

HELMUT SCHMIDT
UNIVERSITÄT

Universität der Bundeswehr Hamburg

Methoden zur simulationsbasierten Absicherung von Steuerungscode fertigungstechnischer Anlagen

Von der Fakultät für Maschinenbau
der Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg
zur Erlangung des akademischen Grades eines Doktor-Ingenieurs (Dr.-Ing.)

genehmigte

DISSERTATION

vorgelegt von

M. Systems Eng. Philipp Puntel Schmidt

aus São Paulo

Hamburg, 2017

Gutachter:

Prof. Dr.-Ing. Alexander Fay

Prof. Dr.-Ing. Christian Diedrich

Vorsitzender:

Prof. Dr. rer. nat. habil. Markus Bause

Tag der mündlichen Prüfung:

24.03.2017

Vorwort des Autors

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter an der Professur für Automatisierungstechnik der Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg von September 2012 bis Oktober 2016.

Mein erster Dank gilt Herrn Prof Dr.-Ing. Alexander Fay, der als fordernder Förderer maßgeblich zum Gelingen dieser Arbeit beigetragen hat. Die vielen Gespräche und auch Anregungen haben die Erstellung dieser Arbeit erst möglich gemacht. Für die Möglichkeit, das Themengebiet international auf Fachkonferenzen vorzustellen zu können sowie die Möglichkeit in diversen Fachauschüssen tätig zu sein, sei gedankt.

Herrn Prof. Dr.-Ing. Christian Diedrich danke ich für das Interesse an meiner Arbeit und die Übernahme des Zweitgutachtens. Herrn Prof. Dr. rer. nat. habil. Markus Bause danke ich für den Vorsitz der Prüfungskommission.

Den ehemaligen Projektpartnern des Forschungsprojekts HIL-IMA, namentlich Herrn Fabian Köslin, Herrn Stephan Diehl und Herrn Willy Riediger danke ich für die hervorragende Zusammenarbeit, die zum Gelingen dieser Arbeit beigetragen hat.

Ich danke zudem Herrn Dr.-Ing. Lars Christiansen und Frau Dr. Dr. Christina Schneider für die kritische Durchsicht der Arbeit. Allen ehemaligen Kollegen am Institut für Automatisierungstechnik danke ich für die vielen interessanten Stunden. Ein Dank sei zudem an meine Masteranden gerichtet.

Meinen Eltern danke ich für die immerwährende Unterstützung auf meinem Weg und den bedingungslosen Rückhalt, der mir meine wissenschaftliche Tätigkeit erst möglich gemacht hat. Tiefe Dankbarkeit gilt meiner Frau Shuling, die mich durch so manche Höhen und Tiefen begleitet hat und dabei so manches ertragen musste. Durch ihre Liebe, ihr Verständnis und ihre Geduld ist die Fertigstellung dieser Arbeit möglich gewesen.

Nürnberg, April 2017

Philipp Puntel Schmidt

Inhaltsverzeichnis

Abkürzungsverzeichnis.....	V
Abbildungsverzeichnis.....	VIII
Tabellenverzeichnis.....	XI
1 Einleitung.....	1
1.1 Motivation und Problemstellung.....	1
1.2 Zielsetzung und Aufbau der Arbeit	3
2 Bedeutung der Inbetriebnahme im Engineering-Prozess fertigungstechnischer Anlagen... 6	6
2.1 Engineering.....	6
2.2 Betrachtung des Engineering-Prozesses fertigungstechnischer Anlagen und beteiligte Gewerke.....	7
2.3 Modellbasiertes Engineering im Kontext des Engineering-Prozesses.....	10
2.4 Auswahl und Dimensionierung von Ressourcen während des Engineering-Prozesses....	16
2.5 Durch Gewerke erstellte Projektergebnisse während des Engineering-Prozesses	20
2.6 Verwendete Werkzeuge im Engineering-Prozess und deren Bedeutung	26
2.7 Aufwands- und Kostenverteilung in Engineeringprojekten.....	28
2.8 Die Inbetriebnahme von fertigungstechnischen Anlagen	30
2.9 Zusammenfassung der Schwachstellen und Defizite des Engineering-Prozesses.....	38
3 Simulation als Methode zur Unterstützung des Engineering-Prozesses	41
3.1 Simulationsbasiertes Testen im Engineering fertigungstechnischer Anlagen.....	41
3.2 Einsatz von Simulation im Rahmen einer virtuellen Inbetriebnahme	42
3.3 Technische Implementierung von Simulationsmodellen	49
3.4 Typische Simulationswerkzeuge	51
3.5 Aufkommende Fragestellungen im Kontext der Umsetzung einer virtuellen Inbetriebnahme	51
4 Stand der Forschung im Bereich virtuelle Inbetriebnahme.....	56
4.1 Ein Vergleich von Engineering-Vorgehen und Produktentwicklungsmethoden	56
4.2 Konventionelle Ansätze zur virtuellen Inbetriebnahme (ohne automatische Modellgenerierung).....	61
4.3 Ansätze mit teil-automatischer Modellgenerierung	73
4.4 Ansätze mit voll-automatischer Modellgenerierung.....	76
4.5 Vergleichende Bewertung	81
5 Handlungsbedarf und Lösungskonzept.....	83
5.1 Identifizierte Defizite	83
5.2 Zielformulierung und Haupttätigkeitsfelder.....	85
5.3 Lösungskonzept, Einschränkung des Scopes und technische Rahmenbedingungen	87

6	Datenmodell für eine virtuelle Inbetriebnahme	98
6.1	Durchgängiges Datenmanagement im Kontext einer virtuellen Inbetriebnahme	98
6.2	Ergänzende Bibliotheken und Methoden für die Modellerstellung	117
7	Definition von Modellarten und Detaillierungsstufen	125
7.1	Grundlegende Betrachtung der Granularität und Detaillierung von Simulationsmodellen	125
7.2	Modellartendefinition und deren wesentliche Aspekte.....	129
7.3	Ableitung der für die virtuelle Inbetriebnahme wesentlichen Detaillierungsstufen.....	137
7.4	Fertigungstechnische Simulationsmodellbibliothek.....	139
8	Applikation der virtuellen Inbetriebnahme	145
8.1	Grundlagen zur Identifizierung des notwendigen Detaillierungsgrads	145
8.2	Systematische Ableitung eines angemessenen Detaillierungsgrads aus Anlagenplanungsdaten	148
8.3	Sicherstellung der Konsistenz von Schnittstellen zwischen Modellen in verschiedenen Detaillierungsstufen	157
8.4	Erstellung des Simulationsmodells	164
9	Fallstudien	169
9.1	Fallstudie Modellfabrik	169
9.2	Fallstudie Schwenk-EHB.....	176
10	Zusammenfassung und Ausblick	182
10.1	Zusammenfassung	182
10.2	Ausblick auf potenzielle Weiterentwicklungsmöglichkeiten.....	184
	Anhang 1 Trennung von Stückgütern und Transport in Simulation	186
	Anhang 2 Kopplungselement	187
	Anhang 3 Anlagenübersicht	189
	Anhang 4 Anlagenstrukturmodell	190
	Anhang 5 Simulationsmodellbibliothek als SystemUnitClassLibrary	191
	Anhang 6 Generiertes Simulationsmodell in Modelica	192
	Anhang 7 Visualisierungslösung.....	193
	Literaturverzeichnis.....	194
	Normen- und Richtlinienverzeichnis.....	205
	Verzeichnis verwendeter Internet-Quellen und erwähnter Software.....	207
	Verzeichnis der Veröffentlichungen des Verfassers	210
	Verzeichnis der betreuten studentischen Arbeiten.....	212
	Lebenslauf	213

Abkürzungsverzeichnis

AIDA	Automatisierungsinitiative Deutscher Automobilhersteller
AML	AutomationML
ASL	Automatisches Lager
AT	Automatisierung
AWL	Anweisungsliste
BMW	Beschreibungsmittel, Methoden, Werkzeuge
CAD	Computer-Aided-Design
CAE	Computer-Aided-Engineering
CAEX	Computer Aided Engineering Exchange
CNC	Computerized Numerical Control
DF	Digitale Fabrik
DMM	Design-Mapping-Matrix
E/A	Eingang(s)-, Ausgang(s)-
EHB	Elektro-Hängebahn
FAT	Factory-Acceptance-Test
FEM	Finite-Element-Methoden
FMI	Functional-Mockup-Interface
FMU	Functional-Mockup-Unit
FPB	Formalisierte Prozessbeschreibung
FTS	Fahrerloses Transportsystem
FUP	Funktionsplan
GMA	(VDI/VDE-)Gesellschaft für Mess- und Automatisierungstechnik
GoM	Grundsätze ordnungsmäßiger Modellierung
HIL	Hardware-in-the-Loop
HMI	Human-Machine-Interface
IC	Interface Class
ICL	Interface Class Library

IE	Internal Element
IH	Instance-Hierarchy
IL	Internal Link
KMU	Kleine und mittlere Unternehmen
KOP	Kontaktplan
MBE	Model-based Engineering
MDA	Model-Driven-Architecture
MDE	Model-Driven-Engineering
MIL	Model-in-the-Loop
MKS	Mehrkörpersimulation
OEM	Original Equipment Manufacturer
PLM	Product-Lifecycle-Management
PLS	Prozessleitsystem
PLT	Prozessleittechnik
PPR	Produkt, Prozess, Ressource
R&I	Rohrleitungs- und Instrumentenfließschema
RCL	Role Class Library
RTI	Real-time Interface
SAT	Site-Acceptance-Test
SIL	Software-in-the-Loop
SPS	Speicherprogrammierbare Steuerung
SRC	Supported Role Class
SUC	System Unit Class
SUCL	System Unit Class Library
TCP/IP	Transmission Control Protocol / Internet Protocol
UML	Unified Modeling Language
VDA	Verband der Automobilindustrie
VDI	Verein Deutscher Ingenieure e. V.
VDMA	Verband Deutscher Maschinen- und Anlagenbau

VDW	Verein Deutscher Werkzeugmaschinenfabriken e. V.
VIBN	Virtuelle Inbetriebnahme
XML	Extensible Markup Language

Abbildungsverzeichnis

Abbildung 1-1: Schematischer Aufbau der Arbeit.....	3
Abbildung 2-1: Engineering-Prozess als Teil des Digitalen Fabrikbetriebs [VDI 4499-2]	8
Abbildung 2-2: Prinzip des modellbasierten Engineering in Anlehnung an [WALÖ10]	13
Abbildung 2-3: Idealisertes, gemeinsames Planungs- und Entwicklungsmodell gemäß [VDI 4499-2].....	15
Abbildung 2-4: Verhaltenskenngrößen von Prozessen als wesentliche Basis für die Dimensionierung und Auswahl von Prozesse umsetzenden Ressourcen.....	17
Abbildung 2-5: HANSE-Modell wie in [SNO00] beschrieben (angelehnt an [VDI 5610-1]).....	19
Abbildung 2-6: Meilensteine des Gewerks Mechanik während des Engineering-Prozesses	21
Abbildung 2-7: Beispiel für einen programminternen Ablauf einer SPS	23
Abbildung 2-8: Aufwandsverteilung bei der Inbetriebnahme automatisierter Anlagen gemäß [ZÄWÜ05]	29
Abbildung 2-9: Stark vereinfachte Übersicht über notwendige Handeingaben und aktiver Verriegelungen über die beschriebenen Betriebsarten.....	32
Abbildung 3-1: Virtuelle Inbetriebnahme und ihr Einsatzzeitpunkt während des Digitalen Fabrikbetriebs (schraffiert).....	42
Abbildung 3-2: Automationspyramide mit Wirkungsbereichen der virtuellen Inbetriebnahme	47
Abbildung 3-3: Aufwände bei der Durchführung einer Simulationsstudie gemäß [ACÉ96]	48
Abbildung 3-4: Übersicht über gebräuchliche Testkonfigurationen der virtuellen Inbetriebnahme in Anlehnung an [VDI 3693-1]	50
Abbildung 4-1: V-Modell als typisches Vorgehensmodell bei der Produktentwicklung (angelehnt an [VDI 2206-1])	58
Abbildung 5-1: Lösungskonzept	90
Abbildung 5-2: Signale und Funktionen bei Nutzung einer FMU mit externen Integrationsverfahren gemäß [RPK ⁺ 14 [#]].....	94
Abbildung 5-3: Aufbau der im Projekt „HIL-IMA“ entwickelten HIL-Simulationsumgebung gemäß [RPK ⁺ 14 [#]].....	95
Abbildung 5-4: Schematische Darstellung der TCP/IP Kopplung HIL-Simulationsplattform/SPS	96
Abbildung 6-1: Hierarchie technischer Systeme in Anlehnung an [HAS11]	104
Abbildung 6-2: Beispielhafte Darstellung einer Anlagenhierarchie	106
Abbildung 6-3: Zusammenhang zwischen Prozessen (gemäß [IEC 61512-1]) und Hierarchieebenen (wie in Abschnitt 6.1.3.1 definiert)	108
Abbildung 6-4: Abbildung von Produkten im Knoten <i>Produkte</i> in Anlehnung an [SCDR09].....	110
Abbildung 6-5: Verbindungen zwischen Produkten, Prozessen und Ressourcen in Anlehnung an [SCDR09].....	111

Abbildung 6-6: Ausschnitt der erstellten Bibliotheken SUCL (links) und ICL (rechts).....	116
Abbildung 6-7: Anlagenstruktur auf Modulebene	117
Abbildung 6-8: Abbildung einer Simulationsmodellbibliothek als SystemUnitClassLibrary	119
Abbildung 6-9: Identifizierung von Simulationsmodellen mit Hilfe von eCl@ss und beschriebenem Konzept	121
Abbildung 6-10: Merkmalleisten in eCl@ss in Anlehnung an [ECL@SSB [®]] und ihre mögliche Verwendung	122
Abbildung 6-11: Konzept für das Mapping von Parametern mit Hilfe von eCl@ss.....	123
Abbildung 6-12: Konzept für das Mapping von Schnittstellen mit Hilfe von eCl@ss.....	124
Abbildung 7-1: Für eine virtuelle Inbetriebnahme relevante Hierarchieebenen in Bezug auf die Anlagenstruktur	128
Abbildung 7-2: Grundlegende Modellarten	128
Abbildung 7-3: Einfache Betrachtung von Abläufen und Bewegungen mittels Totzeitmodellen ...	130
Abbildung 7-4: Betrachtung einer beschleunigungsfreien Bewegung auf definiertem Pfad.....	132
Abbildung 7-5: Modellbeschreibung und wirkende Kräfte/Momente.....	134
Abbildung 7-6: Physikbasierte Betrachtung von Bewegungen auf definierten Pfaden	135
Abbildung 7-7: Funktionsweise der Trennung Stückgut/Förderband in makroskopischem Detaillierungsgrad.....	141
Abbildung 7-8: Funktionsweise der Trennung Stückgut/Förderband in mesoskopisch/dynamischem Detaillierungsgrad	142
Abbildung 7-9: Verlauf der Konvergenz aufgrund der reduzierten Reibkraft gemäß [KPR ⁺ 14 [#]].	144
Abbildung 8-1: Gewichtung des Zusammenhangs von Prozessschritten und Komponenten zur Zuweisung des für eine Simulation notwendigen Detaillierungsgrads	147
Abbildung 8-2: Durchzuführende Teilschritte bei der Erstellung einer Regelbasis zur Ermittlung des notwendigen Detaillierungsgrads einzelner Komponentenmodelle	151
Abbildung 8-3: Beispielhafte Darstellung einer DMM bei der Komponenten (Zeilen) mit Prozessschritten (Spalten) in Beziehung gesetzt werden	155
Abbildung 8-4: Mögliche Inkonsistenzen bei der am Beispiel zweier miteinander verbundener Module	159
Abbildung 8-5: Prinzip des erweiterten Modellverbinder-Konzepts zur Identifizierung von Inkonsistenzen	161
Abbildung 8-6: Exemplarische Anwendung der definierten Regeln 1 und 2	163
Abbildung 8-7: Schnittstellen von Abbildung 8-6, Fallbeispiel 1 im Detail.....	163
Abbildung 8-8: Durchführung einer virtuelle Inbetriebnahme in Anlehnung an [LIN11].....	164
Abbildung 8-9: Generischer Modellgenerierungsalgorithmus.....	168
Abbildung 9-1: Übersicht über die betrachtete Anlage	170

Abbildung 9-2: Ausschnitt Anlagenstrukturmodell.....	171
Abbildung 9-3: Ausschnitt Simulationsmodellbibliothek in Modelica (links), Abbildung als SUC inkl. Schnittstellen in AutomationML (mitte) und der aus Modelica in AutomationML übertragenen Parameter (rechts)	171
Abbildung 9-4: Abbildung in Anlagenstrukturmodell.....	172
Abbildung 9-5: Teilausschnitt des generierten Modelica Simulationsmodells (vgl. Abbildung 9-4 dazu)	172
Abbildung 9-6: Messpunkte	174
Abbildung 9-7: Aufgenommene Messwerte im Vergleich [KPR ⁺ 14 [#]]	174
Abbildung 9-8: Typische Anwendungen von Elektro-Hebebühnen im Automobilbau (Bildübernahme aus [VDI 4441-1])	176
Abbildung 9-9: EHB Typ KB 240 mit Hub-, Fahr- und Schwenkvorrichtung (Bildübernahme mit freundlicher Genehmigung der ROFA INDUSTRIAL AUTOMATION AG [ROFA [®]]) ...	177
Abbildung 9-10: Gesamtanlagenübersicht	178

