieser

Integrationsphase, da alle anderen Abhängigkeiten bereits während der parallelen Detaillierung berücksichtigt werden müssen. Zusammenfassend sind die genannten Phasen und deren Abfolge in Abbildung 4.1 dargestellt.

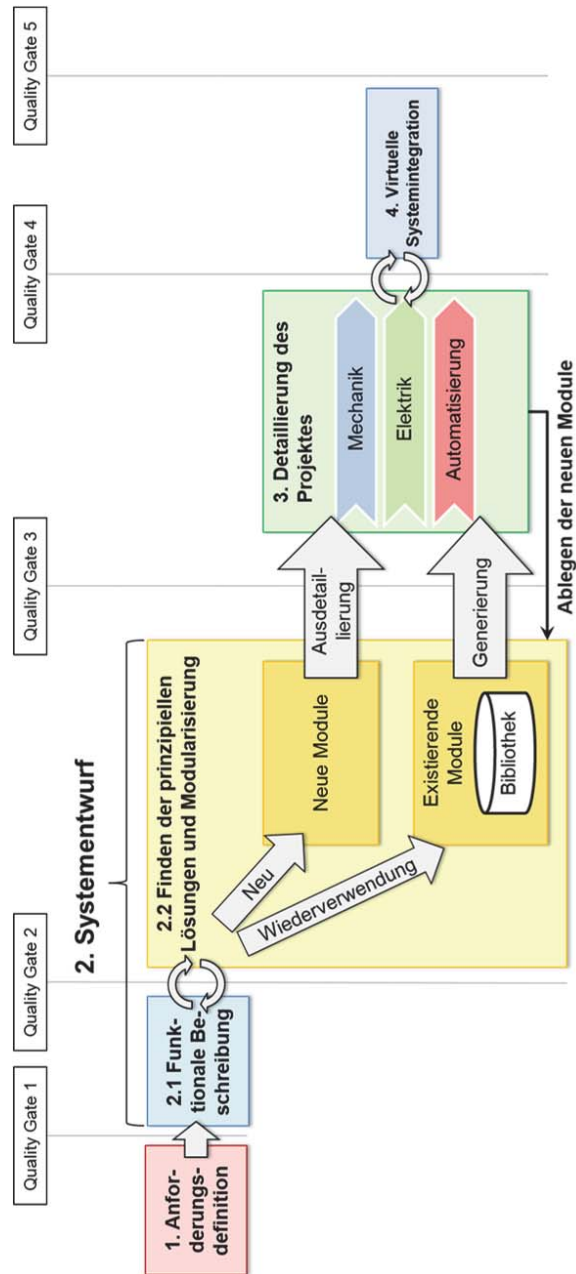


Abbildung 4.1: Vorgehen zur systematischen Entwicklung von Sondermaschinen in Bezug auf die in Kapitel 3.5 eingeführten Quality Gates

4.2 Anforderungsdefinition

Die Integration einer Maschine in eine bestehende Produktionslinie erfordert eine Betrachtung entsprechender Eigenschaften in allen Disziplinen. Böck [2013] zeigt, welche Eigenschaften aus produktionsplanerischer Sicht zur Integration in eine solche Linie betrachtet werden müssen. Diese können auch als Anforderungen an eine zu entwickelnde Maschine gesehen werden. Die statischen Eigenschaften des Produktionsumfeldes liefern die Anforderungen zur maximal nutzbaren Fläche und deren maximalen Bodenbelastung. Dadurch werden direkte Vorgaben an die Mechanik definiert. Diese sind i.d.R. den Layoutplanungen zu entnehmen. Aus dem Entwicklungsvorgehen der Elektrik ist bekannt, dass die fluidtechnischen bzw. elektrischen Energie- und Medienquellen mit den entsprechenden Komponenten zu verschalten sind. Diese Anforderungen des Produktionsumfeldes sind in der Kategorie Energie-/Medienfluss anzugeben. Im Bereich des Informationsflusses sind die Signalschnittstellen der zu entwickelnden Ressource anzuführen. Diese zeigen die Anbindung an ein vorhandenes MES-System oder zu Vor- und Nachfolgemaschinen der Linie auf. Neben der Automatisierungstechnischen Disziplin besitzt auch die Elektrik Interesse an diesen, da hierdurch auch hardwaretechnische Schnittstellen mitdefiniert werden können. Eine der wichtigsten Anforderungen bei der Integration in bestehende Linien ist die Taktzeit. Diese bildet zusammen mit dem Werkstück- bzw. Werkzeugfluss auf der Mikroebene und den Immissionen die produktursächlichen Anforderungen. Produktursächlich bedeutet in diesem Zusammenhang, dass sich diese Anforderungen aus der Produktplanung und -konstruktion ergeben. Die Taktzeit ist hierbei auf die geplante Produktionsmenge und die zur Verfügung stehende Zeitdauer bezogen. Diese Definitionen sind häufig im Rahmen der Produktplanung definiert. Neben dieser Leitlinie sind zur Anforderungserhebung noch prozessspezifische Anforderungen zu betrachten, wie beispielsweise welchen Bewegungspfad das Werkstück zu durchlaufen hat. Die Hauptmerkmale Sicherheit, Ergonomie, Fertigung, Kontrolle, Montage, Transport, Gebrauch, Instandhaltung, Recycling, Kosten und Termin aus Pahl *et al.* [2007] sind in der bisherigen Betrachtung noch nicht eingeflossen. Sie bilden Randbedingungen, welche nur schwer quantifizierbar sind oder aber ihre Ursache in der Organisation liegt. Die Kategorien der produktionsumfeld- und produktbezogenen Anforderungen ergeben zusammen die integrationsrelevanten Anforderungen. Diese werden im Weiteren zur Betrachtung der Anforderungserhebung berücksichtigt. Eine Übersicht der Merkmale und Kategorien zur Erhebung der Anforderungen gibt Abbildung 4.2.

Integrationsrelevante Anforderungen	
Produktionsumfeldbezogene Anforderungen	Randbedingungen hinsichtlich Aufbau, Inbetriebnahme, Gebrauch und Recycling
<ul style="list-style-type: none"> • Statische Eigenschaften <ul style="list-style-type: none"> • Maximale Ressourcenfläche • Maximale Bodenbelastung • Informationsfluss (HW/SW) <ul style="list-style-type: none"> • zu übergeordneter Steuerung • zu Vor- bzw. Nachfolgermaschinen • Energie-/Medienfluss (HW/SW): <ul style="list-style-type: none"> • Quellen: <ul style="list-style-type: none"> • elektrisch • fluidtechnisch • Senken: <ul style="list-style-type: none"> • elektrisch • fluidtechnisch • Emissionen 	<ul style="list-style-type: none"> • Sicherheit • Ergonomie • Fertigung • Kontrolle • Montage • Transport • Gebrauch • Instandhaltung • Recycling • Kosten • Termin
Produktbezogene Anforderungen	Prozessbezogene Anforderungen
<ul style="list-style-type: none"> • Werkstück-/Werkzeugfluss • Taktzeit • Immissionen 	<ul style="list-style-type: none"> • Kinematik: <ul style="list-style-type: none"> • Bewegungspfade • Kräfte • Zustandsgrößen wie: Druck, Temperatur, Feuchtigkeit, Erwärmung, Kühlung, Speicherung, Arbeitsaufnahme, Energie • Vorgeschriebene Werkstoffe

Abbildung 4.2: Sachmerkmal-Leitlinie zur Erhebung von Anforderungen für Sondermaschinen

4.3 Systementwurf

An die Phase des Systementwurfs sind in den vorigen Kapiteln bereits generelle Anforderungen aufgestellt worden. Außerdem leitet Kapitel 3.4.3 die am Ende notwendigen Informationen zur weiteren parallelen Ausdetaillierung her. Dies ist zusammenfassend in Abbildung 4.3 dargestellt. Die Phase der Anforderungsdefinition ist im vorigen Kapitel beschrieben. Dabei sind die zur Integration in das Produktionssystem gestellten Anforderungen auf die jeweiligen Bestandteile so herunterzubrechen, dass eine Integration in das bestehende Produktionssystem lückenlos möglich ist. Weiterhin muss eine abstrakte Problembeschreibung, eine mechatronische Modularisierung bzw. eine Definition des mechatronischen Lösungsprinzips erfolgen. Außerdem ist eine mechatronische Beschreibung des Gesamtsystems erforderlich.

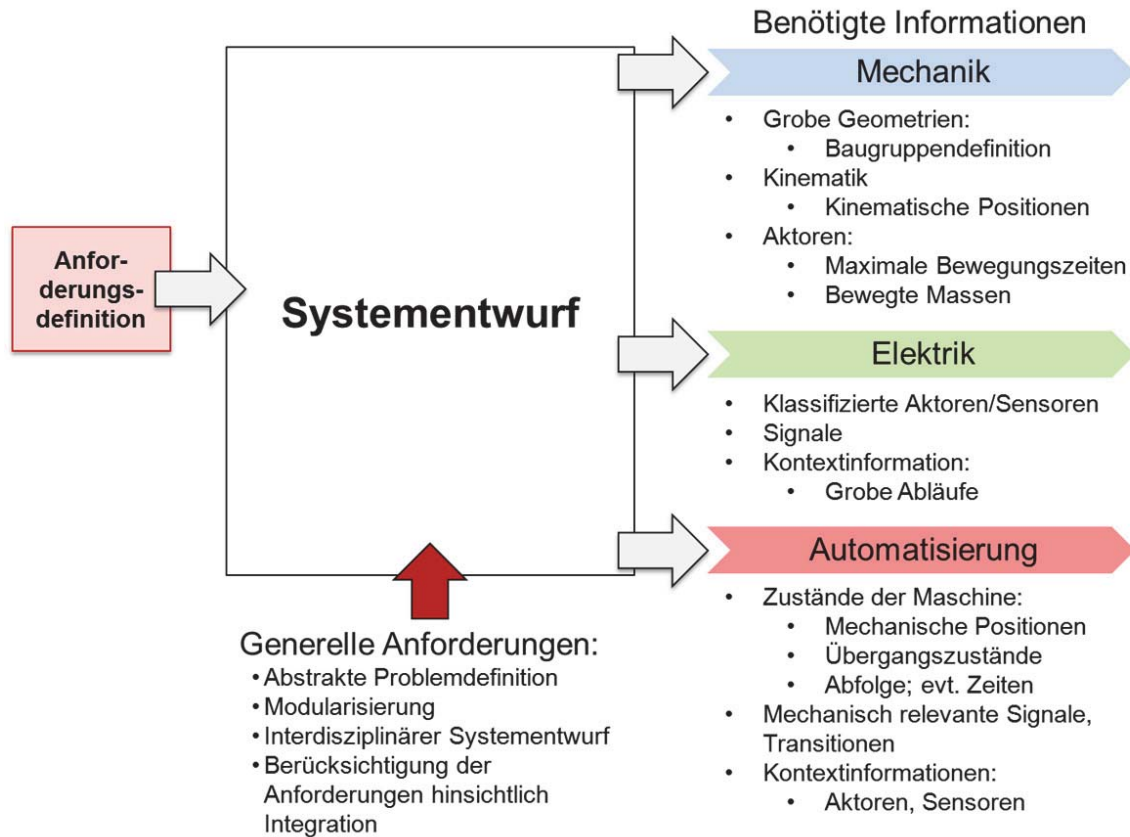


Abbildung 4.3: Anforderungen an die Phase des Systementwurfs

4.3.1 Funktionale Modellierung

Die funktionale Beschreibung dient zunächst der abstrakten Problemformulierung. Dazu wird wie in *Pahl et al.* [2007] zunächst die Hauptfunktion der zu entwickelnden Maschine aufgestellt. Die Formulierung der Funktion erfolgt, mittels einer Substantiv-Verb Kombination. Dadurch wird das Hauptaugenmerk auf die Aktivität der Maschine gelenkt, was einer Abkehr von einer Lösungszentrierung gleichkommt. Es öffnet sich ein großer Lösungssuchraum. Der richtige Abstraktionsgrad kann nach *VDI-Richtlinie 2803* [1996-10] gefunden werden. Im Weiteren erfolgt eine hierarchische funktionale Dekomposition der Hauptfunktion in Teilfunktionen. Dies kann durch die Frage „Wie?“ geschehen.

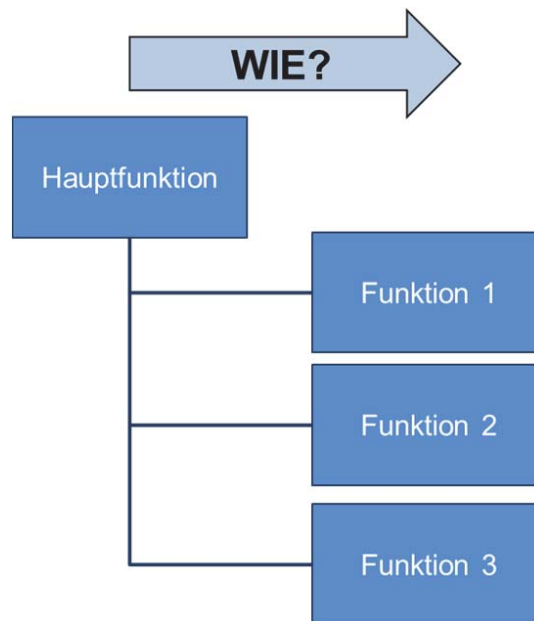


Abbildung 4.4: Funktionale Dekomposition

Zusätzlich sind die Anforderungen zur Integration der Maschine in ein bestehendes Produktionssystem zu berücksichtigen. Dabei können elektrische und fluidtechnische Flüsse als Energieflüsse angesehen werden. Außerdem können Energieflüsse auch mechanischer Natur sein. Die Kategorie der Informationen ist entsprechend mit Informationsflüssen gleichzusetzen. Dadurch wird ersichtlich, dass eine Vernetzung der Funktionen, wie in *Kallmeyer* [1998] vorgestellt, durchgeführt werden kann, wodurch die Kategorien Informations- und Energie-/Medien(fluss) Berücksichtigung finden. Die Anforderungen definieren spezifizierte Anschlusspunkte außerhalb der Maschinensystemgrenze, welche über eine energetische Relation mit einer Funktion innerhalb der Systemgrenze verbunden wird. Dies ist in Abbildung 4.5 dargestellt. Neben Energie- und Informationsverbindungen sind auch Stoffflüsse definiert. Diese können im Bereich des Maschinenbaus (aufgrund der diskreten Produktion) meistens direkt in Materialflüsse überführt werden. Es ist dadurch möglich beispielsweise notwendige Werkstückbewegungen auszudrücken. Innerhalb der Systemgrenze befinden sich nun alle Funktionen auf der untersten Ebene der funktionalen Dekomposition. Diese Funktionen werden soweit möglich mit den Relationen verbunden. Bei *Kallmeyer* [1998] und *Pahl et al.* [2007] sind diese Pfeile als Flüsse benannt. Im Bereich des Systems Engineering oder der Modellierung von Software mittels UML ist bekannt, dass eine Trennung zwischen der statischen Systembeschreibung und der Verhaltensbeschreibung notwendig ist um alle Zusammenhänge zu definieren.

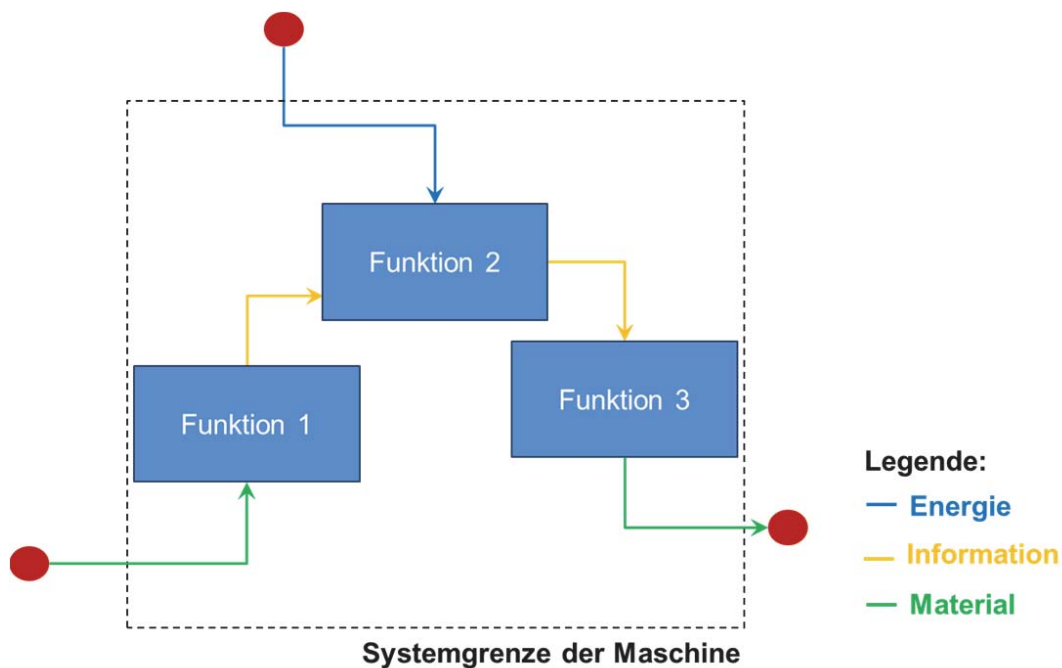


Abbildung 4.5: Vernetzung von Funktionen

Da das Wort „Fluss“ eine zeitliche Abhängigkeit der Relation impliziert, ist diese Trennung bei der Interpretation nach *Kallmeyer* [1998] nicht gegeben. Eine solche Modellierung enthält implizit die Zustandswechsel von einer Funktion zur anderen. Es ist eine Auswertung der Relationsrichtungen notwendig, wobei nicht immer eine eindeutige Rangfolge ermittelt werden kann. Allerdings ist im Bereich des Maschinenbaus eine Vereinfachung für diese Relationstypen möglich. In erster Näherung kann die Zeitdauer der Informationsübertragung und der Energiebereitstellung als zeitdiskret betrachtet werden. Dadurch reduziert sich die Bedeutung der Relationen auf die Definition der Systemgrenzen einzelner Funktionen. In Abbildung 4.5 ergibt beispielsweise die energetische Relation zwischen dem externen Versorgungspunkt und der Funktion zwei, dass die Lösung der Funktion externe Energie benötigt. Hierbei fällt die Systemgrenze der Maschine und der Funktion zusammen. Analog ergibt die informationstechnische Relation von Funktion eins zu Funktion zwei die Systemgrenze für die genannten Funktionen. Eine solche statische Interpretation der funktionalen Vernetzung muss auch für den eingeführten Typ des Materials gelten. Dieser kann im Maschinenbau jedoch nicht als zeitdiskret betrachtet werden. Hinzu kommt die Forderung aus dem Systems Engineering nach einer verhaltensorientierten Systemmodellierung parallel zur aufgezeigten statischen Systembeschreibung. Deshalb muss parallel zum aufgezeigten Modell eine zeitabhängige bzw. verhaltensorientierte (gesteuerte) Systemmodellierung erfolgen. Dadurch ist es

möglich zeitdauerabhängige Materialbewegungen über eine Funktion zu beschreiben. Relationen vom Typ „Material“ geben somit nur eine Materialübergabe im Rahmen des statischen Modells an. Dies berücksichtigt gleichzeitig die Abbildung des Werkstück- bzw. Werkzeugflusses. Die Nutzung einer Substantiv-Verb Kombination kann, wie in Kapitel 2.3.1 bereits beschrieben, als Aktivität betrachtet werden. Solche Aktivitäten können sequenzielle Abhängigkeiten untereinander besitzen. Zur Verkürzung der Taktzeiten ist zu überprüfen, ob einzelne Funktionen parallel ausgeführt werden können. Dabei gibt die Taktzeit an, wie lange der Produktionsprozess innerhalb der Gesamtmaschine zwischen zwei Produkten benötigen darf. Daraus ist ersichtlich, dass alle notwendigen Funktionen zur Umsetzung des Prozesses aufaddiert nur solange wie die geforderte Taktzeitangabe dauern dürfen. Dies bedingt eine zeitdauerbasierte Verhaltensbeschreibung des Systems, um die Gesamtdauer auf einzelne Funktionen herunterzubrechen. Die Verhaltensbeschreibung auf der funktionalen Ebene ist als erster Schritt in Richtung SPS-Programm zu verstehen. Allerdings ist aufgrund der abstrakten „lösungsneutralen“ Modellierung eine weitere Anreicherung mit Informationen, wie Signalen oder deren Typen auf dieser Ebene nicht zielführend. Die „dynamische“ Systembeschreibung, als Gegenstück zur statischen, dient als Eingangsinformation für die SPS-Programmierung. Hierzu bieten sich die Sprachen der *DIN EN 61131 Blatt 3* oder der *UML* an. Ein Vergleich der Beschreibungsformen hinsichtlich der angegebenen Anforderungen veranschaulicht Tabelle 4.1:

	Zeitdauerbasierte Beschreibung	Parallele und sequenzielle Abhängigkeiten	Keine zusätzlichen Informationen zur Modellierung notwendig
Gantt-Charts	●	●	●
Impulsdiagramme	●	●	○
PERT-Charts	●	●	●
Logiknetzwerke	○	○	○
Sequential Function Charts	○	●	◐
Zustandsdiagramme	○	◐	◐
Aktivitätsdiagramme	○	●	●
Sequenzdiagramme	○	●	●
Legende: ○ = nicht erfüllt; ◐ = teilweise erfüllt; ● = vollständig erfüllt			

Tabelle 4.1: Vergleich von Beschreibungssprachen zur Modellierung des Systemverhaltens auf funktionaler Ebene

Wie zu sehen, erfüllen lediglich Gantt- und PERT-Charts alle Anforderungen. Aufgrund des hohen Verbreitungsgrades, wie beispielsweise im Projektmanagement, fällt die Entscheidung zugunsten

von Gantt-Charts aus. Es ergibt sich eine Verhaltensbeschreibung, wie sie in Abbildung 4.6 visualisiert ist.

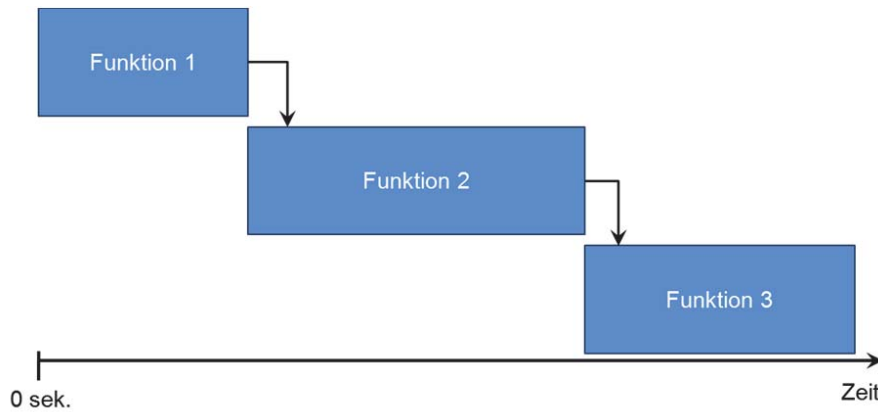


Abbildung 4.6: Gantt-Chart zur Beschreibung des Systemverhaltens

Dabei zeigt die Zeitachse, die Dauer der Funktionen an. Die Anforderung der Taktzeit wird auf der Zeitachse überwacht werden. Zusammenfassend zeigt Abbildung 4.7 die Beschreibung und das Vorgehen zur Funktionsmodellierung auf funktionaler Ebene.

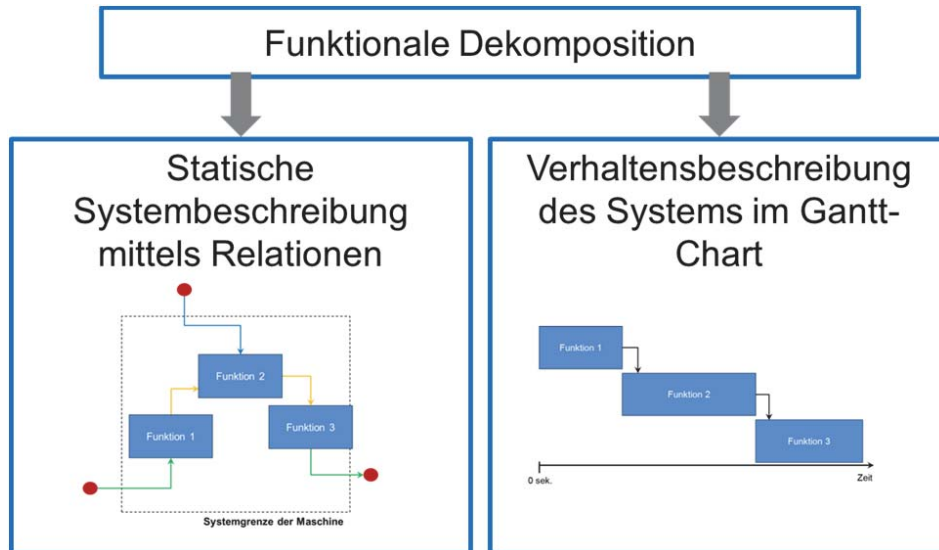


Abbildung 4.7: Übersicht der funktionalen Modellierung

Die Verhaltensbeschreibung ermöglicht ein Herunterbrechen der Taktzeit auf die notwendigen Funktionen der Maschine. Die Schnittstelleninformationen der verfügbaren Energieanschlüsse und der informationstechnischen Schnittstellen werden durch Relationen der Typen Information und

Energie berücksichtigt. Die Systemgrenzen hinsichtlich Werkstück- und Werkzeugübergabe findet sich im Relationstyp Material wieder.

4.3.2 Finden der prinzipiellen Lösung und der mechatronischen Systembeschreibung

Die Beschreibung des Problems durch eine funktionale Modellierung öffnet damit ein großes Lösungssuchfeld. Es ist offensichtlich, dass für die Implementierung einer solch abstrakten Funktion mehrere Lösungsprinzipien in Frage kommen. Zur Suche dieser Prinzipien bieten sich morphologische Kästen an. Abbildung 4.8 zeigt einen solchen.

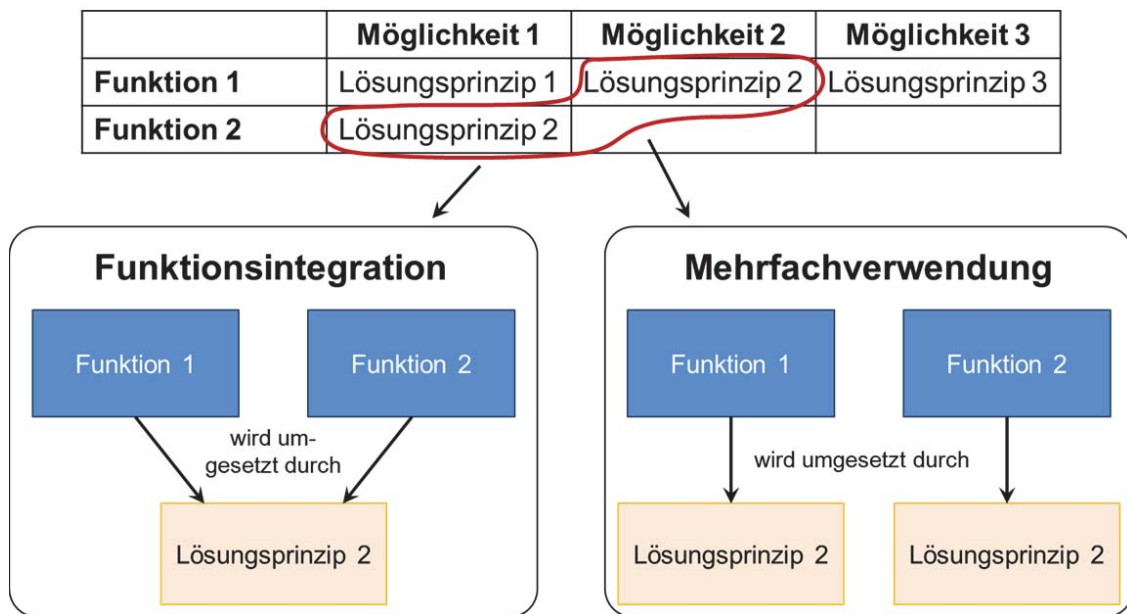


Abbildung 4.8: Morphologischer Kasten und die Ableitung von Funktionsintegrationen und Mehrfachverwendung

Diese Visualisierung eignet sich zur Dokumentation der Suche und des Findens von alternativen Lösungsprinzipien. Die Ermittlung derer kann mit geeigneten Kreativitätsmethoden erfolgen oder auf Basis bereits existierender Implementierungen der verwendeten Funktion in früheren Projekten von einer Software vorgeschlagen werden. Ein Lösungsprinzip ist, wie bereits in Kapitel 3.4.3 beschrieben auf der logischen Ebene einzuordnen. Bei der Suche nach entsprechenden Lösungen ist besonders die Verhaltensbeschreibung der funktionalen Ebene zu berücksichtigen. Oft lassen sich durch den Einsatz geeigneter Prinzipien Funktionen parallelisieren. Wenn diese Abhängigkeit besteht, ist die funktionale Beschreibung nicht vollständig lösungsneutral. Die Angabe, dass Funktionen sequenzielle Abhängigkeiten besitzen schließt weitere Lösungsmöglichkeiten aus.

Dasselbe ist für die Relationsarten Information, Energie und Material der statischen Funktionsbeschreibung gültig. [Frank, et al. 2013c] (vergleiche Abbildung 4.9)

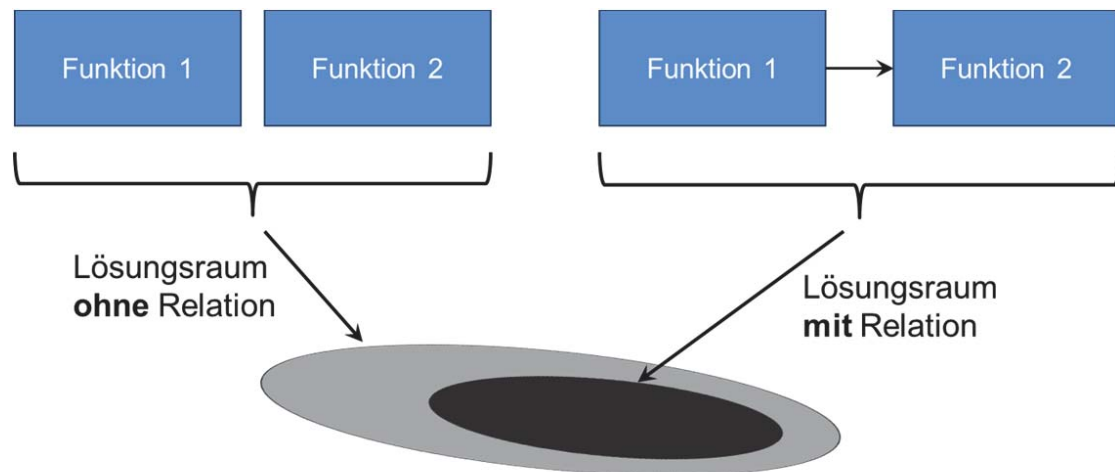


Abbildung 4.9: Reduktion von möglichen Lösungen durch Modellierung von Beziehungen

Dies führt zum Schluss, dass eine reine Top-Down Modellierung ausgehend von Funktionen hin zum Lösungskonzept nicht möglich ist. Deshalb ist die Modellierung iterativ durchzuführen. Gleichbedeutend damit sind durch die Festlegung von Lösungsprinzipien weitere Funktionen oder Relationen im funktionalen Modell hinzuzufügen. Dieser iterative Zyklus ist in Abbildung 4.10 dargestellt. Relationen des Typs Information geben die Abhängigkeiten zwischen verschiedenen Funktionen mit softwaretechnischer Repräsentanz an. Es ist durch dieses iterative Vorgehen auch möglich Relationen zwischen Funktionen zu minimieren, was der Forderung im Rahmen eines strukturierten Steuerungsentwurfs nach geringem Datenaustausch zwischen Softwarebausteinen nachkommt. Gleichzeitig erhöht dies die Chance auf Wiederverwendbarkeit von Softwarebausteinen. Eine effektive Wiederverwendung in mechatronischem Sinne kann nur erfolgen, wenn alle bereits vorhandenen Lösungsprinzipien in Betracht gezogen werden können. Im Rahmen der Softwareunterstützung ist deshalb eine Möglichkeit zum Finden bereits existierender Lösungsprinzipien umzusetzen. Jede Lösung, welche in die engere Auswahl kommt, wird der entsprechenden Funktion zugeordnet und in einer separaten Spalte dokumentiert. Es ist ersichtlich, dass sich dadurch immer eins zu eins Relationen zwischen Funktion und Lösungsprinzip ergeben. Allerdings zeigt die Literatur gerade bei mechatronischen Produkten genügend Beispiele, welche aufzeigen, dass Funktionsintegrationen erhebliche Vorteile bieten. Eine solche Integration entspricht der Umsetzung mehrerer Funktionen mit einem Lösungsprinzip. Ist dies nicht möglich, muss

geprüft werden, ob andere Funktionen mit einem gewählten Prinzip auch umgesetzt werden können. Dadurch wird eine Mehrfachverwendung des gleichen Prinzips in der Maschine herbeigeführt, was zur Aufwandsreduktion führen kann. Dieser Zusammenhang ist in Abbildung 4.8 veranschaulicht. Eine Funktionsintegration ist einer Mehrfachverwendung bei Neuentwicklungen vorzuziehen, da der materielle und entwicklungstechnische Aufwand deutlich geringer ausfällt.

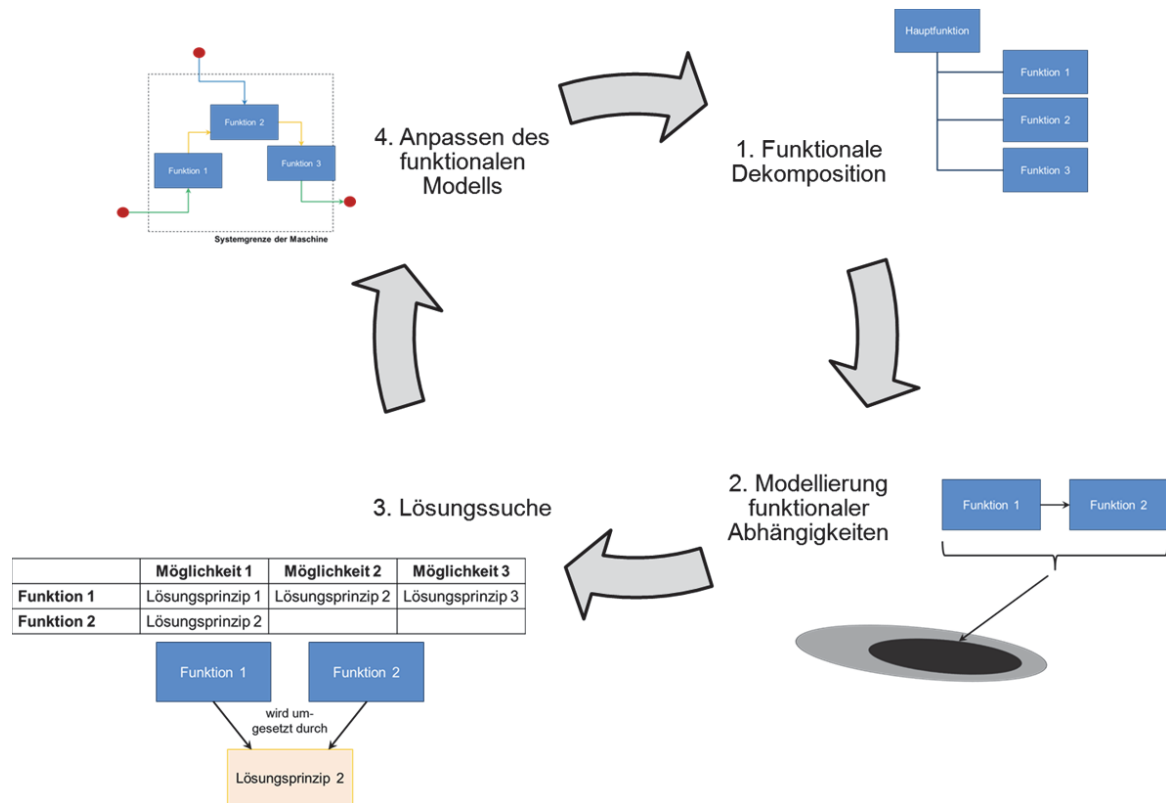


Abbildung 4.10: Iteratives Vorgehen zur funktionalen Modellierung und der Definition von Lösungsprinzipien

Gerade am Beispiel der Funktionsintegration ist zu sehen, dass eine Wiederverwendung nicht alleine durch das Lösungsprinzip entschieden wird, sondern auch welche Funktion(en) damit umzusetzen ist (sind). Ein Förderband kann beispielsweise Produkte speichern und gleichzeitig bewegen. Der Unterschied zwischen einem Förderband, welches beide Funktionen erfüllt und einem, welches nur Produkt bewegen erfüllt liegt häufig in der unterschiedlichen Ansteuerung, also der Ausprägung des Steuerungsprogramms. Es ist ersichtlich, dass eine Wiederverwendung neben den Lösungsprinzipien auch die umzusetzenden Funktionen mit zu berücksichtigen hat. Daher müssen zur Definition von wiederverwendbaren Modulen, Funktionen und das zugehörige Lösungsprinzip

zusammen betrachtet werden. Es sind die Fälle aus Abbildung 4.10 zu unterscheiden. Im einfachsten Fall, einer 1:1-Beziehung zwischen Funktion und Lösungsprinzip, ist die Systemgrenze des wiederverwendbaren Moduls eindeutig bestimmt. Gleiches gilt im Falle der N:1-Beziehung. Gegebenenfalls ist es hierbei möglich eine gemeinsame Elternfunktion der zu verbindenden Funktionen zu identifizieren und über eine 1:1-Relation mit dem Lösungsprinzip zu verbinden. Eine 1:N-Beziehung besitzt theoretisch definierte Systemgrenzen, aus praktischen Gesichtspunkten, würde damit das Ziel einer Modularisierung aber nicht eingehalten. Deshalb muss die Funktion zunächst in zwei separate Funktionen zerlegt werden. Diese können wiederum mit 1:1-Relationen auf die Lösungsprinzipien verbunden werden. N:M-Beziehungen können in gleicher Weise aufgelöst werden. Theoretisch ist auch die Modellanpassung der Lösungsprinzipien möglich. Es wird allerdings empfohlen die funktionale Seite zu adaptieren, da die Funktionen im Weiteren die Eingangsgrößen für die Software darstellen. Alle Modifikationen der funktionalen Seite sind in der Systembeschreibung nachzuziehen.

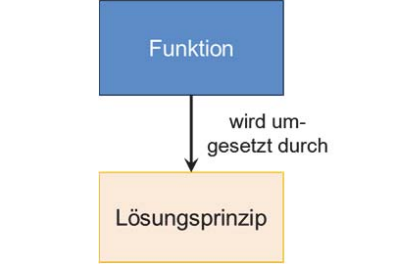
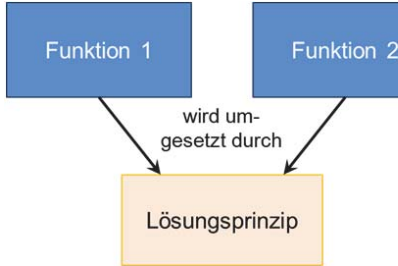
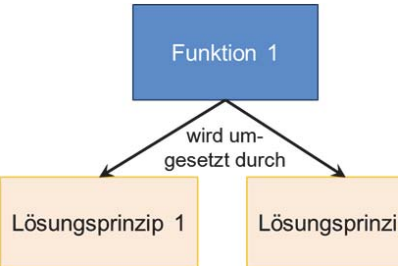
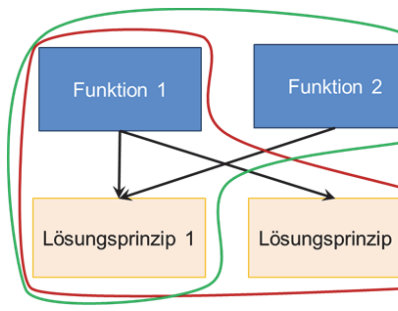
Kardinalität zwischen Funktionen und Lösungsprinzipien		Systemgrenze des Moduls bzw. Handlungsempfehlung zur Definition jener
1:1		Systemgrenze des Moduls eindeutig definiert
N:1		Systemgrenze des Moduls eindeutig definiert. Eventuell ist eine 1:1 Beziehung auf höherer Dekompositionsebene möglich
1:N		Systemgrenze des Moduls eindeutig definiert, aber hinsichtlich der softwaretechnischen Ausrichtung empfiehlt sich eine weitere funktionale Dekomposition
N:M		Systemgrenze nicht eindeutig gegeben (vgl. rotes und grünes Modul). Auflösen der N:M-Beziehung durch weitere funktionale Dekomposition (hier: Funktion 1), bis Systemgrenze eindeutig gegeben ist

Tabelle 4.2: Beziehungen von Funktionen zu Lösungsprinzipien unter dem Gesichtspunkt der Modularisierung zur Wiederverwendung

Wie bereits im Rahmen der statischen Systembeschreibung angedeutet, führen die Relationen zwischen den Funktionen auf die Schnittstellendefinition der Lösungsprinzipien. In Abbildung 4.11 ist zu sehen, dass die informationstechnische Verbindung zwischen Funktion eins und Funktion zwei einen modulinternen Informationsübergang darstellt.

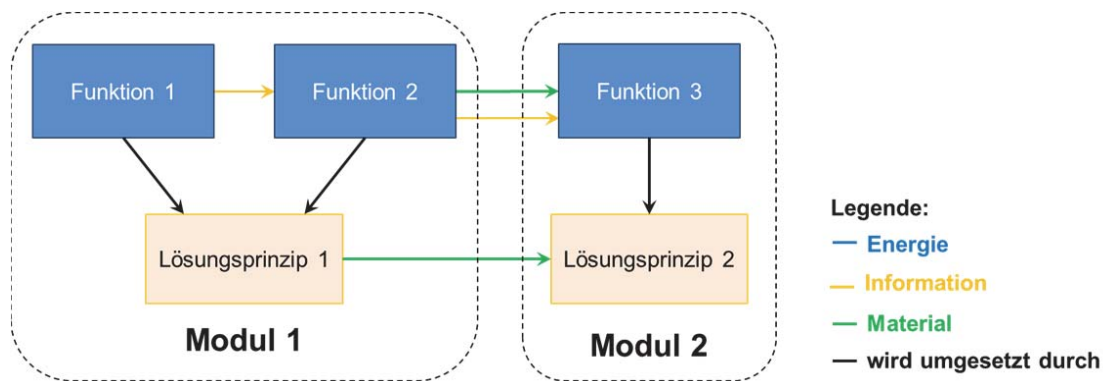


Abbildung 4.11: Statische mechatronische Systembeschreibung für das Gesamtsystem Sondermaschine

Diese Relation kann als Anforderung für die interne Kommunikation, also einer Schnittstellendefinition innerhalb des Moduls gedeutet werden. Die Relation zwischen Funktion zwei und Funktion drei definiert hingegen die modulübergreifende Kommunikation. Eine automatische Ableitung des Maschinenlayouts (Technologieschema), wie es die Elektrik benötigt (vgl. Kapitel 3.2), kann über die Kenntnis der Verknüpfungen zwischen den Funktionen zwei bzw. drei und durch den vom Typ „wird umgesetzt durch“ jeweils zugeordneten Lösungsprinzipien auf die logische Ebene erfolgen. Eine Minimierung der Komplexität zwischen Modulen kann durch die Kommunikationsbeziehungen hergestellt werden. Die energetischen Relationen geben Aufschluss über notwendigen externen Energiebedarf oder -abfuhr. Über den bereits genannten Mechanismus für die automatische Ableitung der Material(fluss)beziehungen auf die logische Ebene ist dies auch für die anderen Verbindungsarten möglich. Allerdings bietet es sich an, beim Übergang von der funktionalen zur logischen Ebene die Relationen detaillierter zu spezifizieren. Eine Materialrelation würde beispielsweise zu einer Rohstoffrelation werden. Die energetische Verknüpfung kann durch die Detaillierung in pneumatisch, hydraulisch, mechanisch oder elektrisch weiter detailliert werden.

4.3.3 Wiederverwendung existierender mechatronischer Module

Jedes Lösungsprinzip ist als Klassifikation eines mechatronisch wiederverwendbaren Moduls anzusehen. Die mechatronischen wiederverwendbaren Alternativen müssen auf ihre Eignung im

konkreten Projekt hin überprüft werden. Außerdem besitzen diese Repräsentanzen in mindestens einer der drei Disziplinen, was im Weiteren als Sichtweisen oder Facetten bezeichnet wird. Der Gesamtzusammenhang ist in Abbildung 4.12 dargestellt.

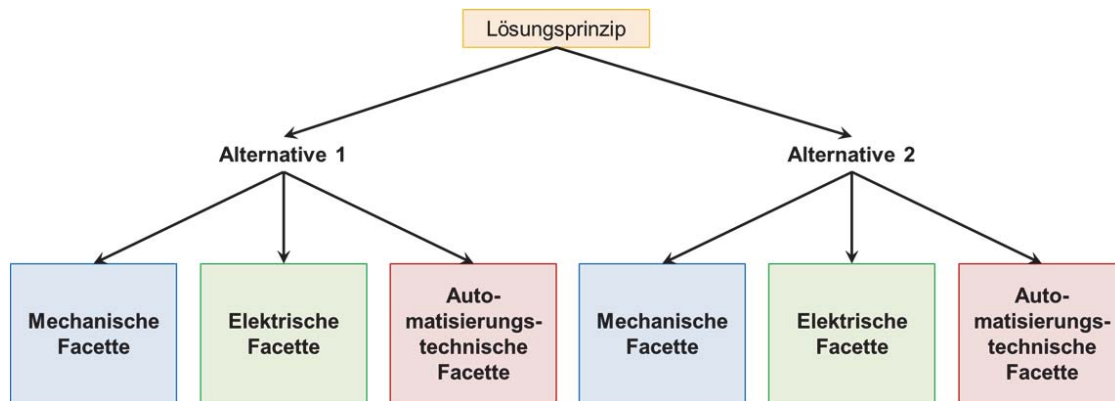


Abbildung 4.12: Lösungsprinzip als Klassifikation verschiedener physikalischer Alternativen

Hinsichtlich der Modulklassifikation bieten sich kinematische Konzepte oder produktionstechnische Verfahren an. Ein kinematisches Konzept, welches in all seinen Varianten exakt beschrieben ist, kann als Top-Down-konfigurierbar angesehen werden. Würde eine solche Konfiguration auf Basis der gegebenen Anforderungen stattfinden, dann wären alle benötigten Teile auf der physikalischen Ebene beschrieben. Somit ermöglicht eine Konfiguration bzw. die Wiederverwendung von mechatronischen Modulen im Wesentlichen einen gezielten Einstieg auf niederen Abstraktionsebenen, was zu einem zeitlichen Sprung im Entwicklungsprozess führt. Dies ist in Abbildung 4.13 am angestrebten Entwicklungsverlauf dargestellt. Die Konfiguration von Maschinen ist im Bereich des Sondermaschinenbaus aufgrund der hohen Kundenindividualität nicht sinnvoll. Der Einstiegspunkt für eine solche Top-Down-Konfiguration ist rot dargestellt. Folglich müssen die Ausprägungen der Lösungsalternativen so gestaltet sein, dass eine maximale Wiederverwendung stattfinden kann.

Dazu müssen alle Teilmodelle, inklusive ihrer Abstraktionsebenen, den Gültigkeitsbereichen (vgl. Kapitel 3.4.1) und deren Systemgrenzen (vgl. Kapitel 3.4.2) in allen Disziplinen berücksichtigt werden. Als Beispiel sei das Hinzufügen eines Servomotors zu einem wiederverwendbaren Objekt aus Sicht der Mechanik angeführt. Dieser Motor ist vollständig definiert. Deshalb kann auch aus elektrischer Sicht dieses Stellelement als Echtgerät Berücksichtigung finden. Dadurch liegt der Einstiegspunkt für diese Komponente neben der mechanischen auch in der elektrischen Sichtweise auf der physikalischen Beschreibungsebene. Werden Objekte der Mechanik nicht zur Wiederver-

Selbiges kann als Handlungsempfehlung auch für die Problematik der unterschiedlichen Systemgrenzen gelten, indem aus mechanischer Sicht Komponenten nicht definiert, aus elektrischer Sicht diese auf einer höheren Abstraktionsebene aber berücksichtigt werden. Dies ermöglicht beispielsweise der Elektrik, Komponenten, welche der Wiederverwendung des Moduls erst noch eine physikalische Auslegung benötigen, auf der logischen Ebene vorzusehen und zu verkabeln. Dies kann auch als unternehmensinterne Regel für die Verkabelung von Antrieben betrachtet werden. Gleichzusetzen ist dies mit der Strategie von internen Standards nach *Litto* [2012]. Abbildung 4.13 visualisiert den Zusammenhang.

Fällt eine Entscheidung für eine Alternative, so ist auf Basis des Gesamtmodells die Kompatibilität mit den anderen Modulen zu überprüfen. Diese Überprüfung kann in drei Fällen, dargestellt in Tabelle 4.3, münden.

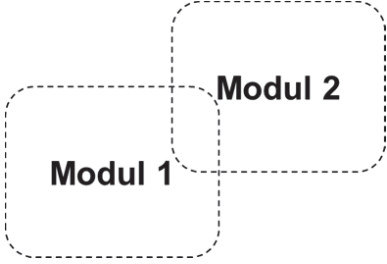
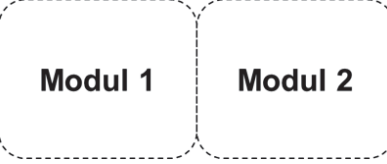

Überlappende Systemgrenzen	
Kompatible Systemgrenzen	
Entfernte oder nicht kompatible Systemgrenzen	

Tabelle 4.3: Mögliche Systemgrenzen zweier Module im Kontext des Maschinenprojekts

Die Systemgrenzen sind gleichbedeutend mit den Schnittstellen. Problematisch ist eine Wiederverwendung bei überlappenden Systemgrenzen. Hierzu muss eines der beiden Module angepasst werden. Als Beispiel kann eine Pumpe zur hydraulischen Versorgung dienen. Ist diese in beiden Modulen enthalten, im Projektkontext wird allerdings nur eine benötigt, ist dies entsprechend anzupassen. Bei entfernten Systemgrenzen ist eine projektspezifische Verbindung beider Module in

der weiteren Detaillierung durchzuführen. Gerade in der Automatisierung wird häufig von inkompatiblen Schnittstellen gesprochen. Diese sind ebenfalls in diesem Fall drei inbegriffen. Hierbei ist zu beachten, dass der technische Aufwand zur Überbrückung größer sein kann, als die Aufwandsreduktion durch Wiederverwendung. Somit ist eine gründliche Überprüfung notwendig. Wiederverwendbare mechatronische Objekte erfordern im Kontext der Sondermaschinen folglich unterschiedliche Facettenmodelle, worin unterschiedliche Systemgrenzen und Abstraktionsniveaus zu berücksichtigt werden können.

4.3.4 Mechatronische Beschreibung neuer Lösungen/Module

Sind keine mechatronischen Module vorhanden, welche wiederverwendet werden können, ist für jedes Lösungsprinzip eine mechatronische Beschreibung zu finden, die es ermöglicht eine Detaillierung des Maschinenprojektes in allen Disziplinen parallel zu starten. Dazu müssen die Aufgabenstellungen der verschiedenen Disziplinen eindeutig beschrieben sein. Bereits in Kapitel 3.4.3 werden die notwendigen Informationen hierfür hergeleitet. Weiterhin wird dort der Einfluss des kinematischen Konzeptes auf die verschiedenen Disziplinen erläutert. Abbildung 4.14 zeigt die Bestandteile einer kinematischen Konzeptdefinition. Wie dort zu sehen, werden bereits grobe 3D-Geometrien definiert, welche es erlauben sowohl erste Baugruppendefinitionen abzuleiten, als auch über die Volumina und den einzusetzenden Materialien Aussagen über die bewegten Massen zu treffen.

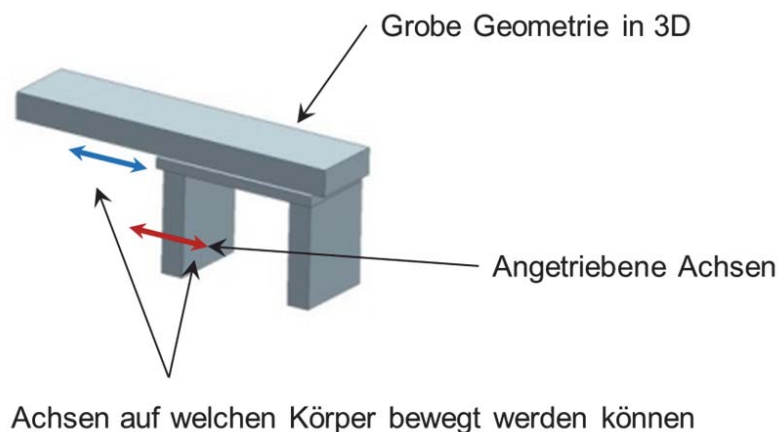


Abbildung 4.14: Notwendige Informationen des kinematischen Konzeptes

Neben diesen Informationen enthält die Beschreibung kinematische Lagerungen, wie rotatorische oder lineare Verknüpfungen zwischen den bewegten Körpern. Weiterhin ist eine Definition der

angetriebenen Achsen notwendig. Dies kann aus Sicht der Automatisierung als Aktor begriffen werden. Das kinematische Konzept selbst enthält noch keine Definition der Wirkelemente auf der physikalischen Ebene, wie z.B. einen Servomotor mit Bestellnummer. Aus Sicht der Elektrik sind aber gerade die mechanischen Stellglieder zur weiteren Detaillierung notwendig. Diese müssen so definiert sein, dass elektrische Symbole dafür identifiziert werden können. Folglich muss jedem notwendigen Aktor ein klassifiziertes Wirkelement der ECAD-Bibliothek zugeordnet werden. Dadurch erfolgt eine Definition des Antriebskonzepts. Werden beispielsweise zwei voneinander abhängige Achsen jeweils durch einen Servomotor angetrieben, ist die Abhängigkeit steuerungstechnisch umzusetzen. Der Mechaniker muss kein Getriebe mit zwei Abtriebswellen vorsehen. Daran wird ersichtlich, dass auch die disziplinspezifische Umsetzung der Funktionen mitdefiniert wird. Mechanische Wandler, wie beispielsweise Getriebe sind für den Elektriker irrelevant. Diese gehen direkt in die Antriebsauslegung ein und beeinflussen dadurch die physikalische Definition des mechanischen Stellgliedes. Neben der Definition der Symbole für die mechanischen Stellglieder gilt dies auch für die Sensorik.

In der Anforderungserhebung aus Kapitel 4.2 wird bereits die Versorgungsperipherie definiert. Daraus ist ableitbar, ob die Maschine beispielsweise eine eigene Druckluftversorgung beim Einsatz von pneumatischen Antrieben benötigt oder ob diese durch die Peripherie des Produktionssystems gegeben ist. Diese Umstände können zu gänzlich unterschiedlichen Antriebskonzepten führen, was bei der Auswahl der Wirkelemente oder der Energieversorgung bzw. deren Aufbau und Verschaltung zu berücksichtigen ist. Weiterhin müssen Signale, welche für die Ansteuerung des mechanischen Stellgliedes zuständig sind, definiert sein. Neben diesen „Ausgangssignalen“ sind auch die „Eingangssignale“ in das Steuerungsprogramm der Sensoren zu modellieren. Zudem ist eine Information bzgl. der physischen Anbindung der Maschine an die Signalübertragung von/zur Vorgänger, Nachfolger und Leitsystem zu planen. Dies alles verkörpert eine vollständige Aufgabenbeschreibung des Elektrikers. Zusammenfassend ist dies in Abbildung 4.15 visualisiert. Die Signale zusammen mit der Verknüpfung ihrer mechanischen Auswirkung (Bewegung der Achse) ist gleichzeitig eine wichtige Information für den Automatisierer (vgl. Kapitel 3.4.1). Neben diesen sind Eingangssignale von Sensoren oder anderen Maschinen bzw. des Leitsystems mit ihrer Verbindung zu den entsprechenden „gemessenen“ Zuständen zu berücksichtigen.

Zusammen mit einer Ablaufbeschreibung, welche die Abfolge der Aktionen zwischen den Zuständen aufweist, ermöglichen die Signale und deren Verknüpfung dem SPS-Programmierer mit der Ausarbeitung des Steuerungsprogramms zu beginnen. Wie in der *DIN EN 61131 Blatt 3*

beschrieben, sind zeitliche Dauern für den SPS-Programmierer nur untergeordnet notwendig, da die Steuerung ereignisbasiert arbeitet. Zeitdauern spielen häufig nur im Fehlerfalle eine Rolle. Dadurch werden häufig SFC's zum Start der Planungen in der Automatisierungstechnik verwendet. Allerdings benötigt die Mechanik zur Auslegung der Antriebe die maximal zur Verfügung stehenden Zeiten zwischen zwei Positionen bzw. Zuständen. Dies führt auf die Notwendigkeit das Verhalten auch in dieser Phase der Maschinenmodellierung zeitdauerbasiert zu modellieren.

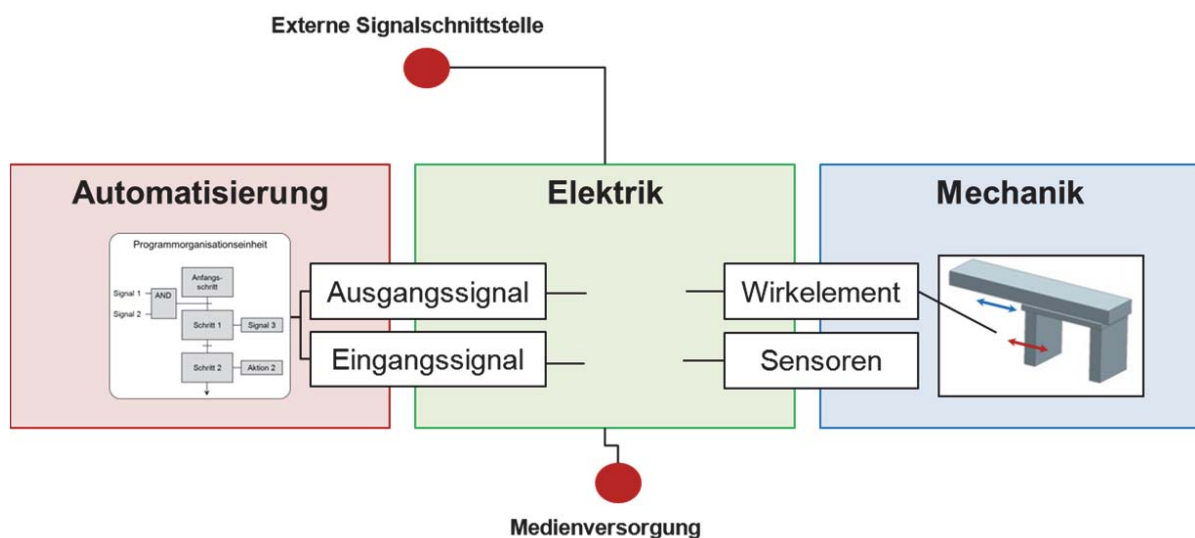


Abbildung 4.15: Startpunkt für die elektrische Konstruktion

Zunächst ist das Sollverhalten der Maschine zu beschreiben, was die Abfolge der einzelnen Aktionen beinhaltet. Dabei definiert eine Aktion, den Ausgangs- und Zielzustand eines Moduls, also einen Zustandsübergang. Die häufigste Art der Zustände sind Positionen, wobei die geometrisch korrekte Position (beispielsweise in mm) zunächst nur eine untergeordnete Rolle spielt. Wichtiger sind aussagekräftige Namen, wie „Drehtisch in Ausgangslage“. Daraus ist ersichtlich, dass die Modellierung des Steuerungsprogramms Zeitdauern, parallele und sequenzielle Abhängigkeiten der Aktionen, Signale, Transitionsdefinitionen und verschiedene Zustände berücksichtigen muss. Dies führt auf den in Tabelle 4.4 dargestellten Vergleich verschiedener Beschreibungssprachen für das Steuerungsverhalten. Es ist ersichtlich, dass Impulsdigramme alle Anforderungen erfüllen.

	Zeitdauerbasierte Beschreibung	Parallele und sequenzielle Abhängigkeiten	Signal- und Transitionsdefinitionen	Zustandsmodellierung, wie Positionen
Gantt-Charts	●	●	○	◐
Impulsdiagramme	●	●	●	●
PERT-Charts	●	●	◐	◐
Logiknetzwerke	○	◐	●	○
Sequential Function Charts	○	●	●	●
Zustandsdiagramme	○	◐	●	●
Aktivitätsdiagramme	○	●	●	●
Sequenzdiagramme	○	●	●	●

Tabelle 4.4: Vergleich von Beschreibungsmittel hinsichtlich des Sollablaufes des kinematischen Lösungsprinzips

Ein solches Impulsdiagramm ist in Abbildung 4.16 unterhalb des funktionalen Ablaufs visualisiert. Dort ist außerdem der Zusammenhang zwischen einer Funktion und dem gewählten Lösungsprinzip aufgezeigt. Die Dauer der Funktion im Gesamtsystem dient als Anforderung für das mit ihr verknüpfte Prinzip. Durch die zeitdauerbasierte Beschreibung der Bewegungen, ist es möglich diese Anforderung auf einzelne Achsbewegungen herunterzubrechen. Gleichzeitig besitzen die Entwickler bei Betrachtung dieses Maschinenteils Gewissheit über die Einhaltung der Gesamttaktzeit,

solange die zeitliche Anforderung erfüllt wird. Ergeben sich aufgrund des gewählten kinematischen Prinzips oder anderer Randbedingungen hier Änderungen, welche auf eine längere funktionale Dauer führen, ist dies direkt im Gantt-Chart der Funktionen abzuändern und die Gesamtzeit gegen die Maschinenanforderungen zu prüfen.

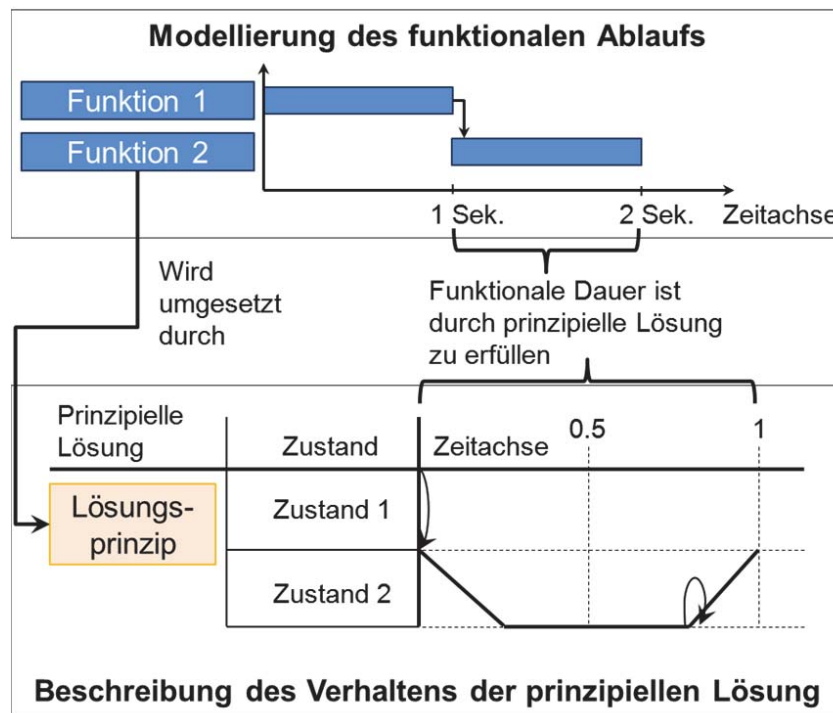


Abbildung 4.16: Zeitdauer der Funktion als Anforderung für den Bewegungsablauf im Impulsdiagramm

Zusammenfassend muss ein Lösungsprinzip Modelle für alle drei Disziplinen gleichermaßen enthalten. Diese sind voneinander abhängig, wie beispielsweise Aktoren, Signale und Positionen, welche in mehr als einer Disziplin benötigt werden. Daraus ist ersichtlich, dass im Rahmen einer Softwareunterstützung eine gemeinsame Beschreibung des Prinzips mit den genannten Beschreibungsmitteln bzw. Elementen notwendig ist. Abbildung 4.17 zeigt dies zusammenfassend noch einmal auf. Farblich-codiert ist, welche Information, in welchem Teilmodell explizit beschrieben ist. Wie auf der funktionalen Ebene können diese Modelle ebenfalls in eine statische und eine verhaltensorientierte Lösungsprinzipbeschreibung eingeteilt werden. Der „statische“ Teil ist durch die Modelle der Elektrik und Mechanik gegeben, den „dynamischen“ Teil steuert das Impulsdiagramm bei. Dabei ist in dieser Phase bereits eine Simulationsunterstützung vorzusehen.

Zu berücksichtigen ist hier, dass diese Ausführungen für jedes logische Objekte bzw. Modul durchzuführen sind, solange diese neu entworfen werden müssen.

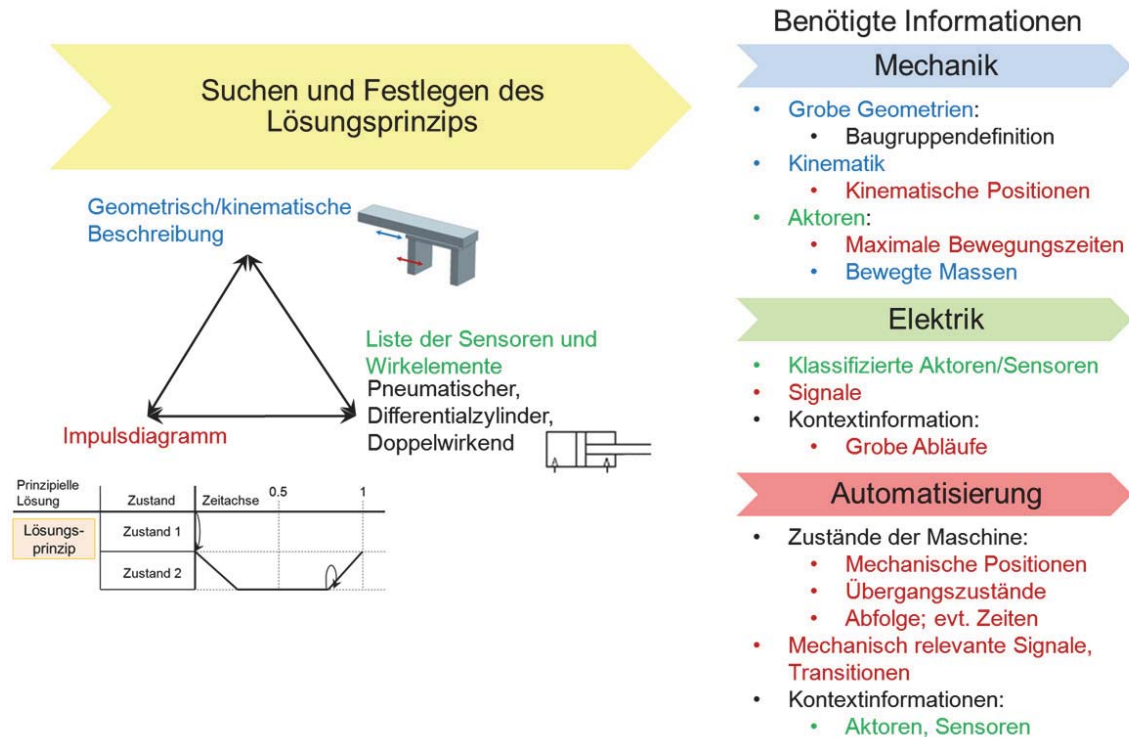


Abbildung 4.17: Mechatronische Beschreibung eines Lösungsprinzips zur parallelen Detaillierung

4.3.5 Mechatronische Beschreibung des Gesamtsystems

Die vorigen Kapitel zeigen, dass zunächst eine funktionale Beschreibung des Gesamtsystems, bestehend aus funktionaler Dekomposition, Verknüpfung und der Verhaltensmodellierung im Gantt-Chart durchgeführt wird. Im Anschluss daran werden Lösungsprinzipien gesucht, welche sich auf der logischen Ebene befinden. Die Verknüpfung von Funktion und Lösungsprinzip definiert ein Modul. Ein Modul enthält eine mechanische Beschreibung in Form von Geometrie und Kinematik, eine elektrische Beschreibung mit definierten Schaltzeichen für Sensoren und Wirkelemente sowie ein Impulsdiagramm für die Ablaufbeschreibung. Abhängig davon, ob ein Modul wiederverwendet wird, sind diese Informationen mehr oder weniger detailliert vorhanden. Neben diesen Informationen werden die Relationen zwischen den Funktionen, auf die logische Ebene heruntergebrochen und weiter spezifiziert. Dies erlaubt die Beziehungen zwischen Modulen zu detaillieren und somit deren Schnittstellen im Gesamtzusammenhang der Maschine zu beschreiben. Außerdem

ist es möglich die integrationsrelevanten Anforderungen auf einzelne Module herunterzubrechen. Dieser Zusammenhang ist in Abbildung 4.18 dargestellt.