Tabellenverzeichnis

Tabelle 1: Typische Testaspekte während der Inbetriebnahme von Anlagen (Beispiele)	35
Tabelle 2: Mögliche Störungen in automatisierten Anlagen in Anlehnung an [WÜN08].....	37
Tabelle 3: Vergleich von Entwicklung gemäß VDI 2206 und Engineering gemäß VDI 4499	60
Tabelle 4: Vergleichende Bewertung aller betrachteten Ansätze	82
Tabelle 5: Aufgeworfene Fragen, Fazit sowie abgeleitete Defizite	83
Tabelle 6: Voraussetzungen für eine automatische Modellgenerierung (angelehnt an [BAR11]) und Vergleich mit den Anforderungen an das Datenmodell der Digitalen Fabrik...	86
Tabelle 7: Nutzbare Anlagenplanungsdaten für die Abbildung eines Anlagenstrukturmodells	101
Tabelle 8: Auswahl möglicher Wirkbeziehungen im Datenmodell der Digitalen Fabrik	113
Tabelle 9: Zusammenhang zwischen Granularität und Detaillierung eines Simulationsmodells.....	127
Tabelle 10: Vergleich der definierten Detaillierungsstufen	138
Tabelle 11: Typische, korrespondierende Potential- und Flussgrößen in a-kausalen Modellierungsansätzen	158
Tabelle 12: Durchzuführende Schritte der Modellerstellung und dafür notwendige Bedingungen hinsichtlich einer teil- oder voll-automatischen Modellgenerierung.	165

1 Einleitung

1.1 Motivation und Problemstellung

Der Trend zur immer höheren Individualisierung von Serienprodukten birgt multiple Herausforderungen [GAN13]. Eine direkte Auswirkung auf das Engineering automatisierter Anlagen zeigt die Automobilherstellung: Waren bis vor wenigen Jahren nur wenige Typenvariationen der gleichen Baureihe üblich, werden heute auf Kundenwunsch immer mehr und auch individualisierte Typen angeboten [KLU10]. Um Investitionskosten niedrig zu halten, werden diese Derivate oftmals auf nur einer Produktionslinie produziert. Verschiedene Derivate können unterschiedlichen Produktlebenszyklen unterliegen. In Kombination bewirkt dies gesteigerte Anforderungen hinsichtlich der Flexibilität von Produktionsanlagen einerseits, wie auch eine zwangsläufig steigende Anzahl von Produktanläufen andererseits [MAN09]. Produktionsunterbrechungen, in denen ein Um- oder Neubau von Anlagen stattfinden kann, werden aufgrund hoher Nachfrage und steigendem Wettbewerb nicht mehr oder nur noch stark verkürzt durchgeführt [DPA14]. Dies reduziert die für die Errichtung und Inbetriebnahme von Produktionsanlagen verfügbare Zeit ganz erheblich, was das Risiko eines nicht erfolgreichen Produktanlaufes erhöhen kann [MAN09]. Durch die Tendenz der immer kürzer werdenden Inbetriebnahmezeiträume sieht sich der Maschinen- und Anlagenbau vor der Herausforderung, sowohl die Qualität und Fehlerfreiheit der gelieferten Automatisierungslösung noch vor der eigentlichen Inbetriebnahme vor Ort sicherzustellen als auch einen Reifegrad selbiger zu erreichen, der die Inbetriebnahme von Anlagen reibungsloser und schneller gestaltet. In diesem Zusammenhang gilt: *„Der Erfolg des Produktionsanlaufes in Hinsicht auf Qualität, Termine und Kosten wird maßgeblich durch die vorgelagerten Projektschritte determiniert“* [SSS08].

Während des Engineerings automatisierter Anlagen können sich, wie in jedem anderen technischen Projekt auch, einerseits in frühen Projektphasen getroffene Entscheidungen als ungeeignete Lösungswege oder andererseits Engineering-Artefakte als fehlerhaft implementiert herausstellen. In anderen technischen Domänen, wie z. B. der Produktentwicklung, werden für diese Fälle spezialisierte Vorgehensmodelle eingesetzt, die Rücksprünge und Iterationen im Entwicklungsprozess explizit vorsehen und so eine Korrektur von Fehlern erlauben, ohne das gesamte Projekt hinsichtlich seiner Kosten- und Zeitziele grundlegend zu gefährden [HWF⁺12]. Oftmals wird die Implementierung hierfür schrittweise vorgenommen, so z. B. beim in der Praxis häufig eingesetzten V-Modell [PHK⁺13]. Durch das im Anlagenbau erzwungene *First-Time-Right* bei der Projektabwicklung ist dies für die Errichtung von automatisierten Anlagen weder möglich noch erwünscht. Der Engineering-Prozess automatisierter, fertigungstechnischer Anlagen folgt vielmehr einem sequenziellen, wasserfallartigen Vorgehensmodell. Die Implementierung einer Anlage wird in der Regel nur singulär und unter Betrachtung des Gesamtsystems vorgenommen, was gerade der Phase der Inbetriebnahme eine besondere Bedeutung zukommen lässt. Während der Inbetriebnahme entsteht in einem Engineering-Projekt erstmals eine Situation, in der die Projektergebnisse aller am Engineering Beteiligten in einer Anwendung gemeinsam implementiert werden – sie stellt in gewisser Weise die *"Stunde der Wahrheit"* für alle Beteiligten dar [ZEU98].

Wasserfallartige Vorgehensmodelle wie das des *Digitalen Fabrikbetriebs* [VDI 4499-2] bedingen gewisse Voraussetzungen für einen reibungslosen Projektablauf. So müssen insbesondere die an das zu entwerfende System gestellten Anforderungen über die gesamte Projektdurchführung hinweg einerseits vollständig bekannt, andererseits stabil sein [PHK*13]. In der Praxis können insbesondere Anforderungen an die Automatisierungslösung und damit an die Umfänge des Gewerkes Automatisierungstechnik diese Kriterien nicht erfüllen. Gründe hierfür können die Unvollständigkeit der erforderlichen Aufgabenspezifikationen zur SPS-Projektierung oder auch missverständliche Formulierungen sein [KIE07]. Eine direkte Auswirkung dieses Umstands ist, dass während der Inbetriebnahme häufig steuerungs-technische Funktionen nachgezogen werden müssen, die dann nur unzureichend bis zum eigentlichen Produktionsanlauf getestet werden können. Kombiniert mit der Tatsache, dass üblicherweise ohnehin bereits mehr als 60% der für eine Inbetriebnahme zur Verfügung stehenden Zeit alleine für das Beheben von Fehlern aus dem Engineering des Automatisierungssystems aufgewendet werden müssen [ZÄWÜ05], ergeben sich nicht zu vernachlässigende Risiken hinsichtlich einer erfolgreichen Durchführung eines Engineering-Projekts im Umfeld des Anlagenbaus.

Als Reaktion auf immer kürzer werdende Inbetriebnahmezeiträume, und um im Sinne einer durchgehenden und effizienten Fabrikplanung die Qualität sowie Termintreue bei Engineering-Projekten einerseits sicherzustellen und andererseits Erprobung und Validierung von Entwurfsentscheidungen schon in bereits frühen Phasen zu ermöglichen, erfolgt im Rahmen des *Digitalen Fabrikbetriebs* die Einbindung der *virtuellen Inbetriebnahme* genannten Methode in den Engineering-Prozess [VDI 4499-2]. Ziel einer *virtuellen Inbetriebnahme* ist das Aufdecken und Beheben von Fehlern, vor allem aus dem Engineering des Automatisierungssystems [VDI 3693-1]. Die Absicherung von Arbeitsergebnissen in Form einer virtuellen Inbetriebnahme kann in diesem Zusammenhang einen ersten wichtigen Zielzustand im Laufe eines Projekts darstellen, da erstmals alle am Engineering Beteiligten und ihre Arbeitsergebnisse in einer Anwendung vereint werden können – noch vor der eigentlichen Inbetriebnahme beim Auftraggeber vor Ort. Simulationsmodelle einer Anlage bilden den Kern einer jeden virtuellen Inbetriebnahme [BPH*15*]. Diese Simulationsmodelle müssen die Anlage dabei in einem Maß abbilden, welches erlaubt, die korrekte Ausführung eines Prozesses unter den bei einem Kunden vorherrschenden Bedingungen zu testen. Bei der Durchführung einer virtuellen Inbetriebnahme muss der Nutzen stets die dafür aufgewendeten Aufwände überwiegen [VDI 4499-2].

Wenngleich mit der virtuellen Inbetriebnahme eine Methode zur Verfügung steht, die Automatisierungslösung vorab validieren zu können, ist die virtuelle Inbetriebnahme derzeit nicht als Standardmethode im Engineering von Anlagen etabliert [BPH*15*]. Den Thesen des VDI/VDE-GMA-Fachausschusses 6.11 „*Virtuelle Inbetriebnahme*“ [GMA FA 6.11[®]] folgend, ist, neben organisatorischen Hürden bei der Einführung, derzeit insbesondere die notwendige Expertise für die Durchführung einer virtuellen Inbetriebnahme ein Hemmnis. Zudem kann der notwendige Aufwand für die Erstellung des zu nutzenden Anlagensimulationsmodells ein positives Aufwand/Nutzen Verhältnis kompromittieren [KHK10]. Insbesondere kleine und mittlere Unternehmen (KMU), die mit einem Anteil von 85 % das Gros der Firmen im deutschen Maschinen- und Anlagenbau stellen [VDM15], können eine Modellerstellung von Hand nicht leisten oder verfügen nicht über die notwendige Expertise und/oder die personellen Ressourcen dafür.

1.2 Zielsetzung und Aufbau der Arbeit

Die Zielsetzung dieser Arbeit ist es, einen Beitrag dazu zu leisten, dass virtuelle Inbetriebnahme als im Engineering-Prozess einzusetzende Methode, insbesondere für KMU, etabliert werden kann. Herausforderung und auch notwendige Bedingung hierfür ist eine systematisierte und hinsichtlich des Kosten/Nutzen-Aufwands optimierte Durchführung der für eine virtuelle Inbetriebnahme notwendigen vorbereitenden Schritte wie der Modellerstellung und damit zusammenhängender Tätigkeiten. Um den Aufwand für die Modellerstellung zu senken, wird in dieser Arbeit ein Konzept für eine automatische Modellgenerierung von Anlagensimulationsmodellen für die virtuelle Inbetriebnahme vorgestellt.

Den Kern der Arbeit bildet die Schaffung der dafür notwendigen Voraussetzung durch die Definition von den Modellerstellungsprozess unterstützenden Methoden einerseits. Andererseits liegt ein Schwerpunkt auf der Definition von für die virtuelle Inbetriebnahme geeigneten Anlagenstrukturbeschreibungen, unter Nutzung von im Engineering bereits etablierten Beschreibungsmitteln. Ziel ist es, Simulationsmodelle automatisch aus diesen Anlagenstrukturbeschreibungen abzuleiten. Die in dieser Arbeit vorgestellten Lösungsbausteine ermöglichen in ihrer Gesamtheit eine automatische Modellgenerierung, die aber nur unter gewissen Voraussetzungen erfolgen kann. Die vorgestellten Methoden und Abbildungsvorschriften haben daher den Anspruch, auch im Falle der Nicht-Erfüllung dieser Voraussetzungen, und so bei einer möglicherweise notwendigen manuellen Modellerstellung, nutzbringend einsetzbar zu sein. Die vorliegende Arbeit unterteilt sich grundlegend in zwei Teile, die durch die Ableitung und Definition eines mehrstufigen Lösungskonzepts, das durch insgesamt 5 Leitfragen charakterisiert wird, miteinander verknüpft werden. Abbildung 1-1 zeigt den schematischen Aufbau dieser Arbeit.

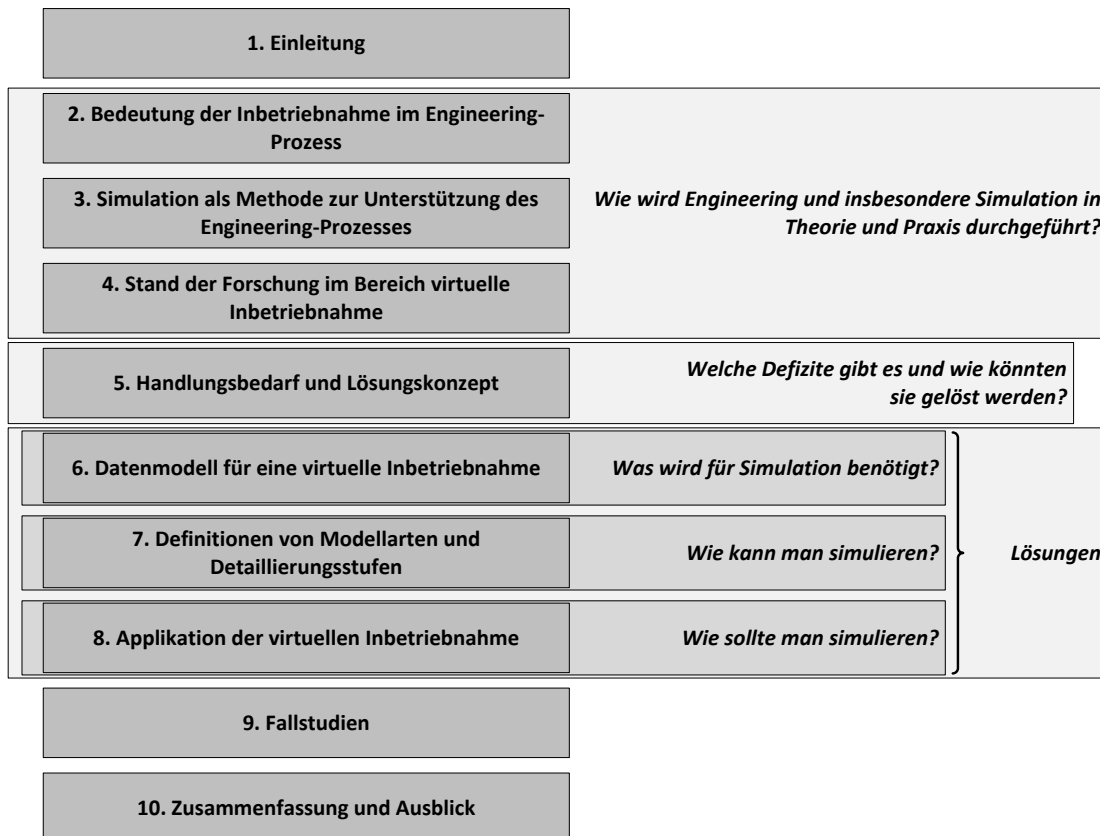


Abbildung 1-1: Schematischer Aufbau der Arbeit

Ausgehend von einer grundlegenden Beschreibung des *Digitalen Fabrikbetriebes* gemäß [VDI 4499-2] sowie seiner Ziele wird in Kapitel 2 detailliert über das derzeitige Vorgehen beim Entwurf automatisierter, fertigungstechnischer Anlagen berichtet. Großer Wert wird auf die Beschreibung des tatsächlichen IST-Zustands in der industriellen Praxis gelegt. Den Schwerpunkt der Untersuchungen bilden typische Vorgehensmuster, wie sie in KMU üblich sind. Ein Schwerpunkt wird auf die Phase der Inbetriebnahme gelegt.

Kapitel 3 befasst sich mit der „*Virtuelle Inbetriebnahme*“ genannten Methode, Steuerungscode noch vor der Inbetriebnahme vor Ort beim Kunden zu testen. Zu diesem Zweck werden allgemeingültige Definitionen und Beschreibungen sowie die derzeitigen Implementierungsstrategien der Methode erläutert. Aufbauend auf den in Kapitel 2 und 3 beschriebenen Vorgehensmustern beim Entwurf von automatisierten Anlagen ergibt sich eine Reihe von Fragen, die in Abschnitt 3.5 thematisiert werden. Diese stellen das Grundgerüst für die sich in Kapitel 4 anschließende Analyse des Stands der Forschung im Bereich virtuelle Inbetriebnahme dar.

In Kapitel 4 werden Ansätze und Methoden analysiert, die sich mit dem Thema virtuelle Inbetriebnahme auseinandersetzen. Hierfür findet zunächst eine allgemein gehaltene, kritische Auseinandersetzung mit Ansätzen der „*klassischen*“ Produktentwicklung, die für eine Verwendung im Anlagenbau postuliert werden, sowie Ansätzen, die sich am Vorgehen des *Digitalen Fabrikbetriebes* gemäß [VDI 4499-2] orientieren, statt. Im Anschluss wird der derzeitige Stand der Wissenschaft zum Thema virtuelle Inbetriebnahme analysiert.

Aus den Untersuchungsergebnissen und den aufgeworfenen Fragestellungen der vorhergehenden Kapitel werden in Kapitel 5 zunächst Defizite herausgearbeitet, deren mögliche Behebung durch das Definieren von jeweils direkt zugeordneten Lösungsbausteinen systematisiert wird. Diesen liegt jeweils eine Leitfrage zugrunde. In Kapitel 5 wird ferner der in dieser Arbeit relevante Anwendungsbereich konkretisiert sowie die für diese Arbeit geltenden technischen Rahmenbedingungen erläutert.

Die in Kapitel 6 erarbeiteten Lösungsvorschläge konzentrieren sich auf die für eine virtuelle Inbetriebnahme benötigte Datenbasis in Form von Informationen, die in einem für diesen Zweck geeigneten Datenmodell hinterlegt sind. Hierfür werden Abbildungsvorschriften und Methoden, die das Datenmodell der *Digitalen Fabrik* konkretisieren und erweitern, vorgestellt und anhand einfacher Beispiele erklärt. In Kapitel 6 wird die Frage beantwortet, welche Informationen für eine Simulation benötigt werden.

Steht eine geeignete Datenbasis zur Verfügung, werden für den Anwendungsfall virtuelle Inbetriebnahme Simulationsmodelle benötigt. Die in Kapitel 7 vorgestellten Lösungsbausteine stellen generelle Gestaltungsprinzipien einer Simulationsmodellbibliothek vor. In Ergänzung dazu werden konkrete Modellarten vorgestellt, die die Simulation in unterschiedlichen Detaillierungsstufen erlauben. In Kapitel 7 wird die Frage beantwortet, wie man simulieren kann.

In Kapitel 8 werden Methoden vorgestellt, die es erlauben, den angemessenen Detaillierungsgrad einzelner Teile eines Anlagensimulationsmodells zu bestimmen. Ebenso wird eine Methode vorgestellt, wie die Konsistenz von Simulationsmodellen, die in heterogenem Detaillierungsgrad und unter Verwendung verschiedener Simulationsmodellerstellungsmethoden erstellt wurden, sichergestellt werden kann. Kapitel 8 schließt mit der Vorstellung eines generischen Modellgenerierungsalgorithmus ab, der alle bisher definierten Lösungsbausteine zu einer Gesamtlösung integriert. In Kapitel 8 wird die Frage beantwortet, wie man simulieren sollte.

In Kapitel 9 werden 2 Fallstudien beschrieben, die die exemplarische Anwendung der erarbeiteten Lösungsbausteine zeigen. Fallstudie 1 konzentriert sich auf die prinzipielle Machbarkeit der automatischen Modellgenerierung sowie eine Gesamtvalidierung des Ansatzes. Ziel ist der Nachweis, dass automatisch generierte Simulationsmodelle dazu geeignet sind, mit ihnen Testfälle im Rahmen einer virtuellen Inbetriebnahme durchführen zu können. Fallstudie 2 untersucht im Speziellen die Methode zur Ermittlung eines adäquaten Detaillierungsgrads. Zu diesem Zweck werden die Betrachtungen auf eine Applikation einer Hängefördertechnik konzentriert.

In Kapitel 10 erfolgen eine kurze Zusammenfassung der Ergebnisse dieser Arbeit sowie ein Ausblick auf Entwicklungsmöglichkeiten des vorgestellten Ansatzes.

2 Bedeutung der Inbetriebnahme im Engineering-Prozess fertigungstechnischer Anlagen

Die folgenden Abschnitte geben einen Einblick in das typische Vorgehen bei der Planung, Dimensionierung und schließlich der Erstellung einer automatisierten, fertigungstechnischen Anlage. Betrachtet wird der Lebenszyklus einer Anlage von der Grobplanung bis hin zur Realisierung, speziell mit dem Fokus auf die Phase der Inbetriebnahme. Es erfolgt eine kritische Würdigung des derzeitigen Stands des Engineerings am Ende des Kapitels.

2.1 Engineering

Der Begriff „*Engineering*“ fasst Teilaspekte der umfangreichen Tätigkeiten eines Ingenieurs in einem Oberbegriff zusammen. Eine genaue Definition oder auch Abgrenzung des Begriffs „*Engineering*“ existiert im deutschsprachigen Raum nicht. Als Begriff wird Engineering vor allem immer dort eingesetzt, wo der Begriff „Entwicklung“ nicht zutreffend oder nicht ausreichend ist, um die Tätigkeiten eines Ingenieurs hinreichend zu beschreiben [FAY09]. Zudem unterliegt der Begriff des Engineerings einem stetigen Wandel: Stand das Engineering im deutschsprachigen Raum zunächst für den schlüsselfertigen Bau von Produktionsanlagen, so stehen heute vor allem anlagenbezogene, gewerkeübergreifende Informationsprozesse über den gesamten Lebenszyklus von automatisierten Anlagen hinweg im Fokus [DIFA13]. Eine allgemein weite Verbreitung hat die vom American Engineers' Council for Professional Development (ECPD), vergleichbar mit dem Verein Deutscher Ingenieure (VDI), bereits 1941 festgelegte Definition des Begriffes Engineering gefunden [EC41]:

“The creative application of scientific principles to design or develop structures, machines, apparatus, or manufacturing processes, or works utilizing them singly or in combination; or to construct or operate the same with full cognizance of their design, or to forecast their behavior under specific operating conditions; all as respects an intended function, economics of operation and safety to life and property.”

Engineering umfasst daher, neben den klassischen Ingenieur Tätigkeiten wie dem Einsatz naturwissenschaftlicher Erkenntnisse zur Erstellung eines technischen Systems, insbesondere die *kreative Anwendung wissenschaftlicher Methoden* beim Entwerfen technischer Systeme. Von [FAY09] wird dies beispielhaft als „[...] *der kreativ schöpferischer Akt, für ein gegebenes Regelungs- oder Steuerungsproblem die geeignete Automatisierungsstruktur und geeignete Geräte auszuwählen*“ für die Domäne Automatisierungstechnik beschrieben. Generell wird der Begriff Engineering mit der Planung und Realisierung automatisierter, industrieller Anlagen gleichgesetzt ([DRA08], [FSM09]) – letztendlich auch, um sich begrifflich von der klassischen Produktentwicklung abzugrenzen. Schnieder [SCH99A] weist in Bezugnahme auf das Engineering automatisierter Anlagen darauf hin, dass es nicht nur der kreative Einsatz von Methoden ist, der das Engineering kennzeichnet, sondern dass es der Dreiklang aus *Beschreibungsmitteln, Methoden* und *Werkzeugen* ist, der, sinnvoll miteinander kombiniert, dem Engineering zugrunde liegen sollte. Schnieder

[SCH99A] bezeichnet dies als *BMW-Prinzip*. Im Folgenden wird eine genauere Charakterisierung des Engineering-Prozesses automatisierter, fertigungstechnischer Anlagen gegeben.

2.2 Betrachtung des Engineering-Prozesses fertigungstechnischer Anlagen und beteiligte Gewerke

Der Begriff *Engineering-Prozess* bezeichnet das generelle Vorgehen bei der Planung und der anschließenden Realisierung einer automatisierten Anlage – unter Einbeziehung und organisatorischer Einordnung der am Engineering-Prozess beteiligten Gewerke (auch Engineering-Organisationen genannt [VDI 3695-1]). Er subsumiert alle Tätigkeiten, die während des Engineerings einer spezifischen Fertigungseinrichtung anfallen, und bietet ein Rahmenwerk für das Engineering [MAN09]. Das Engineering von Anlagen unterscheidet sich von der klassischen Entwicklung technischer Produkte dahin gehend, als die zu entwerfenden Anlagen immer Unikate darstellen, die „[...] *einzelnen beauftragt werden und dann im Rahmen eines Projekts geplant und realisiert werden*“ [FAY09].

Der Engineering-Prozess kann als derjenige Teilaspekt der Fabrikplanung gesehen werden, der im Zusammenhang mit der *Digitalen Fabrik* [VDI 4499-1] als *Digitaler Fabrikbetrieb* bezeichnet wird ([BWG09], [VDI 4499-2]). Der *Digitale Fabrikbetrieb* konkretisiert und bezeichnet „[...] *die Nutzung und das Zusammenwirken von Methoden, Modellen und Werkzeugen der Digitalen Fabrik*“ [VDI 4499-2]. Der Begriff Engineering-Prozess bezieht sich insbesondere auf diejenige Vorgehensebene bei der Projektabwicklung, die eine Koordination und zeitliche Einteilung einzelner Haupttätigkeiten erlaubt (Makro-Logik [HWF*12]). Das von einzelnen Gewerken genutzte Vorgehensmodell in den relevanten Phasen (Mikro-Logik [HWF*12]) spielt bei der Betrachtung des *Digitalen Fabrikbetriebs* keine Rolle, die inhaltliche Ausgestaltung obliegt der Verantwortung der am Engineering-Prozess beteiligten Gewerke [MAN09]. Der Engineering-Prozess kann als konvergentes Vorgehen beschrieben werden, ohne im konkreten auf ein spezifisches Vorgehensmodell festgelegt zu sein.

Im fertigungstechnischen Anlagenbau werden zur Durchführung von Engineering-Projekten in der obersten Vorgehensebene häufig plangetriebene Methoden [HWF*12] genutzt. Diese teilen den Planungsprozess einer automatisierten Anlage in verschiedene, zeitlich aufeinanderfolgende Projektphasen auf ([SCH01], [MAN09]). Sie bieten ein hohes Maß an Struktur – „*im Sinne einer klaren Schrittfolge*“ [HWF*12]). Abbildung 2-1 zeigt den Engineering-Prozess im Rahmen der *Digitalen Fabrikbetriebs*, angelehnt an [VDI 4499-2]. Die in Abbildung 2-1 dargestellte Phase der „Montage- und Fertigungsprozessplanung“ bezieht die Festlegung der Montagereihenfolge eines zu fertigenden Produkts ein.

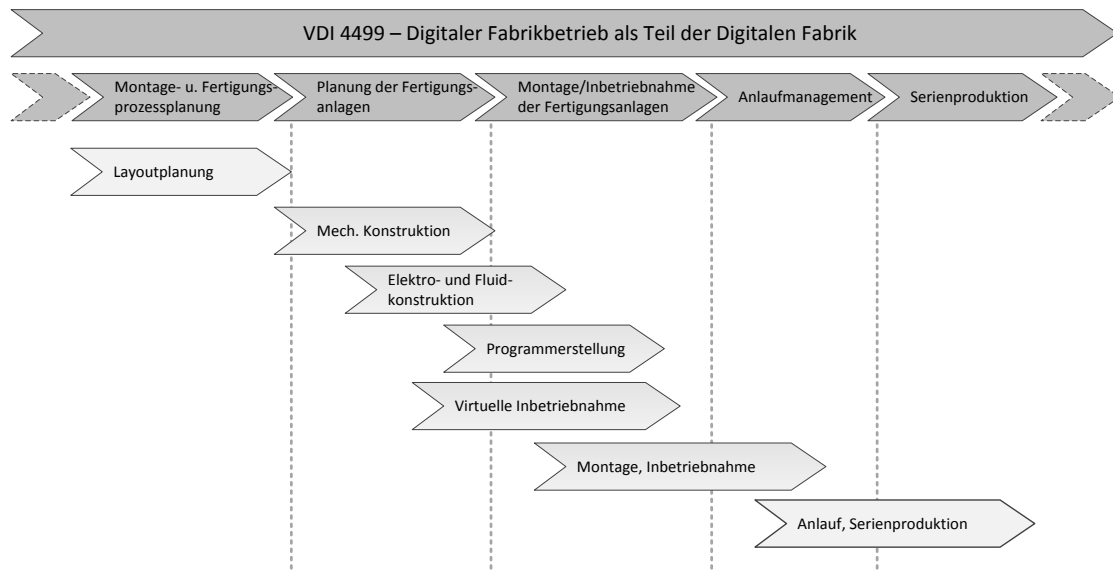


Abbildung 2-1: Engineering-Prozess als Teil des Digitalen Fabrikbetriebs [VDI 4499-2]

Der abgebildete Engineering-Prozess spiegelt die Ziele der Anwendung des *Digitalen Fabrikbetriebs* wieder, die sich im Wesentlichen in drei Zielarten unterteilen lassen: *Zeitziele*, *Qualitätsziele* und *Kostenziele*. Weiterhin zeigt er die typisch zu bearbeitenden Aufgabenkomplexe während der Planung und Errichtung von Fertigungsanlagen in der Makro-Vorgehensebene. *Zeit-*, *Qualitäts-* wie auch *Kostenziele* des *Digitalen Fabrikbetriebs* werden in der [VDI 4499-2] ausschnittsweise wie folgt definiert (Anm.: Es erfolgt eine teils wörtliche Wiedergabe aus [VDI 4499-2]):

Zeitziele des Digitalen Fabrikbetriebs:

- Planungs- und Entwicklungsphasen bis zum Anlauf parallelisieren und die Dauer der gesamten Anlagenplanung und -entwicklung verkürzen
- virtuelle Abnahme von Anlagen oder von Anlagenbereichen/Teilbereichen vorziehen
- Serienanlaufkurven verkürzen

Qualitätsziele des Digitalen Fabrikbetriebs:

- Investitionsentscheidungen absichern, viele Planungsszenarien schnell einander gegenüberstellen
- Planungs- und Umsetzungsfehler reduzieren
- Transparenz von Planungsprozessen erhöhen

Kostenziele des Digitalen Fabrikbetriebs:

- Einsparung von Inbetriebnahme-Aufwand durch Senkung der Anzahl realer Experimente
- Einsparung durch Automatisierung von betriebsbegleitenden Anpassungen

Den Engineering-Prozess unterstützende Methoden wie Simulation (hier verdeutlicht durch die Phase *Virtuelle Inbetriebnahme*) sollen als wichtiges Hilfsmittel zur Erreichung dieser Ziele nicht nur in einzelnen Phasen ein integraler Bestandteil des Engineerings sein, sondern explizit auch als Haupttätigkeiten des *Digitalen Fabrikbetriebs* ([SSA10], [VDI 4499-2]).

Grundsätzlich lassen sich im Maschinen- und Anlagenbau zwei grundlegende Vorgehensmodelle identifizieren: Das bereits beschriebene, sequenzielle Vorgehen des *Digitalen Fabrikbetriebs* sowie Vorgehensmodelle, die sich an dem (z. B. in der [VDI 2206-1] genutzten) V-Modell orientie-

ren. Ein detaillierter Vergleich und eine Ausarbeitung der jeweiligen Vor- und Nachteile beider Vorgehensmodelle finden in Abschnitt 4.1 statt.

2.2.1 Am Engineering beteiligte Gewerke und übliche Organisationsstruktur

Die Planung und Realisierung eines komplexen, d. h. aus vielen verschiedenartigen Elementen bestehenden, automatisierten Systems wird in der Regel nicht „aus einer Hand“ geliefert [FAY05], sondern in Zusammenarbeit vieler Anlagenbauer und Subunternehmer „in projektspezifischer Zusammensetzung“ [RÖD02]. Die Gewerke Mechanik, Elektrotechnik sowie Automatisierungstechnik (als Gewerk, welches den Steuerungscode plant und realisiert) sind, der mechatronischen Sichtweise im Anlagenbau folgend ([BAT06], [KIE07]), an der Realisierung einer fertigungstechnischen Anlage beteiligt. Die mechatronische Sichtweise definiert sich in diesem Zusammenhang als das „[...] synergetische Zusammenwirken der Domänen Maschinenbau, Elektrotechnik und Informationstechnik [...]“ [VDI 2206-1]. Sie arbeiten, gemäß der bereits beschriebenen Ziele des *Digitalen Fabrikbetriebs*, „sequenziell, wo sie inhaltlich auf den Ergebnissen anderer Gewerke aufbauen, [...] ansonsten parallel, um die Projektdauer zu minimieren“ [FAY09]. Erstmals sind alle beteiligten Gewerke während der Inbetriebnahme in einer Anwendung miteinander vereint, was der Inbetriebnahme eine große Bedeutung bei der Umsetzung eines Engineering-Projekts gibt. Findet die Absicherung aller Funktionen und damit die Validierung des Zusammenspiels aus Mechanik, Elektrotechnik und Steuerungscode erst zur realen Inbetriebnahme statt, können u. a. Verzögerungen im Zeitplan, kostspielige Stillstandszeiten und generell ein (stark) erhöhter Aufwand für die Inbetriebsetzung einer Anlage die Folge sein ([ZEU98], [KIE07]).

Die sehr späte Absicherung aller Funktionen einer Anlage lässt sich auf das sequenzielle Engineering-Vorgehen des *Digitalen Fabrikbetriebs* zurückführen. Üblicherweise gibt es in Engineering-Projekten ein führendes Gewerk (oder auch einen Fachbereich), welches den zu realisierenden, technischen Prozess zunächst entwirft und dimensioniert [FAY09]. In der Fertigungstechnik ist dies häufig das Gewerk Mechanik [FAY09]. Ausnahmen von dieser Regel sind möglich und können mit der Konstellation beteiligter Firmen und projektspezifischen Charakteristika begründet sein. Alle dem führenden Gewerk nachgeordneten Gewerke (Elektrotechnik, Automatisierungstechnik) nutzen Unterlagen des führenden Gewerkes als Planungsgrundlage und erweitern diese um die gewerkespezifischen Merkmale. Aus der Organisationsstruktur und dem Vorgehen der beteiligten Gewerke lässt sich ableiten, dass ein *durchgängiges* und auch stringentes *Datenmanagement*, wie es für die *Digitale Fabrik* und insbesondere den *Digitalen Fabrikbetrieb* postuliert wird [VDI 4499-2], von größter Wichtigkeit für den Engineering-Prozess ist. In der Praxis bestehen noch große Defizite bei der effizienten Gestaltung eines durchgängigen Datenmanagements im Engineering ([FAY09], [KÜRO12]).

2.2.2 Projektdurchführung und Synchronisation der beteiligten Gewerke

Beachtet werden sollte, dass das führende Gewerk zwar als führend bei der technischen Gestaltung während des Engineering-Prozesses anzusehen ist und formell auch die Schnittstelle aller beteiligten Gewerke zum Auftraggeber bildet, aber nicht immer die führende Rolle im Projektmanagement übernimmt. Versteht man Projektmanagement als „[...] [die] Summe von organisatorischen und dispositiven Maßnahmen zur Planung, Führung, Überwachung und Steuerung eines Projekts in inhaltlicher, zeitlicher und kostenmäßiger Hinsicht“ [HWF⁺12], konzentriert sich das führende Gewerk auf dispositive Maßnahmen wie die Planung, Überwachung und Koordination

des Projekts in zeitlicher und kostenmäßiger, aber nur eingeschränkt auch in inhaltlicher Hinsicht. Insbesondere ist dies der Fall bei der Planung und Ausführung von spezifischen Lösungen über alle mechatronischen Aspekte einer Anlage hinweg, wie z. B. der Sicherstellung der Kohärenz der mechanischen Apparaturen zu den in der speicherprogrammierbaren Steuerung (SPS) programmierten Funktionsblöcken. In den Phasen Grob- sowie Feinplanung, in denen die Gewerke Elektrotechnik und Automatisierungstechnik in der Praxis noch nicht die dominante Rolle spielen, sondern das Entwerfen des zu realisierenden, technischen Prozesses durch das Gewerk Mechanik im Vordergrund steht, werden wichtige steuerungstechnisch relevante Inhalte nicht immer ausreichend gewürdigt. Dies trifft in der Praxis besonders dann zu, wenn die Domäne Steuerungstechnik nicht am Projektmanagement beteiligt wird. Verallgemeinert kann davon gesprochen werden, dass in diesem Fall das mechatronische Zusammenspiel, wie es in [VDI 2206-1] definiert wird, nicht immer konsequent beachtet wird.