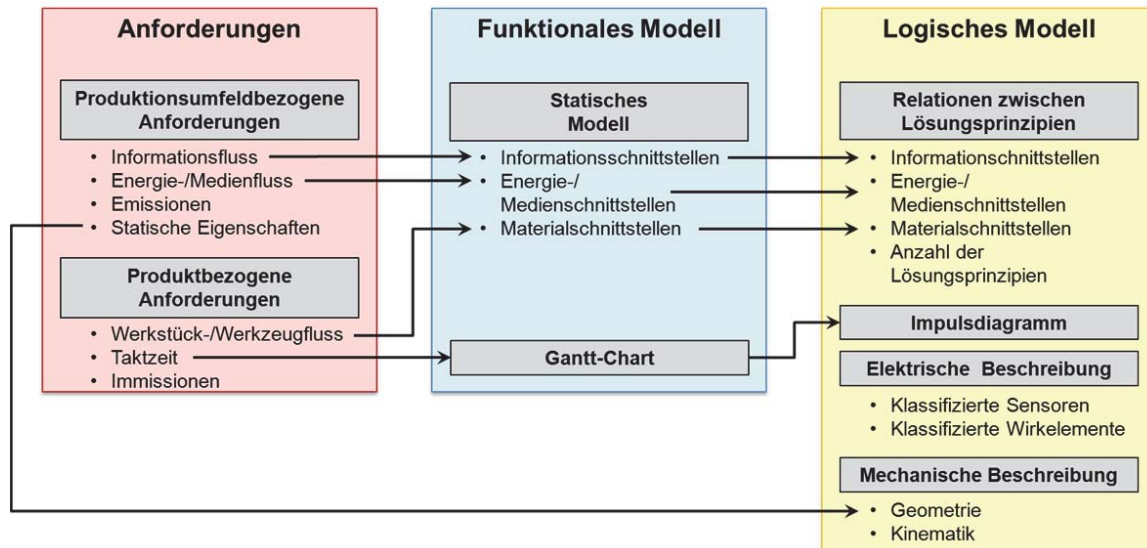


Abbildung 4.18: Herunterbrechen der integrationsrelevanten Anforderungen auf einzelne Maschinenmodule

Das gesamte mechatronische Modell der Maschine am Ende des Systementwurfs zeigt Abbildung 4.19. Es sind zwei grundlegende Bereiche zu unterscheiden: moduldefinierende und modulinteragierende. Die interagierende Beschreibung dient hierbei darzustellen, wie die Verknüpfungen der Module zu einem Gesamtsystem definiert sind. Diese sind als äußerst projektspezifisch zu betrachten und entsprechend auf einer hohen Abstraktionsebene definiert. Die moduldefinierenden Bereiche, wie der Inhalt wiederverwendbarer Module, besitzt eine große innere Abhängigkeit. Diese können aufgrund der äußerst spezifischen Anforderungen im Sondermaschinenbau ebenfalls projektspezifisch sein, sind allerdings projektübergreifend definiert. Im Idealfall tritt der in Tabelle 4.3 dargestellte Fall ein, welcher exakt passende Schnittstellen von Modulen beschreibt, so dass Relationen der interagierenden Ebene zur automatischen Verknüpfung der Module genutzt werden können. Ist dies nicht der Fall, sind diese Verbindungen zu detaillieren und entweder einem Modul zuzuordnen oder projektspezifisch zu belassen.

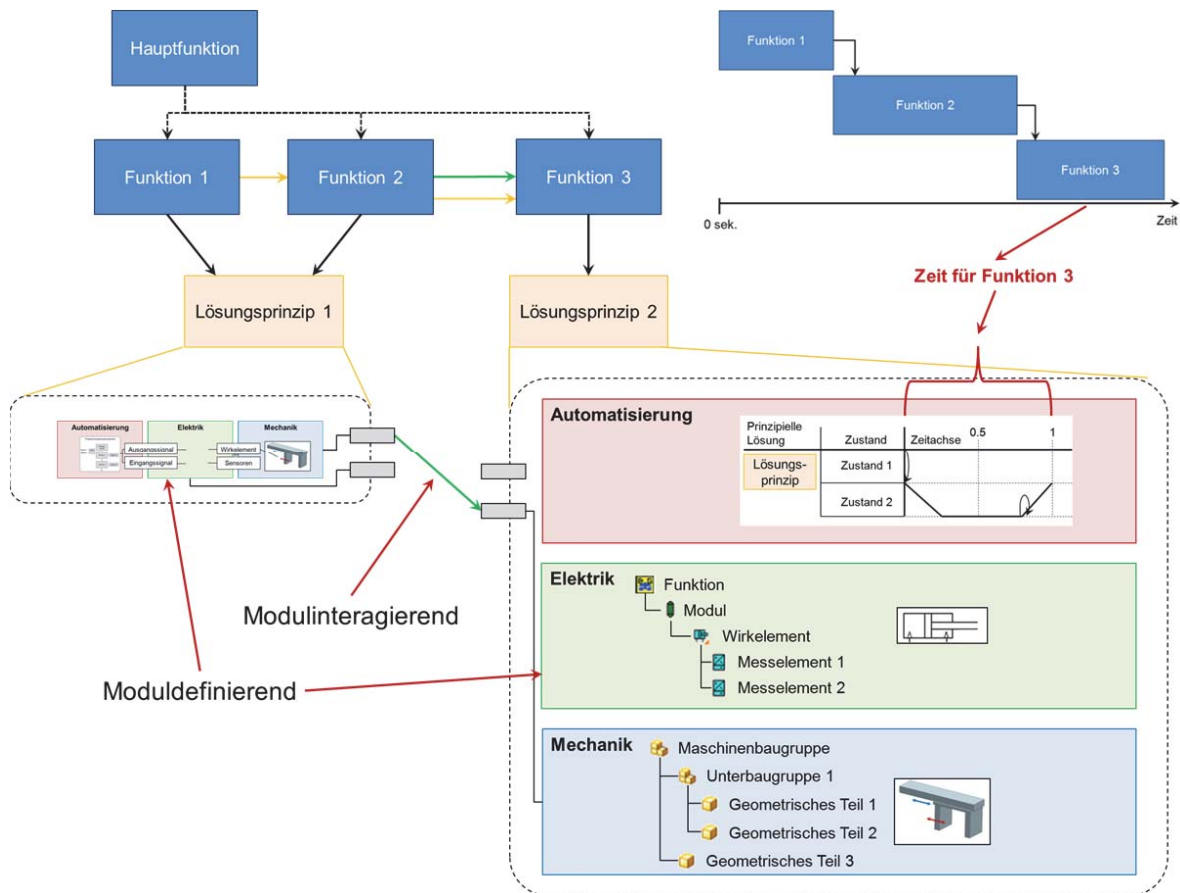


Abbildung 4.19: Mechatronisches Modell der gesamten Maschine am Ende des Systementwurfs

4.4 Parallele Detaillierung

Im vorigen Kapitel ist dargelegt, wie die mechatronische Systembeschreibung definiert sein muss, um parallel in allen Disziplinen die weitere Entwicklung voranzutreiben. Weiterhin ist in Kapitel 3.4.3 aufgezeigt, welche Abhängigkeiten während der Detaillierung auftreten und welche Ergebnisse am Ende dieser Phase vorliegen müssen. Zusammenfassend ist dies in Abbildung 4.20 dargestellt. Das weitere disziplinspezifische Vorgehen unterscheidet sich nicht von den in Kapitel 2.1 vorgestellten disziplinspezifischen Prozessen. Abbildung 4.21 zeigt, welche Tätigkeiten bereits im Rahmen des Systementwurfs abgehandelt wurden und welche in den einzelnen Disziplinen folgen. Dabei ist nur die Abfolge der Tätigkeiten, nicht allerdings deren zeitliche Dauer dargestellt. Zu erkennen ist in Abbildung 4.21, dass in allen Disziplinen die Modelle bereits soweit durch den Systementwurf beschrieben sind, dass diese in den nächsten Phasen direkt die Umsetzung ihrer zugeschriebenen Aufgabe vorantreiben können.

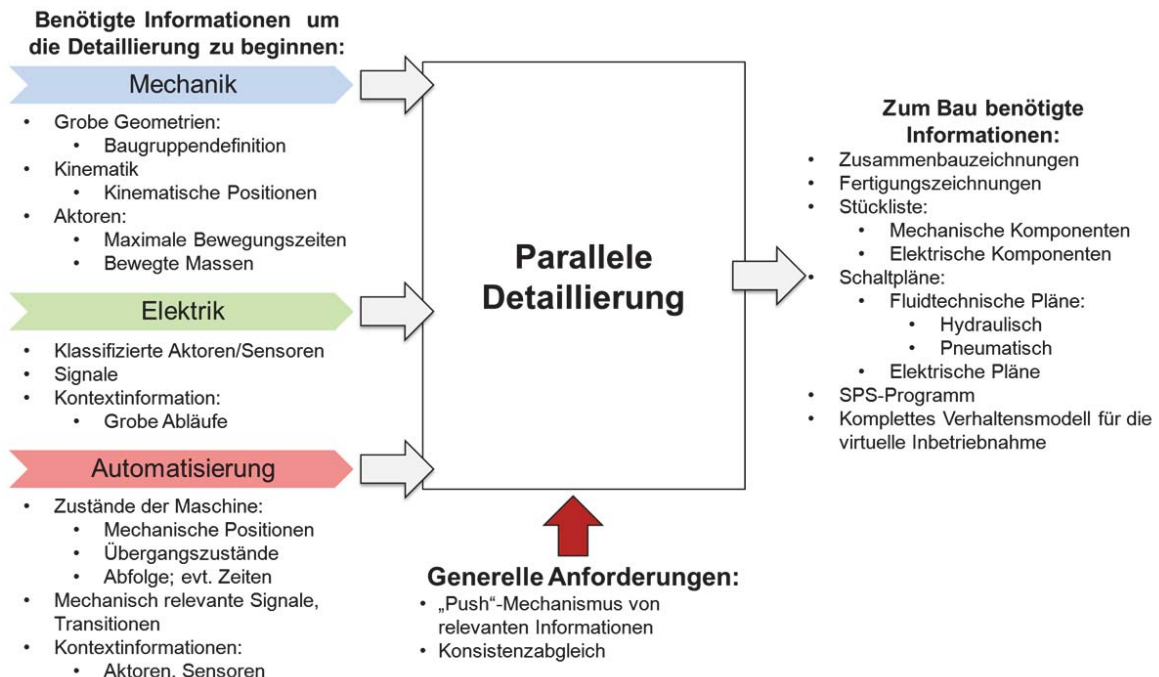


Abbildung 4.20: Anforderungen an die Phase der parallelen disziplinspezifischen Detaillierung

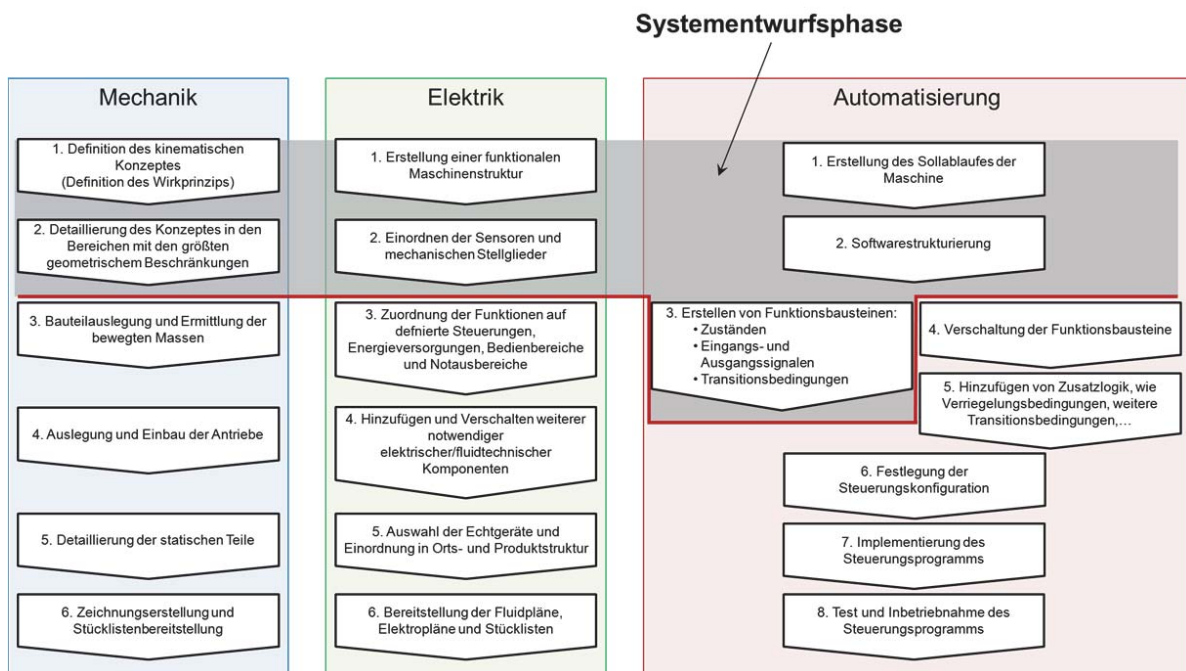


Abbildung 4.21: Aus der Systementwurfsphase definierter Startpunkt in den einzelnen disziplinspezifischen Vorgehen

4.4.1 Simultane Ausdetaillierung

Wie in Kapitel 4.3.5 dargelegt, sind bereits erste Geometrie- und Baugruppendefinitionen für die Mechanik vorhanden. Die Disziplin der Elektrik kann auf einer bestehenden Maschinenstruktur aufbauen. Die Detaillierung in beiden Disziplinen (vgl. Abbildung 4.21) setzt genau auf diese Beschreibungsmittel.

Im Rahmen der SPS-Programmierung verläuft die weitere Detaillierung aber nicht mit Impulsdiagrammen, sondern in Sprachen der *DIN EN 61131 Blatt 3*. Deshalb ist es erforderlich von einem solchen Impulsdiagramm zu einer der normierten Sprachen zu wechseln. Dies kann durch eine automatische Transformation direkt in SFC's ermöglicht werden. [vgl. Drath 2010; Hundt 2012] Darauf aufbauend kann der Steuerungsprogrammierer die Detaillierung in den gewohnten Beschreibungsmitteln durchführen.

Um einen möglichst effizienten Prozess zu gewährleisten, muss jede Information dort einzugeben sein, wo sie entsteht bzw. definiert wird. Das führt dazu, dass sich unterschiedliche Module in unterschiedlichen Phasen der disziplinspezifischen Detaillierung befinden (vgl. Abbildung 4.21). Eine wirklich parallele Arbeitsweise der Disziplinen erfordert folglich, dass Informationen, welche sowohl in der Mechanik und der Elektrik modellierbar sind, auch in beiden Bereichen definierbar sein müssen. Dabei sind sequenzielle Abhängigkeiten zu berücksichtigen. Ein Antrieb muss beispielsweise ausgehend von den bewegten Massen ausgelegt werden bevor dessen Artikelnummer bekannt ist. Allerdings entstehen manche interdisziplinären Informationen aus unterschiedlichen Blickwinkeln. Gerade das Hinzufügen von Signalen geschieht in der Elektrik durch Instanziierung eines Sensors. Die Automatisierung definiert ein solches in der Symboltabelle. Dabei modelliert sie keine Komponente bzw. keinen Sensor. Es entsteht parallel in beiden Modellen ein Signal, welches allerdings im Projektkontext dasselbe ist. Dies ist in Abbildung 4.22 veranschaulicht. Daraus ergibt sich, dass auch nachträglich Objekte als „dieselben“ deklariert werden müssen, um der interdisziplinären Konsistenz gerecht zu werden. Aufgrund dessen werden keine expliziten sequenziellen Abhängigkeiten zwischen den disziplinspezifischen Phasen aus Abbildung 4.21 definiert. Weiterhin wird in Kapitel 3.4.3 dargestellt, dass die interdisziplinären Informationen konsistent zu halten sind. Werden Informationen hinzugefügt, welche als relevant für andere Disziplinen eingestuft sind, müssen diese unmittelbar kommuniziert werden.

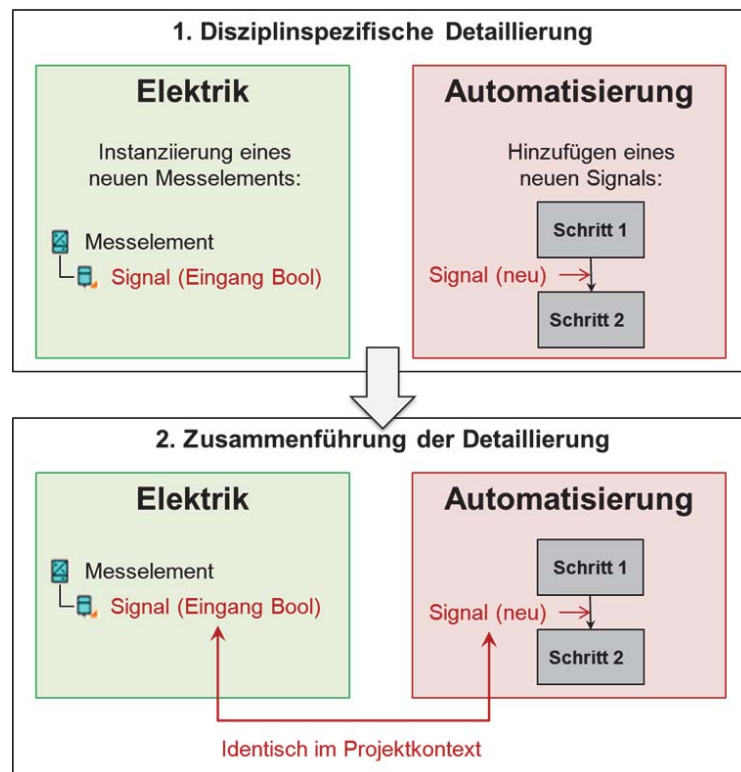


Abbildung 4.22: Entstehung derselben Informationen in unterschiedlichen Kontexten

Außerdem ist deren direkte Weiterverwendung vorzusehen. Änderungen solcher Informationen können, wie in Kapitel 3.4.1 dargelegt, Auswirkungen auf unterschiedliche Modellierungsobjekte in anderen Sichtweisen besitzen, müssen aber nicht. Beispielsweise sei hier eine Änderung der Kabellänge angeführt, welche ein Änderungsereignis auslöst, aber sich immer noch unterhalb der kritischen Grenze bewegt. Im Falle, dass keine automatische Überprüfung dieses Grenzwertes seitens des ECAD-Systems vorgenommen wird, muss der Konstrukteur diese Information auch ohne Modellierungseingriff aufnehmen, überprüfen und den Änderungshinweis löschen können.

Beim Löschen von mechatronisch relevanten Informationen kann es zu Auswirkungen in vielen anderen Bereichen kommen. Als Beispiel sei das Löschen eines Sensors im mechanischen Modell genannt, wodurch auch der Sensor in der Mechanik, als auch alle Signale des Sensors betroffen sind. Folglich sind bei Löschoptionen nicht nur das Objekt selbst, sondern alle mechatronischen Auswirkungen zu kommunizieren.

Am Ende dieser Phase der simultanen Ausdetaillierung müssen alle Unterlagen zum Bau der Maschine zur Verfügung stehen. Durch die Konsistenz der Informationen ist eine aufwandsarme

Bereitstellung jener zu ermöglichen, beispielsweise die Erstellung einer mechatronischen Stückliste um Nichtberücksichtigungen oder Doppelbestellungen von Komponenten zu vermeiden.

4.4.2 Wiederverwendung mechatronischer Komponentenbeschreibungen

Kapitel 4.3.4 zeigt auf, wie neue Module am Ende der Phase des Systementwurfs zu beschreiben sind. Weiterhin ist in Kapitel 4.3.3 dargelegt, dass die Benennung eines Lösungsprinzips gleichzeitig eine Klassifizierung verschiedener Alternativen darstellt. Selbiges kann auf mechatronische Komponenten übertragen werden. Ein Servomotor der Elektrik kann ebenfalls als Klassifikation verschiedener Alternativen angesehen werden. Dieser besitzt sowohl in der Mechanik, mit einer definierten Geometrie, als auch in der Elektrik mit Symbol und Anschlüssen entsprechende Modelle. Daraus ist ersichtlich, dass neben Modulen auch Komponenten mechatronisch beschrieben werden können und müssen. Durch eine Definition einer mechatronischen Komponente auf der logischen Ebene (wie Klasse Servomotor) schränkt sich der Raum, welche physikalischen Komponenten eingesetzt werden können weiter ein. So schließt die Definition eines Servomotors den Einsatz eines Pneumatikzylinders aus. (vgl.: Abbildung 4.9) Es ist ersichtlich, dass mögliche Alternativen solcher Komponenten ebenso wiederverwendet werden müssen, wie die mechatronischen Module aus Kapitel 4.3.3. Durch Nutzung der logischen Beschreibung als Klassifizierung wird gleichzeitig der Suchaufwand eingeschränkt.

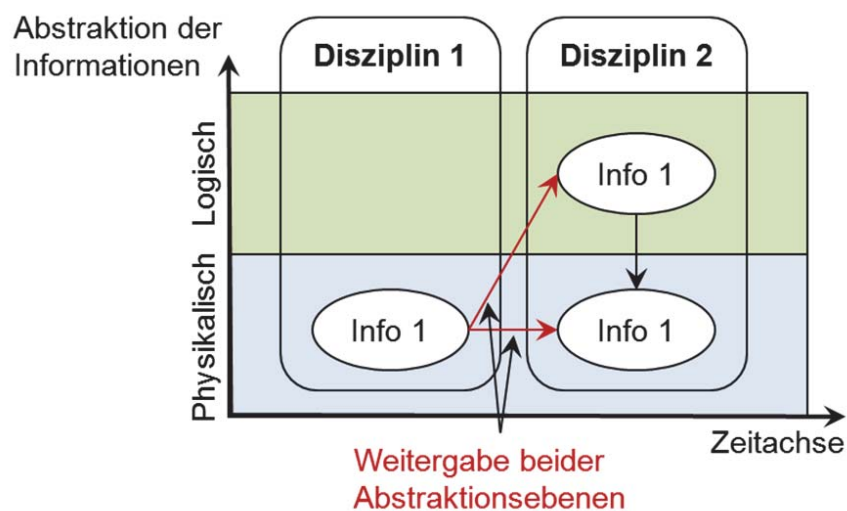


Abbildung 4.23: Aufwandsreduktion durch Weitergabe von abstrahierten Informationen

Während der parallelen Detaillierungsphase kann im Rahmen der Wiederverwendung ein weiterer Fall auftreten. Während die bisherige Wiederverwendung über die Klassifizierung von Alternativen

Top-Down von statten geht, kann es durch eine Definition physikalischer Komponenten geschehen, dass Bottom-Up logische Modelle erzeugt werden müssen. Durch die Bestellnummer sind alle Sichtweisen definiert. Der Einbau eines mechatronisch vollständig beschriebenen Servomotors auf der physikalischen Ebene enthält neben der Bestellnummer auch das Symbol der Elektrik. Folglich muss, wie in Abbildung 4.23 dargestellt, auch die logische Repräsentanz in der anderen Sichtweise bei Wiederverwendung instanziiert werden.

4.5 Virtuelle Systemintegration

Kapitel 4.4.1 zeigt, dass die verschiedenen Disziplinen bereits Tätigkeiten der Systemintegration während der parallelen Phase durchführen. Beispielsweise fügt die Mechanik verschiedene Baugruppen dem Gesamtprojekt hinzu. Selbiges geschieht durch die Verkabelung und das Hinzufügen weiterer Logik im Rahmen der Automatisierung. Durch den Konsistenzabgleich ist es möglich einen großen Teil der Systemintegration direkt während der Detaillierung konsistent durchzuführen.

Aufgrund des Unikatcharakters von Sondermaschinen ist eine virtuelle Inbetriebnahme zur Fehlererkennung im Steuerungsprogramm vorzusehen, da kein Prototyp gebaut wird. Dazu ist neben den bereits betrachteten Modellen ein Verhaltensmodell der Maschine notwendig. Dieses muss alle vom SPS-Programm benötigten Signale bedienen, um einen Durchlauf und somit den Test des Programms zu ermöglichen. Der Aufwand zur Modellierung des Verhaltens muss sich aber in Grenzen halten. Das Verhalten der Maschine ist sowohl aus mechanischer als auch der elektrischen Sicht zu beschreiben, da sich manche Signale nach Kapitel 3.3 auf Zustände der elektrotechnischen Geräte beziehen. Der Aufbau von Sondermaschinen zeigt, dass die Elektrik auf Zulieferteile zurückgreift, während die Kinematik bzw. mechanische Ausprägung bei jedem Projekt neu definiert werden muss. Weiterhin sind in der Elektrik alle Signale in den Schaltplänen modelliert. Abbildung 4.24 zeigt diese Situation auf. Daraus kann abgeleitet werden, dass das erforderliche Verhaltensmodell auf Basis der elektrischen Schaltpläne und der geometrisch kinematischen Beschreibung bereitzustellen ist. Dadurch ist es möglich die Verantwortung einer korrekten Verhaltensmodellierung aus organisatorischer Sicht den konstruierenden Disziplinen zuzuordnen. Nur diese besitzen das entsprechende Wissen, um das Verhalten korrekt zu beschreiben. Der Test des aktuellen Steuerungsprogramms muss zu jedem Zeitpunkt möglich sein. Dabei ist zu berücksichtigen, dass evtl. einige im Steuerungscode benötigten Signale nicht modelliert sind. Andersherum können aber auch Signale bereitgestellt werden, welche im SPS-Programm „noch“ nicht genutzt werden. Aufgrund des Konsistenzabgleichs während der parallelen

Detaillierung sind beide Fälle vor dem Steuerungstest ersichtlich. Weiterhin ist es notwendig, Fehler während der Simulation bewusst herbeizuführen um die richtige Reaktion der Software zu testen.

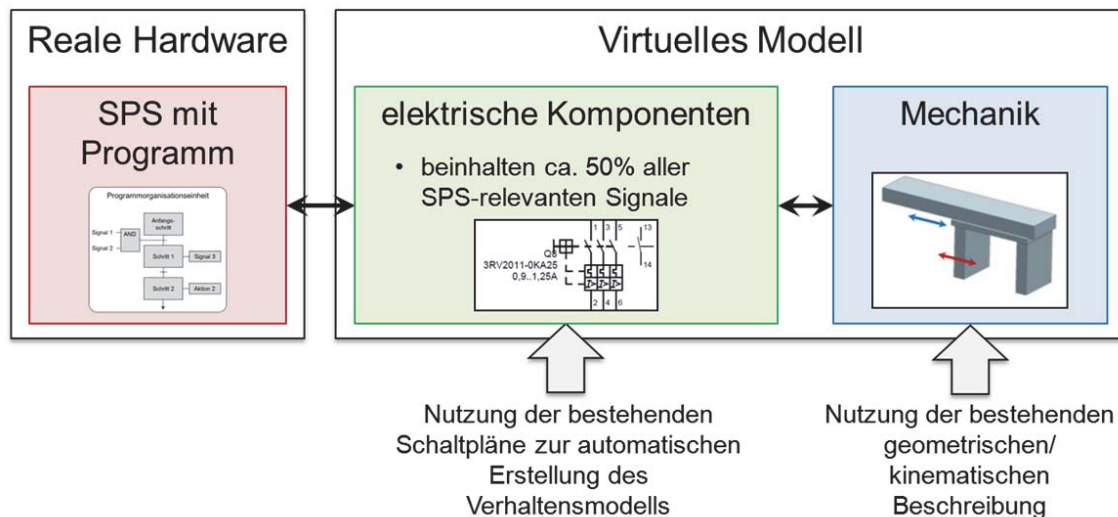


Abbildung 4.24: Modellerstellung zur virtuellen Inbetriebnahme durch Nutzung der Schaltpläne und des kinematisierten mechanischen Modells.

Der Test des Steuerungsprogramms kann sowohl als Software- als auch Hardware-in-the-loop-Simulation ausgeführt werden.

4.6 Ausschneiden und Ablegen mechatronischer Objekte

Die Kapitel 4.3.3 und 4.4.2 zeigen die Wiederverwendung mechatronischer Objekte auf. Es ist ersichtlich, dass mechatronische Objekte mehrere disziplinspezifische Sichtweisen enthalten können. Generisch betrachtet, muss mindestens eine Sichtweise ein Modell für ein mechatronisches Objekt aufweisen. Allerdings kann aus praktischen Gesichtspunkten die Aussage getroffen werden, dass jedes elektrische Gerät als mechatronisches Objekt angesehen werden kann. Dabei ist die logische Modellierung als Klassifikation zu betrachten. Die geometrische Beschreibung ist gleichzeitig die mechanische Repräsentanz. Wie bereits einleitend dargestellt, ist eine Definition dieser Objekte vor Beginn des Projektes nicht möglich. Deshalb muss der Aufwand zur Definition derer minimal sein. Aufgrund der Forderungen aus Kapitel 3.4 ist außerdem ersichtlich, dass unterschiedliche Systemgrenzen und Abstraktionsniveaus zur Definition eines mechatronischen Objekts benötigt werden. Dies kann am Ende des Projektes oder währenddessen geschehen. Zur

Erstellung müssen eine Klassifikation des Objektes und die entsprechenden Sichtweisen gemeinsam abgelegt werden.

4.7 Zusammenfassung

Abbildung 4.25 zeigt zusammenfassend alle Phasen und die hierfür definierten Beschreibungsmittel auf. Dabei ist zu beachten, dass die Phase der Systemintegration in diesem Zusammenhang nur die Aufgabe zur Verifikation des SPS-Codes besitzt, was als virtuelle Inbetriebnahme bezeichnet wird. Dies ist auf die disziplinspezifische Integration während der simultanen Detaillierung zurückzuführen. Weiterhin ist es für den Sondermaschinenbau essenziell, nicht Top-Down alle Module mechatronisch korrekt zu beschreiben, sondern aus kleinen mechatronischen Einheiten schnell und effizient zum Ziel zu kommen. Deshalb kann nicht von einer reinen vorwärtsgerichteten Durchführung ausgegangen werden. Es ist viel mehr gerade im Kleinen ein Springen zwischen Abstraktionsebenen. Dabei grenzen höhere Abstraktionsebenen den Lösungsraum für niedrigere Ebenen entsprechend ein.

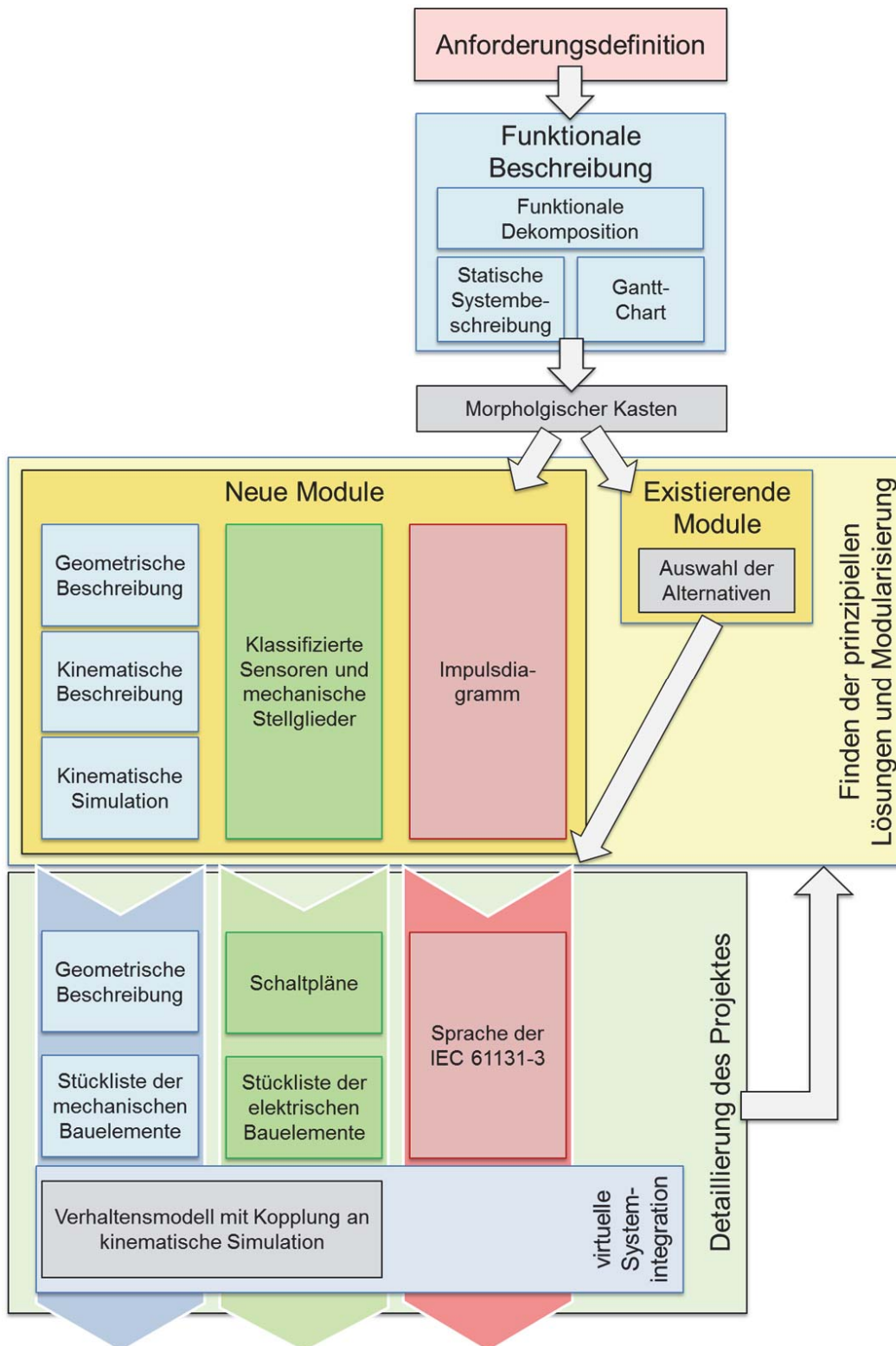


Abbildung 4.25: Phasen und ihre Beschreibungsmittel der vorgestellten Methode

5 Softwaretechnische Unterstützung

In Kapitel 3 sind die Abhängigkeiten zwischen den Disziplinen dargestellt. Darauf aufbauend zeigt Kapitel 4 die Methode zur mechatronischen Sondermaschinenentwicklung auf. Beide stellen Anforderungen an eine durchgängige softwaretechnische Unterstützung. Im Folgenden wird zunächst die informationstechnische Grundlage zur Integration von Softwarewerkzeugen gelegt, bevor diese Anforderungen in die Architektur mechatronischer Objekte überführt werden. Darauf aufbauend werden die benötigten Knoten- und Kantentypen aufgezeigt, welche bei der Umsetzung in den darauffolgenden Teilkapiteln benötigt werden.

5.1 Softwareintegration auf Basis semantischer Netze

Wie bereits ausgeführt, sind nicht nur die Daten an sich, sondern auch deren Bedeutung im interdisziplinären Kontext für eine integrierte Unterstützung des mechatronischen Entwicklungsprozesses wichtig (vgl. Abbildung 3.6). Weiterhin wird aufgezeigt, dass nicht alle Daten, beispielsweise Symbole der Elektrik, in den anderen Disziplinen relevant sind und übertragen werden müssen. Daraus kann Abbildung 5.1 abgeleitet werden. Folglich sind für die Interoperabilität zwischen verschiedenen Applikationen nur die Informationen zu berücksichtigen, welche sich im interdisziplinären Bereich befinden. Die übrigen Informationen können als privat aus Sicht der einzelnen Applikationen betrachtet werden.

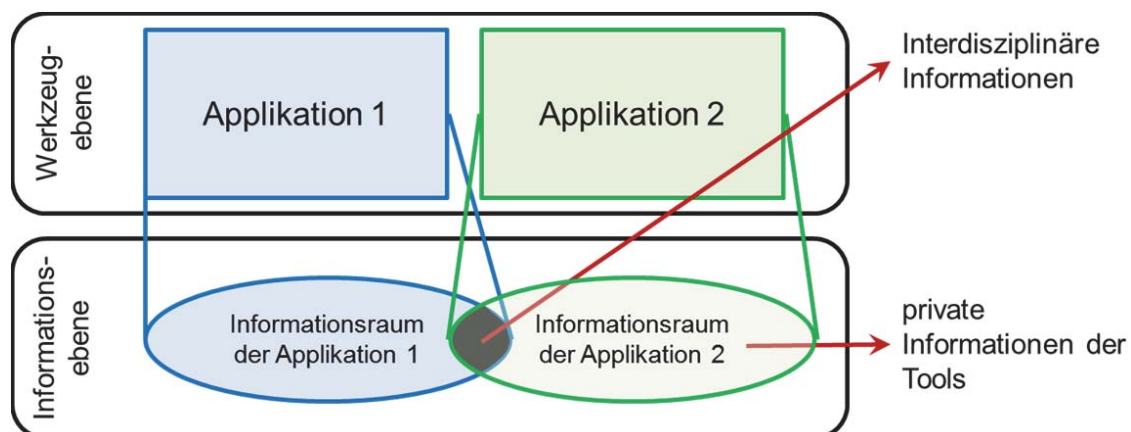


Abbildung 5.1: Interdisziplinäre und proprietäre Informationen von Softwarewerkzeugen [vgl. Frank, et al. 2011]

Abbildung 5.2 zeigt die Basisarchitektur, worin zu sehen ist, dass die proprietäre Datenhaltung der Applikationen beibehalten wird. Somit ist gewährleistet, dass jedes Softwarewerkzeug seiner Funktion entsprechend, Daten speichern und halten kann. Lediglich die Informationen, welche sich im interdisziplinär relevanten Bereich befinden, werden in das semantische Netz geschrieben. Eine Nachverfolgung zum konkreten proprietären Objekt der Softwarewerkzeuge aus dem semantischen Netz heraus wird mit Pointern in die proprietäre Datenhaltung sichergestellt.

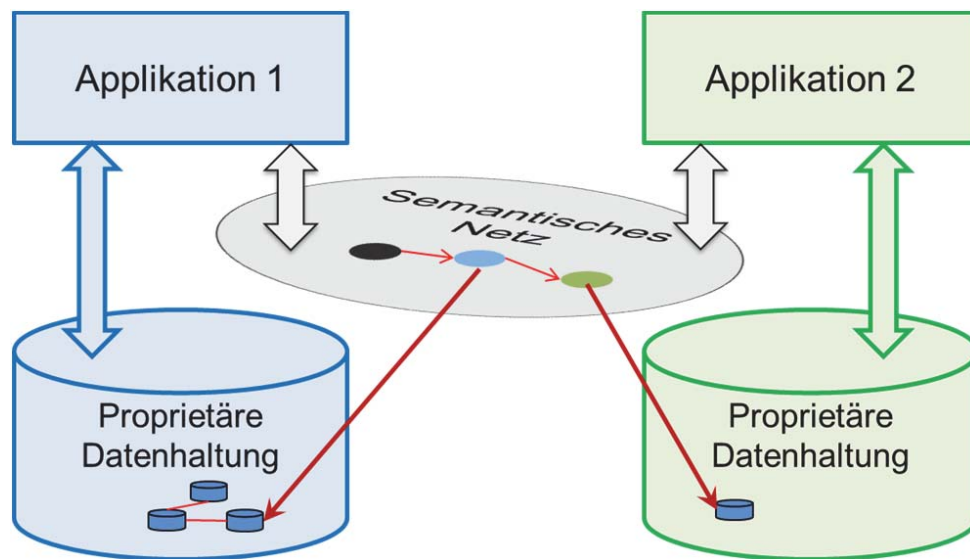


Abbildung 5.2: Grundarchitektur zur Interoperabilität zweier Applikationen auf Basis eines semantischen Netzes

Für jedes Entwicklungsprojekt wird ein neues semantisches Netz erstellt. Ein solches ist schematisch in Abbildung 5.3 zu sehen. Dabei ist zu beachten, dass nicht jede Applikation alle Knoten- und Kantentypen des semantischen Netzes interpretieren muss und kann. Ein Softwarewerkzeug interpretiert beispielsweise nur Knoten des roten und orangefarbenen Typs, sowie die rot eingefärbten Kanten. Wird nun durch einen Anwender ein Modell in einer Applikation erstellt, werden interdisziplinär relevante Informationen im Netz veröffentlicht. Dies kann beispielhaft durch die Knoten mit den Nummern eins, zwei, drei und vier umgesetzt sein. Neben dem Schreiben in das Netz, müssen verschiedene Softwarewerkzeuge auch davon lesen und dieses interpretieren. Nach dem Hinzufügen der genannten Knoten, wäre am Beispiel in Abbildung 5.3 für Applikation zwei zu erkennen, dass zwei orangefarbene Knoten hinzugefügt wurden. Diese sind als relevant für das entsprechende Softwarewerkzeug eingestuft. Im Weiteren muss es aus dieser Sichtweise heraus möglich sein, die Information der Knoten weiter zu nutzen, was durch ein Ziehen

der Kanten eins und zwei umgesetzt sein kann. Ein Hinzufügen dieser Kanten, könnte am Ende beispielsweise für den User ein weiteres Objekt in seinem applikationsspezifischen Projektbaum entstehen lassen. Diese Hierarchie stellt dadurch eine Sichtweise bzw. einen View des Netzes dar. Dadurch muss nicht das gesamte semantische Netz dem User visualisiert werden. Ein solches Beispiel stellt der blaue Baum in Abbildung 5.3 dar, welcher durch die Interpretation von Applikation eins über dem semantischen Netz entsteht. Die netzartige Struktur führt dazu, dass es prinzipiell irrelevant ist, ob zunächst der blaue Weg von Knoten eins bis zu Knoten vier oder entsprechend der rote eingeschlagen wird. Dies ist speziell im Falle der Signale (vgl. Kapitel 4.4.1) zu berücksichtigen, welche aus unterschiedlichen Kontexten heraus im Rahmen einer wirklich parallelen Detaillierungsphase entstehen können. Neben dem Fall des Hinzufügens von Informationen, müssen während eines Entwicklungsprojektes, solche auch wieder gelöscht werden können.

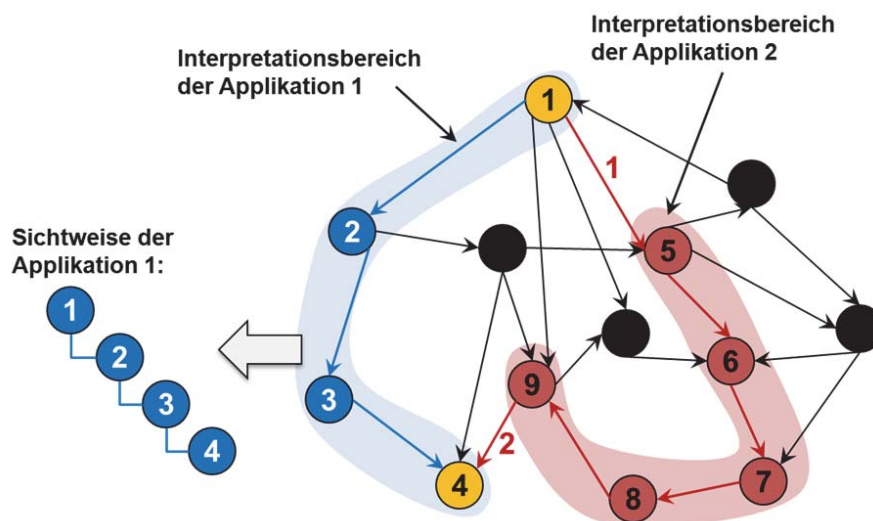
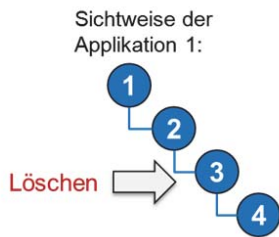


Abbildung 5.3: Interpretationsbereiche von Applikationen auf dem semantischen Netz

Abbildung 5.4 veranschaulicht einen Löschvorgang. Im applikationsspezifischen blauen Baum, wird der blaue Knoten 3 gelöscht. Üblicherweise entfällt dadurch auch der blaue Knoten vier in diesem Kontext. Gleichzeitig wird dieser allerdings auch im Rahmen des roten Softwarewerkzeugs verwendet. Dieses Löschen muss dem Anwender des roten Softwarewerkzeugs mitgeteilt werden, damit dieser entsprechende Maßnahmen einleiten kann. Äquivalentes gilt bei Änderungen an Knoten. Besitzen Kanten als Quelle und Ziel unterschiedliche Interpretationsbereiche, ist das Löschen oder Ändern sowohl an den Quell- als auch den Zielknoten entsprechend zu verfolgen.

Solche Konsistenznachverfolgungen sind in vielen Fällen zunächst einmal durch Userzustimmung umzusetzen. Prinzipiell ist eine automatische Ableitung der Maßnahmen sowohl bei Änderungen, als auch durch automatische Interpretation denkbar.

1. Löschen eines Knotens



2. Auswirkungen auf andere Applikationen

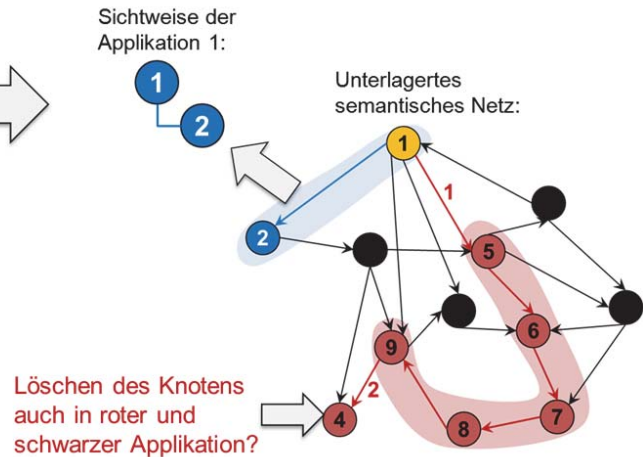


Abbildung 5.4: Löschen von Knoten in semantischen Netzen und die Auswirkung auf andere Applikationen

5.2 Architektur mechatronischer Objekte zur Wiederverwendung

Eine Interpretation der Wiederverwendung nach Kapitel 4.3.3 führt dazu, dass ein mechatronisches Objekt (MO) nicht unbedingt auf der physikalischen Ebene definiert sein muss. Nach Abbildung 4.12 ist das logische Objekt gleichzeitig die Klassifizierung verschiedener Alternativen und bildet damit bereits eine erste Information über ein MO. Ein solches Objekt enthält Modelle aus mindestens einer Sichtweise, ohne die Anzahl theoretisch zu begrenzen. Im Rahmen dieser Arbeit sind praktisch gesehen zwei oder drei Disziplinen zu berücksichtigen.

Das mechatronische Objekt selbst, kennt die ihm zugehörigem Teilmodelle, was in Abbildung 5.5 über den Relationstyp „gehört zu“ ausgedrückt ist. Die Ansätze aus Kapitel 2.5.2 nutzen die bis hier beschriebenen Relationstypen für die mechatronische Objektdefinition. Allerdings sind gerade die Verbindungen zwischen den Modellen (vgl. u.a. Abbildung 3.6) aber essenziell. Dadurch kann eine schnelle Einpassung in das neue Projekt gewährleistet und Änderungen bzw. Löschungen nachvollzogen werden, um ein konsistentes Projekt zu erhalten. Außerdem ist eine nahtlose Weiterverwendung von Informationen direkt möglich. Das mechatronische Objekt muss folglich die Relationen zwischen den Modellen kennen, was durch die roten Pfeile dargestellt ist. Diese sind

ebenfalls im semantischen Netz gespeichert. Daraus kann abgeleitet werden, dass ein mechatronisches Objekt im Wesentlichen einen Ausschnitt eines semantischen Netzes zuzüglich der privat zugewiesenen Informationen der Applikationen darstellt. In Abbildung 5.5 wird kein expliziter Unterschied bzgl. der interdisziplinären Informationen und den privaten auf den disziplinspezifischen Facetten dargestellt. Lediglich die roten Pfeile geben einen Hinweis auf interdisziplinäre Informationen. Jede Disziplin bzw. Applikation repräsentiert dadurch mit seinem Modell und dem zugehörigen Ausschnitt des semantischen Netzes seine eigene Facette im mechatronischen Objekt. Das mechatronische Objekt an sich kennt somit alle Facetten, als auch die Relationen zwischen diesen. Weiterhin ist die Klassifikation als logisches Objekt oder Lösungsprinzip definiert. Diese kann im Falle einer physikalischen Direktinstanziierung zur Instanziierung der logischen Ebene herangezogen werden. (vgl. Kapitel 4.4.2)

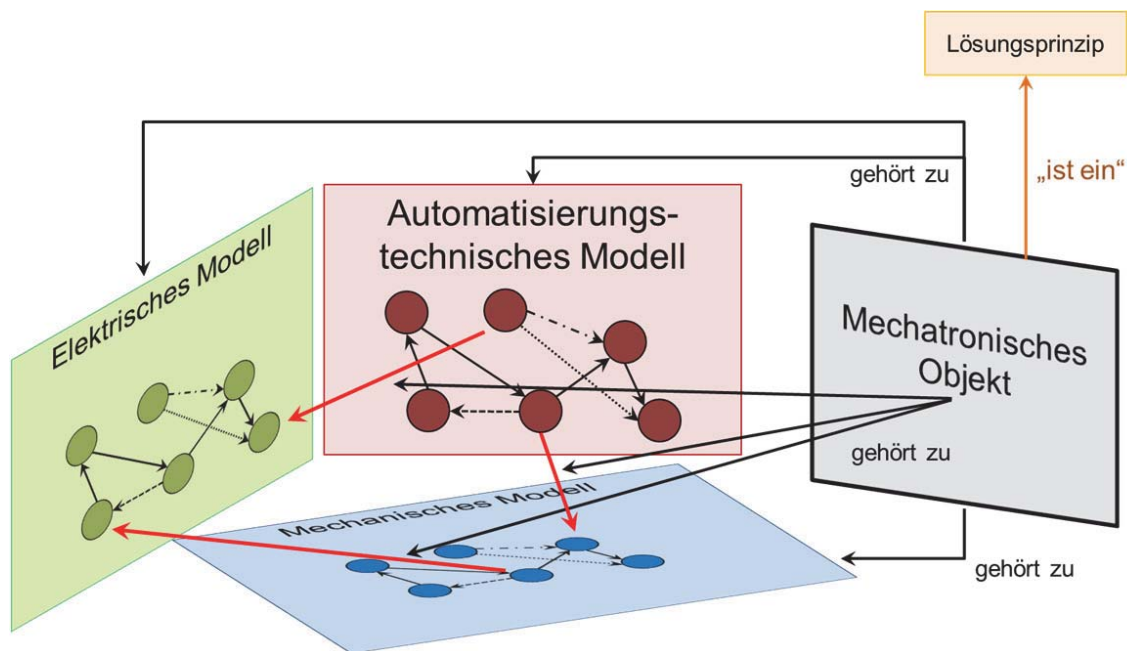


Abbildung 5.5: Architektur des mechatronischen Objektes

Die Eigenschaft mechatronischer Objekte, semantische Netze zu verkörpern führt dazu, dass eine Definition dieser Objekte durch Ausschneiden aus einem zugrundeliegenden Projekt erfolgen kann. Dazu fügt jede Disziplin jene Modellbestandteile hinzu, welche sie als wiederverwendbar in weiteren Projekten unter dem genannten logischen Objekt identifiziert. Handlungsempfehlungen hierzu zeigt Kapitel 4.3.3 auf. Dabei besteht die Schwierigkeit in der Bestimmung der Schnittgrenze, da durch Hinzufügen eines Knotens auch immer alle zugehörigen Kanten und die

damit verbundenen Knoten betrachtet werden müssen. Es ist beispielsweise offensichtlich, dass Anschlüsse von elektrischen Geräten bei deren Ausschneiden ebenfalls abzulegen sind. Folglich sind Bedingungen zu definieren, welche angeben, wann ein verbundener Knoten auszuschneiden ist. Abbildung 5.6 zeigt die Betrachtung dieser Abbruchbedingung im interdisziplinären Kontext auf. Danach sind Kanten zwischen den Facetten nur zu berücksichtigen, wenn sowohl das Quell- als auch das Zielobjekt ausgewählt sind.

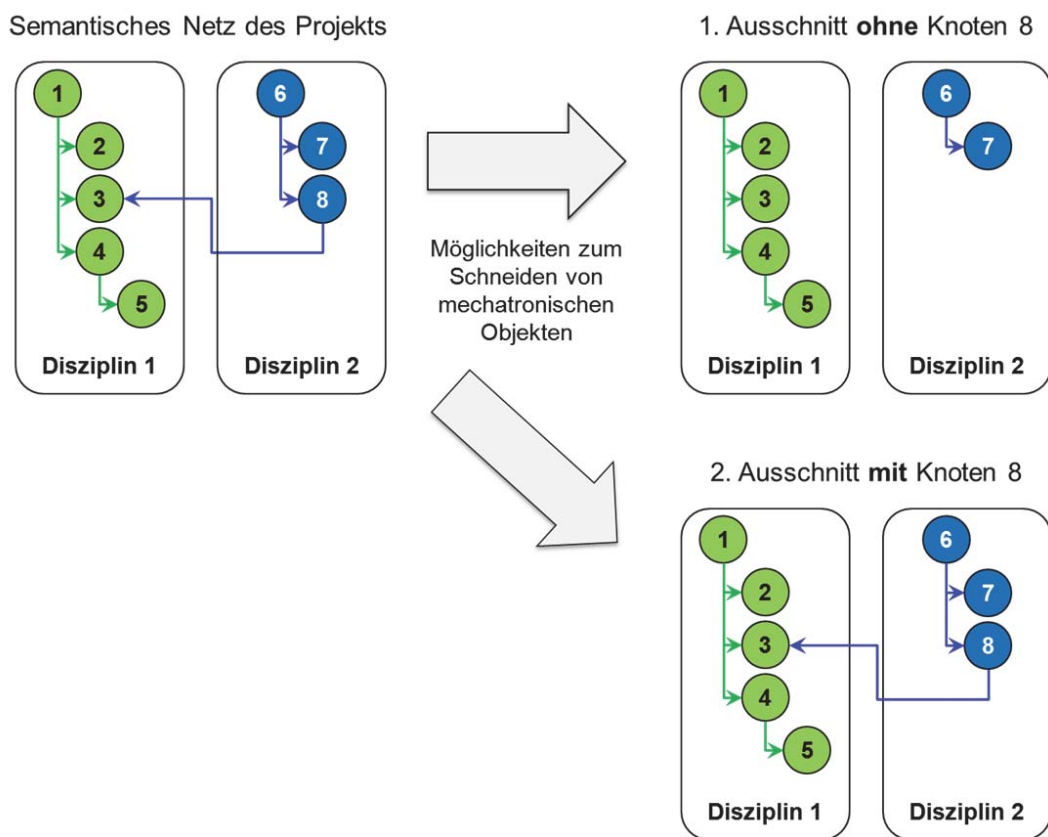


Abbildung 5.6: Schneiden mechatronischer Objekte auf Basis eines bestehenden Projektes

Die Wiederverwendung von mechatronischen Objekten erfolgt durch Wiederinstanzierung der ausgeschnittenen Teilmodelle. Der Einstieg zur Wiederverwendung konkreter Alternativen im Rahmen des Entwicklungsprozesses ist häufig über die Definition eines logischen Objektes gegeben. Dies ist gleichzeitig die Klassifikation des mechatronischen Objektes. Dadurch muss die Softwareunterstützung dem Anwender alle möglichen Alternativen aufzeigen können. Nach Auswahl einer Alternative durch den Nutzer werden alle Knoten automatisch in das semantische Netz instanziiert. Das mitinstanziierte Netz enthält, wie bereits dargestellt, alle (auch die

interdisziplinären) Abhängigkeiten. Dadurch können die in Kapitel 5.1 dargestellten Mechanismen direkt genutzt werden.

5.3 Ermittlung von Knoten- und Kantentypen im interdisziplinären Bereich

Die Ermittlung der Knoten- und Kantentypen auf Basis der in Kapitel 4 vorgestellten Methode ist bis zu einer gewissen Detaillierungstiefe möglich. Dieser Top-Down-Ansatz definiert, welche Typen durch welche Disziplinen behandelt werden müssen. Im Anschluss daran ist es möglich geeignete Werkzeuge auszuwählen. Dabei ist zu beachten, dass alle notwendigen interdisziplinären Informationen darin modelliert werden können. Deshalb muss die Semantik der Daten in den Softwarewerkzeugen extrahiert werden. Schlussendlich erfolgt ein Abgleich mit den Top-Down ermittelten notwendigen disziplinspezifischen und den interdisziplinären Informationen. Dieses Vorgehen ist in Abbildung 5.7 dargestellt.

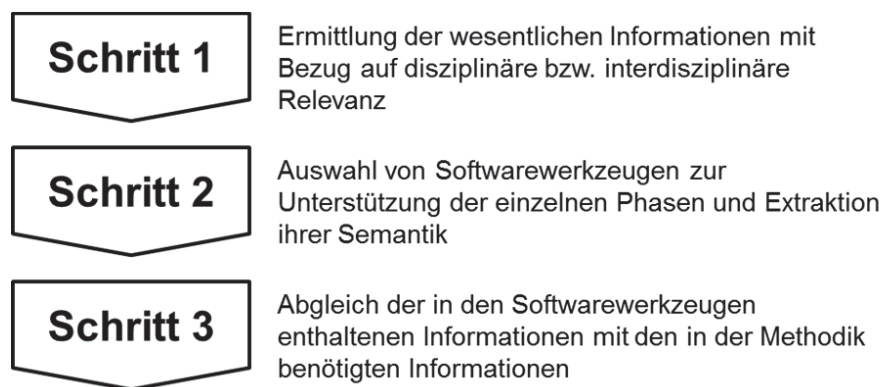


Abbildung 5.7: Vorgehen zur Ermittlung der Knoten- und Kantentypen

Dadurch ist es möglich eventuelle Über- oder Unterdeckungen zwischen Disziplinen und Applikationen zu identifizieren und zu beheben.

5.3.1 Definition der Knoten- und Kantentypen auf Basis der vorgestellten Methode

Die Methode aus Kapitel 4 definiert die wesentlichen Knoten- und Kantentypen. Diese sind in Abbildung 5.8 dargestellt. Manche Knoten- und Kantentypen, wie die Verbindungen der Informations-, Material- oder Energieübergabe zwischen Funktionen sind schon durch die aufgestellte Methodik eindeutig identifizierbar. Die Verhaltensbeschreibung mittels Gantt-Chart ist mit der Kante „erfolgt vor“ angedeutet, welche eine sequenzielle Abhängigkeit zwischen Funktionen darstellt. Die funktionale Dekomposition wird über den Relationstyp „wird erfüllt durch“ abgebildet. Die Restriktion, dass diese Dekomposition eine Hierarchie verkörpert, was bedingt, dass

maximal eine einzige Kante auf die entsprechende Funktion zeigen darf, wird in der Applikation definiert. Allerdings muss gewährleistet sein, dass diese Restriktionen nicht mit anderen Anwendungen kollidieren. Die einfachste Lösung in einem solchen Fall ist die Einführung eines weiteren Kantentyps für das entsprechende Softwarewerkzeug. Die Auswahl und Definition der prinzipiellen Lösung in Bezug auf die Funktionen ist durch den Kantentyp „wird umgesetzt durch“ veranschaulicht. Dies markiert den Übergang von der funktionalen Modellierung zur logischen Ebene in der Phase des Systementwurfs. Darin sind Abläufe durch Impulsdiagramme beschrieben. Diese werden im nächsten Entwicklungsschritt in Sequential Function Charts automatisch transformiert. Der Übersichtlichkeit wegen sind die Impulsdiagramme deshalb nicht explizit aufgeführt. Der interdisziplinär gekennzeichnete Bereich enthält alle Informationen, welche in Kapitel 3.4.3 als Abhängigkeiten zwischen den Disziplinen definiert sind, was auch durch die Überlappung verschiedener Disziplinen sichtbar ist. Dabei ist zu berücksichtigen, dass Informationen in verschiedenen Kontexten entstehen und im Verlauf des Projektes als äquivalent zu deklarieren sind (vgl. Kapitel 4.4.1). Deshalb wird die Kante vom Typ „ist äquivalent“ eingeführt. Informationen, wie beispielsweise Symbole der Elektrik, sind auf Basis der entsprechenden disziplinspezifischen Entwicklungsvorgehen und Modellierungen definiert. Folglich markiert der hellgraue Bereich aus Sicht der Methodik alle interdisziplinären Informationen, welche im semantischen Netz abgebildet werden müssen. Für die Bereitstellung einer mechatronisch bereinigten Stückliste als Forderung aus Kapitel 3.4.3, wird die Kante vom Typ „ist äquivalente Komponente zu“ eingeführt.

Ebenfalls zu erkennen ist, dass einige Disziplinen Informationen aus anderen benötigen, nicht um diese direkt weiter zu nutzen sondern um Kontextinformationen bereitzustellen. Diese ermöglichen ein besseres Systemverständnis. Als Beispiel hierfür sei die Schrittkette der Automatisierung genannt, worin der Elektrokonstrukteur herauslesen kann, ob Aktionen parallel oder sequenziell verlaufen. Bekanntlich besitzt dies Einfluss auf die Verschaltung. Folglich ist auch der Ablauf von Aktionen als interdisziplinär zu deklarieren.

5.3.2 Auswahl von Softwarewerkzeugen und Extraktion ihrer Semantik

Ausgehend von der in Kapitel 4 aufgezeigten Methode und ihren Beschreibungsmitteln, wird nach geeigneten und eingeführten Softwarewerkzeugen gesucht. Abbildung 5.9 zeigt die ausgewählten Werkzeuge, welche bereits einen Großteil der benötigten Phasen und Übergänge abdecken. Alle grün unterlegten Beschreibungsmittel sind mit den angegebenen Softwarewerkzeugen direkt abbildbar. Aus der dargestellten Situation sind zwei wesentliche Defizite erkennbar, einerseits eine unvollständige Softwareunterstützung der abzudeckenden Phasen und andererseits fehlende Kopplungen bzw. Übergänge zwischen den Softwarewerkzeugen oder Phasen.

Für die Anforderungsdefinition ist prinzipiell eine textuelle Beschreibung ausreichend, welche die Anforderungen auflistet. Allerdings sind diese mit den ermittelten Funktionen zu verknüpfen. Die in der Methode definierten statischen und dynamischen Beschreibungsmittel auf funktionaler Ebene sind derzeit noch in keinem Softwarewerkzeug implementiert. Deshalb ist hierfür ein Softwareprototyp zu entwickeln. Da die statische Verknüpfung der Funktionen auf gerichteten Graphen basiert, wird als Basis der Grapheditor Yed [yWorks, 2012] genutzt. Die Erstellung des Gantt-Charts zur dynamischen Funktionsbeschreibung erfolgt mit Excel. Der morphologische Kasten und die Auswahl bestehender Module sind aus softwareunterstützender Sicht sehr eng miteinander verbunden. Deshalb wird dies in einer eigens entwickelten Applikation berücksichtigt. Diese stellt auch das Ausschneiden der mechatronischen Objekte bereit. Die elektrische Konstruktion erfolgt in ComosET [Siemens Industry Software, 2010b], einem am Markt erhältlichen ECAD-System, welches alle dargestellten Beschreibungsformen der Elektrik unterstützt. Zur Ablaufbeschreibung in der Systementwurfsphase wird ein Impulsiagramm verwendet. Dies bietet das Softwarewerkzeug Sequence Designer [Siemens Industry Software, 2010a], welches als Teil von Automation Designer erhältlich ist. Eingepägt ist hier bereits die Generierungsfunktion in Step7 einer IEC-konformen SPS-Programmiersprache für Siemens-Steuerungen. Für die mechanische Ausdetaillierung wird ein MCAD-System benötigt. Diese Systeme bieten die Möglichkeit einer geometrischen Beschreibung, jedoch keine explizite Modellierung der Kinematik und deren Simulation. Hierzu kann das Softwarewerkzeug Mechatronics Concept Designer (MCD) [Siemens PLM Software, 2009a; Siemens PLM Software, 2009b; Siemens PLM 2010] genutzt werden, welches sowohl die kinematische Beschreibung, als auch deren Simulation unterstützt. Die Applikation ist zudem direkt auf dem MCAD-System NX integriert. Gleichzeitig stellt die Anwendung eine Möglichkeit zur Beschreibung von Anforderungen und deren Verknüpfung auf eine funktionale Dekomposition bereit. Das Simulations-

modell für die virtuelle Inbetriebnahme muss neben dem kinematischen Modell auch alle anderen Verhalten, wie beispielsweise die Status der elektrischen Geräte, berücksichtigen. Daher werden diese im Softwarewerkzeug SIMIT [Siemens AG, 2004] für die virtuelle Inbetriebnahme modelliert. Zusammen ergibt sich Abbildung 5.9.