Die notwendige Synchronisation zwischen entworfenen mechanischen Apparaturen, ausgewählten Sensoren und Aktoren, und des zu entwickelnden Steuerungscode (insbesondere wenn in Form von Firmenstandards vorgegeben), die wesentlich eine Grundlage des *Digitalen Fabrikbetriebs* ist, findet in der Praxis oftmals nicht zufriedenstellend statt, „[...] so dass bei der Integration verstärkt Probleme aufgrund unberücksichtigter Abhängigkeiten auftreten“ [HEL13]. Auftraggeber und führende Gewerke beschränken sich in vielen Fällen bei der Planung steuerungstechnischer Aspekte auf bereits bestehende, funktionierende Referenzanlagen (mitunter mehrere, undifferenziertes „Rosinen-Herauspicken-Prinzip“) oder Firmen- respektive Werksstandards, ohne die genauen Charakteristika der neu zu errichtenden Anlage zu berücksichtigen. Der *beste Lösungsweg*, mit dem Fokus auf die Umsetzung der von der Automatisierungstechnik zu liefernden Umfänge, ist so nicht immer von Anfang an klar und muss vom Gewerk Automatisierungstechnik im Laufe des Projekts erst noch „gesucht“ werden. Da erst das Zusammenspiel aller Domänen respektive der Ergebnisse aller am Engineering-Prozess beteiligten Gewerke das für eine Anlage charakteristische Verhalten aufzeigt, welches gegen die Anforderungen des Auftraggebers getestet werden kann, führt dies zu Planungsfehlern im Engineering-Prozess, die sich erstmals während der Integration aller von den beteiligten Gewerken erstellten Planungsartefakten zur Inbetriebnahme beim Kunden vor Ort zeigen. Im Folgenden werden die einzelnen Projektergebnisse der am Engineering beteiligten Gewerke näher betrachtet. Dies schließt den Weg zur Erarbeitung dieser Ergebnisse mit ein.

2.3 Modellbasiertes Engineering im Kontext des Engineering-Prozesses

Modellbasierte Ansätze stellen im Engineering ein Mittel zur Darstellung von Projektergebnissen dar und haben große Verbreitung gefunden. Diedrich und Fay [DIFA13] formulieren in diesem Kontext: Modelle „[...] prägen die gedankliche Vorstellung [im Engineering], mit ihnen werden die Anforderungen und das zu automatisierende System beschrieben“. Wagner und Löwen [WALÖ10] schreiben ihnen sogar die Eigenschaft eines „[...]wesentliche[n] technische[n] Hebel[s] zur Integration und Optimierung der Engineering-Prozesse im industriellen Anlagenbau“ zu. In den folgenden Abschnitten erfolgt zunächst eine grundlegende Differenzierung zwischen modellbasierten und modellgetriebenen Ansätzen im Engineering (→ Abschnitt 2.3.1). Auf das eng damit zusammenhängende Datenmanagement im Kontext der *Digitalen Fabrik* wird gesondert in Abschnitt 2.3.2 eingegangen.

2.3.1 Vergleich modellbasierter und modellgetriebener Ansätze im Engineering

Die Ursprünge der Begriffe „modellbasiert“ und „modellgetrieben“ liegen in der Softwareentwicklung. Bereits die Suffixe „-basiert“ und „-getrieben“ lassen die Schlussfolgerung zu, dass Modelle eine insgesamt große Rolle spielen. Unter einer *modellgetriebenen Softwareentwicklung (Model-Driven-Engineering, MDE)* versteht man die Nutzung von Modellen, um Teile eines Softwaresystems auf einem höheren Abstraktionsniveau zu beschreiben [StV05]. Insbesondere die Unified Modeling Language (UML) [BRJ05] kommt für Beschreibungen derartiger Modelle zum Einsatz. Generell werden diese Modelle als *Plattform-Independent-Models* bezeichnet. Bei der modellgetriebenen Softwareentwicklung liegen diese Modelle gleichberechtigt neben dem Quellcode; letztendlich wird der Quellcode aus den erstellten (Plattform-Independent-) Modellen (automatisiert) für die jeweilige Zielplattform abgeleitet [ME04]. *„Das Adjektiv ‚getrieben‘ soll dabei die zentrale Rolle der Modelle im Entwicklungsprozess hervorheben“* [Scho08] – es sind die Modelle, die die Entwicklung treiben, nicht die konkreten Lösungen, die aus den Modellen erzeugt werden. Dienen Modelle im Gegensatz dazu „nur“ zum Design oder zur Dokumentation und ggf. dem Test einer Software, wird dies als *modellbasierte Softwareentwicklung (Model-Based-Engineering)* bezeichnet [StV05]. Zwischen dem Modell und der Implementierung besteht lediglich eine *„gedankliche Verbindung“*. Ein Modell würde so nur den Ablauf eines Prozesses beschreiben, während die Implementierung des Ablaufs in der jeweiligen Programmiersprache direkt erfolgt, ohne dass eine direkte Beziehung zwischen Modell und händisch erzeugtem Code besteht.

Betrachtet man das Engineering automatisierter Anlagen, so basiert das Engineering ganz wesentlich sowohl auf der Erstellung wie auch der Nutzung multipler Modelle ([VDF*13], [Wal010]). Konkret formulieren [Wal010] dies so: *„Die beim Engineering einer Anlage erstellten Modelle spezifizieren Eigenschaften des technologischen Prozesses, einzelner Komponenten oder des technischen Systems (= der Anlage) aus verschiedenen technischen Sichten [...], z. B. in Form von Verfahrensdiagrammen, CAD- und CAE-Plänen, mathematischen Modellen, räumlichen Modellen, Funktionsplänen etc.“*. Insbesondere für die Automatisierungstechnik und der dort im Vordergrund stehenden Umsetzung von Automatisierungsfunktionen in Form von IEC 61131-3 Steuerungscode [IEC 61131-3] ist es charakteristisch, zahlreiche Modelle aus anderen Gewerke zu verwenden, welche die zu automatisierenden Systeme beschreiben [VDF*13]. Modelle können prinzipiell für jeden Teilschritt beim Engineering eines Automatisierungssystems benötigt oder genutzt werden, *„[...] u. a. für den Entwurf, die Implementierung, den Test, die Optimierung und die Diagnose“* [VDF*13]. In diesem Kontext und unter Beachtung der im Engineering vorherrschenden Rahmenbedingungen lassen sich für das Engineering von Anlagen allgemein mehrere Nutzungsaspekte und Einsatzgebiete für Modelle identifizieren [Wal010]:

- Modelle als Transportmittel der Wertschöpfungsleistung: *„Modelle unterstützen den Planungsprozess, indem sie die Ergebnisse [...] strukturiert und ggf. auf die wesentlichen Informationen abstrahiert dokumentieren, und für weitere Schritte bereitstellen“* [Wal010].
- Modelle zur Automatisierung bei der Leistungserbringung (Automation der Automation, [SSE09]): Automatisierte Durchführung einzelner Teilaufgaben, die sich algorithmisch aus Modellierungsdaten ableiten lassen, sowie automatisierte Erstellung neuer Modelle (z. B. Simulationsmodelle).
- Modelle zur rechnergestützten Aus- und Bewertung von Lösungseigenschaften: *„Durch Modelle können Informationen in strukturierter und formalisierter Weise dargestellt werden, so dass eine rechnergestützte Auswertung unter Zuhilfenahme mathematischer*

Paradigmen erfolgen kann, z. B. die Verwendung eines CAD-Modells in einer FEM-Simulation. [...] Neben Simulation gibt es auch Auswertungen in Richtung der mathematischen Optimierung von Systemen (z. B. Betriebsoptimierung, Optimierung von Komponentenparametern)“ [WALÖ10].

- Modelle als Basis eines integrierten Engineerings: *„Die erstellten Engineering-Ergebnisse basieren [derzeit] auf meist spezifischen, isolierten und inkongruenten Modellen, welche die spezifischen technischen Sichten auf die Anlage widerspiegeln. [...] [Ziel ist es,] eine bessere Systematisierung, Verzahnung, Absicherung und Optimierung der Engineering-Prozesse zu erreichen“ [WALÖ10].*

Definiert man diese Nutzaspekte und Einsatzgebiete als Basis für die Verwendung von Modellen während eines Engineering-Prozesses, sind es, im Gegensatz zu einem MDE Ansatz in der Softwareentwicklung, nicht die entstehenden Modelle, die das Engineering „treiben“ – vielmehr entsprechen die erzeugten Modelle den (Teil-)Ergebnissen des Engineering-Prozesses. Modelle und durch Gewerke erstellte (Teil-)Lösungen können daher als kongruent angesehen werden. Modelle weisen im Engineering vielfache Einsatzgebiete auf, die (automatisierte) Ableitung einer konkreten Lösung (also das Überführen eines Modells aus höherem Abstraktionsniveau in eine plattform-spezifische, konkrete Lösung) gehört üblicherweise nicht dazu. Unberührt davon kann die Ableitung von Modellen für verschiedene Einsatzbereiche, z. B. Simulation, aus bereits bestehenden Modellen durchgeführt werden. Ausdrücklich sei in diesem Kontext darauf hingewiesen, dass aus dieser Aussage kein Rückschluss über den Reifegrad einer Lösung respektive eines Modells möglich ist – den Reifegrad und damit die Qualität einer Lösung sicherzustellen kann in einer Simulation erfolgen, für die wiederum die während des Engineering-Prozesses entstandenen Modelle eine geeignete Basis sein können (ggf. um weitere Modelle ergänzt).

Die Begriffe *modellbasiert* und *modellgetrieben* sollten für die Verwendung beim Engineering automatisierter Anlagen und im Speziellen unter Beachtung des Gewerks Automatisierungstechnik scharf voneinander abgegrenzt werden, auch wenn diese zum Teil synonym für Ansätze im Engineering genutzt werden [WALÖ10]. Zu beachten ist, dass die im Software-Engineering üblichen *Model-Driven-Engineering (MDE)* Ansätze, insbesondere in der Automatisierungstechnik und der dort im Vordergrund stehenden Erstellung von Steuerungscode auf Basis der [IEC 61131-3], so nicht durchsetzbar sind [VBK⁺11]. Die während einer Inbetriebnahme mit an Sicherheit grenzender Wahrscheinlichkeit notwendigen Änderungen und Korrekturen am Steuerungscode müssen direkt auf der Zielplattform vor Ort ausgeführt werden können – zu zeitintensiv ist die ausschließliche Änderung an einem Modell und die anschließende Prüfung und Transformation in einen lauffähigen Steuerungscode [VDF⁺13]. Modellgetriebene Ansätze im Engineering automatisierter Anlagen, insbesondere mit Bezug auf das modellgetriebene Engineering des Steuerungscode, wie sie u. a. von [FRTH11] und [OBV15] propagiert werden, eignen sich derzeit nur für Spezialanwendungen und stellen einen maßgeblichen Paradigmenwechsel für die Arbeitsweise des Gewerks Automatisierungstechnik dar. Letztendlich erfordert der Einsatz modellgetriebener Methoden, neben den in *Model-Driven-Architecture (MDA)* Ansätzen üblichen hohen Anforderungen an den Formalismus wie auch an die Definition des Abstraktionsgrads der Modelle [MEL04], spezifisches methodisches Fachwissen. Es ist in diesem Zusammenhang zu beachten, dass es üblicherweise keine Informatiker mit entsprechendem Ausbildungshintergrund sind, die IEC 61131-3 Steuerungscode erstellen, sondern Angehörige des Gewerks Automatisierungstechnik mit (ingenieurs-)technischer Ausbildung [VDF⁺13].

Unter Anbetracht der derzeitig erfolgenden und mit Bezug zur *Digitalen Fabrik* postulierten gewollten Verwendung von Modellen im Engineering-Prozess kann das Vorgehen der Nutzung von Modellen im Engineering daher am besten als *modellbasiertes Engineering (Model-Based-Engineering, MBE)* bezeichnet werden. Zu beachten ist, dass „[...] [modellbasiertes] Engineering im industriellen Anlagen- und Lösungsgeschäft [...] weniger auf die Schaffung neuer integrierter Funktionalitäten im Umfeld der Anlagentechnik ab[zielt], sondern auf die Integration und Verbesserung des Zusammenspiels projektbezogener Engineering-Prozesse nach mechatronischen Gesichtspunkten“ [WALÖ10]. Abbildung 2-2 zeigt das Prinzip des modellbasierten Engineerings, aufbauend auf den Untersuchungen von [WALÖ10].

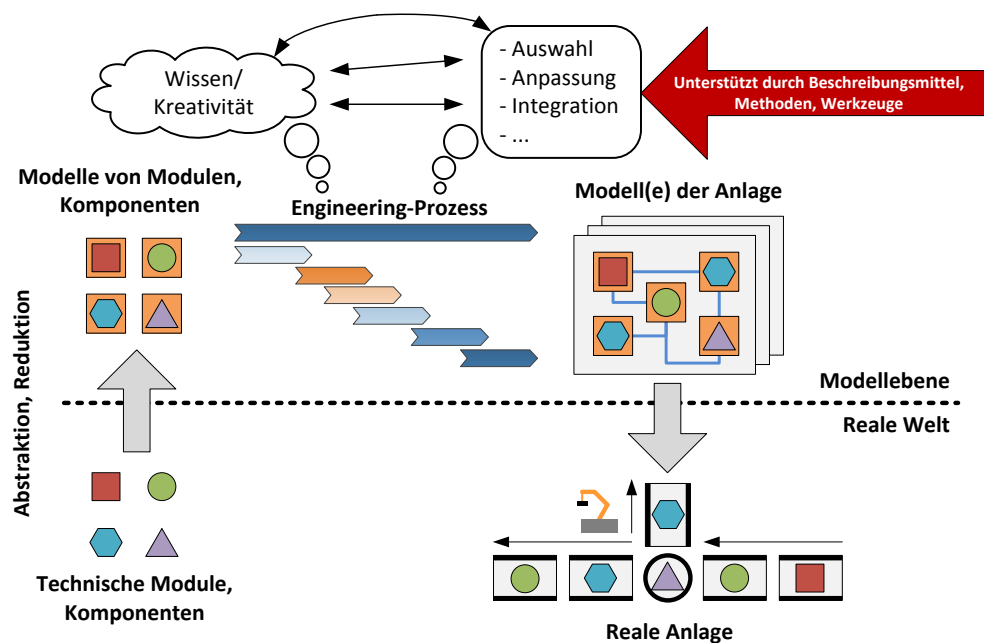


Abbildung 2-2: Prinzip des modellbasierten Engineering in Anlehnung an [WALÖ10]

2.3.2 Das Produkt-, Prozess-, Ressource-Prinzip als mögliche Basis des Datenmodells der Digitalen Fabrik

Das Bestreben des *Digitalen Fabrikbetriebs* ist es, ein für alle Planer gemeinsames Planungs- und Entwicklungsmodell zur Verfügung zu stellen, um möglichst paralleles Arbeiten zu ermöglichen, und damit eine Zeitersparnis im Engineering-Prozess zu erreichen [VDI 4499-2]. Das Bereitstellen von Modellen kann nur mit Hilfe eines *durchgängigen Datenmanagements* effizient durchgeführt werden [FAY09]. In diesem Kontext spielt einerseits die *Datenhaltung* (der Ort, an dem Daten gespeichert werden), andererseits das verwendete *Datenmodell* (wie die Daten abgespeichert werden) eine große Rolle. Insbesondere bei einem *durchgängigen Datenmanagement* kommt dem zugrunde liegenden Datenmodell eine besondere Bedeutung zu [VDI 4499-1]. Die *Datenhaltung* wie auch das zugrunde liegende *Datenmodell* haben einen großen Einfluss auf Art und Anzahl von Schnittstellen zwischen den für das Engineering genutzten CAE-Werkzeugen. Grundlegend lassen sich in der *Digitalen Fabrik* zwei Arten der *Datenhaltung* wie auch zwei verschiedene Arten von *Datenmodellen* unterscheiden [VDI 4499-2]:

- Die *dezentrale Datenhaltung*, bei der jedes Gewerk prinzipiell immer seine eigenen Daten im Sinne der Deutungshoheit behält. Die Änderungsrechte liegen ausschließlich bei dem Gewerk, das die Daten erstellt hat. Es gilt: „Zum [Daten-]Austausch zwischen den Gewerken müssen Schnittstellen und Methoden zur Synchronisation bereitgestellt werden“ [VDI 4499-2]. Bei einer dezentralen Datenhaltung wird zusätzlich zwischen einem dezentralen und einem zentralen Datenmodell unterschieden. Die dezentrale Datenhaltung, verbunden mit einem dezentralen Datenmodell, stellt in den meisten Fällen den derzeitigen Ist-Zustand im Engineering dar ([DFB11], → Abschnitt 2.6).
- Die *zentrale Datenhaltung*, bei der alle Gewerke auf einer gemeinsamen (zentralen) Datenbasis arbeiten. Hier gilt: „Sie greifen alle auf ein und dieselbe gemeinsame Datenbasis zu, in der die relevanten Daten aktuell gehalten werden“ [VDI 4499-2]. Die zentrale Datenhaltung beinhaltet immer auch ein *zentrales Datenmodell*. Integrierte Toolsuiten bieten in vielen Fällen eine zentrale Datenhaltung, während in einer heterogenen Werkzeuglandschaft die zentrale Datenhaltung eher unüblich ist.

Das im Rahmen der *Digitalen Fabrik* unter Beachtung der mechatronischen Sichtweise auf ein System und vor allem unter dem Paradigma der Verknüpfung von *Produkt-, Prozess- und Ressourcendaten* zu einem gemeinsamen Planungs- und Entwicklungsmodell postulierte *PPR-Konzept* ([FEL97], [VDI 4499-1]) kommt im *modellbasierten Engineering* eine besondere Bedeutung zu. Ziel des PPR-Konzepts ist es, alle für die Fabrikplanung und damit im Wesentlichen auch für die Errichtung einer fertigungstechnischen Anlage sowie von im Engineering-Prozess zur Unterstützung eingesetzten Methoden (wie Simulation) benötigten Daten in einem gemeinsamen Datenmodell zu beschreiben. Der Begriff *Produkte* repräsentiert in diesem Zusammenhang alle für das eigentliche Produkt relevanten Daten wie Maße, Gewicht und weitere relevante Eigenschaften. Unter dem Begriff *Prozesse* werden alle für die Herstellung wesentlichen Aspekte, die die konkrete Fertigung eines Produkts betreffen, zusammengefasst. Als *Ressourcen* werden die den eigentlichen Fertigungsprozess ausführenden technischen Apparaturen bezeichnet. Diese Daten sollen allen beteiligten Gewerken zur Verfügung stehen und von allen beteiligten Gewerken mit Daten gefüllt werden können. Das PPR-Konzept kann sowohl bei einer *dezentralen* als auch einer *zentralen Datenhaltung* zum Einsatz kommen. Das *Datenmodell der Digitalen Fabrik*, wenn unter dem Paradigma des PPR-Konzepts gesehen, sollte *spezifische Anforderungen* erfüllen, um während des Engineerings eingesetzt werden zu können [VDI 4499-1]:

- Das Datenmodell sollte eine *objektorientierte Modellierung* der Anlage und aller enthaltenen Elemente ermöglichen. Dies bedeutet, dass die abgebildete Anlagenstruktur (Ressourcen) in Kombination mit den relevanten Produkt- und Prozessdaten auf der Abstraktion und Klassifizierung von realen Objekten basieren sollte. Diese können dafür in Bibliotheken vorgehalten werden und entsprechend in dem Teil des Datenmodells, welcher die Anlagenstruktur beschreibt, instanziiert werden (Instanziierung von Klassen). Diese Instanzen sollten beliebig miteinander und auch hierarchisch kombiniert werden können (Aggregation von Objektinstanzen). Die Abstraktion von Objekten in Objektklassen kann in diesem Zusammenhang eine große Rolle bei der Übersichtlichkeit spielen, wofür [BAR11] folgendes Beispiel beschreibt: „So können beispielsweise Feldgeräte, welche sich in ihrer baulichen Ausprägung, [...] Leistungsdaten [und weiteren] unterscheiden, derart abstrahiert werden, dass sie aufgrund der resultierenden reduzierten Aspekte einer einzigen Klasse zugeordnet werden können“. Auf die Vorteilhaftigkeit einer durchgehen-

den Objektorientierung im Engineering automatisierter Anlagen weisen u. a. [KIE95] und [FAY06] hin, eine Anwendung zeigt u. a. [BAR11] sowie [LSB12].

- Aus der objektorientierten Abbildung folgt, dass sich Beziehungen und Abhängigkeiten zwischen Objekten im Datenmodell abbilden lassen müssen (Assoziationen zwischen Objektinstanzen). So ist die Verknüpfung zwischen Prozessobjekten, Produktobjekten und Ressourcenobjekten sowie die Verknüpfung der einzelnen Objektklassen untereinander notwendig, um ein gesamtheitliches Bild einer Anlage im Sinne des *PPR-Konzepts* abbilden zu können ([GÜFA08], [SCDR09]).

[BWG09] weisen zusätzlich darauf hin, dass die *allgemeinen Anforderungen* an das Datenmanagement (und damit an das Datenmodell) der *Digitalen Fabrik* in Analogie zu den nach [BAEB01] definierten Anforderungen an Datenbankmanagementsysteme zu sehen sind. Für das Datenmodell einer (fertigungstechnischen) Anlage sind insbesondere die Sicherstellung der für den jeweiligen Anwendungskontext vorhandenen *Datenaktualität*, *Datenvalidität* und auch *Datengenaugkeit* (sprich: Granularität) sowie der *Korrektheit*, *Konsistenz* und *Vollständigkeit* der Daten bedeutend. Abbildung 2-3 zeigt in idealisierter Form die Verknüpfung einzelner Teilergebnisse, die von am Engineering-Prozess beteiligten Gewerken erzeugt werden, in einem gemeinsamen Planungs- und Entwicklungsmodell. Es veranschaulicht ferner potenzielle Anwendungsfelder wie die *virtuelle Inbetriebnahme*. Angemerkt sei, dass nicht alle der aufgelisteten Aspekte für jeden Teil des Engineerings von gleicher Bedeutung sind. Vielmehr stellt Abbildung 2-3 eine Sammlung möglicher infrage kommender Aspekte, die im *Datenmodell der Digitalen Fabrik* abgebildet werden können, dar.

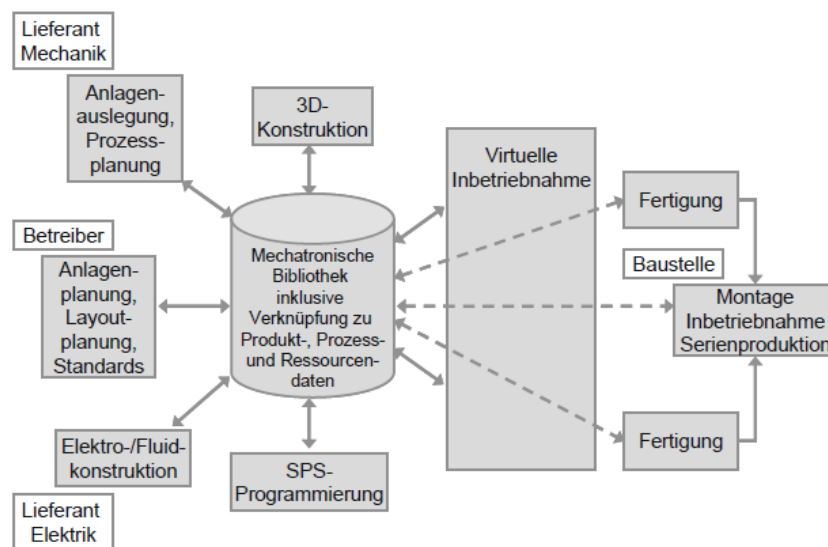


Abbildung 2-3: Idealisiertes, gemeinsames Planungs- und Entwicklungsmodell gemäß [VDI 4499-2]

Die als Modelle vorliegenden Projektergebnisse werden in den nächsten Abschnitten, beginnend mit einer sehr grundlegenden Beschreibung des Entstehungsvorgangs dieser Modelle, genauer betrachtet.

2.4 Auswahl und Dimensionierung von Ressourcen während des Engineering-Prozesses

Die Auswahl und Dimensionierung von Prozesse ausführenden Ressourcen stellt den Kern des kreativ-schöpferischen Vorgangs dar, der dem Engineering zugrunde liegt [FAY09]. Janschek [JAN10] definiert für mechatronische Systeme als generelles Ziel eines Systementwurfs die „[...] *Sicherstellung von Leistungseigenschaften eines technischen Systems [...] in Bezug zu [den] gestellten Anforderungen*“. Diese Anforderungen werden von einem Auftraggeber/Bauherren in Form eines Lastenhefts [VDI 3694] für das spezifische Projekt zur Verfügung gestellt und ausgeschrieben. Das Lastenheft entspricht in der Praxis typischerweise den formalen Anforderungen an ein Lastenheft, wie sie in [VDI 3694] vorgegeben werden. In der Praxis erfüllt das Lastenheft aber nicht immer die zu berücksichtigenden inhaltlichen Aspekte, die in [VDI 3694] definiert werden. Die spezifischen Kundenanforderungen an die umzusetzenden Funktionen/Prozesse einer automatisierten Anlage haben naturgemäß eine starke Auswirkung auf die Gestaltung und Dimensionierung der zu entwerfenden Anlage. Der potenzielle Lösungsraum (d. h. welche Ressourcen zu welchem Zweck und wie implementiert zum Einsatz kommen dürfen) wird oft von bauherrenspezifischen Lieferrichtlinien in Form von Best-Practices (v. a. im klassischen Maschinenbau, → Abschnitt 2.5.1), Geräte-Freigabelisten (vornehmlich im Bereich der Elektrotechnik, → Abschnitt 2.5.2) und vorgegebenen Softwarestandards (für die Automatisierungstechnik, → Abschnitt 2.5.3) eingeschränkt. Die hieraus resultierende Schnittmenge wird weiterhin durch gerätetechnische Eigenschaften und Randbedingungen der jeweiligen Anwendungsdomäne eingeschränkt [DIFA13].

Um eine anforderungsgerechte Dimensionierung und Auswahl der einzusetzenden Ressourcen treffen zu können, sollten die (vom Kunden aufgestellten und) im Lastenheft hinterlegten Anforderungen in quantifizierter Form vorliegen. Janschek [JAN10] bezeichnet diese quantifizierten Anforderungen im Kontext eines mechatronischen Systementwurfs als *Verhaltenskenngrößen*. Verhaltenskenngrößen repräsentieren die messbaren Anforderungsspezifikationen an eine automatisierte Anlage und beschreiben die *quantifizierten Verhaltenseigenschaften* von mechatronischen Systemen. Kundenanforderungen wie umzusetzende Best-Practices, einzuhaltende Positioniergenauigkeiten und weitere sowie insbesondere Prozessbeschreibungen liegen innerhalb eines Lastenhefts im Allgemeinen nicht in formalisierter, sondern in Prosaform vor. Hinzu kommt, dass Anforderungen in der Praxis oftmals als nicht vollständig oder als nicht stabil eingeordnet werden können. Nicht stabil bedeutet, dass sich Anforderungen während des Engineerings oder sogar noch während des Anlaufs einer Anlage ändern können. Es ist meistens Aufgabe der am Engineering-Prozess beteiligten Gewerke, Verhaltenskenngrößen aus den textuellen Anforderungen des Kunden zu extrahieren sowie die dort implizit vorhandenen *Verhaltenseigenschaften* zu quantifizieren, und sie somit im Sinne einer späteren Testbarkeit zu gestalten. Die Quantifizierung der *Verhaltenseigenschaften* selber kann durch spezielle *Verhaltensmetriken* während des Engineering-Prozesses erfolgen [JAN10], um zeitveränderliche, stochastische Einfluss- und Systemgrößen spezifizierbar und somit messbar und überprüfbar zu machen. Idealerweise geschieht dies unter Einbeziehung aller am Engineering-Prozess beteiligten Gewerke. Zusätzlich sollte der Bauherr mit einbezogen werden. In den folgenden Betrachtungen wird nicht im Speziellen auf die Quantifizierung der Verhaltenseigenschaften eingegangen, diese werden für das hier im Fokus stehende Gewerk Automatisierungstechnik als gegeben und somit durch die den eigentlichen technischen Prozess entwerfenden und dimensionierenden Gewerke (Mechanik, Elektrotechnik) als bereits erstellt vorausgesetzt.

[JAN10] führt weiter aus, dass im einfachsten Fall eine simple ja/nein Abfrage ausreicht, um funktionale Anforderungen testen zu können (z. B. Gerät ausgeschaltet oder nicht). Oftmals zielen Anforderungen aber auf Verhaltenskenngrößen stationärer oder dynamischer Natur ab [JAN10]. Beispiele hierfür können u. a. die Positioniergenauigkeit, Ausrichtungsstabilität, Übergangszeiten für den Kraftaufbau und mögliche Freiheitsgrade sein. Letztlich haben diese Verhaltenskenngrößen den entscheidenden Einfluss auf die Dimensionierung und endgültige Umsetzung der zu entwerfenden Anlage. Sie bilden die entscheidenden *Einflussfaktoren* bei der Auswahl und Dimensionierung von Ressourcen, die während des Engineerings vorgenommen wird. Sinngemäß entstehen dadurch im Laufe des Engineering-Prozesses Verbindungen bzw. ein Zusammenhang zwischen den von Bauherren vorgegebenen und spezifizierten Prozessen und den jeweiligen projektierten Ressourcen. Verhaltenskenngrößen zeigen aufgrund ihrer Charakteristik über alle mechanischen Aspekte hinweg Anlagenteile auf, die besonderer Betrachtung bedürfen. Abbildung 2-4 zeigt eine Darstellung dieses Prinzips unter Zuhilfenahme der *Formalisierten Prozessbeschreibung* gemäß [VDI 3682-1]. Der vom Kunden vorgegebene Prozess wird während des Engineerings um die aus den Kundenanforderungen abgeleiteten Verhaltenskenngrößen in Form von Merkmalen ergänzt (soweit nicht direkt vom Bauherren vorgegeben), die die Basis für die von den Gewerken Mechanik und Elektrotechnik vorgenommene Dimensionierung und Auswahl geeigneter Ressourcen darstellen. Das Gewerk Automatisierungstechnik setzt auf dieser Basis den Steuerungscode anhand der Kundenvorgaben und den Projektergebnissen der vorhergehenden Gewerke um.

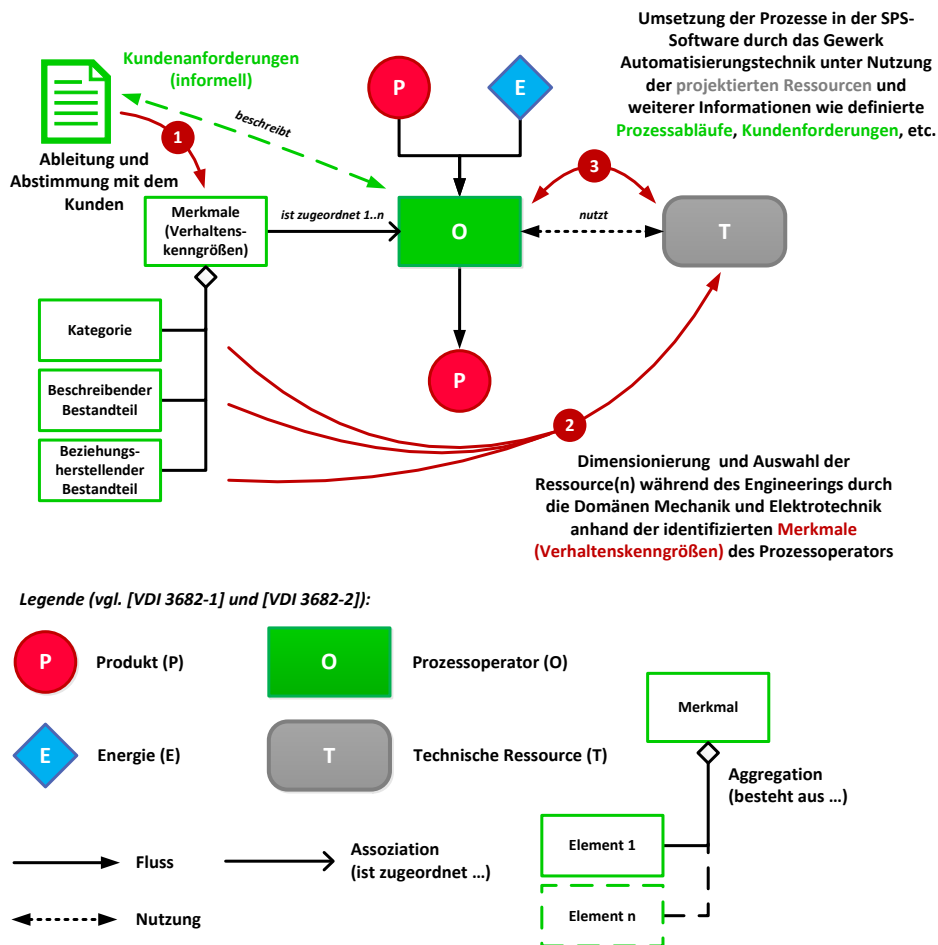


Abbildung 2-4: Verhaltenskenngrößen von Prozessen als wesentliche Basis für die Dimensionierung und Auswahl von Prozesse umsetzenden Ressourcen

2.4.1 Durch Gewerke eingesetztes Wissen

Die wertschöpfende Leistung eines Ingenieurs besteht nach [WALÖ10] „[...] in der Erbringung planerischer Konzepte zur Ausführung eines technologischen Prozesses in einer Anlage mittels geschickter Nutzung bzw. Kombination der Eigenschaften von Geräten, Maschinen oder Teilsystemen“. Während eines Engineering-Prozesses werden die vom Kunden vorgegebenen und zu automatisierenden Prozesse daher modelliert und das (informelle) Prozessmodell durch die beteiligten Gewerke schrittweise mit Wissen angereichert – Wissen um die konstruktive Umsetzung von Prozesse ausführenden Ressourcen, wie mechanischen Grundsystemen, und um die benötigten (automatisierungstechnischen) Geräte wie Sensoren und Aktoren, Feldgeräte wie E/A Baugruppen etc. Das mechatronische System wird durch den vom Gewerk Automatisierungstechnik programmierten Steuerungscode komplettiert. Das spezifische Wissen der beteiligten Gewerke wird daher im Laufe des Engineering-Prozesses zu einer Gesamtlösung integriert [SJF11].

Das von den Gewerken genutzte und zu einer Gesamtlösung integrierte Wissen kann als *implizites Wissen* (an Personen gebunden, schwer kommunizierbar, kaum formalisierbar, [VDI 5610-1]), einerseits, und *explizites Wissen* (auf verschiedenen Ebenen formalisierbar, damit kommunizierbar und speicherbar, [VDI 5610-1]) andererseits, charakterisiert werden [NON91]. Gemäß des HANSE-Modells [SNO00] teilt sich dieses Wissen in 5 unterschiedliche Komponenten auf, die die verschiedenen Wissensbedürfnisse eines Unternehmens (respektive eines am Engineering beteiligten Gewerks) darstellen können. Diese fünf Komponenten können nicht genau implizitem oder explizitem Wissen zugeordnet werden, sie zeigen vielmehr den fließenden Übergang zwischen implizitem und explizitem Wissen auf (→ Abbildung 2-5). Die fünf Komponenten des HANSE Modells können wie folgt charakterisiert werden (*Anm.: Es erfolgt eine teils wörtliche Übernahme aus [SNO00]*):

Heuristiken: Heuristiken können als „Daumenregeln“ oder Schätzungen beschrieben werden und stellen eine Möglichkeit dar, mit beschränkten Informationen effizient eine Entscheidung zu einem bestimmten Sachverhalt treffen zu können (wobei diese Entscheidung nicht zwingend die bestmögliche Entscheidung sein muss). Vornehmlich in Situationen, in denen Anforderungen als nicht bekannt oder als nicht stabil einzustufen sind, stellen Heuristiken ein wichtiges Mittel zur Entscheidungsfindung im Engineering dar [WEFR95].