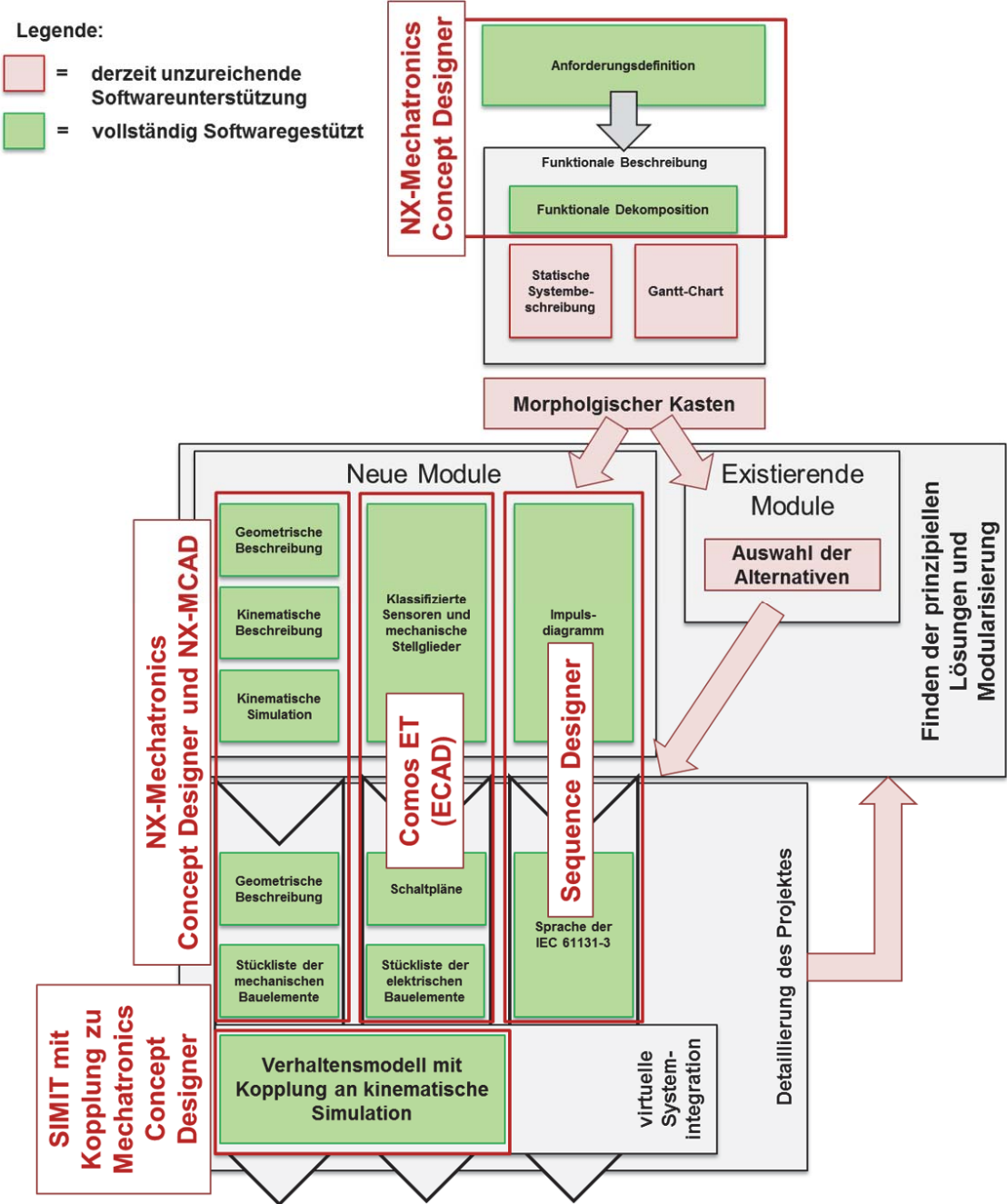


Abbildung 5.9: Abdeckung der Phasen durch am Markt erhältliche Softwarewerkzeuge

Um das zweite Defizit der fehlenden Kopplungen und Übergänge zwischen den Softwarewerkzeugen zu schließen, sind alle interdisziplinär relevanten Abhängigkeiten zu unterstützen. Mechatronics Concept Designer setzt derzeit ein Gantt-Chart zur Beschreibung und Simulation des gewollten Maschinenablaufs ein. Dieses muss durch ein Impulsdiagramm ersetzt werden. Der Übergang von der elektrischen Konstruktion zur Simulation bzw. virtuellen Inbetriebnahme ist ebenfalls noch nicht gelöst. Neben diesen Kopplungen sind auch die interdisziplinären Abhängigkeiten nach Kapitel 3.4.3, während der simultanen Ausdetaillierung zu berücksichtigen. Allerdings sind diese von den semantischen Modellierungsmöglichkeiten der eingesetzten Softwarewerkzeuge abhängig. Dies soll am Beispiel von sogenannten Kollisions-sensoren der kinematischen Simulation in Mechatronics Concept Designer aufgezeigt werden. Solche Sensoren geben bei Kollision mit einem Partner, in Abbildung 5.10 durch eine Box dargestellt, ein Signal über dieses Ereignis aus. Die Interpretation dieses Kollisionsereignisses im mechatronischen Kontext ist die Ausgabe eines Signals des „Sensors“ für die Steuerung, welches gleichzeitig eine Position repräsentiert.

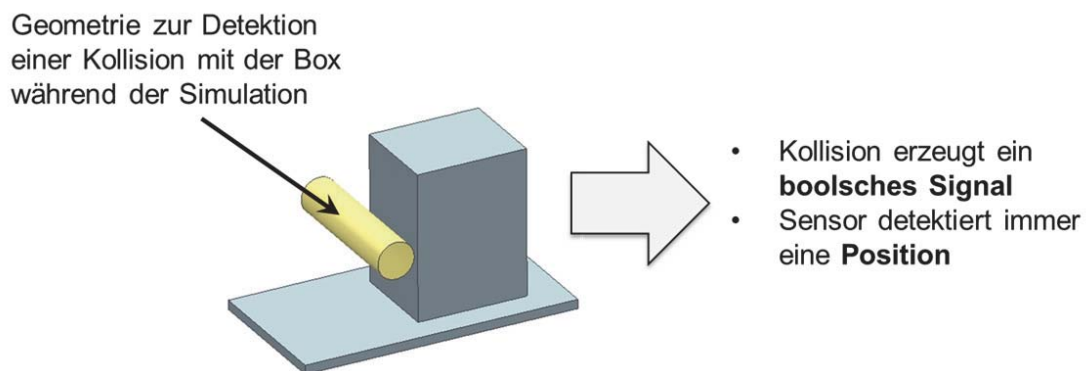


Abbildung 5.10: Semantische Extraktion eines „Collision-Sensors“ des Mechatronics Concept Designers zur Integration in das semantische Netz

Die Modellierung eines Knotens im semantischen Netz vom Typ „Sensor“ ist aufgrund dieser Interpretation nicht ausreichend, denn Positionen stellen gleichzeitig Zustände dar, die von der Steuerung berücksichtigt werden können. Deshalb werden Kollisionssensoren im semantischen Netz durch einen Positionsknoten, einen Signalknoten und die Verbindung zwischen diesen beiden abgebildet. Dieses Beispiel zeigt die Vorgehensweise zur Extraktion der werkzeugspezifischen Semantik auf, welche die bereitgestellten Knoten- und Kantentypen der Softwarewerkzeuge

Das Softwarewerkzeug Mechatronics Concept Designer ist in diesem Zusammenhang als Erweiterung der MCAD-Systeme zu sehen.

5.3.3 Abgleich der benötigten und bereitgestellten Informationen

Die ideale Informationsabdeckung der Disziplinen und des interdisziplinären Bereichs ist in Kapitel 5.3.1 dargestellt. Kapitel 5.3.2 zeigt die von den Softwarewerkzeugen bereitgestellten Informationen auf. Ein Vergleich der entsprechenden Abbildungen (Abbildung 5.8 und Abbildung 5.11) lässt erkennen, dass zwischen den Bereichen bei einer optimalen Unterstützung und den derzeit bereitgestellten einige Abweichungen bestehen.

Die Kollisionssensoren des Mechatronics Concept Designers für die Phase des Systementwurfs aus Kapitel 5.3.2 zeigen auf, dass eine Position über ein Messelement zu detektieren und als boolesches Signal an die Steuerung auszugeben ist. Allerdings besteht keine Information darüber, welcher elektrisch relevante Sensortyp für diese Positionserkennung genutzt wird. In der Phase des Systementwurfs ist diese Information aber zu definieren. Dadurch ist eine semantische Anreicherung des Sensormodells mit einem elektrischen Gerätetyp notwendig, um die interdisziplinäre Kommunikation zur Elektrik und des ECAD-Systems zu unterstützen. Eine weitere Lücke stellen kinematische Positionen dar, welche das MCAD-System oder der MCD nicht explizit modellieren können. Selbiges gilt aus mechanischer Perspektive für die Anschlüsse und Längen bzgl. Kabel, Rohre und Schläuchen. Das ECAD-System trifft keine Unterscheidung seiner Geräte hinsichtlich Sensoren, Wirkelementen oder anderer elektrischer Geräte. Diese ist allerdings notwendig um mit der Mechanik und Automatisierung kommunizieren zu können. Die Verhaltensmodellierung der elektrischen Komponenten ist häufig auf Gerätetypenebene ausreichend, muss derzeit aber in einem separaten Softwarewerkzeug umgesetzt werden. Die automatisierungstechnische Sichtweise bringt mit dem Sequence Designer eine gute Abdeckung mit ein.

Aus der beschriebenen Situation ist ersichtlich, dass die einzelnen Applikationen ihre zugeordneten Aufgaben sehr gut unterstützen, im interdisziplinären Kontext aber sehr große Abdeckungslücken aufweisen, welche derzeit durch implizite Modelltransformationen der einzelnen Entwickler übertragen werden. Die Schließung dieser Lücken kann nur durch eine explizite semantische Anreicherung der Modelle in den einzelnen Softwarewerkzeugen für alle Bereiche, welche nicht deckungsgleich mit Abbildung 5.8 sind, erreicht werden.

5.4 Architektur der Softwareunterstützung

Kapitel 5.3 zeigt das benötigte Informationsmodell auf, welches die Basis für die dargelegte Methode und die eingesetzten Softwarewerkzeuge bildet. Gleichzeitig ist dies eine Betrachtung der semantischen Datenmodellierung. Wie auch das Vorgehen zum Entwurf von Datenbanksystemen (vgl. Kapitel 2.6.2) zeigt, erfolgt im Anschluss an die logische Modellierung die physikalische Implementierung. Die Architekturübersicht zur Umsetzung des aufgezeigten Informationsmodells hinsichtlich einer integrierten Sondermaschinenentwicklung zeigt Abbildung 5.12. Die Basis bildet, wie bereits in Kapitel 5.1 dargelegt, ein semantisches Netz. Dieses basiert auf gerichteten Graphen. Zur Beschreibung solcher Graphen bietet sich GraphML an, eine auf XML-basierende Beschreibungssprache. [Brandes, et al. 2002] Vorteilhaft ist eine Visualisierung dieser Graphen direkt in Yed. Alle Knoten- und Kantentypen werden dort definiert. Zur physikalischen Verknüpfung mit anderen Softwarewerkzeugen bieten manche Anwendungen entsprechende Programmierschnittstellen, sogenannte Application Programming Interfaces (API) an. NX, welches im Rahmen dieser Arbeit die Basis der Applikationen des Mechatronics Concept Designers und des MCAD-Systems bildet, stellt eine solche bereit.

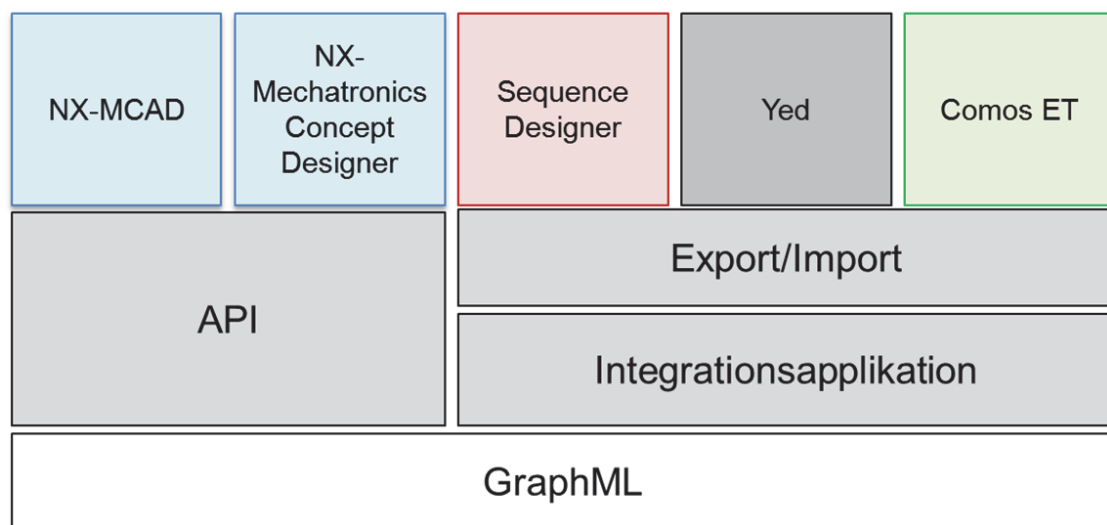


Abbildung 5.12: Architektur der prototypischen Umsetzung

Damit werden die Arbeitsbereiche und Regeln der beiden Softwarewerkzeuge zur Arbeit mit dem semantischen Netz definiert. Andere Programme können nur über Export-/ Importschnittstellen angebunden werden. Aufgrund der technischen Ausprägung von ComosET und Sequence Designer, sowie der in diesem Rahmen nicht zur Verfügung stehenden API von Yed, können diese nicht

direkt an GraphML angebunden werden. Deshalb erfolgt deren Anbindung über XML-basierte Export- und Importschnittstellen. Die Transformation dieser Daten in und aus dem semantischen Netz übernimmt die im Rahmen dieser Arbeit entwickelte Integrationsapplikation. Diese stellt auch die Funktionalität des morphologischen Kastens zur Auswahl Definition und Instanziierung der mechatronischen Objekte bereit. Daraus ist abzuleiten, dass neben den semantischen Lücken zwischen den Disziplinen ebenfalls technische Anbindungsprobleme zur Verknüpfung der Softwarewerkzeuge bestehen.

5.5 Umsetzung der Softwareunterstützung

5.5.1 Umsetzung der Phasen Anforderungsdefinition und Systementwurf

Die Dokumentation der Anforderungen erfolgt üblicherweise in Form von Office-Dokumenten. Eine Unterstützung hierfür bietet Mechatronics Concept Designer in der Form an, dass auf Basis der Kapitelüberschriften in Word-Dokumenten Anforderungsobjekte importiert werden können. Dies führt auf eine hierarchische Struktur der Anforderungen. Im Weiteren ist eine Definition der funktionalen Hierarchie, also eine Dekomposition direkt in dieser Applikation möglich. Zusätzlich können Anforderungen mit entsprechenden Funktionselementen verknüpft werden. Dadurch bildet die funktionale Dekomposition den Ausgangspunkt der entwickelten Integrationsapplikation.

Abbildung 5.13 zeigt links das Userinterface zur Wahl der entsprechenden Modellierungsebenen. Dies hilft gleichzeitig sich im Entwicklungsprozess zu Recht zu finden und ist gleichbedeutend mit den beschriebenen Abstraktionsebenen. Von dort gelangt der Anwender in den Bereich der funktionalen Modellierung. Hierbei ist zu erkennen, dass die funktionale Verknüpfung, basierend auf Yed durch Export- und Import der GraphML-Datei an das projektspezifische semantische Netz angebunden ist. Über den Button der „Lösungssuche für Funktionen“ gelangt der Nutzer zum morphologischen Kasten und zu der Definition der Lösungsprinzipien. Gleichzeitig bildet dies den Übergang zur logischen Modellierung.

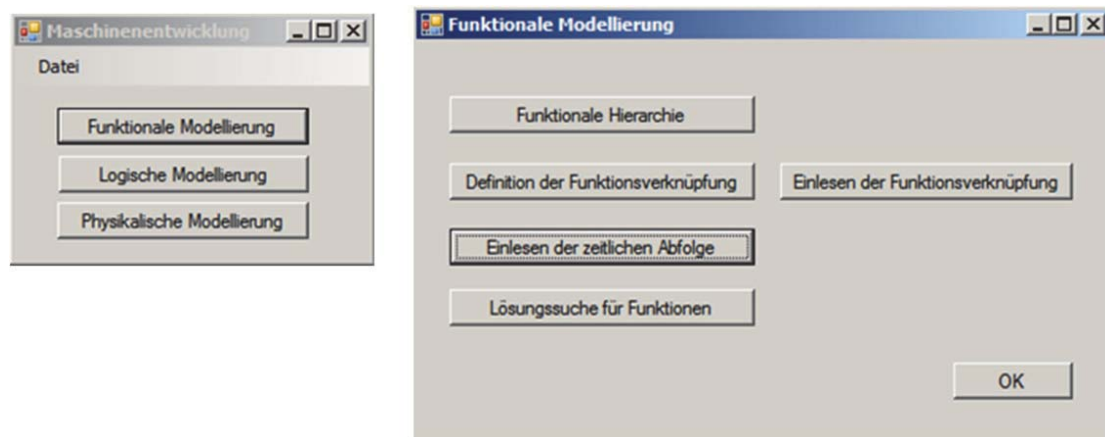


Abbildung 5.13: Userinterface zur funktionalen Modellierung

Zu Beginn der logischen Modellierung (siehe Abbildung 5.14) erfolgt der Aufbau der Maschinenstruktur auf logischer Ebene, vergleichbar der Maschinenstruktur im ECAD-System. (vgl. Abbildung 3.2) Dabei können auf Basis des funktionalen Modells und durch die Spezifikation logischer Objekte die Signal- und Energieschnittstellen der einzelnen Module definiert werden. Analog funktioniert die Ableitung des Materialflusses. Im Falle einer nicht durchgeführten funktionalen Modellierung kann die Maschinenstruktur auch manuell erstellt werden. Die Beschreibung der Lösungsprinzipien enthält die Auswahl bereits existierender physikalischer Alternativen. Außerdem führt die „Mechatronische Lösungsbeschreibung“ zur Anbindung des Impulsdigramms aus dem Sequence Designer an das Modell des Mechatronics Concept Designer. Eine vollständige Beschreibung eines neuen Lösungsprinzips kann, wie bereits in Kapitel 5.3 dargestellt, nur durch eine semantische Anreicherung der genutzten Wirk- und Messelemente über die Angabe ihres elektrischen Gerätetyps erreicht werden.

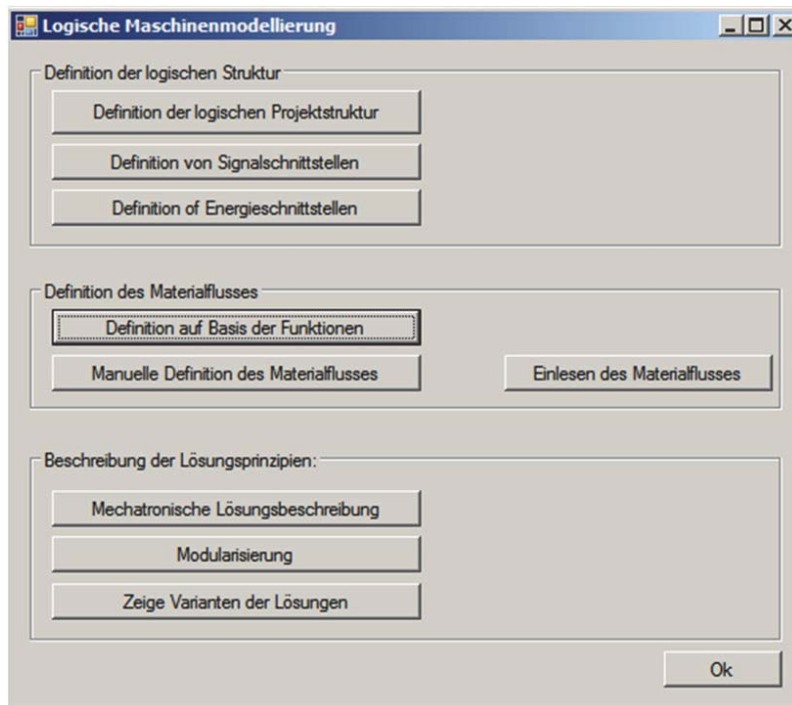


Abbildung 5.14: Userinterface zur logischen Modellierung

Eine solche Anreicherung geschieht im Rahmen der Definition eines Kollisionssensors für das Modell des Mechatronics Concept Designers. Dazu steht eine entsprechende Erweiterung der Dialoge bereit. Diese ist in Abbildung 5.15 dargestellt. Darin stehen für Kollisionssensoren alle ECAD-Sensorobjekte zur Auswahl, welche eine Position detektieren und ein boolesches Signal ausgeben. Sollen andere Zustände, wie beispielsweise eine Temperatur gemessen werden, müssten andere Simulationsobjekte in der Simulationsumgebung integriert werden. Andere Signaltypen können nur mittels Transformation zwischen dem booleschen Signal des Kollisionssensors und der Signalschnittstelle implementiert werden.

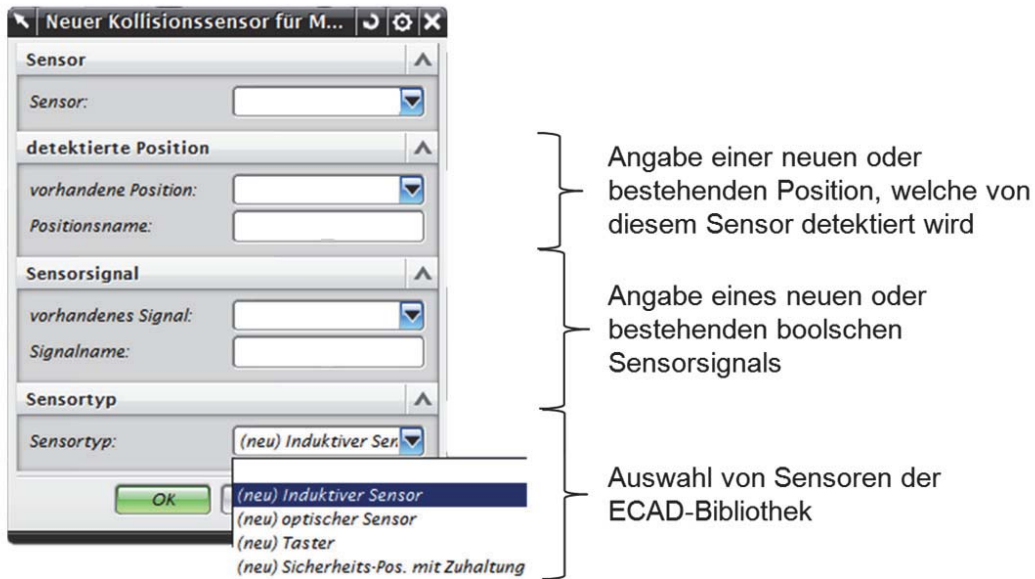


Abbildung 5.15: Beispiel zur semantischen Anreicherung bei der Erstellung eines „Collision Sensors“ in Mechatronics Concept Designer

5.5.2 Parallele Detaillierung

Während der parallelen Detaillierungsphase müssen die entsprechenden interdisziplinären Abhängigkeiten berücksichtigt werden. Hinzugefügte, gelöschte oder geänderte Objekte betreffen nicht in jedem Fall alle Disziplinen. Um die disziplinspezifischen Konstrukteure nicht zu überlasten, müssen solche Objekte gefiltert werden. Dies ist über disziplinspezifische Navigatoren auf der physikalischen Modellierungsebene realisiert. Zu sehen sind diese in Abbildung 5.16.

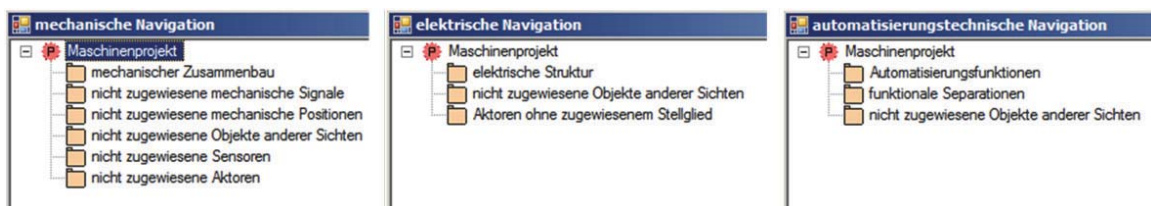


Abbildung 5.16: Navigatoren für die parallele Detaillierung mit Ordner für interdisziplinär relevante Informationen

Sie dienen der interdisziplinären Kommunikation, indem alle relevanten Informationen aus den anderen Bereichen auflaufen und entsprechende Weiterverarbeitungsfunktionen bereitgestellt werden. Jeder Navigator enthält bestimmte Ordner für entsprechende Objekte und Informationen der anderen Sichtweisen. Außerdem ist der jeweilige disziplinspezifische Projektkontext angegeben.

Der mechanische Konstrukteur bezieht sich hierbei auf die Baugruppenstruktur, welche im ersten Ordner visualisiert wird. Die Elektrokonstruktion nutzt für diesen Kontext die Maschinenstruktur, während die Automatisierung die logischen Module mit den umzusetzenden Funktionen und den zur Verfügung stehenden Zuständen und Signalen enthält. In den entsprechend anderen Ordnern werden die interdisziplinären Informationen bereitgestellt. In den übrigen Ordnern werden jene Informationen bereitgestellt, welche im interdisziplinären Bereich der entsprechenden Disziplin liegen. Im Ordner „nicht zugewiesene Objekte anderer Sichten“ des mechanischen Navigators werden alle elektrischen Geräte der Elektrik dargestellt, welche noch keine geometrische Repräsentanz in der Baugruppenstruktur besitzen. Auf Basis dieser Information kann der mechanische Konstrukteur dann eine solche in seinem Modell hinzufügen. Durch eine Zuweisung des Bauteils oder der Baugruppe auf das elektrische Objekt wird die Relation vom Typ „ist äquivalente Komponente“ im semantischen Netz gezogen.

Kinematische Positionen werden im Navigator kenntlich gemacht. Dadurch ist es möglich diese entsprechend mit einem Signal zu verknüpfen. Dies ermöglicht die Anforderungsdefinition für einen Sensor. Die Verknüpfung des Signals und der Position mit dem entsprechenden Kollisionssensor erfolgt dann im Dialog des Softwarewerkzeuges. (vgl. Abbildung 5.15)

Entstehen Informationen, welche sich im Nachhinein als äquivalent herausstellen (vgl. Kapitel 4.4.1), ist deren nachträgliche Verknüpfung im Navigator durchzuführen. Dies ist möglich indem ein Objekt per Drag n' Drop auf das andere gezogen wird. Geänderte Objekte werden am Symbol mit einem Ausrufezeichen markiert. Eine entsprechende Behandlung dieser, kann über ein Menü erfolgen, beispielsweise durch die Angabe, dass diese nicht für das Modell relevant ist.

Das Löschen von Objekten wird an Stelle eines Ausrufezeichens durch ein rotes Kreuz visualisiert. Auch hier steht dem Anwender neben dem Löschen des Objektes auch die Möglichkeit zur Verfügung, dieses abzulehnen. Dadurch werden alle in Abbildung 3.15 aufgezeigten Abhängigkeiten der Disziplinen berücksichtigt.

5.5.3 Simulationsunterstützung der Maschinenentwicklung

Die aufgestellten Anforderungen aus Kapitel 2.7.1 an diese Arbeit, sehen eine durchgängige Maschinensimulation entlang des gesamten Entwicklungsprozesses vor. Die Simulation trägt hierbei bereits in der Phase des Systementwurfs zum interdisziplinären Verständnis der kinematischen Konzepte bei. (vgl. Kapitel 4.3.4). Diese Phase adressiert das ausgewählte Softwarewerkzeug des Mechatronics Concept Designers. Dabei wird durch eine Anbindung des Sequence Designers das bisherige Gantt-Chart zur Beschreibung des Steuerungsablaufes substituiert. Dies

ermöglicht zunächst eine rein zeitbasierte Simulation des kinematischen Ablaufs auf Basis der angegebenen geometrischen Positionen. Dies ist in Abbildung 5.17 auf der linken Seite dargestellt.

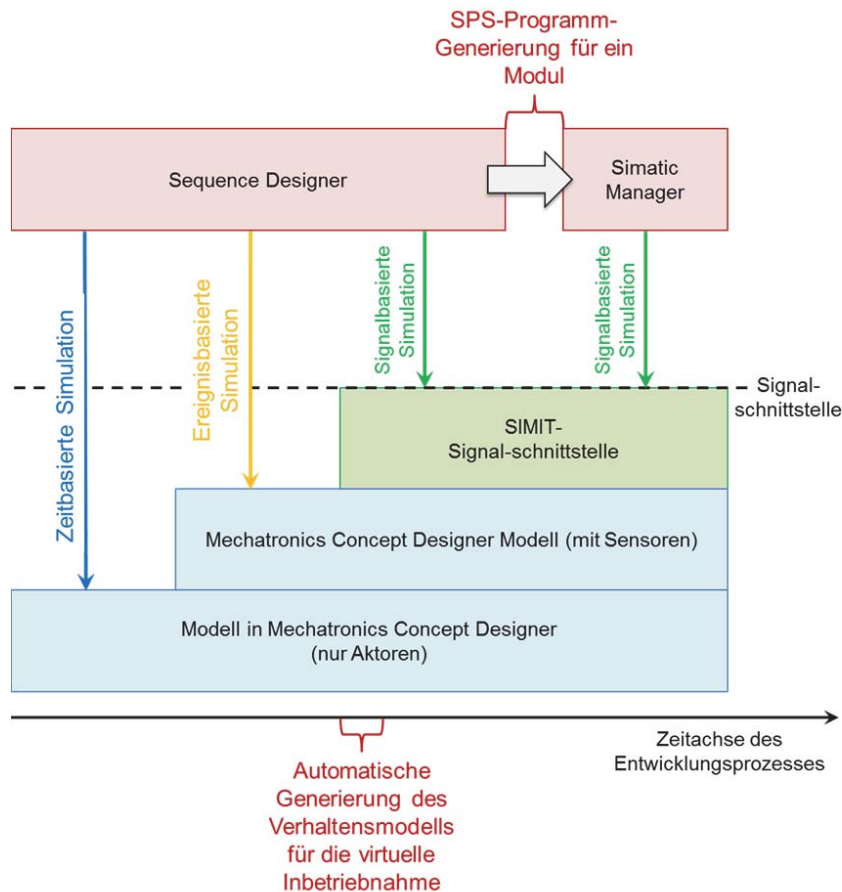


Abbildung 5.17: Stufen der virtuellen Inbetriebnahme entlang des Entwicklungsprozesses

Hierbei werden die Zustandsübergänge durch die zeitliche Begrenzung der Aktionen ausgelöst. In der ersten konzeptionellen Lösungsfindung ist dies ausreichend. Neben dem reinen Ablauf hilft die physikbasierte Simulation des Mechatronics Concept Designers auch physikalische Effekte, wie die Gewichtskraft und Reibung, beispielsweise bei der Beförderung von Werkstücken, besser zu verstehen. Gleichzeitig ermöglicht das Softwarewerkzeug ein Eingreifen des Nutzers zur Herbeiführung von Störfällen.

Zu einer vollständigen mechatronischen Konzeptbeschreibung werden auch Sensoren benötigt. Eine Zuweisung der Signale dieser Sensoren als Transitionsbedingung für entsprechende Aktionen führt auf die ereignisbasierte Simulation. Dabei ist zwischen zeit- und ereignisbasierter Simulation ein fließender Übergang möglich. Es ist damit nicht relevant, ob alle Aktionen ereignisbasiert

modelliert sind, oder nicht. Im Zweifel greift der zeitbasierte Mechanismus aus der ersten Stufe. Wird ein Sensor eingebaut und seiner Transition zugeordnet, beginnt die nachfolgende Aktion nicht aufgrund der begrenzten Zeitdauer der vorherigen, sondern durch das Auslösen des Sensorereignisses. Ausgehend von dieser Beschreibung ist eine parallele Entwicklung in den einzelnen Disziplinen möglich.

Die Kopplung des Simulationsmodells in Mechatronics Concept Designer mit der Beschreibung des Ablaufes in Sequence Designer erfolgt durch die Integrationsapplikation. Diese liest die Sensor-Aktor-Definitionen des MCD-Modells und die XML-Datei des Sequence Designers ein. Eine Zuweisung der beschriebenen Positionen in Sequence Designer auf die entsprechenden Aktoren und einer solchen für die Sensoren auf die entsprechenden Positionen, bildet die Basis für eine Generierung einer MCD-kompatiblen Steuerungsdatei. Diese Datei kann direkt in MCD angegeben und für die Simulation genutzt werden.

Aus Kapitel 3.3 ist bekannt, dass neben den Signalen, welche direkt das mechanische Grundsystem beeinflussen oder dessen Wirkungsweise messen, weitere Signale zur Steuerung der Maschine aus den elektrischen Komponenten herangezogen werden müssen. Um ein vollständiges Verhaltensmodell zu erhalten, wird bereits in Kapitel 4.5 dargelegt, dass eine Generierung dessen auf Basis der Schaltpläne eine enorme Aufwandsreduktion darstellt. Abbildung 5.18 zeigt das zu Grunde liegende Prinzip. Dabei wird jeder Komponente der ECAD-Bibliothek ein Verhaltensmodell zugeordnet. Jedes Verhaltensmodell besitzt exakt jene Anschlüsse, welche auch in den Schaltplänen dargestellt werden. Durch Interpretation der Schaltpläne ist es möglich das Verhaltensmodell entsprechend von ComosET nach SIMIT zu generieren. Eine Kopplung von SIMIT mit Mechatronics Concept Designer ist über die gegebene OPC-Schnittstelle möglich. Ein Mapping der Wirk- und Messelemente bietet folglich ein umfangreiches Simulationsmodell inklusive 3D-Visualisierung und Einbeziehung physikalischer Effekte. Die Generierung des SPS-Programms auf Basis des Sequence Designer-Modells ermöglicht eine signalbasierte Simulation unter Einbindung der realen Steuerung.