Artefakte: Der Begriff *Artefakte* umschreibt das explizite Wissen und/oder die in irgendeiner Weise kodierte Information in Form von Daten, die einer Organisation/einem Gewerk zur Verfügung steht respektive von einem Gewerk während des Engineerings erzeugt wird.

Natürliche Begabung: Ist generell nicht übertragbar und an die jeweilige Person gebunden.

Skills: Beschreiben diejenigen Aspekte im Wissen von Personen, die mittels konkreter Tests gemessen werden können. Skills beschreiben das Sachverständnis, die praktische Befähigung oder auch die Fertigkeit einer Person.

Erfahrungen: Erfahrung kann als das wertvollste, gleichzeitig aber auch als das am schwierigsten zu handhabende Vermögen eines Unternehmens/Gewerks angesehen werden. Erfahrung ist immer an bestimmte Personen oder Gruppen gebunden.

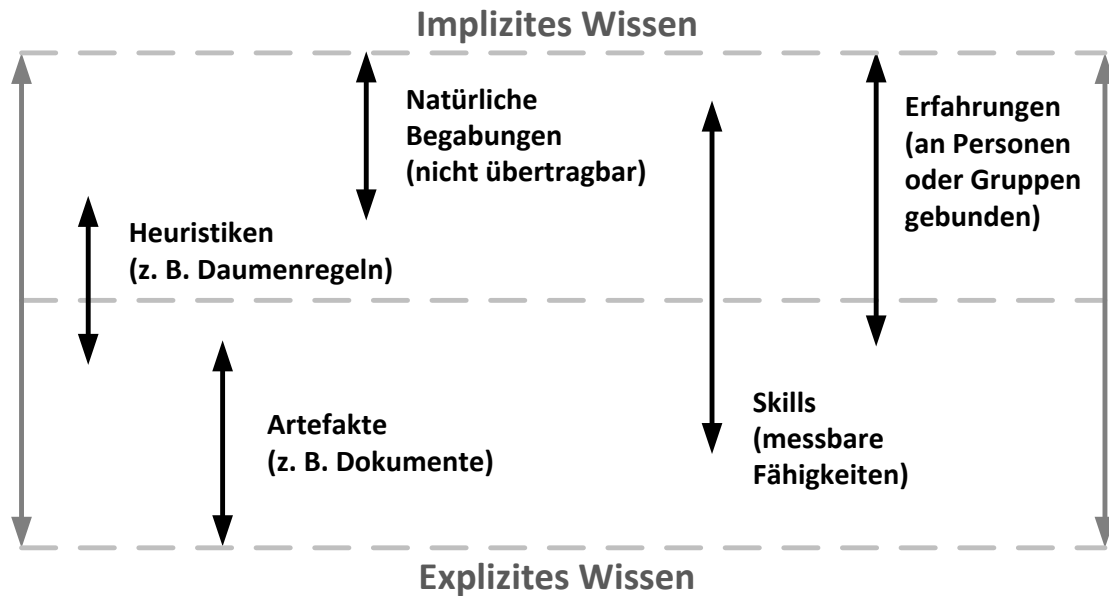


Abbildung 2-5: HANSE-Modell wie in [Sno00] beschrieben (angelehnt an [VDI 5610-1])

Es ist die Kombination einzelner Wissenskomponenten, die die Erstellung neuer und um gewerkespezifisches Wissen bereicherte Projektergebnisse (→ Abschnitt 2.5) ermöglicht [VDI 5610-1]. Aufbauend auf vorhandenen und von vorhergehenden Gewerken erstellten Artefakten wie Anforderungslisten, CAD-Zeichnungen, Excel-Tabellen werden durch die Anwendung von Heuristiken, natürlicher Begabung, Skills und Erfahrung neue und verfeinerte Modelle (→ Abschnitt 2.5) geschaffen, welche die zu planende, und zu dimensionierende Anlage immer feiner beschreiben.

Wissensbasierte Systeme, als Teilgebiet der künstlichen Intelligenz, versuchen in diesem Kontext das explizit wie implizit vorhandene Wissen mit Methoden der Wissensrepräsentation und Wissensmodellierung abbildbar und damit systematisiert nutzbar zu machen [BEKE06]. Die Bündelung von Wissen in wissensbasierten Systemen respektive Expertensystemen (u. a. [RUD06], [RFS⁺11]) stellt im Engineering eine Möglichkeit dar, multiple Problembereiche wie die Analyse (z. B. [SCH08]), die Diagnose (z. B. [CHR15]) und die Gestaltung von Maschinen und Anlagen (z. B. [GÜT12]) mit dem Ziel eines insgesamt effizienteren Engineerings zu unterstützen ([NgSc89], [RUN11]). Insbesondere KMU können die Verwaltung, den Aufbau und die Pflege derartiger Systeme aber nicht immer leisten [LANo01]. Stattdessen sind „[...] wertvolle Erfahrungen, Fachkenntnisse und Kundenwissen [...] oft in den Köpfen von wenigen Experten“ [LANo01]. KMU sind gemäß der Aussagen von [LANo01] in diesem Zusammenhang stärker als Großunternehmen von den Kenntnissen und Fähigkeiten einzelner Mitarbeiter abhängig.

2.4.2 Repräsentation des Wissens in Form von Projektergebnissen

Nach [WALÖ10] repräsentieren die eigentlichen, während des Engineerings erzeugten Modelle (→ Abschnitt 2.3) das *Kernwissen* eines Gewerks. Sie bilden damit ein Schlüsselglied in den Arbeitsprozessen des Engineering-Prozesses. Im Engineering-Prozess sind daher die erzeugten Artefakte in Form von 3D-CAD Daten, elektrischen Schaltplänen, (informellen) Prozessbeschreibungen und weiteren Modellen die eigentlichen Wissens- und Informationsträger, die als Basis zur Anwendung von Wissen dienen können. Es ist vornehmlich das *Erfahrungswissen* der Angehörigen des Gewerks Automatisierungstechnik (insbesondere des jeweiligen Projektleiters), kombiniert

mit *Heuristiken*, dass dem Gewerk Automatisierungstechnik die Interpretation von Anlagenplanungsdaten erlaubt. Das Gewerk Automatisierungstechnik sollte daher darin unterstützt werden, mit den zur Verfügung stehenden knappen (zeitlichen, monetären wie personellen) Ressourcen Anlagenplanungsdaten interpretieren zu können. Insbesondere gilt es zu erkennen, wo viel Aufwand bei der Konstruktion einzelner Teilelemente einer Anlage betrieben wurde, wo entsprechend weniger Aufwand und damit weitgehend Standardelemente verbaut wurden, und welche Konsequenzen dies auf die durch das Gewerk Automatisierungstechnik eingesetzten Methoden wie z. B. einer Simulation zur Absicherung von Arbeitsergebnissen haben könnte (→ Abschnitt 8.1). Im folgenden Abschnitt werden die einzelnen Projektergebnisse der am Engineering beteiligten Gewerke vorgestellt.

2.5 Durch Gewerke erstellte Projektergebnisse während des Engineering-Prozesses

Projektergebnisse liegen zumeist in Form von Modellen vor (→ Abschnitt 2.3). Sie werden mittels für den Einsatzzweck geeigneten Engineering-Werkzeugen erstellt (→ Abschnitt 2.6). Die folgenden Abschnitte bieten einen Überblick über die wesentlichen Projektergebnisse der am Engineering-Prozess beteiligten Gewerke.

2.5.1 Projektergebnisse des Gewerks Mechanik

Das Gewerk Mechanik spielt im Anlagenbau bei der Planung und konkreten Realisierung einer Anlage eine tragende Rolle (→ Abschnitt 2.2.1). Es übernimmt, unter enger Abstimmung mit dem Bauherren, die Konstruktion und Auslegung all derjenigen Elemente einer Anlage, die unter dem Begriff *mechanische Apparaturen* und *Antriebs Elemente* zusammengefasst werden können. Es greift dafür auf die Ergebnisse der Grobplanungsphase (in der Regel von Kundenseite ausgeführt) zurück. Ergebnis sind Modelle in Form von (3D-)CAD Konstruktionsdaten, die sich (im Sinne konstruktiver Gestaltung wie auch Umsetzung im jeweiligen CAD-Werkzeug, [VDA 09], [BWG09]) an den Best-Practices und Konfigurationen des jeweiligen Bauherren oder hauseigener Best-Practices orientieren. Die Projektergebnisse des Gewerks Mechanik entstehen nicht im Sinne eines *First-Time-Right*, sondern sukzessive während des gesamten Engineering-Prozesses von der Feinplanung bis hin zur Abnahme mit stetig steigendem Reifegrad. Grundsätzlich werden die aus der Grobplanungsphase resultierenden „*unscharfen Planungsdaten*“ [BMW08] vom Gewerk Mechanik zu konkreten Ausführungslayouts weiter verfeinert [BMW08]. Am Beispiel von Liefervorschriften für automatisierte Anlagen deutscher Automobilhersteller [ABV11] soll dies exemplarisch verdeutlicht werden. Grundsätzlich werden vier Meilensteine definiert, die sich über den gesamten Engineering-Prozess verteilen (*Anm.: Es erfolgt eine teils wörtliche Wiedergabe aus den Liefervorschriften gemäß* [BMW08] *sowie* [ABV11]):

Meilenstein 1 (Auftragslayout): Kurz nach der Beauftragung ist dem Auftraggeber ein Grob-Layout zu liefern, das den aktuellen Planungsstand als Referenz für die Integration in die benachbarten Anlagen zeigt. Nach Meilenstein 1 erfolgt die eigentliche *Feinplanungsphase*.

Meilenstein 2 (Freigabelayout): Nach Abschluss der *Feinplanungsphase* erfolgt mittels des *Freigabelayouts* die Freigabe des Konstruktionsentwurfs durch den Auftraggeber. Ist die Freigabe erfolgt, kann die *Fertigung der Anlage* erfolgen.

Meilenstein 3 (Ausführungslayout): Vor der Errichtung/Montage der Anlage beim Auftraggeber ist ein Ausführungslayout zu liefern, das die Änderungen seit der Freigabe des Konstruktionsentwurfs dokumentiert. Anschließend kann (nach entsprechender Abnahme) die *Montage und Inbetriebnahme* der Anlage beim Auftragnehmer erfolgen.

Meilenstein 4 (Bestandslayout): Zur (Vor-)Abnahme des Lieferumfangs ist ein aktualisiertes Layout zur Dokumentation des Zustands „as built“ zu liefern.

Abbildung 2-6 zeigt diese Meilensteine und wann sie während des Engineering-Prozesses erreicht werden.

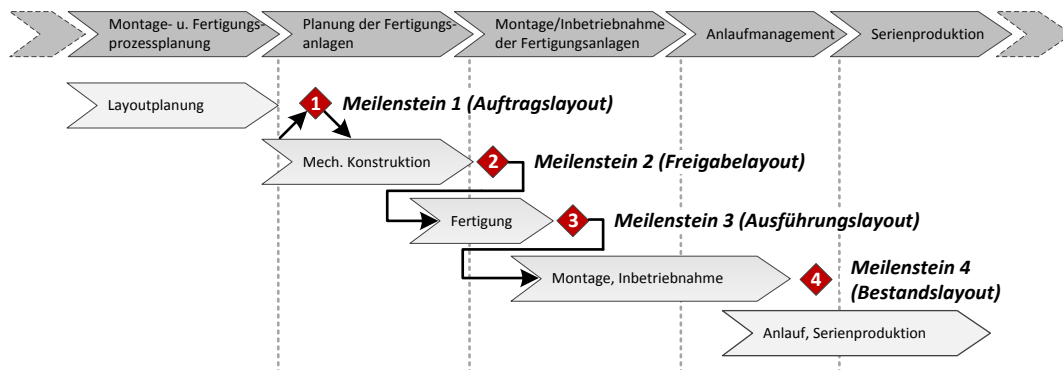


Abbildung 2-6: Meilensteine des Gewerks Mechanik während des Engineering-Prozesses

Zu beachten ist, dass es sich bei den o. g. Layouts (Meilenstein 1 – 4) nicht um die detaillierten Konstruktionsdaten aus dem zum Einsatz kommenden Konstruktionswerkzeug handelt, sondern um daraus abgeleitete grob granular aufgelöste Daten, die einen Überblick über die Anlage verschaffen sollen. Einzelne Objekte innerhalb einer Anlage (Förderbänder, Roboter, Sensoren, Aktoren) finden sich in den Layouts wie in den Konstruktionsdaten daher in unterschiedlicher Detaillierung und auch Granularität hinsichtlich ihres Aufbaus (→ Abschnitt 2.6). Grundsätzlich handelt es sich bei den Konstruktionsdaten wie auch den aus diesen abgeleiteten Layouts um Modelle, die anderen Gewerken im Laufe des Engineerings zur Verfügung gestellt werden. Der Begriff Modell lässt sich in diesem Kontext weit fassen: Ein Modell der Anlage kann ein geplottetes Anlagenlayout sein, eine handschriftliche Skizze, oder auch aus vielen einzelnen, heterogenen Modellen in unterschiedlichsten Datenformaten bestehen, die nicht oder nur wenig miteinander in Beziehung stehen.

Die Projektergebnisse des Gewerks Mechanik gehen über die Erstellung detaillierter Konstruktionsdaten und daraus abgeleiteten Layouts weit hinaus. Projektergebnisse des Gewerks Mechanik beinhalten u. a. auch Stücklisten über zu verbauende, mechanische Komponenten. Zusätzlich können auch Daten über Berechnungen zur Statik und Dynamik des betrachteten Systems [BWG09] beinhaltet sein wie auch eine Beschreibung der durch die Anlage auszuführenden Prozesse.

Diese Prozessbeschreibungen sollten technische Parameter wie Rampenzeiten und zu erreichende bzw. maximal zulässige Beschleunigungen umfassen. Im Gegensatz zur Verfahrenstechnik und der dort im Vordergrund stehenden prozesstechnischen Funktion stehen in der Fertigungstechnik (dreidimensionale) Struktur, Mechanik und Elektrik der Apparate im Vordergrund [MBS*11]. Dies ist von Relevanz, da bei fertigungstechnischen Anlagen, im Gegensatz zu einer verfahrenstechnischen Anlage, der Produktfluss nicht aus einer ununterbrochenen Kette von Komponenten nachvollzogen werden kann [GÜT12]. Dies hat zur Folge, dass eine Beschreibung dieser Prozesse (res-

pektive Produktflüsse) notwendig für das Verständnis einer Anlage sein kann. Es existiert für die Darstellung von Prozessen in der Fertigungsindustrie aber keine formale, anlagenunabhängige Prozessbeschreibung [MBS⁺11]. Die Darstellung erfolgt mittels einfacher Tabellen in Excel oder auch Gantt-Charts [DRA10]. Auch wenn mit der *Formalisierten Prozessbeschreibung* [VDI 3682-1] ein in der Abbildung von Prozessen skalierbares, neutrales Beschreibungsmittel zur Verfügung steht, dass sich auch für die Abbildung diskreter (fertigungstechnischer) Prozess eignet [HJS⁺15], findet dies derzeit noch keine praktische Anwendung in der Industrie. Eine kompakte Anlagenübersicht wie das aus der Verfahrenstechnik bekannte R&I-Fließbild, das prägnant wesentliche Inhalte einer (in diesem Fall verfahrenstechnischen) Anlage darstellen kann, ist in der Fertigungstechnik nicht bekannt.

2.5.2 Projektergebnisse des Gewerks Elektrotechnik

Wesentliche Ergebnisse, die aus dem Engineering-Prozess durch das Gewerk Elektrotechnik resultieren, sind der Stromlaufplan, Daten über die Schaltschrankkonstruktion sowie die technische Realisierung des Automatisierungssystems. In der Praxis ist es nicht ungewöhnlich, dass beauftragte (Unter-)Lieferanten die Elektrotechnik und den Steuerungscode aus einer Hand liefern – es erfolgt in einer derartigen Konstellation in fast allen Fällen aber eine klare organisatorische Trennung in verschiedene Fachabteilungen und Projektteams, die sich nur mit der Elektrokonstruktion und, davon getrennt, dem Steuerungscode beschäftigen. In dieser Arbeit wird der Fokus des Gewerks Elektrotechnik daher auf die Strukturierung wie auch elektrotechnischen Umsetzung des Automatisierungssystems, nicht auf der softwaretechnischen Umsetzung der auszuführenden Prozesse gelegt.

Zu den Aufgaben des Gewerks Elektrotechnik gehört gemäß [WEBr06] neben der Erstellung des Stromlaufplans das *Erstellen von Schutzfunktionen für Personal und Anlage* gemäß [DIN EN ISO 12100] sowie [DIN EN ISO 13849-1]. Im Vordergrund stehen nicht das Entwerfen und Entwickeln neuer Schutzfunktionen, sondern der zielgerichtete Einsatz und die Kombination von bereits vorkonfektionierten und geprüften Geräten wie Lichtvorhängen, Lichttastern etc. Dazu gehören auch geeignete SPS-Komponenten, wie fehlersichere Steuerungen und fehlersichere Kommunikationsperipherie. Zu den Aufgabenbereichen des Gewerks Elektrotechnik gehört ferner die Auslegung des genutzten Feldbusses. Entsprechend der genutzten Technologie (z. B. PROFINET, PROFIBUS, AS-Interface) entwirft das Gewerk Elektrotechnik die zugrunde liegende Busstruktur und stellt die elektrotechnische Funktionsfähigkeit der projektierten AT-Geräte sicher. Die technische Machbarkeit, die Anforderungen an den Datendurchfluss und die beiden wichtigen Eigenschaften Echtzeitfähigkeit sowie Ausfallsicherheit sind Entscheidungsgrößen, die für die Auswahl eines geeigneten Feldbusses beachtet werden.