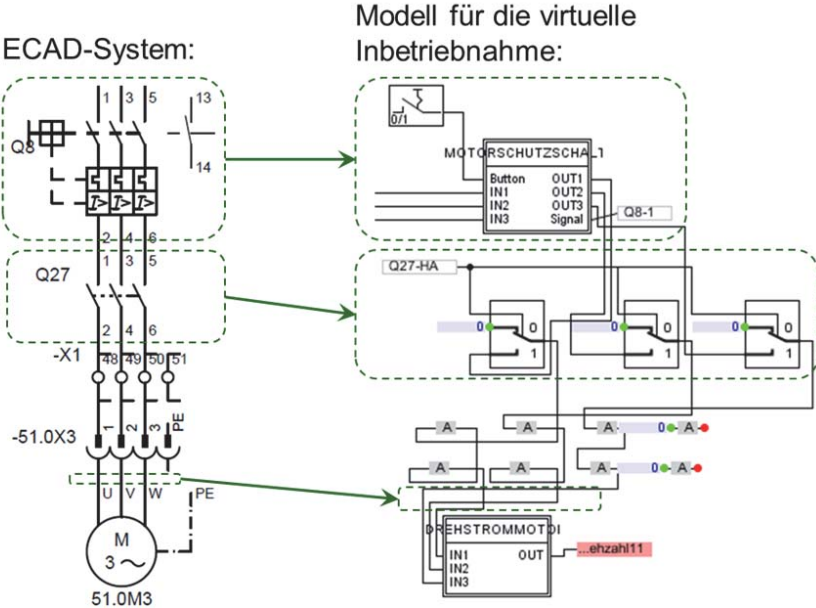


Abbildung 5.18: Generierung von Modellen für die virtuelle Inbetriebnahme auf Basis von Schaltplänen

6 Validierung am Beispiel einer Firmwarelademaschine

Als Validierungsbeispiel dient eine Sondermaschine zur Produktion von speicherprogrammierbaren Steuerungen der Siemens AG. An dieser Maschine wurden die einzelnen Schritte der Methode inklusive ihrer Beschreibungsmittel, der Softwareunterstützung und der Abhängigkeiten zwischen den Disziplinen erprobt.

6.1 Fallbeispiel: Entwicklung einer Firmwarelademaschine

Die Firmwarelademaschine wird im Rahmen der Produktion von speicherprogrammierbaren Steuerungen eingesetzt. Dabei ist es notwendig, die Firmware auf den Chip der Steuerung aufzuspielen. Gleichzeitig muss überprüft werden, ob alle Kontakte auf der Leiterplatte des Produkts „SPS“ richtig montiert sind. Beide Prozesse sind äußerst produktabhängig und stellen keine Standardverfahren dar. Daher sind Maschinen zur Automatisierung dieser Aufgaben nicht am Markt erhältlich. Weiterhin muss die Maschine in den Ablauf des Produktionssystems der Montage integriert werden. Dies führt auf eine klassische Ausgangssituation für die Entwicklung von Sondermaschinen.

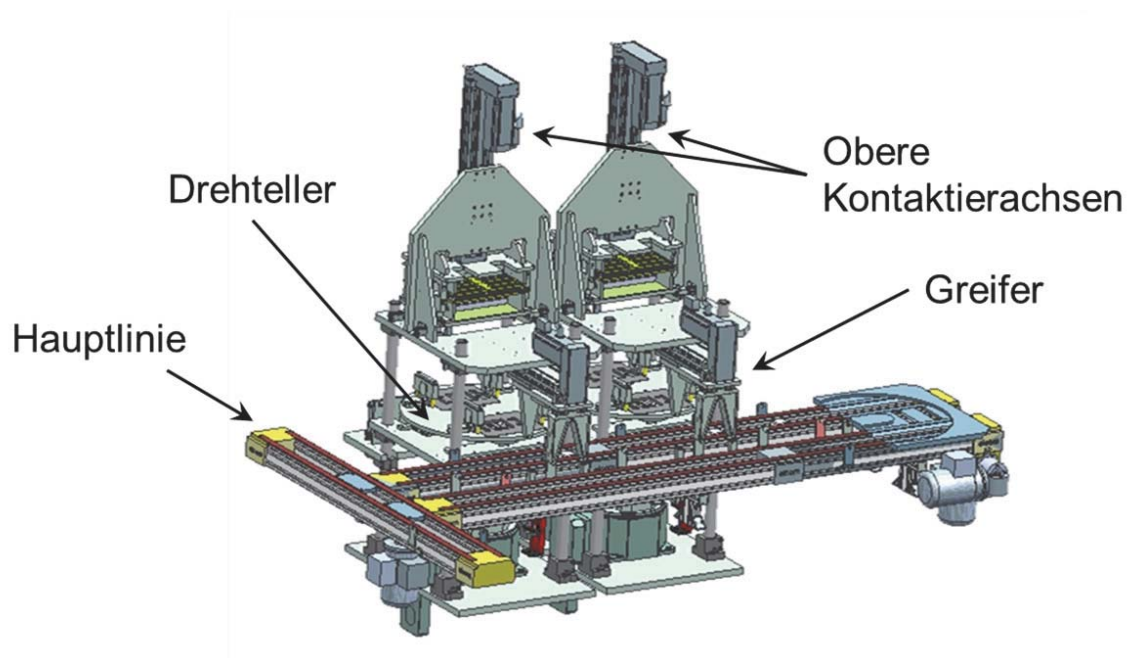


Abbildung 6.1: Firmwarelademaschine

Abbildung 6.1 zeigt das mechanische Modell der Maschine auf. Das Förderband auf der linken Seite stellt gleichzeitig einen Teil der Hauptlinie dar. Ausgehend davon werden die Steuerungen in die Maschine ausgeleitet und am Ende um 180° gedreht. Weiterführend werden die Steuerungen samt Werkstückladeträger über einen Lift und einen Greifer auf den Drehteller gesetzt. Dieser positioniert die Steuerungen unter zwei vertikalen Achsen, je eine unterhalb und eine oberhalb der Steuerung, zur Überprüfung der Kontakte. Nach Gutbefund wird die Firmware auf die Steuerung geladen, diese über den Drehteller wieder auf das Förderband gesetzt und der Hauptlinie zugeführt.

6.2 Entwicklungsvorgehen

Ausgehend von der anfangs in Kapitel 6.1 dargestellten Situation zur Integration dieser Maschine in die Montagelinie, ergibt sich die Anforderung, die zu entwickelnde Maschine gleichzeitig als Puffer zu nutzen. Weiterhin werden die Anforderungen gemäß der Sachmerkmalleitlinie aufgestellt. Besonders die integrationsrelevanten Randbedingungen sind hierbei zu klären. Ausgehend davon und der Aufgabenstellung kann die Hauptfunktion zu „Software aufspielen“ formuliert werden. Die nächste Dekompositionsebene ergibt die Funktionen „Produkt bereitstellen“, „Produkt speichern“, „Kontakte überprüfen“, „Firmware aufspielen“ und „Produkt zurückführen“. Hieraus kann auch direkt die Aufgabenstellung ersehen werden. Die gesamte funktionale Dekomposition zeigt Abbildung 6.2.

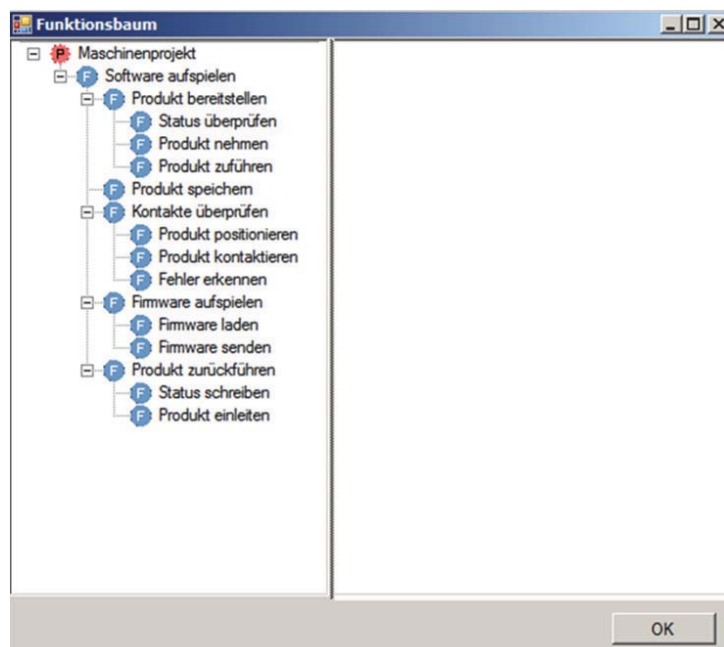


Abbildung 6.2: Funktionale Struktur der Firmwarelademaschine in der Integrationsapplikation

Im Weiteren erfolgt eine funktionale Vernetzung in Yed. Gleichzeitig wird im Gantt-Chart auf Basis der definierten Elementarfunktionen der grobe Ablauf der Maschine modelliert. Gerade diese Modellierung zeigt auf, dass neben der reinen Dauer auch Parallelitäten zwischen Funktionen auf neue Lösungen führen können. In Abhängigkeit davon, werden prinzipielle Lösungen über den morphologischen Kasten (Abbildung 6.3) gesucht. Dabei ist es möglich sowohl bereits existierende Prinzipien anzugeben oder neue zu definieren. Die Festlegung der Gesamtlösung erfolgt über die Definition der eins zu eins Zuweisung von Funktion auf Lösungsprinzip oder durch funktionale Integration.

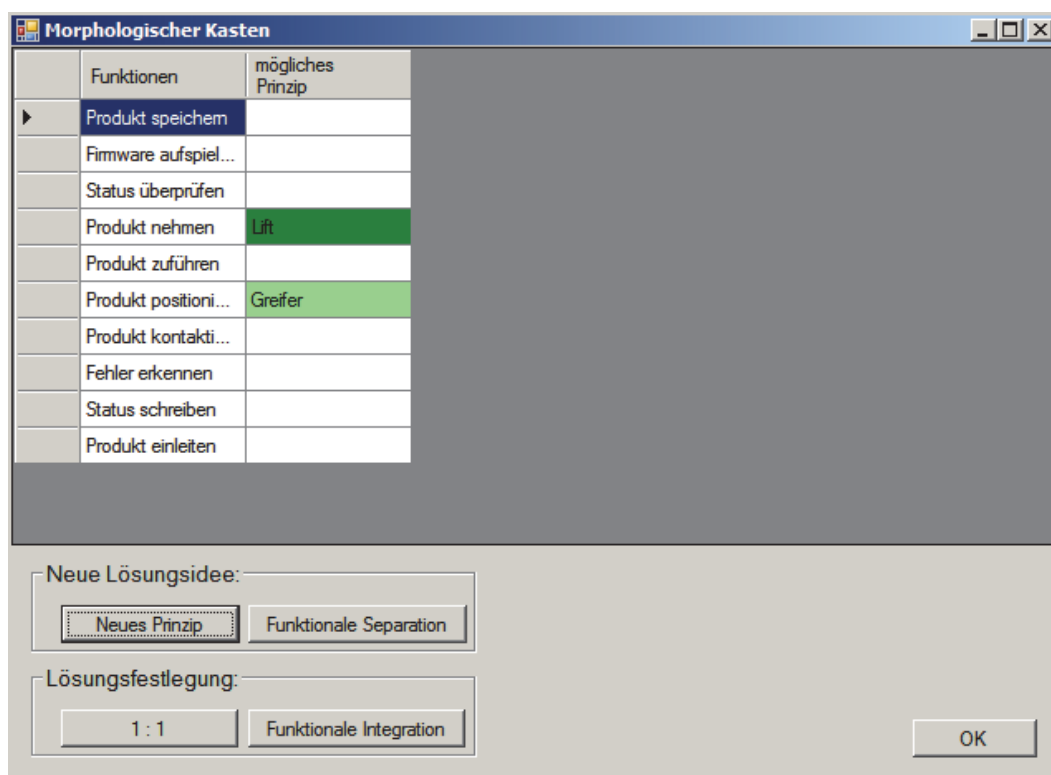


Abbildung 6.3: Morphologischer Kasten zur Suche und Definition von Prinziplösungen

Werden schon existierende Lösungsklassifizierungen genutzt, kann über das User Interface im logischen Modell nachgeschaut werden, ob bereits verschiedene Alternativen entwickelt worden sind. Dieses Menü ist in Abbildung 6.4 dargestellt. Eine Wiederverwendung der aufgezeigten Alternative kann über das Kontextmenü und die Auswahl „Verwende dieses mechatronische Objekt“ erfolgen. Dadurch werden die dargestellten Objekte im semantischen Netz instanziiert, was nach Kapitel 4.3.3 einem Zeitsprung in der Entwicklung entspricht. Die Visualisierung aller

Alternativen zu einem logischen Objekt, ermöglicht auch Konstrukteuren Zugang zu bereits existierenden Lösungen, die bis zu diesem Zeitpunkt keine Kenntnis darüber hatten.

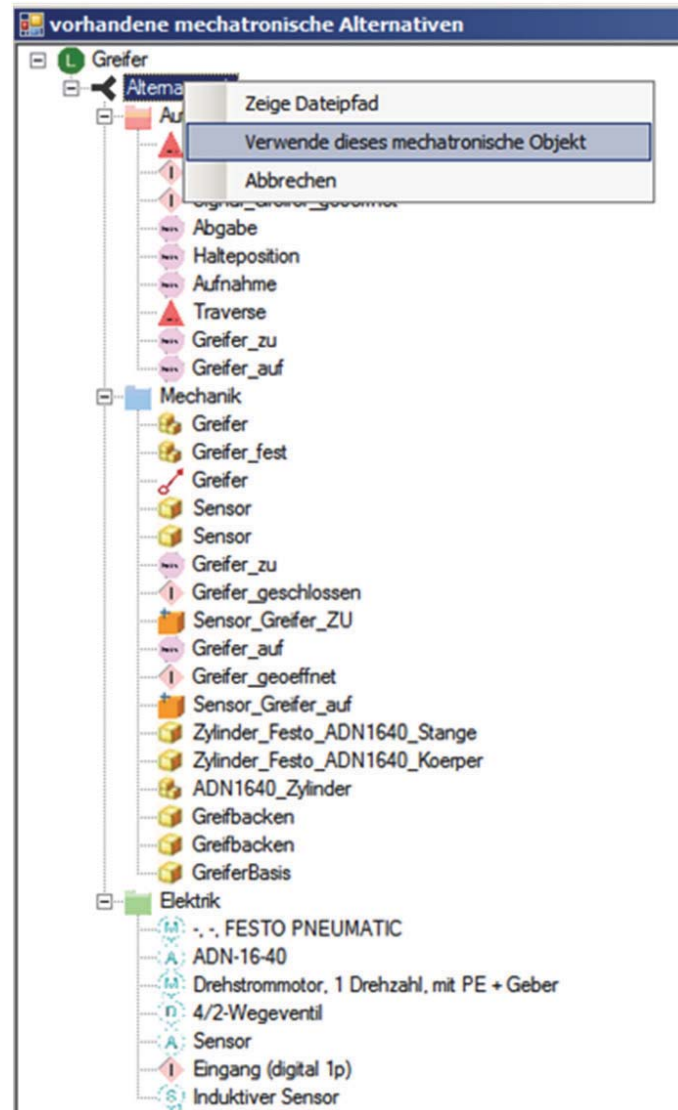


Abbildung 6.4: Wiederverwendung eines Greifers

Neuentwicklungen, wie beispielsweise die eines Greifers, werden zunächst kinematisch in Mechatronics Concept Designer beschrieben. Abbildung 6.5 zeigt den Greifer der Firmwarelademaschine in einem frühen Entwicklungsstand auf. Weiterhin ist darin die Angabe des Antriebskonzeptes für die Traversenachse zu sehen. Dies verkörpert das Antriebskonzept, welches im Rahmen einer weiterführenden Elektrokonstruktion notwendig ist.

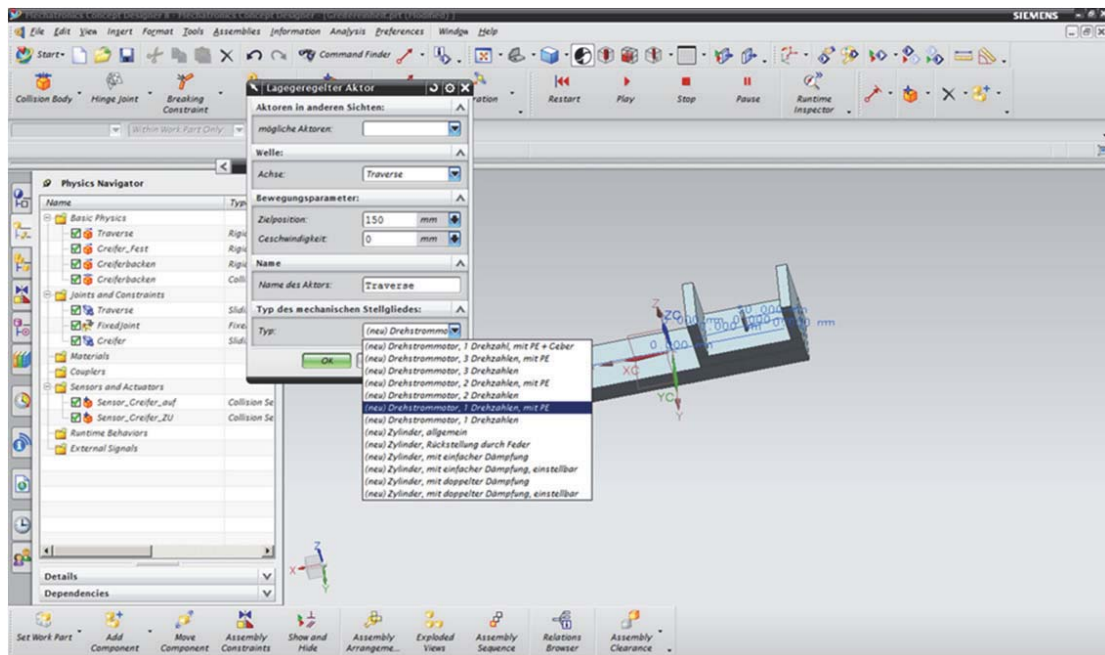


Abbildung 6.5: Kinematisches Konzept eines Greifers in Mechatronics Concept Designer mit Angabe des Antriebskonzepts

Die Ablaufbeschreibung geschieht in Sequence Designer (siehe Abbildung 6.6). In einem ersten Schritt müssen keine Signale modelliert werden. Zur Simulation sind die Beschreibungen des Greifers in Mechatronics Concept Designer und Sequence Designer zu koppeln.

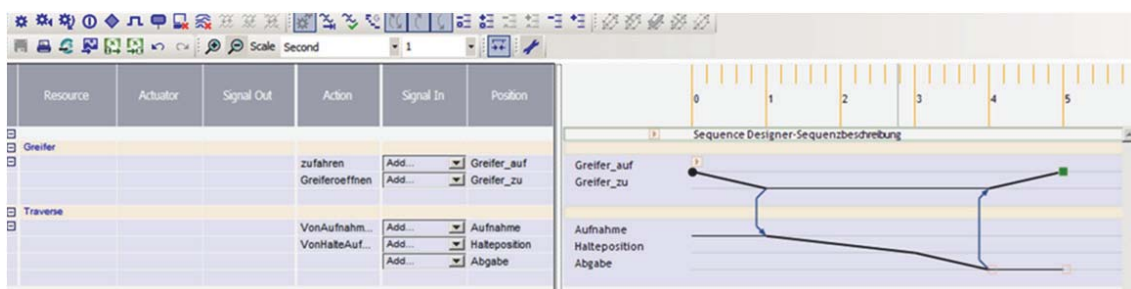


Abbildung 6.6: Mechanische Ablaufbeschreibung in Sequence Designer für den Greifer

Abbildung 6.7 stellt links einen Teil des gesamten semantischen Netzes zur Beschreibung der Greifeinheit dar. Das Löschen eines Sensors durch die Elektrokonstruktion ist darin ebenfalls veranschaulicht. Diese Aktion ist im Maschinenprojektbaum durchzuführen und löscht somit neben dem eigentlichen Sensorobjekt (grün mit weißer Schrift) auch das zugehörige Attribut der Bestellnummer (weißer Knoten mit grüner Schrift) und den zugehörigen Signaleingang (rote

Raute). Mechanische Bauteile sind als orangefarbene Rechtecke dargestellt, Positionen als Ellipsen und Kollisionssensoren als grünfarbene Rechtecke mit schwarzer Schrift. Daraus kann erkannt werden, dass die Aktion des Löschsens Auswirkungen auf alle Kanten besitzt, welche zwischen weiter bestehenden und zu löschenden Knoten führen. Rechts ist die Auswirkung auf die Objekte im mechanischen Navigator gezeigt.

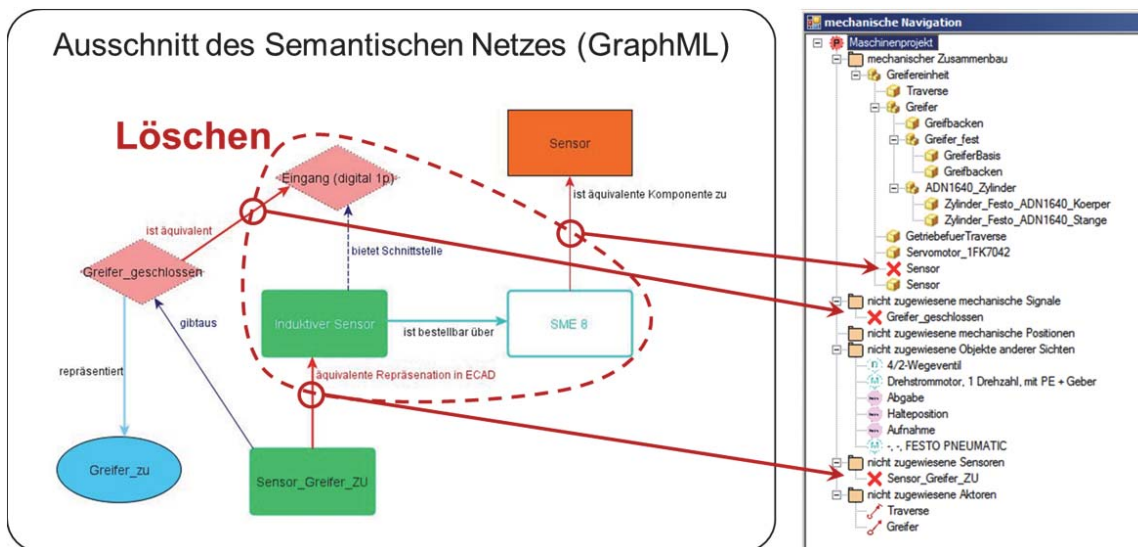


Abbildung 6.7: Löschen eines Sensors in interdisziplinärem Umfeld aus dem ECAD-System heraus

Zu erkennen ist, dass ein Löschen des Sensors im ECAD-System auch den Kollisionssensor inklusive dessen Signals im Modell des Mechatronics Concept Designer, sowie die geometrische Repräsentanz im MCAD-System in Frage stellt. Der Anwender kann über das Kontextmenü entscheiden, ob er bestimmte Objekte daraufhin ebenfalls löschen möchte oder ob dieses Löschen keine Auswirkungen auf sein Modell besitzt. Dieses ist beispielhaft in Abbildung 6.8 dargestellt. Analog hierzu ist die Änderung von Objekten umgesetzt, wobei zusätzlich abgerufen werden kann, welche Attribute des Knotens sich geändert haben. Weiterhin können alle Aussagen zu diesem Objekt angezeigt werden. Diese geben dann alle Tripel des semantischen Netzes aus, in welchen der Objektknoten entweder Quelle oder Ziel darstellt. Objekte auf logischer Ebene, wie der dargestellte Drehstrommotor der Elektrik, besitzen weiterhin die Möglichkeit bereits einmal zugewiesene physikalische Repräsentanzen anzuzeigen. Die verschiedenen Alternativen werden analog zu Abbildung 6.4 auch für Komponenten dargestellt. Folglich ist auch eine optimierte Wiederverwendung für Komponenten möglich.

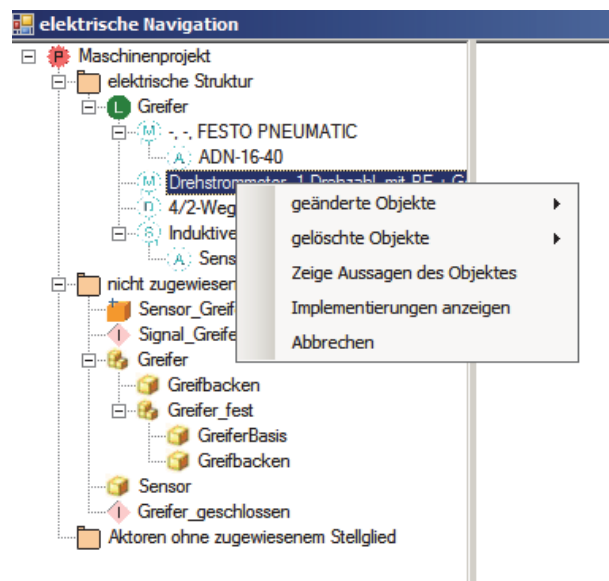


Abbildung 6.8: Kontextmenü der Navigatoren

Der in Abbildung 6.7 dargestellte Kantentyp „ist äquivalente Komponente zu“ wird zwischen der Bestellnummer des ECAD-Modells und eines mechanischen Bauteils oder Baugruppe gezogen. Eine Auswertung dieses Kantentyps ermöglicht die Bereitstellung einer mechatronisch bereinigten Stückliste. Die Generierung des SPS-Programms für den Greifer erfolgt auf Basis der Sequence Designer-Beschreibung. Ein vollständiges Verhaltensmodell der Maschine wird auf Basis der Schaltpläne, wie in Kapitel 5.5.3 aufgezeigt, automatisch erzeugt. Dadurch ist eine virtuelle Inbetriebnahme der Maschine mit Hardwareeinbindung möglich.

Um den entwickelten Greifer als weitere Alternative, wie in Abbildung 6.4 zu erhalten, muss dieses Modul aus dem Projekt ausgeschnitten werden. Dazu enthält die Integrationsapplikation die Funktion der mechatronischen Objektdefinition. Diese ist in Abbildung 6.9 dargestellt. Rechts sind alle Objekte der verschiedenen Disziplinen des Projektes aufgezeigt, welche noch keinem Lösungsprinzip zugewiesen sind. Durch Drag n' Drop dieser Objekte von rechts unter ein logisches Objekt wird die Zuweisung durchgeführt. Dabei ist es möglich beispielsweise die dargestellte Bestellnummer „Servomotor_1FK...“ in der Elektrik zuzuweisen, was automatisch auch dessen logische Repräsentanz „Drehstrommotor. 1 Drehzahl mit PE ...“ mitdefiniert. Umgekehrt ist nur eine Zuweisung der logischen Ausprägung möglich. Dadurch wird eine Definition von mechatronischen Objekten mit unterschiedlich abstrahierten Modellen und Systemgrenzen erreichbar.

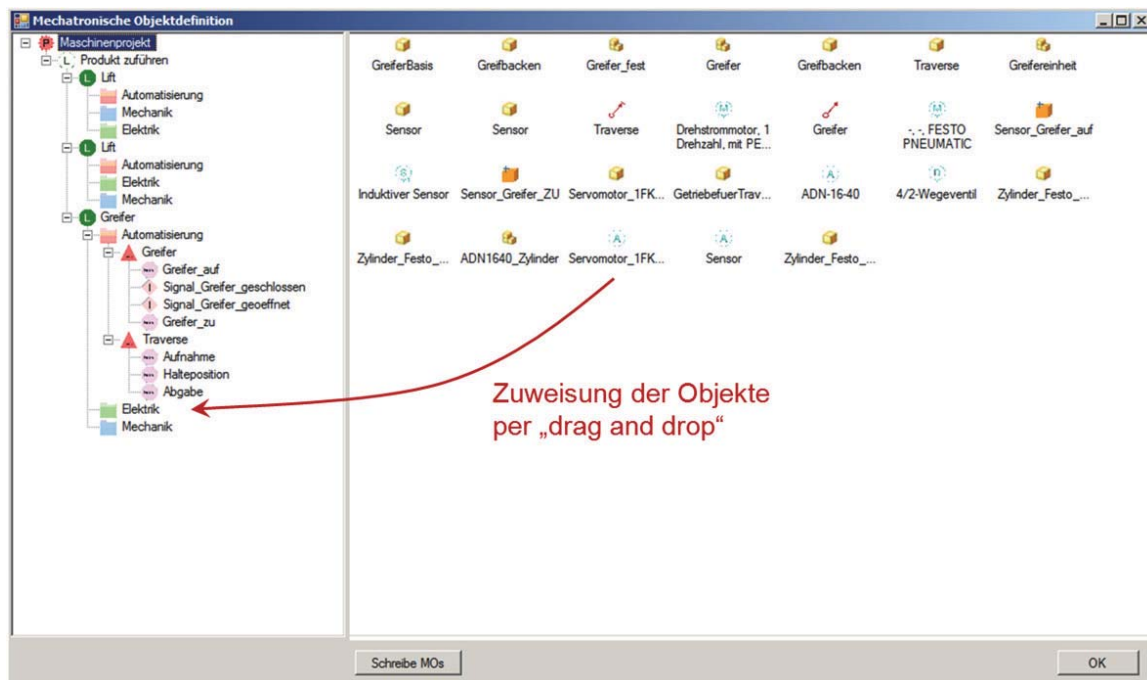


Abbildung 6.9: Schneiden mechatronischer Objekte

6.3 Erkenntnisse der Validierung

6.3.1 Erkenntnisse aus methodischer Sicht

Die Anforderungsdefinition für eine Sondermaschine konnte aufgrund der gegebenen Sachmerkmaleitlinien alle notwendigen Informationen aus mechatronisch integrativer Sicht ermitteln. Die Analyse der disziplinspezifischen Vorgehensweise und Modellierungen ergab, dass eine parallele Entwicklung nur durch Einbindung aller Disziplinen bereits auf der logischen Ebene möglich ist. Dies führt auf einen interdisziplinären Systementwurf mit der aufgezeigten mechatronischen Systembeschreibung. Dadurch wird neben einem mechatronischen Optimum des Konzeptes auch ein gemeinsames Systemverständnis erreicht und die Aufgabenbeschreibung in der weiteren Detaillierung ausreichend genau definiert. Dabei konnten Entwickler der Automatisierungstechnik und Mechanik im Gegensatz zu denen der Elektrik gut mit abstrakten Funktionen umgehen. Der Übergang von der funktionalen zu logischen Beschreibung ist von iterativem Vorgehen geprägt. Dabei war häufig eine Benennung des logischen Objektes ohne Beschreibung dessen Wirkungsweise, also wie dies ein kinematisches Konzept umsetzt, schwierig. Dadurch ergeben sich in der Modellierung Sprünge, nicht nur zwischen den Disziplinen, sondern auch zwischen den Abstraktionsebenen. Gleichzeitig ebnen die logischen Objekte als Klassifizierung von

mechatronischen Lösungsprinzipien einen Einstieg in die mechatronische Wiederverwendung. Ein Ausschneiden der mechatronischen Objekte, basierend auf einem bestehenden Projekt, minimiert den Erstellungsaufwand. Eine Modellierung der Module vorab ist deshalb nicht notwendig. Ein Schneiden der mechatronischen Objekte, um wirklich wiederverwendbare Module zu erhalten, ist häufig vom Einzelfall abhängig. Dazu ist ein zusammenhängendes mechatronisches Verständnis notwendig.

Im Rahmen der simultanen Ausdetaillierung wurde ein sogenannter Push-Mechanismus von Informationen umgesetzt, um für eine konsistente interdisziplinäre Modellierung über alle Disziplinen zu sorgen. Die Anwender behalten hierbei immer noch ihren Handlungsspielraum, um Entscheidungen über Auswirkungen auf ihr Modell zu übernehmen oder abzuweisen. Dadurch werden die Auswirkungen disziplinspezifischer Arbeiten auf andere Disziplinen sichtbar. Dies fördert auch die interdisziplinäre Zusammenarbeit. Außerdem wird eine solche weiterhin durch das gemeinsame Systemverständnis und eine gemeinsame logische Systemmodellierung unterstützt. Gerade das gemeinsame Systemverständnis kann auch auf die frühzeitige kinematische Simulation zurückgeführt werden, welche zudem direkt die automatisierungstechnische Beschreibung mit einbezieht. Gleichzeitig ermöglicht diese eine durchgängige Simulation der Planungen.

Das generierte Verhaltensmodell auf Basis der Schaltpläne und den entsprechenden Details kann für einen äußerst umfangreichen Softwaretest herangezogen werden. Grund hierfür ist u.a., dass auch die Energiebereitstellung für viele Geräte mit berücksichtigt ist. Dabei ist eine Modellierung des Komponentenverhaltens oftmals auf Gerätetypenebene möglich, was den Modellerstellungsaufwand deutlich reduziert.

Gespiegelt an den aufgestellten Anforderungen ist eine sehr gute Erfüllung derer durch die Methode festzustellen. Die Bewertung ist in Abbildung 6.10 dargestellt.

Anforderungsdefinition	Systementwurf		Detaillierung		Simulationsunterstützung entlang des gesamten Prozesses	Wiederverwendung von mechatronischen Modulen	Nutzung eingeführter Werkzeuge und Beschreibungsmittel
Integration in Produktionssystem	abstrakte Problembeschreibung	Interdisziplinäre Maschinenbeschreibung und Detaillierung in allen Disziplinen	Konsistenzabgleich				
●	●	●	●	●	●	●	◐

○ Keine Berücksichtigung ◐ Teilweise berücksichtigt ● Vollständig berücksichtigt

Abbildung 6.10: Erfüllung der aufgestellten Anforderungen durch die in dieser Arbeit entwickelte Methode

Prinzipiell sind alle Beschreibungssprachen aus der gängigen Lehre bekannt. Die Kombination zur Beschreibung der funktionalen Ebene führt allerdings auf notwendige Schulungen. Diese sind auch deshalb notwendig, da im Rahmen der methodischen Konstruktion keine explizite funktionale Modellierung durchgeführt wird. Bezogen auf die in Kapitel 2.7.1 abgeleitete Anforderung zur ausschließlichen Nutzung bestehender Beschreibungsmittel, um eine möglichst breite Akzeptanz zu erreichen, kann dies nicht als „vollständig berücksichtigt“ bewertet werden.

6.3.2 Erkenntnisse bei der Verknüpfung von Softwarewerkzeugen

Eine Verknüpfung der Softwarewerkzeuge sollte zu einem minimierten Aufwand und konsistenten Modellen der verschiedenen Disziplinen führen. Die erarbeitete Methode zeigt auf, wie alle notwendigen Beschreibungen durchgängig genutzt werden können. Allerdings kann festgestellt werden, dass für ein gemeinsames Systemverständnis auch die disziplinübergreifende Bereitstellung von Modellen in anderen Disziplinen notwendig ist. So benötigt die Elektrokonstruktion grundlegende Kenntnisse über den Aufbau der Maschine, welche üblicherweise in einem Technologie- oder Anlagenschema enden. Dieses muss derzeit von Hand erstellt werden, was allerdings auf die mangelnde Integrationsfähigkeit des genutzten ECAD-Systems zurückzuführen ist.

Hinsichtlich der Aufwandsreduktion werden, wenn möglich, existierende Dokumente oder Modelle genutzt um Generierungen durchzuführen. Dies vermindert den Aufwand gerade bei der Erstellung der Verhaltensmodelle für die virtuelle Inbetriebnahme erheblich. Die Implementierung der Sequence Designer-Anbindung an das Softwarewerkzeug Mechatronics Concept Designer zur Simulation zeigte, dass ein fließender Übergang zwischen zeitdauer- und eventbasierter Simulation möglich ist. Die Verknüpfung der Softwarewerkzeuge über das semantische Netz ermöglichte zudem die direkte Weiterverwendung von Informationen.

Gerade die Nutzung der logischen Ebene zur Klassifizierung der physikalischen Objekte öffnet die Möglichkeit neben größeren Modulen auch auf Komponentenebene Wiederverwendung zu betreiben. Um während der parallelisierten Entwicklung in den Disziplinen alle Informationsentstehungsmöglichkeiten zu unterstützen, um keine Informationen mehrmals eingeben zu müssen, war eine Ausprogrammierung aller möglichen Kombinationen der relevanten Knoten- und Kantentypen für jede Applikation notwendig. Dadurch besitzt der Anwender die Möglichkeit jeden Weg zur Lösung zu beschreiten. Dies berücksichtigen die Anwendungen selbst oft nicht. So lassen Applikationen häufig keine Änderung von Modellen zu, wie beispielsweise eine Typänderung eines Signals. Viele Softwarewerkzeuge zwingen den Anwender zunächst zum Löschen eines Signals und

anschließend zu einer Neuinstanziierung. Dies ist in Abbildung 6.11 dargestellt. In einer mechatronischen Umgebung führt dies zu zwei Notifikationen, eine zur Löschung und eine der Neuinstanziierung im mechatronischen Umfeld. Die Absicht des Anwenders ist allerdings nur eine Änderung. Folge ist eine Vielzahl von Benachrichtigungen.

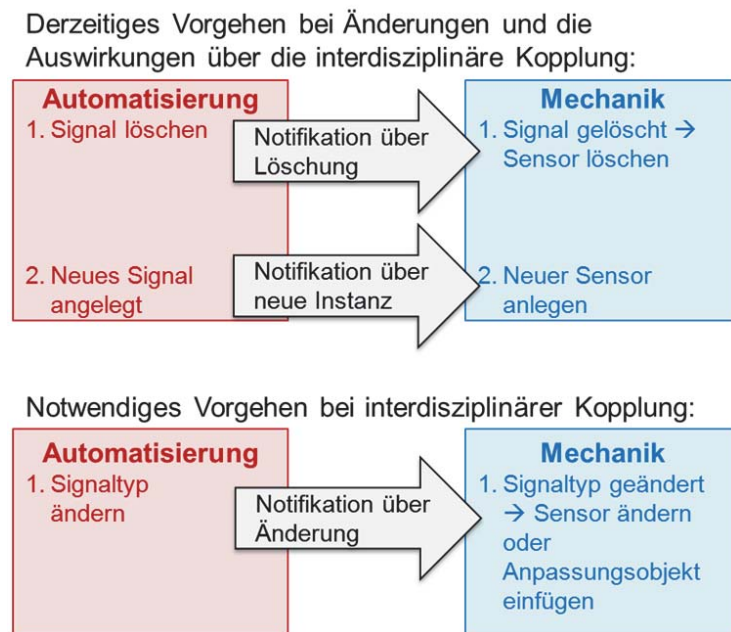


Abbildung 6.11: Integrierte Umgebungen verlangen nach mehr Änderungsmöglichkeiten

Die bisher implizit durchgeführten Modelltransformationen zwischen und in den Disziplinen müssen in einer optimierten Entwicklung zugänglich sein. Daher ist eine Veröffentlichung dieser notwendig. Gleichzeitig führt dies auf ein Vielfaches zusätzlicher Knoten- und Kantentypen und derer Instanzen in einem Projekt, was die Leistungsanforderungen an die Rechnerinfrastruktur deutlich erhöht.

7 Zusammenfassung und Ausblick

Kürzere Produktlebenszyklen und individuellere Produkte verlangen bei produzierenden Unternehmen eine schnellstmögliche Anpassung ihres Produktionssystems hinsichtlich einer optimierten Prozesskette. Durch neue Produktionsverfahren und integrierte Prozesse wird ein erhebliches Optimierungspotenzial freigesetzt. Dies führt zu einem steigenden Bedarf an kundenindividuellen Maschinen. Gleichzeitig wird ein Fachkräftemangel im Bereich der Konstruktion und Entwicklung vorausgesagt. Diese Situation führt dazu, dass die Sondermaschinenentwicklung höchst effizient gestaltet werden muss.

Im Wesentlichen können zwei Kategorien von Ansätzen zur schnelleren Maschinenentwicklung identifiziert werden. Die erste versucht, Maschinen strikt zu modularisieren, wodurch die Modulentwicklung von der Auftragsabwicklung getrennt wird. Eine Beschleunigung der Maschinenbereitstellung erfolgt durch Konfiguration verschiedener Module. Dabei muss eine kundenspezifische Anpassung bereits vorab abgebildet werden. Die zweite Kategorie fokussiert die Beschleunigung des Entwicklungsprozesses. Allerdings fehlt derzeit eine ganzheitliche Betrachtung für den Maschinenbau und insbesondere für den Sondermaschinenbau. Zudem vernachlässigen diese Ansätze die Rationalisierungspotenziale einer Wiederverwendung von bereits entwickelten Maschinenbestandteilen zur schnelleren Maschinenbereitstellung. Daher wurde in dieser Arbeit als Ziel die Entwicklung einer durchgängigen softwaregestützten Methode zur integrierten Sondermaschinenentwicklung gesetzt, wobei eine Parallelisierung der Disziplinen und eine Wiederverwendung von bereits entwickelten Modulen berücksichtigt werden sollte. Dazu wurden zunächst die derzeitige Entwicklung von Sondermaschinen betrachtet, Grundlagen erläutert und Begrifflichkeiten geklärt. Im Weiteren erfolgte eine Betrachtung der bestehenden problem-lösungsorientierten Ansätze der Produktentwicklung, der produktionsprozessbasierten Ansätze mit Ursprung in der Fabrikfeinplanung und den informationstechnischen Ansätzen zur Verknüpfung verschiedener Softwarewerkzeuge. Auf Basis der bestehenden Defizite bei der Entwicklung von Sondermaschinen wurden Anforderungen an eine solche Methode und deren Softwareunterstützung abgeleitet. Diese bildeten die Basis zur Bewertung der beschriebenen Ansätze und zur Ableitung des Handlungsbedarfs.

Darauf aufbauend wurden in einem ersten Schritt die disziplinspezifischen Entwicklungs- und Modellierungsvorgehen und deren Abhängigkeiten untereinander analysiert. Hierbei zeigte sich, dass die Modelle der Disziplinen unterschiedliche Abstraktionsgrade besitzen, welche in drei

Kategorien eingeteilt werden können, der funktionalen, der logischen und der physikalischen. Eine wirklich parallele Entwicklung in den verschiedenen Disziplinen ist möglich, wenn ein mechatronischer Systementwurf bereits auf logischer Ebene erfolgt, da hier die wesentlichen systemarchitektonischen Randbedingungen für alle Disziplinen definiert werden. Weiterhin wurden Abhängigkeiten auch in einer simultanen Ausdetaillierungsphase identifiziert.

Darauf aufbauend wurde eine Methode zur mechatronischen Entwicklung von Maschinen erarbeitet. Zunächst wird darin eine Anforderungserhebung anhand einer definierten Sachmerkmaleitlinie zur Unterstützung einer Integration der Maschine in eine bestehende Linie durchgeführt. Anschließend erfolgt eine funktionale Modellierung in Form eines statischen und dynamischen Modells zur Beschreibung der Gesamtaufgabenstellung. Die Suche nach möglichen Lösungsprinzipien markiert den Übergang zur logischen Ebene und gleichzeitig zum mechatronischen Systementwurf. Dabei bildet die Benennung der Lösungsprinzipien die Basis für eine mögliche Wiederverwendung von Modulen. Die eigentliche Moduldefinition geschieht auf Basis des aktuellen Entwicklungsprojektes durch Zuweisung der jeweiligen disziplinspezifischen Modelle zum Lösungsprinzip. Dieser Schritt muss unter Berücksichtigung mechatronischer Abhängigkeiten geschehen. Gleichzeitig fördert die Benennung des genutzten Lösungsprinzips im Rahmen eines Entwicklungsprojektes bereits existierende mechatronische Modulalternativen zu Tage, die im Kundenprojekt verwendet werden können. Modulneuentwicklungen werden, unterstützt durch Simulation, interdisziplinär so beschrieben, dass alle Disziplinen gleichberechtigt sind und diese Beschreibung nahtlos für die weitere Detaillierung nutzen können. Die Phase der Ausdetaillierung, simultan in den verschiedenen Disziplinen, ist begleitet von der Kommunikation interdisziplinär relevanter Informationen, welche zur mechatronischen Modellkonsistenz führt. Am Ende der Phase sind alle notwendigen Komponenten für den Bau der Maschine definiert und die Schaltpläne vorhanden. Anschließend erfolgt der Test des Steuerungsprogramms im Rahmen einer virtuellen Inbetriebnahme.

Unterstützt wird die Methode durch eine Verknüpfung am Markt erhältlicher Softwarewerkzeuge auf Basis eines semantischen Netzes. Methodenbestandteile, für welche keine Software erhältlich war, wurden durch im Rahmen dieser Arbeit entwickelten Prototypen softwaretechnisch unterstützt. Dabei wurde Wert auf eine Weiterverwendung bestehender Informationen gelegt. So kann beispielsweise das Verhaltensmodell der Maschine zur Simulation im Rahmen der virtuellen Inbetriebnahme auf Basis der Schaltpläne generiert werden. Allerdings zeigte sich in der prototypischen Umsetzung, dass zwar eine semantische Verknüpfung zur Unterstützung einer

mechatronischen Entwicklung möglich ist, eine praktikable Umsetzung allerdings durch fehlende Schnittstellen der Applikationen begrenzt wird.

Die Methode wurde anhand eines Industriebeispiels validiert. Die vorgestellte Arbeit liefert somit neben einer durchgängigen Methode zur Entwicklung von SPS-gesteuerten Sondermaschinen gleichzeitig ein Informationsmodell zur parallelen Entwicklung in den beteiligten Disziplinen. Dieses Modell ist auf alle SPS-gesteuerten Maschinen übertragbar.

Die analysierten Abhängigkeiten zwischen den Disziplinen gelten gerade im Kontext der Wiederverwendbarkeit auch im Rahmen konfigurationsbasierter Ansätze. Diese gehen gegenüber der dargestellten Vorgehensweise „Top-down“ und nicht „Bottom-up“ vor. Dadurch kann eine Vererbungshierarchie zwischen Modul und Alternativen beschrieben werden, was einer hohen Standardisierung entgegenkommt. Dies bietet für weitere Arbeiten den Freiraum zu untersuchen, ob der Aufbau einer solchen Vererbung auch „Bottom-up“ erfolgen kann. In diesem Zusammenhang ist gleichzeitig die Rolle eines „mechatronischen Standardisierers“ in der Maschinenentwicklung zu betrachten. Dessen Aufgaben beinhalten unter anderem Themen, wie ein mechatronisches Bibliotheksmanagement und eine Lifecyclebetrachtung mechatronischer Objekte.

Die vorliegende Arbeit befasst sich mit der Entwicklung von Maschinen. Zu Beginn des Entwicklungsprozesses werden Informationen aus der übergeordneten Ebene der Zellen- und Linienebene in Bezug auf die Fabrikstruktur benötigt. Dazu sind Erweiterungen hinsichtlich der Informationsbereitstellung im Rahmen der Linienplanung denkbar. Gleiches gilt für die Prozessentwicklung, welche die verfahrenstechnisch erreichbaren Parameter definiert. Außerdem ist die Einbeziehung weiterer involvierter Bereiche im Rahmen einer Maschinenentwicklung zu berücksichtigen, wie beispielsweise Informationen des Vertriebs oder Änderungen des Aufbaus während der Montage.

Aus organisatorischer Sicht ist zu überlegen, ob die disziplinspezifischen Entwickler und deren Softwarewerkzeuge eine gemeinsame Aufgabengrenze besitzen müssen. Im Beispiel der Verkabelung wäre es in Zukunft auch denkbar, dass bestimmte geometrische Aufgabenstellungen nicht mehr in der Verantwortung der mechanischen Konstruktion liegen. Dies führt zu Fragen im Bereich der Applikationsbereitstellung, Nutzung und der Organisation der Verantwortungsbereiche. Im Rahmen der informationstechnischen Ergebnisbeurteilung können weitere Forschungsaspekte aufgezeigt werden. So führen die bisher implizit und nun explizit durchgeführten Modelltransformationen zu einer vielfachen Knoten- und Kantenanzahl. Diese müssen in vertretbaren Antwortzeiten angefragt und ausgewertet werden. Weiterhin sind Usability-Aspekte einer

integrierten mechatronischen Entwicklung zu untersuchen. Konkret betrifft dies die Darstellung der Hinweise bezüglich Änderungen im mechatronischen Kontext. Eine Ausgestaltung einzelner Module direkt in deren Kontext, würde zu einer Definition von mechatronischen Objekten führen, ohne eine Phase der Moduldefinition explizit durchschreiten zu müssen. Dadurch entfällt ein Mehraufwand zur Modulerstellung.

Eine Umsetzung des dargestellten Ansatzes versetzt Maschinenhersteller somit in die Lage äußerst effizient Maschinen zu entwickeln. Gleichzeitig wird ermöglicht, unterschiedliche Geschäftsmodelle umzusetzen, beispielsweise durch eine möglichst schnelle Bereitstellung einer modular aufgebauten Maschine oder einer energetisch und kundenindividuell optimierten Maschine durch Nutzung der eingesparten Entwicklungszeit für Optimierungen. Dadurch wird es für produzierende Unternehmen einfacher die Produktion sehr schnell an die gegebenen Randbedingungen anzupassen und höchst effizient zu gestalten. Zudem kann die Maschinenentwicklung zukünftig völlig anders organisiert werden, da disziplinspezifische Applikationen in einem übergeordneten mechatronischen oder disziplinübergreifenden Maschinenkontext enger zusammenwachsen. Dies kann die eigentlichen Aufgaben der Entwickler entlang des Entwicklungsprozesses in den Vordergrund rücken, indem alle benötigten Informationen zugänglich sind und sofort weiterverwendet werden können.

Summary

Short Product Lifecycles and customized products force manufacturers to adapt and optimize their production systems as fast as possible. Thereby the optimization is focused on the process chain. Using new production methods and integrated processes, a huge potential for optimization is raised. This leads to an increasing demand of specialized production machines. Concurrently a skilled worker shortage for the area of engineering is predicted. This situation needs a highly efficient design for the engineering process of specialized production machines. Therefor two basic approaches for accelerating machine engineering can be identified. The first one focuses on modularizing and separating strictly the development of modules and the order procedure. The acceleration by providing the machine is hereby done through configuration of different modules. Thereby a customer specific adaption has to be described upfront. The second category tries to accelerate the engineering process itself. However, a holistic consideration regarding machine-building and in particular for the specialized production machines is missing. Additionally, these approaches neglect the rationalization potential of reusing already existing machine components and modules for a quite faster provision of the machine. Based on this, the goal of this thesis is to provide a seamless method for integrated specialized production machine engineering, considering the parallelization of disciplines and reuse of already existing modules. Therefor the current development of specialized machines was regarded, basics were annotated and the terminology was clarified. Further on, a consideration of existing approaches with focusing on solving problems coming from product development, production-process based approaches originated within factory planning and information technology for connecting different software-tools was done. Existing deficits have been worked out and have been formulated as requirements for the engineering method of specialized production machines and the needed software assistance. An Evaluation of the existing approaches was done and the demand for action was derived. Subsequently an analysis of the discipline specific methods and their dependencies to each other has been done. Thereby one result shows, that the models of the disciplines can be categorized into different degrees of abstractions, a functional, a logical and a physical one. A really parallelized engineering is possible, if the mechatronic concept is done on logical level, because there the basic system-architectural constraints are defined for all disciplines. Additionally dependencies between the disciplines within a parallel detailing phase have been identified. Based on that a method for mechatronic engineering of machines is presented. The first phase shows the requirements definition based on defined class

list of characteristics for integrating machines into existing production systems. Subsequently a static and dynamic functional model is used for describing the task. Searching for different alternatives of solution principles marks the transition from functional level to the logical one and coincidentally the mechatronic system design. The naming of the solution principles builds the base for reusing modules. The module definition is done within the project by assigning the discipline specific models to the principle. This step has to be done by considering mechatronic dependencies between disciplines. Additionally the naming of the principle is used by providing already existing alternative mechatronics modules to the user, which he is able to reuse within the project. If new modules are needed, they have to be described in an interdisciplinary way, which is supported by simulation. All disciplines have equal rights within the definition of new modules, so they are able to use the information seamlessly within the further detailing. This detailing-phase is done simultaneously within the different disciplines, accompanied by using communication of interdisciplinary relevant information to get a mechatronic consistency of the models. At the end of the phase all necessary components for building up the machine are defined and the wiring diagrams are available. Subsequently virtual commissioning is used to validate the controller program.

The method is assisted by connecting different commercial software-tools based on a semantic net. For parts of the method, where no software-tool was available, prototypes have been implemented and used. Thereby the reuse of existing information was focused. For example the behavior-model of the machine for virtual commissioning was generated based on wiring diagrams. Certainly, the semantic connection was possible as a prototype, but missing interfaces on different applications restrict a feasible usage.

The method was validated based on an industrial example. This thesis provides a seamless method for engineering PLC-driven specialized production machines and an information model for parallelizing disciplines. This model can be transferred to all PLC-controlled machines. The given dependencies between the disciplines are valid, especially within the context of reuse also within configuration based approaches. These approaches proceed in a way “Top-down” in comparison to the presented way “bottom-up”. Thereby a hierarchy for inheritance between module and alternatives can be described, which would lead to a high level of standardization. This can be used for further work, if such inheritances can be done also bottom-up. This association leads to the need of considering a role for standardizing in a mechatronic way for machine builders. Therein tasks

like the management of mechatronic libraries and the lifecycle of mechatronics objects can be summarized.

The presented thesis focuses on engineering machines. At the beginning of the engineering process information from overlaying levels of cells and lines, related to the structure of factories are needed. Therefor extensions regarding the provision of information within line planning are thinkable. The same is true for the development of production methods, where reachable parameters are defined.

Additionally there should be an information integration of other departments involved within machine engineering, like sales or production into account. Based on an organizational perspective it should be thought about, if the engineers and the used tools have to have a common border for tasks. For example in future it would be thinkable, that the responsibility of specific geometrical like wiring-tasks is not based within mechanical engineering anymore. This leads to questions on providing applications, using them, organization and areas of responsibilities.

Within the informatics results, further scientific aspects can be shown. Currently implicit and explicit model transformation leads to a huge amount of nodes and relations between them. They have to be asked and answered within reasonable response-times. Further on usability aspects within mechatronic engineering should be researched. Concretely notices of changes within the mechatronics context have to be regarded. A detailing of individual modules directly in their context, would lead to the definition of mechatronic ones, without an explicit phase of module definition. Based on that an additional effort for building modules is no longer required.

An application of the presented approach enables machine-builders to engineer machines most efficient. Simultaneously, this allows different business models, for example a model for providing a modular machine as fast as possible or another one for optimized machine in an energetic or customized way of using the shortened duration of engineering for optimizations. Thus enables manufacturing companies much easier to adapt their production very fast and efficient to given external constraints. Additionally the machine engineering can be organized quite different in future, because discipline specific applications grow together within an overlaying mechatronic and machine context. This enables the engineers to focus on the given tasks along the engineering process because all information needed are available and can be used directly.

Literaturverzeichnis

- Albers, et al. 2013 Albers, Albert; Zingel, Christian, 2013. Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML. In: Abramovici, Michael, Hrsg. *Smart product engineering: Proceedings of the 23rd CIRP Design Conference*, 11.-13.03.2013, Bochum. Berlin: Springer, S. 83–92.
ISBN 978-3-64-230816-1
- Alvarez Cabrera, et al. 2010 Alvarez Cabrera, Andres Alberto; Foeken, Marten.; Tekin, Aydin; Woestenenk, Krijn; Erden, Mustafa; De Schutter, Bart; van Tooren, Michael .J.L.; van Houten, Fred .J.A.M.; Tomiyama, Tetsuo, 2010. Towards automation of control software: A review of challenges in mechatronic design. *Mechatronics* [online]. **20** (7), S. 876–886.
ISSN 2196-677X
DOI 10.1016/j.mechatronics.2010.05.003
- Arnold, et al. 2012 Arnold, Peter; Rudolph, Stephan, 2012. Bridging the gap between product design and product manufacturing by means of graph-based design languages. In: Horváth, Imre Hrsg.: *Tools and methods of competitive engineering - Proceedings of the ninth International Symposium on Tools and Methods of Competitive Engineering*, 7.-11.05.2012, Karlsruhe. Delft: University of Technology, 2012.
ISBN 978-90-5155-081-8
- Barnes, et al. 2008 Barnes, Mark; Finch, Ellen Levy, April 2008. *Collada - Digital Asset Schema Release 1.5.0 – specification*. Clearlake Park: Khronos Group,
http://www.khronos.org/files/collada_spec_1_5.pdf (16.09.2013)
- Bathelt 2007 Bathelt, Jens, 2007. *Entwicklungsmethodik für SPS-gesteuerte mechatronische Systeme*, Düsseldorf: VDI-Verlag.

- ISBN 978-3-18-340520-6
Zürich, Eidgenössische Technische Hochschule, Diss., 2006
- Baudisch, et al. 2012 Baudisch, Thomas; Frank, Gernot; Schlögl, Wolfgang;
Westkämper, Engelbert, 2012. Concept for seamless engineering
tool chain for PLC-controlled specialized production machines. In:
International Aset, Hrsg. *Proceedings of International Conference
on Mechanical Engineering and Mechatronics*, 16.-18.08.2012,
Ottawa, 2012, S. 192-1-192-8
- Bauernhansl 2012 Bauernhansl, Thomas, 2012. Wie der Wandel gelingt -
Nachhaltigkeit als Treiber der Markt- und Ressourcenstrategie.
In: Gesellschaft für Fertigungstechnik, Hrsg. *Stuttgarter Impulse -
Produktionstechnik für den Wandel: FTK 2012 -
Fertigungstechnisches Kolloquium*, 25.-26.09.2012, Stuttgart,
S. 271–294.
ISBN 978-3-925589-59-1
- Bayrak, et al. 2007 Bayrak, G.; Wannagat, Andreas; Vogel-Heuser, Birgit, 2007.
Echtzeit- und Regelungstechnische Aspekte bei der automatischen
Transformation von Matlab/Simulink in SPS-basierten
Steuerungscode. In: Holleczeck, Peter, Hrsg. *Mobilität und Echtzeit:
Fachtagung der GI-Fachgruppe Echtzeitsysteme*, 6.-7.12.2007,
Boppard. Berlin: Springer.
DOI 10.1007/978-3-540-74837-3_5
- Bayrhammer, et al. 2013 Bayrhammer, Eric; Kennel, Matthias, 2013. Virtuelles Teachen zur
Erstellung von Steuerungsprogrammen. In: Schenk, Michael, Hrsg.
*Digitales Engineering zum Planen, Testen und Betreiben
technischer Systeme*: 10. Fachtagung - IFF-Wissenschaftstage,
18.06.-20.06.2013, Magdeburg. Stuttgart: Fraunhofer Verlag,
S. 63–68
- Beek, et al. 2010 Beek, Thom van; Erden, Mustafa; Tomiyama, Tetsuo, 2010.
Modular design of mechatronic systems with function modeling.
Mechatronics [online]. **20** (7), S. 850–863

- ISSN 0957-4158
DOI 10.1016/j.mechatronics.2010.02.002
- Bellalouna 2009 Bellalouna, Fahmi, 2009. *Integrationsplattform für eine interdisziplinäre Entwicklung mechatronischer Produkte*. Bochum, Ruhr-Univ., Diss., 2009
- Bergert, et al. 2008 Bergert, Martin; Diedrich, Christian, 2008. Durchgängige Verhaltensmodellierung von Betriebsmitteln zur Erzeugung digitaler Simulationsmodelle von Fertigungssystemen. *atp - Automatisierungstechnische Praxis*. **50** (7), S. 61–66.
ISSN 2190-4111
- Bertsche, et al. 2007 Bertsche, Bernd; Bullinger, Hans.-Jörg, 2007. *Entwicklung und Erprobung innovativer Produkte - Rapid Prototyping - Grundlagen, Rahmenbedingungen und Realisierung*. Berlin: Springer.
ISBN 978-3-540-69879-1
- Betschon, et al. 2008 Betschon, Lukas; Dierssen, Stefan, 2008. Die virtuelle Maschine als Kommunikationsbrücke in der mechatronischen Produktentwicklung. In: Kompetenznetzwerk Mechatronik BW e.V., genehmigte Sonderausgabe, Hrsg. *Internationales Forum Mechatronik*, 22.-23.09.2008, Stuttgart. Göppingen, S. 256–268.
ISBN 978-3-00-025245-7
- Blass 1997 Blass, Eckhart, 1997. *Entwicklung verfahrenstechnischer Prozesse - Methoden, Zielsuche, Lösungssuche, Lösungsauswahl*. 2. Aufl., Berlin: Springer.
ISBN 3-540-61823-6
- Bleisteiner 2010 Bleisteiner, Klaus, 2010. *Die Vision wird Mission - Generierung von Automatisierungsgewerken*. Vortrag im Rahmen der Veranstaltung: Siemens AG: Comos Anwendertage

- Blumauer, et al. 2006 Blumauer, Andreas; Pellegrini, Tassilo, 2006. Semantic Web und semantische Technologien: Zentrale Begriffe und Unterscheidungen. In: Pellegrini, Tassilo, Hrsg. *Semantic Web - Wege zur vernetzten Wissensgesellschaft*. Berlin: Springer. ISBN 3-540-29324-8, S. 7–25
- Böck 2013 Böck, Jochen, 2013. *Methode zur Integration nicht-konventioneller Verfahren in flexible Fertigungs- und Montagelinien*. Stuttgart: Fraunhofer Verlag. ISBN 978-3-8396-0590-5
Stuttgart, Univ., Diss., 2013
- Botaschanjan, et al. 2009 Botaschanjan, Jewgenij; Hummel, Benjamin; Lindworsky, Alexander, 2009. Interdisziplinäre Funktionsmodellierung im Anlagenbau. *ZWF - Zeitschrift für wissenschaftlichen Fabrikbetrieb*. **104** (1-2), S. 71–74. ISSN 0947-0085
- Brandes, et al. 2002 Brandes, Ulrik; Eiglsperger, Markus; Herman, Ivan; Himsolt, Michael; Marshall, Scott, 2002. GraphML Progress Report - Structural Layer Proposal. In: Mutzel, Petra, Hrsg. *Proceedings of 9th International Symposium - Graph Drawing*, 23.-26.09.2001, Wien. Berlin: Springer, 2002, S. 501–512. DOI 10.1007/3-540-45848-4_59
- Brecher, et al. 2008 Brecher, C.; Fedrowitz, C.; Herfs, W.; Kahmen, H.; Lohse, W.; Rathjen, O.; Vitr, M., 2008. Durchgängiges Production Engineering - Potentiale der digitalen Fabrik. In: Brecher, Christian, Hrsg. *Wettbewerbsfaktor Produktionstechnik: Aachener Perspektiven Aachener Werkzeugmaschinenkolloquium*, 05.-06.06.2008, Aachen. Aachen: Apprimus, 2008, S. 231–261. ISBN 978-3-940565-04-4

- Brecher, et al. 2013 Brecher, Christian; Özdemir, Denis; Ecker, Christian; Eilers, Jan; Lohse, Wolfram, 2013. Modellbasierte Rekonfigurierbarkeit. *wt Werkstattstechnik online* [online]. **103** (2), S. 157–161. ISSN 1436-4980
- Brudniok 2007 Brudniok, Sven, 2007. *Methodische Entwicklung hochintegrierter mechatronischer Systeme am Beispiel eines humanoiden Roboters*. Karlsruhe, Univ., Diss., 2007
- Campean, et al. 2013 Campean, Felician; Henshall, Edwin; Yildirim, Unal; Uddin, Amad; Williams, Huw, 2013. A Structured Approach for Function Based Decomposition of Complex Multi-disciplinary Systems. In: Abramovici, Michael, Hrsg. *Smart product engineering: Proceedings of the 23rd CIRP Design Conference, 11.-13.03.2013*, Bochum. Berlin: Springer, S. 113–122. ISBN 978-3-64-230816-1
- Conrad 2010 Conrad, Jan, 2010. *Semantische Netze zur Erfassung und Verarbeitung von Informationen und Wissen in der Produktentwicklung*. ISBN 3-930429-78-0 Saarbrücken, Univ. des Saarlandes, Diss., 2010
- DIN 1219 Blatt 1 DIN ISO 1219 Blatt 1 2007-12. *Fluidtechnik – Graphische Symbole und Schaltpläne – Teil 1: Graphische Symbole für konventionelle und datentechnische Anwendungen*
- DIN 1325 Blatt 1 DIN EN 1325 Blatt 1 1996-11. *Value Management; Wertanalyse; Funktionenanalyse Wörterbuch*
- DIN 7000 DIN ISO 7000, 2008-12. *Graphische Symbole auf Einrichtungen – Index und Übersicht*
- DIN 10303-210 DIN EN ISO 10303-210, 2002-12. *Produktdatenerstellung und -austausch*
- DIN 60617 Blatt 2 DIN EN 60617 Blatt 2 1997-08. *Graphische Symbole für Schaltungsunterlagen*

- DIN 61131 Blatt 3 DIN EN 61131 Blatt 3 2003-12. *Speicherprogrammierbare Steuerungen - Teil 3: Programmiersprachen*
- DIN 61346 Blatt 1 DIN EN 61346 Blatt 1 1997-01. *Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte; Strukturierungsprinzipien und Referenzkennzeichnung - Teil 1: Allgemeine Regeln*
- DIN 61346 Blatt 2 DIN EN 61346 Blatt 2 2000-12. *Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte; Strukturierungsprinzipien und Referenzkennzeichnung - Teil 2: Klassifizierung von Objekten und Kodierung von Klassen*
- DIN 62424 DIN EN 62424, 2010-01. *Darstellung von Aufgaben der Prozessleittechnik – Fließbilder und Datenaustausch zwischen EDV-Werkzeugen zur Fließbilderstellung und CAE-Systemen*
- DIN 81346 Blatt 1 DIN EN 81346 Blatt 1 2010-05. *Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte – Strukturierungsprinzipien und Referenzkennzeichnung - Teil 1: Allgemeine Regeln*
- DIN 81346 Blatt 2 DIN EN 81346 Blatt 2 2010-05. *Industrielle Systeme, Anlagen und Ausrüstungen und Industrieprodukte – Strukturierungsprinzipien und Referenzkennzeichnung - Teil 2: Klassifizierung von Objekten und Kodierung von Klassen*
- Diedrich, et al. 2011 Diedrich, Christian; Lüder, Arndt; Hundt, Lorenz, 2011. Bedeutung der Interoperabilität bei Entwurf und Nutzung von automatisierten Produktionssystemen. *at - Automatisierungstechnik*. **59** (7), S. 426–439.
ISSN 2196-677X
DOI 10.1524/auto.2011.0937
- Dohmen 2002 Dohmen, Winfried, 2002. *Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer Systeme*, München: Herbert Utz.
ISBN 3-8316-0214-X
München, Techn. Univ., Diss., 2002

- Doll 2010
Doll, Ullrich, 2010. Einsatz von mechatronischen Designwerkzeugen der digitalen Fabrik für die Anlagenprojektierung. In: Gesellschaft für Fertigungstechnik, Hrsg. *Stuttgarter Impulse - Fertigungstechnik für die Zukunft: FTK 2010 - Fertigungstechnisches Kolloquium*, 29.-30.09.2010, Stuttgart, S. 631–648.
ISBN 978-3-925589-51-5
- Drath 2010
Drath, Rainer, 2010. *Datenaustausch in der Anlagenplanung mit AutomationML - Integration von CAEX, PLCopen XML und COLLADA*, Berlin: Springer.
ISBN 978-3-642-04673-5
- Drath, et al. 2011a
Drath, Rainer; Barth, Mike, 2011. Concept for interoperability between independent engineering tools of heterogeneous disciplines. In: Mammeri, Zoubir, Hrsg. *16th Conference on Emerging Technologies & Factory Automation*, 05.-09.09.2011, Toulouse.
ISBN 978-1-4577-0017-0
- Drath, et al. 2011b
Drath, Rainer; Fay, Alexander; Barth, Mike, 2011. Interoperabilität von Engineering Werkzeugen. *at - Automatisierungstechnik*. **59** (7), S. 451–460.
ISSN 2196-677X
DOI 10.1524/auto.2011.0938
- Eberle 2012
Eberle, Eckhard, 2012. Herausforderungen und Chancen für die Produktionsautomatisierung. In: Gesellschaft für Fertigungstechnik, Hrsg. *Stuttgarter Impulse - Produktionstechnik für den Wandel: FTK 2012 - Fertigungstechnisches Kolloquium*, 25.-26.09.2012, Stuttgart, S. 1–10.
ISBN 978-3-925589-59-1
- ecl@ss e.V.
ecl@ss e.V.: *eCl@ss - Classification and Product Description*. 2013. Köln: ecl@ss e.V.,