Oftmals werden diese Konfigurationen bereits durch den Bauherren vorgegeben. Der Auswahlprozess für spezifisch einzusetzende elektrotechnische Komponenten wird daher in der Regel durch (bauherrenspezifische) Freigabelisten, bei denen die zu verwendenden Komponenten aufgeschlüsselt sind, vereinfacht. Der Lösungsraum an verwendbaren automatisierungstechnischen Komponenten wird so mitunter stark eingeschränkt. Die Umsetzung der elektrotechnischen Konstruktion erfolgt aufbauend auf den vom Bauherren vorgeschriebenen Best-Practices und Richtlinien. Sie baut, im Gegensatz zu den Arbeiten des Gewerks Mechanik, nicht auf einem vom Bauherren während der Grobplanung erstellten (elektrotechnischen) Layout auf, sondern auf leeren

Standardprojekten, die insbesondere Einstellungen im verwendeten Engineering-Werkzeug, Beschriftungs- wie Farbkonventionen und weiteres vorgeben.

Diese Standardprojekte werden anschließend sukzessive, unter üblicherweise starker Wiederverwendung einzelner Teilaspekte aus vorangegangenen Projekten, gefüllt. Abhängig davon, ob Betreiber oder Gewerke, die eine Anlage in Verkehr bringen, für den laufenden Betrieb verantwortlich sind, können Abweichungen von landestypischen Vorschriften möglich sein. Ähnlich zu den Projektergebnissen des Gewerks Mechanik entstehen die Projektergebnisse des Gewerks Elektrotechnik nicht im Sinne eines *First-Time-Right*, sondern sukzessive während des gesamten Engineering-Prozesses. Die erstellten Elektropäne weisen daher unterschiedliche Reifegrade, je nach Projektfortschritt und ähnlich zu den vom Gewerk Mechanik erstellten Layouts, auf. Üblicherweise wird parallel zur Programmierung des Steuerungscode ein aktualisierter Elektropän für das Gewerk Automatisierungstechnik ausgegeben, welcher in einen vorläufigen Baustellenstand des Elektropäns mündet. Der Baustellenstand ermöglicht ein Inbetriebnehmen der Anlage, ist in der Praxis aber in keinem Fall bereits vollständig und muss daher in jedem Fall nochmals überarbeitet werden. Im Laufe der Installation und der Inbetriebnahme wird der Baustellenstand zum endgültigen Planungsstand verfeinert. Aus diesem müssen die zumeist umfangreichen händisch vorgenommenen Korrekturen in den Abgabepänstand nachgezogen werden.

2.5.3 Projektergebnisse des Gewerks Automatisierungstechnik

Das Projektergebnis des Gewerks Automatisierungstechnik ist derjenige [IEC 61131-3] Steuerungscode, der auf der Steuerung einer automatisierten Anlage zum Einsatz kommt und die Anlage gemäß den Anforderungen des Bauherren steuert und überwacht. Zum Lieferumfang gehört in vielen Fällen auch noch die Bedien- und Beobachtungssoftware mittels Human-Machine-Interface (HMI) – in Form von Bedienpanels, die über die Anlage verteilt sind und ein Eingreifen durch Bediener in den Produktionsprozess vor Ort ermöglichen. Grundsätzlich wird in allen Sprachen der [IEC 61131-3] programmiert, wobei Kontaktplan (KOP) aufgrund der zugrunde liegenden Ähnlichkeit zu Stromlaufplänen und der generell einfachen Nachvollziehbarkeit von Verriegelungen und/oder Schaltbedingungen oftmals bevorzugt wird. In der Praxis wird die Verwendung von KOP vielfach von den Bauherren vorgeschrieben. Die grundsätzliche Struktur des Steuerungscode kann in guter Näherung als *prinzipiell immer gleich aufgebaut* beschrieben werden. Ein Beispiel für einen derartigen Aufbau ist in Abbildung 2-7 gegeben.

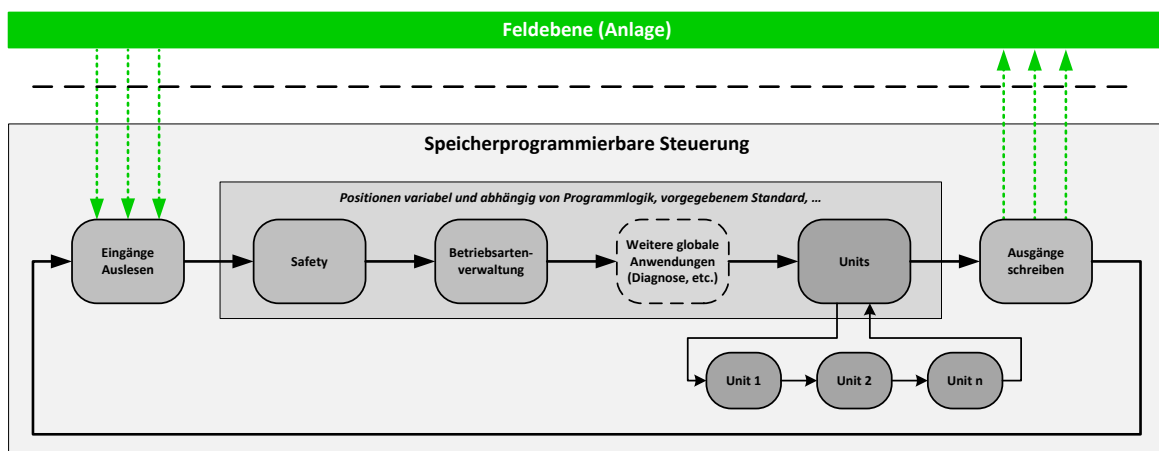


Abbildung 2-7: Beispiel für einen programminternen Ablauf einer SPS

Entscheidend für die Funktionalität einer Anlage ist derjenige Programmcode, der den eigentlichen technischen Prozess steuert, und der im Teil der „Unit“ hinterlegt ist. Dieser enthält „[...] den Sollablauf für den Abarbeitungsprozess“ [Güt12]. Der dort hinterlegte Steuerungscode unterscheidet sich von Anlage zu Anlage stark, was auch für den Safety-Teil des Programmcodes gilt. Andere Teile wie die Betriebsartenumschaltung und weitere, möglicherweise auf vorgegebenen Standards basierende Programmteile sind anlagenübergreifend zumindest ähnlich – lediglich die Parametrierung wird anlagenspezifisch vorgenommen.

Grundsätzlich lassen sich beim Engineering von [IEC 61131-3] Steuerungscode zwei Herangehensweisen unterscheiden, die eine direkte Auswirkung auf die Erstellung wie auch auf den Reifegrad in Hinsicht auf die Inbetriebnahme des Programmcodes haben: Die *freie inhaltliche Gestaltung des Programmcodes* sowie die Nutzung von bereits *vorgefertigten Funktionsbausteinen* in Form von Firmen-/Werkstandards.

Freie inhaltliche Gestaltung des Programmcodes

Die freie inhaltliche Gestaltung des Programmcodes sichert dem Gewerk Automatisierungstechnik die maximale Flexibilität beim Engineering zu. Viele Anlagenbetreiber/Bauherren überlassen den programminternen Ablauf in der Hand des Zulieferers und schreiben höchstens Bezeichnungen von Operanden und Betriebsmitteln und die grobe Struktur des Programms vor. In der Regel unterteilt diese Strukturvorgabe das Programm in spezifische Funktionsblöcke für einzelne Anlagenteile (Abbildung 2-7, Units), um die Übersichtlichkeit zu wahren. Zudem wird dadurch die spätere Instandhaltung der Anlage vereinfacht. Außerhalb der Units werden bestimmte, zur Überwachung/Steuerung der Produktion verwendeten Technologien in Form von Firmen-/Werkstandards zur Verfügung gestellt. Ein Beispiel ist das von der Firma BMW in der Produktion verwendete Prozessleitsystem *Legato*, dort *IPS-T/L/Q* genannt (für Technik/Logistik/Qualität, [MABU06], [HAP08]), das in den PLx-Standards ausgegeben wird. Ein *frei gestalteter* Programmcode kann zielgerichtet auf die Planungsergebnisse der vorhergehenden Gewerke angepasst oder erst auf Grundlage dieser Projektergebnisse entworfen werden. Die Vorteile eines derartigen Vorgehens liegen in der bestmöglichen Einbeziehung des Wissens des Gewerks Automatisierungstechnik. Das gesammelte Erfahrungswissen der beteiligten Projektingenieure kann in das Projekt eingebracht werden. Des Weiteren besteht die Möglichkeit, eine im Idealfall bereits herstellerübergreifend zum Einsatz gekommen Lösung, die üblicherweise einen hohen Reifegrad besitzt, zu verwenden. Ein möglicher Nachteil liegt im potenziell erhöhten Schulungsaufwand für die Instandhaltung, die erst an den für sie fremden Code gewöhnt werden muss. Zudem besteht die Möglichkeit, dass die Instandhaltung vermehrt Änderungswünsche mit einbringt, die potenziell große Änderungen am Programmcode notwendig machen können.

Gestaltung des Programmcodes aufbauend auf vorgegebenen Funktionsblöcken (Werkstandards)

Bauherren und Anlagenbetreiber liefern oftmals bereits vorgefertigte Bibliotheken mit einzusetzenden IEC 61131-3 Bausteinen in den bereits beschriebenen Werkstandards aus. Diese sind für die Projektierung und Programmierung des Steuerungscode entsprechend zu verwenden. Im Idealfall ist nur noch das Aufrufen und Parametrieren dieser Bausteine notwendig (je nach gewünschter Funktionalität). Dies soll nachfolgend Anhand eines einfachen Beispiels erläutert werden.

In einer Fördertechnikapplikation gebe es Bausteine für die Steuerung von Förderbändern mit nur einer Richtung, zwei Richtungen, spezielle Bausteine für Eckumsetzer, Hubtische, usw. Aus dieser Lösung hervorgehende Vorteile können wie folgt identifiziert werden:

1. Eine potenziell durchgehende Standardisierung des Steuerungscode über Anlagengrenzen hinweg begünstigt positive Nebeneffekte wie Schaffung von Synergieeffekten bei der anlagenübergreifenden Instandhaltung.
2. Eine prinzipiell hohe Wiederverwendbarkeit bereits bestehender Steuerungscode-Artefakte.
3. Potenziell weniger Aufwand bei der Programmierung.

Eine Voraussetzung für die genannten Vorteile ist ein hohes Maß an Synchronisation der mechanischen wie elektrotechnischen Umsetzung bei der Planung der Fertigungseinrichtung – so muss die mechanische Apparatur in Zusammenspiel mit den geplanten Sensoren und Aktoren so konstruiert sein, dass eine Steuerung durch die vorgegebenen Standard-Funktionsblöcke möglich ist. Die Sicherstellung der Kohärenz der mechanischen Apparaturen zu den vorgegebenen Funktionsblöcken ist daher eine maßgebliche Herausforderung während des gesamten Engineering-Prozesses, die aber oftmals vernachlässigt wird (→ Abschnitt 2.2). „Best-Practices“ in Form von vorgegebenen Funktionsbausteinen stellen daher Insellösungen dar, die oftmals nicht über alle mechatronischen Aspekte hinweg synchronisiert sind. Die möglichen Vorteile im Vergleich zur bereits beschriebenen freien inhaltlichen Lösung können so nicht immer zum Tragen kommen, sondern werden vielmehr ins Gegenteil verkehrt: Der Aufwand, die zu verwendenden Bausteine überhaupt erst nutzbar zu machen, kann so hoch sein, dass das eigentlich positive Nutzen/Aufwand Verhältnis umgekehrt wird. Letztendlich muss umständlich um die Bausteine „herum programmiert“ werden, um deren Einsatz überhaupt erst zu ermöglichen. Zudem besteht die Möglichkeit, dass Werkstandards und -vorschriften auf veralteten Technologien basieren und so z. B. keine Multiinstanziierung erlauben (wenn auch technisch machbar), was die [IEC 61131-3] Softwareerstellung wie auch Pflege stark beeinträchtigen kann. Ausnahmen hiervon sind höchstens bei einfachen Standardlösungen wie z. B. Zuführ-Fördertechnik festzustellen, die in der Regel auf die einfachst mögliche Weise implementiert werden.

Die Qualität (oder auch der Reifegrad) des Steuerungscode, der in der Phase der Programmerstellung programmiert wird, spielt eine entscheidende Rolle bei der Inbetriebnahme, wie in Abschnitt 2.8 dargelegt werden wird. Unabhängig davon, ob der Steuerungscode inhaltlich frei oder aus vorgegebenen Funktionsbausteinen umgesetzt wird, hängt der Reifegrad des Steuerungscode naturgemäß in hohem Maß von der Erfahrung und dem Wissen der am Projekt beteiligten Ingenieure ab. Entscheidend ist, ob verstanden wurde, wie die vom Auftraggeber geforderten Prozesse ablaufen sollen. Durch das Nicht-Vorhandensein einer standardisierten und formalen Prozessbeschreibung ist dieses Verständnis erschwert. Die Softwareerstellung wird ferner dadurch verkompliziert, dass bestimmte Teile der Software nur auf der Baustelle (also während der Inbetriebnahme) programmiert werden können ([VDW97], [WÜN08]). Beispiele für derartige Codefragmente können Handhabungsbausteine zur Ansteuerung von Feldgeräten sein, deren genaue Funktionalität erst auf der Baustelle bekannt wird (weil erst dann feststeht, welche Geräte wirklich zum Einsatz kommen) oder ganz generell Abläufe, die sich aus der textuell vorliegenden Prozessbeschreibung nicht klar identifizieren lassen und erst „auf der Baustelle“ fertiggestellt werden, so dass kein frühzeitiger Test stattfinden kann.

Im folgenden Abschnitt werden die für die Erstellung der in diesem Abschnitt beschriebenen Projektergebnisse genutzten Werkzeuge näher beschrieben.

2.6 Verwendete Werkzeuge im Engineering-Prozess und deren Bedeutung

Gemäß dem BMW-Prinzip spielen Werkzeuge und deren zielgerichteter Einsatz eine wesentliche Rolle während des gesamten Lebenszyklus einer automatisierten Anlage [SCH99A]. Jedes am Engineering-Prozess beteiligte Gewerk bringt verschiedene Engineering-Werkzeuge unterschiedlicher Hersteller (mitunter auch Eigenentwicklungen) in den Engineering-Prozess mit ein [FAY05], um den gewerkespezifischen Aufgabenkomplex zu lösen. Jedes Werkzeug steht für sich alleine – allen gemeinsam ist, dass sie Daten respektive Modelle erzeugen, die von anderen Gewerken zur Aufgabenlösung potenziell benötigt werden.

Einen Überblick über gängige Werkzeuge bei der Grobplanung von Fertigungseinrichtungen (Fabriklayout, → Abschnitt 2.5.1) bietet der VDA-Leitfaden *Informationen zum 3D-Datenaustausch in der Fabrikplanung* [VDA 09], der für die weiteren Betrachtungen beispielhaft herangezogen wird. Allgemein wird dort bei der Planung des grob granularen [BWG09] Fabriklayouts zunächst in spezifische Werkzeuge für *Architektur* (z. B. Bentley Architecture [BENTLEYARCH[®]]), *Stahlbau* (z. B. TRICAD BM), *technische Gebäudeausrüstung* (z. B. TRICAT MS) und *Fördereinrichtungen* (z. B. TRICAD FT) unterschieden [TRICAD[®]]. Der Detaillierungsgrad der Daten der Grobplanung ist „[...] so zu wählen, dass eine rasche Übersicht für die Planer und Entscheider ermöglicht wird, um die Funktionsweise der Fabrik zu verstehen, und zu überprüfen“ [BWG09]. Nicht jedes einzelne CAE-Werkzeug respektive jeder betrachtete Aspekt der Layoutplanung ist gleichermaßen bedeutend für die anschließende Anlagenkonstruktion. Wichtig für die eigentliche (mechanische) Konstruktion sind vor allem Daten der CAE-Werkzeuge für Fördereinrichtungen sowie den Stahlbau. Für das Gewerk Elektrotechnik können Daten der technischen Gebäudeausrüstung eine größere Rolle spielen. Die im Zusammenhang mit der Layoutplanung eingesetzten Werkzeuge eignen sich nicht zwangsläufig für die detaillierte Auskonstruktion von Anlagen bei der Feinplanung, so dass hierfür spezielle Werkzeuge eingesetzt werden [BWG09]. Die verschiedenen Werkzeuge der Layoutplanung ermöglichen daher zunächst nur die Erstellung von Basislayouts für die im Engineering-Prozess nachfolgende Konstruktion spezifischer Fertigungseinrichtungen (→ Abschnitt 2.5.1). Sie stellen im *Digitalen Fabrikbetrieb* den ersten grundlegenden Schritt beim Engineering von Fertigungseinrichtungen dar. Zur Absicherung der Layoutplanung können ferner Simulationswerkzeuge zum Einsatz kommen, mit deren Hilfe das Anlagenlayout vor der detaillierten Auskonstruktion der Anlage abgesichert werden kann. Eine genauere Betrachtung von Simulationsmöglichkeiten findet gesondert in Kapitel 3 statt.