- <http://www.eclass.de/eclasscontent/standard/overview.html.de>
(16.09.2013)
- EPLAN 2013 EPLAN, Software & Service GmbH & Co. KG, 2013: *EPLAN Engineering Center. Die Basis für funktionales Engineering*. Monheim am Rhein: EPLAN Software & Service GmbH & Co. KG, <http://www.eplan.de/de/loesungen/mechatronik/eplan-engineering-center/> (16.09.2013)
- Epple 2011 Epple, Ulrich Prof., 2011. Merkmale als Grundlage der Interoperabilität technischer Systeme. *at-Automatisierungstechnik*. **59** (7), S. 440–450.
ISSN 2190-4111
DOI 10.1524/auto.2011.0939
- Esswein, et al. 2010 Esswein, Werner; Lehrmann, Sina; Kunze, Günther; Gubsch, Ines; Penndorf, Timo, 2010. Integrative Produktentwicklung mit virtuellen Prototypen. *wt Werkstattstechnik online* [online]. **100** (1/2), S. 30–36.
ISSN 1436-4980
- Faatz 2004 Faatz, Andreas, 2004. *Ein Verfahren zur Anreicherung fachgebietspezifischer Ontologien durch Begriffsvorschläge*. Darmstadt, Univ., Diss., 2004
- Farines, et al. 2011 Farines, Jean-Marie; da Rocha, Vinicius Gravina; Carpes, Ana Maria M.; Vernadat, Francois; Cregut, Xavier, 2011. A model-driven engineering approach to formal verification of PLC programs. In: Mammeri, Zoubir, Hrsg. *16th Conference on Emerging Technologies & Factory Automation*, 05.-09.09.2011, Toulouse.
ISBN 978-1-4577-0017-0
- Flath 2002 Flath, Martin, 2002. *Methode zur Konzipierung mechatronischer Produkte*. Paderborn: Heinz Nixdorf-Institut.
ISBN 3-935433-17-4
Paderborn, Univ., Diss., 2002

- Frank, et al. 2011 Frank, Gernot; Schlögl, Wolfgang; Lenord, Matthias, 2011. Continuous Machine Design from Concept to Operation based on Mechatronic Objects. In: Bittanti, Sergio, Hrsg. *Proceedings of 18th IFAC-World Congress*, 28.08.-02.09.2011, Mailand, S. 1620-1621.
ISBN 978-3-902661-93-7
- Frank, et al. 2012 Frank, Gernot; Baudisch, Thomas; Schlögl, Wolfgang; Westkämper, Engelbert, 2012. Automatic model generation for virtual commissioning of specialized production machines. *Softwaretechnik Trends* [online]. **32** (2).
ISSN 0720-8928.
http://pi.informatik.uni-siegen.de/stt/32_2/03_Technische_Beitraege/MAT12/mat2012_submission_5.pdf (27.08.2013)
- Frank, et al. 2013a Frank, Gernot; Böck, Jochen; Westkämper, Engelbert; Schlögl, Wolfgang, 2013. A continuous method for integrating non-conventional technologies into existing production-lines from concept to resource. In: Gouvea da Costa, Sergio E., Hrsg. *Challenges for Sustainable Operations: 22nd International Conference on Production Research*, 28.07.-01.08.2013, Iguassu Falls, Brasilien
- Frank, et al. 2013b Frank, Gernot; Lenord, Matthias; Schlögl, Wolfgang; Westkämper, Engelbert, 2013. Durchgängige Taktzeitplanung von SPS-gesteuerten Sondermaschinen. In: Schenk, Michael, Hrsg. *Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme: 10. Fachtagung - IFF-Wissenschaftstage*, 18.06.-20.06.2013, Magdeburg. Stuttgart: Fraunhofer Verlag, S. 55–62
- Frank, et al. 2013c Frank, Gernot; Westkämper, Engelbert; Schlögl, Wolfgang; Lenord, Matthias, 2013. System design of PLC-controlled specialized production machines. In: Abramovici, Michael, Hrsg. *Smart product engineering: Proceedings of the 23rd CIRP Design*

- Conference, 11.-13.03.2013, Bochum. Berlin: Springer,
S. 613-622.
ISBN 978-3-64-230816-1
- Frank, et al. 2013d Frank, Timo; Eckert, Karin; Hadlich, Thomas; Fay, Alexander;
Diedrich, Christian; Vogel-Heuser, Birgit, 2013. Erweiterung des
V-Modells für den Entwurf von verteilten
Automatisierungssystemen. *at - Automatisierungstechnik*. **61** (2),
S. 79–91.
ISSN 2196-677X
DOI 10.1524/auto.2013.0009
- Gehrke 2005 Gehrke, Matthias, 2005. *Entwurf mechatronischer Systeme auf
Basis von Funktionshierarchien und Systemstrukturen*.
Paderborn, Univ., Diss., 2005
- Göhner, et al. 2013 Göhner, Peter; Rauscher, Michael, 2013. Konsistenzprüfung im
frühen mechatronischen Entwurf. *at - Automatisierungstechnik*.
61 (2), S. 109–113.
ISSN 2196-677X
DOI 10.1524/auto.2013.0012
- Grätz 2006 Grätz, Florian M., 2006. *Teilautomatische Generierung von
Stromlauf- und Fluidplänen für mechatronische Systeme*.
München: Herbert Utz.
ISBN 3-8316-0643-9
München. Techn. Univ., Diss., 2006
- Grundig 2006 Grundig, Claus-Gerold, 2006. *Fabrikplanung - Planungssystematik
- Methoden - Anwendungen*. 2., aktual. Aufl., München: Hanser.
ISBN 3-446-40642-5
- Haberfellner, et al. 2002 Haberfellner, Reinhard; Daenzer, Walter F., 2002. *Systems
engineering - Methodik und Praxis*. 11. Aufl., Zürich: Verlag
Industrielle Organisation.
ISBN 3-85743-998-X

- Hehenberger, et al. 2010 Hehenberger, Peter; Poltschak, F.; Zeman, Klaus; Amrhein, W., 2010. Hierarchical design models in the mechatronic product development process of synchronous machines. *Mechatronics* [online]. **20** (7), S. 864–875
ISSN 0957-4158
DOI 10.1016/j.mechatronics.2010.04.003
- Helmis 2009 Helmis, Steven, 2009. *Webbasierte Datenintegration - Ansätze zur Messung und Sicherung der Informationsqualität in heterogenen Datenbeständen unter Verwendung eines vollständig webbasierten Werkzeuges*. Wiesbaden: Vieweg+Teubner.
ISBN 978-3-8348-9280-5
- Hensel 2011 Hensel, Thomas, 2011. *Modellbasierter Entwicklungsprozess für Automatisierungslösungen*. München: Herbert Utz.
ISBN 978-3-8316-4167-3
München, Techn. Univ., Diss., 2011
- Hergula 2003 Hergula, Klaudia, 2003. *Daten- und Funktionsintegration durch föderierte Datenbanksysteme*. Univ. Kaiserslautern, Fachbereich Informatik, Daten.
ISBN 978-3-9368-9028-0
Kaiserslautern, Techn. Univ., Diss., 2003
- Hundt 2012 Hundt, Lorenz, 2012. *Durchgängiger Austausch von Daten zur Verhaltensbeschreibung von Automatisierungssystemen*, Berlin: Logos-Verlag.
ISBN 978-3-8325-3258-1
Magdeburg, Univ., Diss., 2012
- Hüsemann, et al. 2000 Hüsemann, Bodo; Lechtenböcker, Jens; Vossen, Gottfried, 2000. Conceptual Data Warehouse Design. In: *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW)*, 05.-06.06.2000, Stockholm, Schweden, S. 3–9. <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-28/paper6.pdf> (16.09.2013)

- Kallmeyer 1998 Kallmeyer, Ferdinand, 1998. *Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme*. Paderborn: Heinz Nixdorf-Institut.
ISBN 3-931466-41-8
Paderborn, Univ., Diss., 1998.
- Kemper, et al. 2010 Kemper, Hans-Georg; Baars, Henning; Mehanna, Walid, 2010. *Business Intelligence - Grundlagen und praktische Anwendungen - Eine Einführung in die IT-basierte Managementunterstützung*. 3. Aufl. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage.
ISBN 978-3-8348-9727-5
- Kemper, et al. 2011 Kemper, Alfons; Eickler, André, 2011. *Datenbanksysteme - Eine Einführung*. 8. Aufl. München: Oldenbourg.
ISBN 978-3-486-59834-6
- Kiefer 2007 Kiefer, Jens, 2007. *Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau*.
ISBN 978-3-930429-72-1
Saarbrücken, Univ. des Saarlandes, Diss., 2007
- Kiefer, et al. 2009 Kiefer, Jens; Ollinger, Lisa; Bergert, Martin, 2009. Virtuelle Inbetriebnahme - Standardisierte Verhaltensmodellierung mechatronischer Betriebsmittel im automobilen Karosseriebau. *atp - Automatisierungstechnische Praxis*. **60** (7), S. 40–46.
ISSN 2190-4111
- Kleiner 2003 Kleiner, Sven, 2003. *Föderatives Informationsmodell zur Systemintegration für die Entwicklung mechatronischer Produkte*. Aachen: Shaker.
ISBN 3-8322-2284-7
Darmstadt, Techn. Univ., Diss., 2003
- Kleiner 2013 Kleiner, Sven, 2013. Model Based Design with Systems Engineering Based on RFLP Using V6. In: Abramovici, Michael, Hrsg. *Smart product engineering: Proceedings of the 23rd CIRP Design Conference*, 11.-13.03.2013, Bochum.

- Berlin: Springer, S. 93-102.
ISBN 978-3-64-230816-1
- Klemm, et al. 2006 Klemm, Peter; Rüdele, H.; Weimer, Tobias, 2006. Durchgängiges Engineering von mechatronischen Systemen. In: Gesellschaft für Fertigungstechnik, Hrsg. *Stuttgarter Impulse - Fertigungstechnik für die Zukunft*: FTK 2006 - Fertigungstechnisches Kolloquium, 20.-21.09.2006, Stuttgart, S. 101–131.
ISBN 3-00-019764-8
- Kövari 2011 Kövari, Lars, 2011. *Konzeption und Realisierung eines neuen Systems zur produktbegleitenden virtuellen Inbetriebnahme komplexer Förderanlagen*. Karlsruhe: KIT Scientific Publishing.
ISBN 978-3-86644-624-3
Karlsruhe, Karlsruher Institut für Technologie, Diss., 2010
- Kufner 2012 Kufner, Annika, 2012. *Automatisierte Erstellung von Maschinenmodellen für die Hardware-in-the-Loop-Simulation von Montagemaschinen*. Heimsheim: Jost-Jetter.
ISBN 978-3-939890-94-2
Stuttgart, Univ., Diss., 2011
- Kühnle 2010 Kühnle, Hermann, 2010. *Distributed manufacturing*, London: Springer.
ISBN 978-1-84882-706-6
- Lacour 2012 Lacour, Frédéric-Felix, 2012. *Modellbildung für die physikbasierte Virtuelle Inbetriebnahme materialflussintensiver Produktionsanlagen*. München: Herbert Utz.
ISBN 978-3-8316-4162-8
München, Techn. Univ., Diss., 2011
- Lechler, et al. 2008 Lechler, Armin; Kircher, Christian; Verl, Alexander, 2008. Simulationsgestütztes mechatronisches Engineering. *wt Werkstattstechnik online* [online]. **98** (5), S. 377–383.
ISSN 1436-4980

- Leonard, 2001 Leonard, John, 2001. *Systems engineering fundamentals: supplementary text*, Fort Belvoir, Virginia: Defense Acquisition University Press
- Lercher 2008 Lercher, Bernd, 2008. *Konzeption und System einer Integrationsplattform zur Entwicklung von Werkzeugmaschinen* München, Techn. Univ., Diss., 2008
- Lindworsky 2008 Lindworsky, Alexander, 2008. Virtuelle Inbetriebnahme - Nutzenmaximierung durch Automatisierung der Modellerstellung. In: Kompetenznetzwerk Mechatronik BW e.V., genehmigte Sonderausgabe, Hrsg. *Internationales Forum Mechatronik*, 22.-23.09.2008, Stuttgart. Göppingen, S. 410–424. ISBN 978-3-00-025245-7
- Lindworsky 2011 Lindworsky, Alexander, 2011. *Teilautomatische Generierung von Simulationsmodellen für den entwicklungsbegleitenden Steuerungstest*. München: Herbert Utz. ISBN 978-3-8316-4125-3 München, Techn. Univ., Diss., 2011
- Litto 2004 Litto, Marco, 2004. *Baukastenbasiertes Engineering mit Föederal - Ein Leitfaden für Maschinen- und Anlagenbauer*, Frankfurt am Main: VDMA-Verlag. ISBN 3-8163-0478-8
- Litto 2010 Litto, Marco, 2010. *Aquimo - Adaptierbares Modellierungswerkzeug und Qualifizierungsprogramm für den Aufbau firmenspezifischer mechatronischer Engineeringprozesse - Ein Leitfaden für Maschinen- und Anlagenbauer*, Frankfurt am Main: VDMA-Verlag. ISBN 978-3-8163-0589-7
- Litto, et al. 2012 Litto, Marco; Buck, Raphael, 2012. Durchgängige Auftragsbearbeitungsprozesse für Maschinen und Anlagen vom Vertrieb bis zur Inbetriebnahme. In: Gesellschaft für Fertigungstechnik, Hrsg. *Stuttgarter Impulse - Produktionstechnik*

- für den Wandel: FTK 2012 - Fertigungstechnisches Kolloquium*,
25.-26.09.2012, Stuttgart, S. 205–226.
ISBN 978-3-925589-59-1
- Lüder, et al. 2010 Lüder, Arndt; Hundt, Lorenz; Föhr, Matthias; Holm, Timo;
Wagner, Thomas; Zaddach, Jorgos-Johannes, 2010. Manufacturing
System Engineering with Mechatronical Units. In: Institute of
Electrical and Electronics Engineers, Hrsg. *15th IEEE-Conference
on Emerging Technologies and Factory Automation*,
13.-16.09.2010, Bilbao.
DOI 10.1109/ETFA.2010.5641167
- Lüder, et al. 2011 Lüder, Arndt; Föhr, Matthias; Hoffmann, Martin; Hundt, Lorenz;
Langer, Yvonne; Frank, Stefanie, 2011. Aggregation of
engineering processes regarding the mechatronic approach In:
Mammeri, Zoubir, Hrsg. *16th Conference on Emerging
Technologies & Factory Automation*, 05.-09.09.2011, Toulouse.
ISBN 978-1-4577-0017-0
- Müller, et al. 2009 Müller, Verena; Verl, Alexander; Haubelt, Andreas, 2009.
Baukastenbasiertes simulationsgestütztes Engineering. In: *A&D
Kompendium*. 2009-2010, S. 32–34.
<http://www.aud24.net/pi/index.php?StoryID=189&articleID=16407>
7 (26.08.2013)
- Müller, et al. 2010 Müller, Verena; Verl, Alexander, 2010. Wiederverwendbare
Modelle zur disziplin- und domänenübergreifenden Simulation. In:
Brecher, Christian, Hrsg. *Hybride Technologien in der Produktion*
Düsseldorf: VDI-Verlag, 2010, S. 9-21
- Nieding 2012 Nieding, Philipp, 2012. *Aufwandsoptimierte mechatronische
Simulation von Werkzeugmaschinen auf Basis mitwachsender
Modelle*. Aachen: Shaker.
ISBN 978-3-84401317-7
Karlsruhe, Karlsruher Institut für Technologie, Diss., 2012

- North 2011 North, Klaus, 2011. *Wissensorientierte Unternehmensführung - Wertschöpfung durch Wissen*. 5. Aufl., Wiesbaden: Gabler. ISBN 978-3-8349-2538-1
- Obersamer 2013 Obersamer, Lukas, 2013. Durchgängiges Engineering für den Maschinenbau – Schnittstellen Moderner Engineering Werkzeuge aus der Automatisierungstechnik. In: Schenk, Michael, Hrsg. *Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme*: 10. Fachtagung - IFF-Wissenschaftstage, 18.06.-20.06.2013, Magdeburg. Stuttgart: Fraunhofer Verlag, S. 13–18
- Pahl, et al. 2007 Pahl, Gerhard; Beitz, Wolfgang; Feldhusen, Jörg; Grote, Karl-Heinz, 2007. *Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung, Methoden und Anwendung*. 7. Aufl., Berlin: Springer. ISBN 3-540-34060-2
- Pörnbacher 2011 Pörnbacher, Clemens, 2011. *Modellgetriebene Entwicklung der Steuerungssoftware automatisierter Fertigungssysteme*. München: Herbert Utz. ISBN 978-3-83164108-6 München, Techn. Univ., Diss., 2011
- Possel-Dölken 2010 Possel-Dölken, Frank 2010. Mechatronische Modularisierung im Sondermaschinenbau. In: Gausemeier, Jürgen; Hrsg. *Entwurf mechatronischer Systeme*: 7. Paderborner Workshop Entwurf mechatronischer Systeme, 18.-19.03.2010, Paderborn, S. 357–370
- Pritschow 2006 Pritschow, Günter, 2006. *Einführung in die Steuerungstechnik*, München: Hanser. ISBN 3-446-21422-4
- Reichenberger 2010 Reichenberger, Klaus, 2010. *Kompendium semantische Netze - Konzepte, Technologie, Modellierung*. Heidelberg: Springer. ISBN 978-3-642-04315-4

- Reinhart, et al. 2007 Reinhart, Gunther; Hensel, Thomas; Lindworsky, Alexander; Spitzweg, Michael, 2007. Teilautomatisierter Aufbau von Simulationsmodellen. *wt Werkstattstechnik online* [online]. **97** (9), S. 663–667.
ISSN 1436-4980
- Reitbauer 2006 Reitbauer, Alois, 2006. IT Konsolidierung und Informationsintegration. In: Pellegrini, Tassilo, Hrsg. *Semantic Web - Wege zur vernetzten Wissensgesellschaft*. Berlin: Springer. ISBN 3-540-29324-8, S. 387–404
- Reuter, et al. 2010 Reuter, Axel; Müller, Verena; Verl, Alexander, 2010. Disziplinübergreifendes Engineering. *wt Werkstattstechnik online* [online]. **100** (5), S. 399–406.
ISSN 1436-4980
- Rodica Maga, et al. 2011 Rodica Maga, Camelia; Göhner, Peter; Jazdi, Nasser, 2011. Requirements on Engineering Tools for Increasing Reuse in Industrial Automation. In: Mammeri, Zoubir, Hrsg. *16th Conference on Emerging Technologies & Factory Automation*, 05.-09.09.2011, Toulouse.
ISBN 978-1-4577-0017-0
- Röhrig 2002 Röhrig, Martin, 2002. *Variantenbeherrschung mit hochflexiblen Produktionsendstufen*. Düsseldorf: VDI-Verlag.
ISBN 3-18-362202-5
Hannover, Univ., Diss., 2002
- Scheibler 2010 Scheibler, Thorsten, 2010. *Ausführbare Integrationsmuster*. Stuttgart, Univ., Diss., 2010.
<http://elib.uni-stuttgart.de/opus/volltexte/2010/5498>
- Schilke 2010 Schilke, Martin, 2010. *Einsatz von Produktdatenmanagement-Systemen im Sondermaschinenbau für die Automobilindustrie*. ISBN 978-3-930429-76-9
Saarbrücken, Univ. des Saarlandes, Diss., 2009

- Schlögl 2008
Schlögl, Wolfgang, 2008. Digitale Fabrik 2.0. In: Gesellschaft für Fertigungstechnik, Hrsg. *Stuttgarter Impulse - Fertigungstechnik für die Zukunft: FTK 2008 - Fertigungstechnisches Kolloquium*, 10.-11.09.2008, Stuttgart, S. 453–462.
ISBN 978-3-00-025623-3
- Schlögl 2011
Schlögl, Wolfgang, 2011. *Mit digitalem Engineering durchgängig von der digitalen Fabrik in den Anlagenbetrieb*. Vortrag im Rahmen der Veranstaltung: Graduate School of advanced Manufacturing Engineering (GSaME) der Univ. Stuttgart: Innovation und Produktion - Zukunftsfähigkeit durch neue Wege, 11.02.2011, Stuttgart
- Schreck 2007
Schreck, Bernhard, 2007. Integrierte Servoantriebe - Vorschub für den modularen Verpackungsmaschinenbau. In: Verl, Alexander, Hrsg. *Tagungsband zur SPS/IPC/Drives 2007*, 27.-29.11.2007, Nürnberg. Berlin: VDE-Verlag, S. 577–585.
ISBN 978-3-8007-3069-8
- Schreiber, et al. 2011
Schreiber, Werner; Zimmermann, Peter, 2011. *Virtuelle Techniken im industriellen Umfeld - Das AVILUS-Projekt - Technologien und Anwendungen*, Berlin: Springer.
ISBN 978-3-642-20636-8
- Seitz 2012
Seitz, Matthias, 2012. *Speicherprogrammierbare Steuerungen für die Fabrik- und Prozessautomation - Strukturierte und objektorientierte SPS-Programmierung, Motion Control, Sicherheit, vertikale Integration*. 3. Aufl., München: Fachbuchverlag Leipzig.
ISBN 978-3-446-43325-0
- Siemens 2010
Siemens AG, 2010. *Wie optimieren wir unsere Anlagenplanung ganzheitlich? - SIMATIC Automation Designer based on Comos hebt alle Potenziale im Digital Engineering, um Zeit und Kosten zu sparen*. Nürnberg.

- http://w3.siemens.dk/home/dk/dk/comos/comos_automation/Documents/SIMATIC_aut_designer.pdf (27.08.2013)
- Siemens 2011 Siemens Industry Software, 2011. *Process Simulate - Manufacturing process verification in powerful 3D environment*. Köln. <http://url9.de/LsO> (27.08.2013)
- Siemens AG, 2004 Siemens AG, 2004: *SIMIT: Software zum Testen von Steuerungsprogrammen*, Version 4, Computer-Software: Nürnberg: Siemens AG
- Siemens Industry Software, 2010a Siemens Industry Software, 2010. *Automation Designer based on a Comos: Software zur elektrischen und automatisierungstechnischen Planung, Konstruktion von Anlagen und Maschinen*, Version: 9.1 Service Pack 100, Computer Software: Essen: Siemens
- Siemens Industry Software, 2010b Siemens Industry Software, 2010: *Comos ET: Software zur elektrischen und automatisierungstechnischen Planung, Konstruktion von Anlagen und Maschinen*, Version: 9.1 Service Pack 100, Computer Software: Essen: Siemens AG
- Siemens PLM 2010 Siemens PLM Software, 2010. *Mechatronics Concept Designer - Ein funktionsorientierter Ansatz für den Maschinen- und Anlagenbau*. Köln. <http://url9.de/LsM> (27.08.2013)
- Siemens PLM 2011 Siemens PLM Software, 2011. *Open product lifecycle data sharing using XML - White Paper*. <http://url9.de/LsK> (27.08.2013)
- Siemens PLM Software, 2009a Siemens PLM Software, 2009: *Mechatronics Concept Designer: Software für eine funktionsorientierte mechatronische Maschinenentwicklung*, Version 8.5, Computer Software: Köln: Siemens PLM Software
- Siemens PLM Software, 2009b Siemens PLM Software, 2009: *NX-Modeling: Software zur mechanischen Konstruktion*, Version 8.5, Computer Software: Köln: Siemens PLM Software

- Silver 2011 Silver, Bruce, 2011. *BPMN Method and Style*. 2. Aufl. Aptos, California, Cody-Cassidy Press.
ISBN 978-0-9823681-1-4
- Spiegelberger 2011 Spiegelberger, Bernd, 2011. *Anwendergerechte Gestaltung mechatronischer Entwicklungsprozesse für kleine und mittlere Unternehmen im Maschinenbau*. Göttingen, Sierke.
ISBN 978-3-86844-359-2
München, Techn. Univ., Diss., 2010
- Steden, et al. 2013a Steden, Frank; Weyrich, Michael, 2013. Produktionssysteme modulbasiert simulieren. *wt Werkstattstechnik online* [online].
103 (2), S. 162–167.
ISSN 1436-4980
- Steden, et al. 2013b Steden, Frank; Weyrich, Michael, 2013. Automated Configuration of a Machine Simulation Based on a Modular Approach. In: Abramovici, Michael, Hrsg. *Smart product engineering: Proceedings of the 23rd CIRP Design Conference*, 11.-13.03.2013, Bochum. Berlin: Springer, S. 603-612.
ISBN 978-3-64-230816-1
- Stich, et al. 2011 Stich, Peter; Reinhart, Gunther; Wunsch, Georg, 2011. Effiziente Methoden zum digitalen Engineering im Maschinen- und Anlagenbau. In: Bender, Klaus Hrsg. *Tagungsband zur SPS/IPC/Drives 2011*, 22.-24.11.2011, Nürnberg. Berlin: VDE-Verlag, S. 159–167.
ISBN 978-3-8007-3379-8
- Stork, et al. 2011 Stork, Detlef; Kühn, Johannes, 2011. Maschinen modular in der Software abbilden. In: Bender, Klaus, Hrsg. *Tagungsband zur SPS/IPC/Drives 2011*, 22.-24.11.2011, Nürnberg. Berlin: VDE-Verlag, S. 223–232.
ISBN 978-3-8007-3379-8
- Suh 2001 Suh, Nam P., 2001. *Axiomatic design - Advances and applications*. New York, Oxford University Press.

- ISBN 0-19-513466-4
- Thramboulidis, et al. 2011 Thramboulidis, Kleanthis; Frey, Georg, 2011. Towards a Model-Driven IEC 61131-Based Development Process in Industrial Automation. *Journal of Software Engineering and Applications (JSEA)* [online]. **4** (4), S. 217–226.
ISSN 1945-3124
DOI 10.4236/jsea.2011.44024
- US PRO 1996 U.S. Product Data Association (US PRO), 1996. *IGES - Initial Graphics Exchange Specification - IGES 5.3*, Charleston.
http://www.uspro.org/documents/IGES5-3_forDownload.pdf/view
(27.08.2013)
- VDI 2206 VDI-Richtlinie 2206 2004-06. *Entwicklungsmethodik für mechatronische Systeme*.
- VDI 2221 VDI-Richtlinie 2221 1993-05. *Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*.
- VDI 2222 Blatt 1 VDI-Richtlinie 2222 Blatt 1 1997-06. *Konstruktionsmethodik - Methodisches Entwickeln von Lösungsprinzipien*.
- VDI 2803 Blatt 1 VDI-Richtlinie 2803 Blatt 1 1996-10. *Funktionenanalyse - Grundlagen und Methode*.
- VDI 3633 VDI-Richtlinie 3633 1996-11. *Simulation von Logistik-, Materialfluß- und Produktionssystemen - Begriffsdefinitionen*.
- VDI 3633 Blatt 8 VDI-Richtlinie 3633 Blatt 8 2007-04. *Simulation von Logistik-, und Materialfluss- und Produktionssystemen - Maschinennahe Simulation*.
- VDI/VDE 3695 Blatt 2 VDI/VDE-Richtlinie 3695 Blatt 2 2009-05. *Engineering von Anlagen - Evaluieren und optimieren des Engineerings - Themenfeld Prozesse*.
- VDI 5610 Blatt 1 VDI-Richtlinie 5610 Blatt 1 2009-03. *Wissensmanagement im Ingenieurwesen Grundlagen, Konzepte, Vorgehen*.

- VDW
Forschungsvereinigung Werkzeugmaschinen und
Fertigungstechnik e.V. VDW-Richtlinie, 2002.
Funktionsbeschreibung, Frankfurt am Main
- Vogel-Heuser, et al. 2005
Vogel-Heuser, Birgit; Witsch, Daniel; Katzke, Uwe, 2005.
Automatic Code Generation from a UML model to IEC 61131-3
and system configuration tools. In: Institute of Electrical and
Electronics Engineers (IEEE), Hrsg. *Proceedings of International
Conference on Control and Automation*, 26.-29.06.2005, Budapest,
S. 1034-1039.
ISBN 0-7803-9137-3
DOI 10.1109/ICCA.2005.1528274
- Voß 2012
Voß, Verena, 2012. *Wiederverwendbare Simulationsmodelle für
die domänen- und disziplinübergreifende Produktentwicklung*.
Heimsheim: Jost-Jetter.
ISBN 978-3-939890-98-0
Stuttgart, Univ., Diss., 2012
- Weilkiens 2006
Weilkiens, Tim, 2006. *Systems engineering mit SysML/UML -
Modellierung, Analyse, Design*. 1. Aufl. Heidelberg: dpunkt.Verlag
ISBN 3-89864-409-X
- Westkämper, et al. 2009
Westkämper, Engelbert; Jovane, Francesco; Williams, David,
2009. *The ManuFuture Road - Towards Competitive and
Sustainable High-Adding-Value Manufacturing*, Berlin: Springer.
ISBN 978-3-540-77011-4
- Westkämper 2012a
Westkämper, Engelbert, 2012. Engineering Apps.
wt Werkstattstechnik online [online]. **102** (10), S. 718–722.
ISSN 1436-4980
- Westkämper 2012b
Westkämper, Engelbert, 2012. Mit digitalen Werkzeugen den
Wachstumsbremsen begegnen.
wt Werkstattstechnik online [online]. **102** (3), S. 79.
ISSN 1436-4980

- Weyrich, et al. 2011a Weyrich, Michael; Klein, Benjamin; Laurowski, Martin; Wang, Yongheng, 2011. Mechatronic Engineering of Novel Manufacturing Processes Implemented by Modular and Sensor-Guided Machinery. In: Mammeri, Zoubir, Hrsg. *16th Conference on Emerging Technologies & Factory Automation*, 05.-09.09.2011, Toulouse.
ISBN 978-1-4577-0017-0
- Weyrich, et al. 2011b Weyrich, Michael; Steden, Frank; Wolf, Jochen; Scharf, Matthias, 2011, Identification of Mechatronic Units based on an Example of a Flexible Customized Multi Lathe Machine Tool. In: Mammeri, Zoubir, Hrsg. *16th Conference on Emerging Technologies & Factory Automation*, 05.-09.09.2011, Toulouse.
ISBN 978-1-4577-0017-0
- Weyrich, et al. 2012 Weyrich, Michael; Klein, Philipp, 2012. Modulbasiertes Engineering von Produktionsanlagen.
wt Werkstattstechnik online [online]. **102** (9), S. 592–597.
ISSN 1436-4980
- Wiendahl 2005 Wiendahl, Hans-Peter, 2005. Planung modularer Fabriken - *Vorgehen und Beispiele aus der Praxis*. München: Hanser.
ISBN 3-446-40045-1
- Wiendahl, et al. 2007 Wiendahl, Hans-Peter; ElMaraghy, Hoda; Nyhuis, Peter; Zäh, Michael; Wiendahl, Hans-Hermann; Duffie, Neil; Brieke, Michael, 2007. Changeable Manufacturing - Classification, Design and Operation. *CIRP Annals - Manufacturing Technology* [online]. **56** (2), S. 783–809.
ISSN 00078506
DOI 10.1016/j.cirp.2007.10.003
- Willmann 2011 Willmann, Christian, 2011. *Konzeption einer kontextbasierenden Wissensumgebung für die digitale Fabrik*. Aachen: Shaker.
ISBN 978-3-8440-0203-4
Kassel, Univ., Diss., 2010

- Witte 2012
Witte, Martin, 2012. System Engineering, Plant Engineering and Functional Models. *Softwaretechnik Trends* [online]. **32** (2). ISSN 0720-8928
http://pi.informatik.uni-siegen.de/stt/32_2/03_Technische_Beitraege/MAT12/mat2012_submission_15.pdf (27.08.2013)
- Wünsch 2008
Wünsch, Georg, 2008. *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. München: Herbert Utz. ISBN 978-3-8316-0795-2
München, Techn. Univ., Diss., 2008
- Würslin, et al. 2008
Würslin, Rainer; Wascher, Florian, 2008. Mechatronischer Baukasten - Vorgehensmodell und Modellierungsmethodik für ein mechatronisches Entwicklungswerkzeug. In: Kompetenznetzwerk Mechatronik BW e.V., genehmigte Sonderausgabe, Hrsg. *Internationales Forum Mechatronik*, 22.-23.09.2008, Stuttgart. Göppingen, S. 330–353. ISBN 978-3-00-025245-7
- yWorks, 2012
yWorks, 2012: *Yed Graph-Editor: Software zur Visualisierung von gerichteten Graphen*. Version: 3.9.2, Computer Software: Tübingen: yWorks
- Zäh, et al. 2006
Zäh, Michael; Wünsch, Georg; Hensel, Thomas; Lindworsky, Alexander, 2006. Feldstudie - Virtuelle Inbetriebnahme. *wt Werkstattstechnik online* [online]. **96** (10), S. 767–771. ISSN 1436-4980
- Zäh, et al. 2009
Zäh, Michael; Botaschanjan, Jewgenij; Hensel, Thomas; Hummel, Benjamin; Lindworsky, Alexander, 2009. Simulationsmodelle für die virtuelle Inbetriebnahme. *ATZproduktion*. **2** (5/6), S. 18–22. ISSN 1865-4908

Gerade in Hochlohnländern ist unter den Gesichtspunkten stark verkürzter Produktlebenszyklen und einer Individualisierung von Produkten eine immer größere Nachfrage an kundenspezifischen Maschinen auszumachen. Gleichzeitig steigt dort aber der Fachkräftemangel. Um dennoch kurze Auftragsdurchlaufzeiten bei Sondermaschinen zu erreichen, ist ein äußerst effizientes Engineering notwendig. Dazu wird eine Methode vorgestellt, welche von der Anforderungsaufnahme bis zur Inbetriebnahme der Maschine die beteiligten Disziplinen Mechanik, Elektrik und Automatisierung parallelisiert und eine Wiederverwendung von mechatronischen Modulen ermöglicht. Unterstützt wird die Methode durch eine softwaretechnische Verknüpfung der disziplinspezifischen Softwarewerkzeuge auf Basis eines semantischen Netzes.

ISBN 978-3-8396-0968-2



FRAUNHOFER VERLAG