In der eigentlichen Anlagenkonstruktion im Sinne der Feinplanung von Fertigungseinrichtungen kommen „klassische“ CAD-Werkzeuge wie CATIA [CATIA[®]], Pro/ENGINEER [PRO/ENGINEER[®]] oder auch NX [NX[®]] zum Einsatz [VDA 09], wiederum in bauherrenspezifischer Konfiguration. Der notwendige Datenaustausch zwischen CAE-Werkzeugen der Layoutplanung und CAE-Werkzeugen für die Anlagenkonstruktion erfolgt in beide Richtungen [VDA 09] (sofern möglich), wobei die Rücktransformation von detaillierten (Anlagen-) Konstruktionsdaten in Werkzeuge des Fabriklayouts mit einer Reduktion des Detaillierungsgrads verbunden ist [VDA 09]. Die detaillierte Abbildung von Schrauben oder Normteilen einer komplexen Maschine spielt im Fabriklayout naturgemäß keine Rolle [BWG09]. Ferner können Werkzeuge für die Berechnung der Statik und Dynamik einer Anlage (z. B. während eines „*Virtuellen Engineerings*“ [DSK⁺13]) sowie zur Simulation grundlegen-

der Prozesse (z. B. zur Sicherstellung der Kollisionsfreiheit einer mechanischen Apparatur, → Kapitel 3) zum Einsatz kommen [WSR92]. Zu beachten ist, dass nicht jedes zur Konstruktion genutzte Engineering-Werkzeug nach dem Paradigma der Objektorientierung arbeitet. Auch wenn das Engineering-Werkzeug Objektorientierung unterstützt, wird in der Praxis häufig noch das rein *Feature-basierte Modellieren* [VDI 2218] angewendet. *Feature-basiertes Modellieren* bedeutet, „[...] dass ein rechnerinternes Produkt oder Bauteilmodell vom Konstrukteur [...] direkt mit Hilfe der als Features vordefinierten Gestaltelemente erzeugt wird [...] und nicht nur durch das Zusammenfügen von Linien, Flächen und Grundkörpern“ [VAJ94]. Objektorientierung kann *Feature-basierte Modellierung* zwar unterstützen, ist aber keine zwingende Voraussetzung dafür [VDI 2218].

Die vom Gewerk *Elektrotechnik* eingesetzten Werkzeuge nutzen Daten der Ergebnisse der Layoutplanung (insbesondere der technischen Gebäudeausrüstung) sowie Daten aus der Anlagenkonstruktion (Feinplanung, insbesondere Auslegungsdaten für elektrische Antriebe und andere Stellgeräte), um den spezifischen Aufgabenkomplex des Gewerks zu lösen. Typische zum Einsatz kommende Werkzeuge sind E-CAD Werkzeuge wie *EPLAN* [EPLAN[®]] zur Erstellung von Schaltplänen und zur Schaltschrankkonstruktion sowie zur Planung der benötigten Kabellängen. Obwohl der Markt weitere Werkzeuge kennt, ist die Verbreitung des Werkzeugs *EPLAN* (insbesondere im deutschsprachigen Raum) als so hoch einzuschätzen, dass von einem Quasi-Standard gesprochen werden kann und andere Werkzeuge in der Praxis nur eine untergeordnete Rolle spielen. Auf eine Betrachtung alternativer Werkzeuge (wie z. B. *WSCAD* [WSCAD[®]], *ESplan* [ESPLAN[®]] oder *ELTIME* [ELTIME[®]]) wird daher verzichtet, da sich nur die Ausprägung der erstellten Daten und Modelle unterscheidet, am Ende das Projektergebnis aber dasselbe ist. Im Folgenden werden die Begriffe Schaltplan und *EPLAN* synonym genutzt.

Bei den Arbeiten des Gewerks *Automatisierungstechnik* kommen Werkzeuge zum Einsatz, die das Programmieren, Parametrieren und Konfigurieren der eingesetzten Steuerung und der angeschlossenen dezentralen Peripherie (automatisierungstechnische Komponenten wie E/A-Baugruppen oder ähnliche) ermöglichen. Hersteller automatisierungstechnischer Komponenten und Steuerungen wie Siemens, ABB, Beckhoff und weitere bieten hierfür proprietäre Lösungen an, die aufgrund der unterschiedlichen Implementierung der in der [IEC 61131-3] definierten Sprachen nicht untereinander kompatibel sind. Typische, im Projektgeschäft zum Einsatz kommende Werkzeuge sind *Step 7* [STEP 7[®]], *CODESYS* [CODESYS[®]], *PCWorx* [PCWORX[®]], *e!COCKPIT* [E!COCKPIT[®]]. Hinzu kommt eine große Anzahl an firmenspezifischen Parametrier- wie auch Konfigurationswerkzeugen für einzelne automatisierungstechnische Komponenten wie Feldumrichter, E/A-Baugruppen, sicherheitstechnische Einrichtungen wie Lichtvorhänge, Laserscanner und weitere. Zusätzlich werden durch das Gewerk *Automatisierungstechnik* Werkzeuge zur Erstellung des HMI wie *WinCC* [WINCC[®]], *WinCC Flexible* [WINCCFLEXIBLE[®]] oder *Zenon* [ZENON[®]] eingesetzt.

Gewerke, die an der Planung und Errichtung einer fertigungstechnischen Anlage beteiligt sind, sehen sich in der Praxis vor die Tatsache gestellt, nicht frei zwischen Werkzeugen wählen zu können ([VDA 09], [BWG09]). Planer von Fertigungsanlagen (interne wie externe) respektive am Engineering-Prozess beteiligte Gewerke sind zumeist angehalten, die vom jeweiligen Bauherren vorgegebenen CAE-Werkzeuge für die Planung zu nutzen. Dies ist vorwiegend darin begründet, die Datenintegrität, welche die höchste Priorität im Planungsprozess genießt, in jedem Fall sicherzustellen [BWG09]. Unter Datenintegrität wird in diesem Zusammenhang vor allem die Korrektheit und Vollständigkeit der zu übermittelnden Daten verstanden. Begründet liegt dies in den nicht vorhandenen oder nur rudimentären Schnittstellen zwischen verschiedenen Engineering-

Werkzeugen, so dass ein effizientes Engineering im Sinne der Interoperabilität zwischen verschiedensten Werkzeugen [DFB11] im Hinblick auf die verbreitete, dezentrale Datenhaltung derzeit nicht immer möglich ist. Ein fehlerfreier Datenaustausch gelingt derzeit nur im nativen Datenformat zwischen gleichen Systemen, deren Konfiguration durch Standardisierung identisch ist [VDA 09]. Das im *Digitalen Fabrikbetrieb* als wesentliches Merkmal beschriebene *durchgängige Datenmanagement* [VDI 4499-2] ist derzeit effizient nur innerhalb geschlossener Werkzeugketten mit a) einer zentralen Datenhaltung mit zentralem Datenmodell ([SCH99B], [VDI 4499-2]) und/oder b) aufeinander abgestimmten und konsistenten Schnittstellen zwischen in einer integrierten Toolsuite enthaltenen Werkzeugen möglich ([KIE07], [DFB11]). Beispiele für integrierte Toolsuiten können die Siemens PLM Software [SIEMENSPLM[®]], Siemens Teamcenter [TEAMCENTER[®]] oder auch die 3DEXPERIENCE Plattform [3DEXPERIENCE[®]] der Firma Dassault sein.

Die spezifisch einzusetzenden Werkzeuge und Konfigurationen können sich von Bauherren zu Bauherren, sogar von bauherrenspezifischer Fachabteilung zu Fachabteilung stark unterscheiden [VDA 09]. Nur vereinzelt existieren Bestrebungen, diese für Teilaspekte des Anlagenbaus zu vereinheitlichen (in Verbänden wie dem VDA [VDA[®]] oder VDI [VDIA[®]], siehe z. B. [ABD⁺08]). Konzepte neutraler Datenaustauschformate wie AutomationML [DRA10] zielen darauf ab, den Datenaustausch zwischen unterschiedlichen Engineering-Werkzeugen zu ermöglichen, und so die Interoperabilität in heterogenen Werkzeugketten wie potenziell auch in integrierten Toolsuiten sicherzustellen [DFB11]. Allerdings stoßen diese Ansätze in der Praxis aufgrund der Vielzahl an unterschiedlichsten Konfigurationen, diametralen Auslegungen der Kernkonzepte der *Digitalen Fabrik* (insbesondere Auslegung des zentralen Datenmodells) und divergenten Strategien von Werkzeugherstellern an ihre Grenzen. Letztlich bedeutet dies, dass selbst beim Einsatz integrierter Toolsuiten keine inhärente bauherren- respektive fachabteilungsübergreifende Konsistenz erreicht werden kann. Der Verbund von Insellösungen im Engineering kann mit Hilfe eines integrierten Engineering-Ansatzes gelöst werden ([KIE07], [MAN09]). Ein integrierter Engineering-Ansatz lässt sich aber nicht zwangsläufig auch bauherrenübergreifend etablieren. Es bilden sich stattdessen neue, bauherren- respektive fachabteilungsspezifische Insellösungen ([KKS⁺06], zitiert nach [KIE07]). In diesem Zusammenhang kann auch von *vielen verschiedenen Digitalen Fabriken* und entsprechend vielen *Datenmodellen der Digitalen Fabrik* gesprochen werden. Dies stellt nicht nur große Engineering-Organisationen, sondern insbesondere KMU vor große organisatorische wie auch monetäre Herausforderungen, da im Regelfall mehr als ein Bauherr bedient wird und so die Anzahl an eingesetzten Werkzeugen und Konfigurationen wie auch zur Verfügung gestellten Datenmodellen sehr hoch sein kann. Aktuelle Forschung widmet sich insbesondere der Entkopplung des Werkzeugzwangs durch die Sicherstellung der Interoperabilität zwischen verschiedenen Engineering-Werkzeugen, so dass jedes Gewerk seine favorisierten Werkzeuge [DFB11] einsetzen kann. Diese Entwicklung erfordert zwei Aspekte: Auf der einen Seite müssen unterstützende Methoden und Beschreibungsmittel entwickelt werden ([DFB11], [ScDR09], [JCS⁺12]), die ein durchgängiges Engineering unterstützen und erst möglich machen. Auf der anderen Seite ist eine Standardisierung bei der Abbildung von Daten im Datenmodell der Digitalen Fabrik, wie sie beispielhaft von [GÜFA08] vorgestellt wurde, notwendig.

2.7 Aufwands- und Kostenverteilung in Engineeringprojekten

Eine im Jahr 2015 veröffentlichte Studie des VDMA [OERE15] geht bei der Kostenverteilung im Maschinen- und Anlagenbau von einer Teilung von 70% Kosten für Mechanik und 30% für Elektro-

technik/Software aus. Diese beispielhafte Aufteilung stellt einen Mittelwert über die in der Studie betrachteten Anlagenbauprojekte dar und kann, abhängig von der umzusetzenden Anlage, der verwendeten Technologie und dem Innovationsgrad der umzusetzenden Lösung, erheblich schwanken. Eine Studie der AIDA (*Automatisierungsinitiative Deutscher Automobilhersteller*, zitiert nach [FAY09]) zeigt für den Anteil Steuerungstechnik/Software am Beispiel einer Rohbaufertigungszelle im Automobilbau einen Anteil von 55% für die Planung, Programmierung und Inbetriebnahme, während Kaufteile und Fertigung (inkl. Schaltschrankbau) nur 39% der gesamt anfallenden Kosten ausmachen. Die restlichen Aufwände verteilen sich auf Projektmanagement und die (speziell im Rohbau übliche) Vorinbetriebnahme im Sinne eines Factory-Acceptance-Tests.

Legt man für den Zeitanteil der Inbetriebnahme die Studie des VDW ([VDW97], auch [ZÄWÜ05], [LIN11]) zur „Abteilungsübergreifende[n] Projektierung komplexer Maschinen und Anlagen“ zugrunde, nimmt die Inbetriebnahme alleine ca. 15-25 % der gesamten Projektdauer ein. Bis zu 90% dieser Zeit werden für die Inbetriebnahme der Elektro- respektive Steuerungstechnik aufgebracht, die restliche Zeit für das Beheben mechanischer Probleme und mechanischer Feinjustage, insofern diese noch nicht während der Montage durchgeführt werden konnten. Laut Studie werden alleine ca. 70% der auf die Steuerungstechnik und Elektrik fallenden Inbetriebnahmezeit für das Beheben von Softwarefehlern benötigt. Dies bezieht sich auf fehlerhafte Programmierung an sich, falsche Parametrierung, falsche Anbindung von Sensoren/Aktoren und Ähnliches. Dies bedeutet, dass fast ein Sechstel der gesamten Projektzeit darauf verwendet wird, Softwarefehler, die in der vorhergehenden Planungsphase entstanden sind, zu beheben. Es sei an dieser Stelle angemerkt, dass die Studie nicht explizit die nach der Inbetriebnahme folgende Optimierungsphase und die vom Gewerk Automatisierungstechnik zur Sicherstellung des störungsfreien Betriebs und damit der Anlagenverfügbarkeit genutzte Produktionsbegleitungsphase betrachtet. Diese kann in der Praxis potenziell um mehrere Größenordnungen über der reinen Inbetriebnahmezeit liegen. Aus den Studien lässt sich folgern, dass zeitliche und monetäre Aufwände gesenkt werden können, wenn es gelingt, den Anteil an teuren Inbetriebnahmestunden wie auch die Verweilzeit des Gewerks Automatisierungstechnik an der Anlage insgesamt zu senken (Personalkosten im Feld sind in der Regel deutlich teurer als Bürostundensätze). Abbildung 2-8 zeigt eine Übersicht über die verschiedenen Zeitanteile einzelner Tätigkeiten am Engineering.

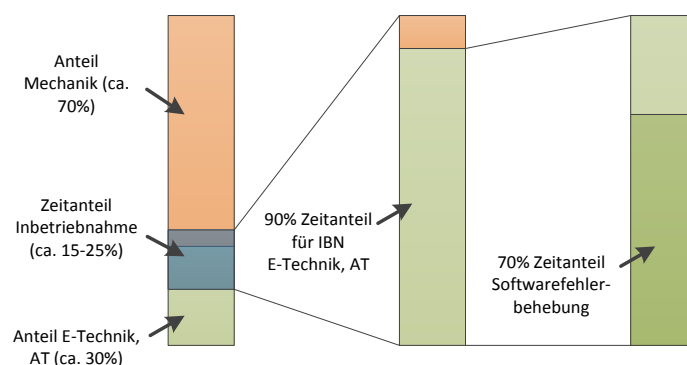


Abbildung 2-8: Aufwandsverteilung bei der Inbetriebnahme automatisierter Anlagen gemäß [ZÄWÜ05]

Der Inbetriebnahme einer Gesamtanlage und dem dort stattfindenden Test des Steuerungscode kommt daher eine maßgebliche Bedeutung während des Engineering-Prozesses und damit der Realisierung einer Anlage zu. In den folgenden Abschnitten erfolgt daher eine Erläuterung der einzelnen Vorgänge während einer Inbetriebnahme und, einhergehend mit dem großen Zeitaufwand, der für die Behebung von Softwarefehlern notwendig ist, eine Darstellung der – neben Planungsfehlern – dafür verantwortlichen Ursachen.

2.8 Die Inbetriebnahme von fertigungstechnischen Anlagen

Nach derzeit geltender europäischer Maschinenrichtlinie [EU 2006/42/EG] definiert sich der Begriff der Inbetriebnahme als „[...] die erstmalige bestimmungsgemäße Verwendung einer von dieser Richtlinie erfassten Maschine“. Diese sehr knappe Definition trifft bereits den Kern dessen, was eine Inbetriebnahme ausmacht: Aus einer Ansammlung mechanischer und elektrischer Objekte, die auf von Ingenieuren während des Engineering-Prozesses definierte Art und Weise miteinander interagieren sollen, entsteht durch die Inbetriebnahme erstmals eine Situation, in der die Projektergebnisse aller am Engineering beteiligten Gewerke in einer Anwendung miteinander vereint werden. Hierdurch kommt das charakteristische Verhalten einer Anlage erstmals zum Vorschein. Weitere Definitionen werden in der [VDI 5600-1] oder auch in [WÜN08] vorgenommen, und decken sich weitgehend mit der erwähnten Maschinenrichtlinie.

Darüber hinausgehend ist der Begriff der Inbetriebnahme in der Literatur häufig mehrdeutig und eher heterogen genutzt. Insbesondere die zeitliche Einordnung in den Engineering-Prozess sowie der Umfang der Tätigkeiten werden unterschiedlich bewertet. Zeugträger [ZEU98] beschreibt die Inbetriebnahme als „[f]unktionsgerechtes Einschalten der Anlage in Verbindung mit dem Prozess und Hochfahren der Leistung auf das geforderte Niveau“. Ziel der Inbetriebnahme ist „[...] die Überführung des fertig montierten Systems in einen betriebsfähigen Zustand gemäß den Vertragsbedingungen“ [ZEU98]. Kiefer [KIE07] sieht generell eine Phase des Produktionsanlaufs, die, nach der eigentlichen Fertigung und Montage des Produktionssystems, eine Brückenfunktion zwischen der Produktionsplanung und dem Produktionsbetrieb einnimmt. Die Inbetriebnahme ist ein Teil des Hochlaufprozesses. In diesem Zusammenhang weisen [ZEU98] und [KÖV10] darauf hin, dass insbesondere die Montage und Inbetriebnahme in der Praxis nicht genau gegeneinander abgegrenzt werden kann: „Während einige Bereiche noch montiert werden, können in anderen Anlagenteilen schon [...] Teilfunktionen überprüft und einzelne Systeme in [Tests] [...] voreingestellt werden“ [ZEU98]. Selbiges gilt sinngemäß für die nach der Inbetriebnahme folgende Anlauf- und auch Produktionsphase [KÖV10]. Oft werden die Phasen Inbetriebnahme und anschließender Anlauf daher auch zu einer sogenannten „Anlaufphase“ subsumiert.

Nachfolgend wird die Inbetriebnahme als die Summe aller Tätigkeiten verstanden, die notwendig sind, um ein System vom Zustand der „funktionsfähigen Anlage“ in einen Zustand der „produktionsfähigen Anlage“ zu überführen [NA 35]. Während der Inbetriebnahme ist daher die Funktionsfähigkeit der zu liefernden/umzusetzenden Prozesse entsprechend der Anforderungen des Auftraggebers nachzuweisen. Dies beinhaltet das Optimieren von Betriebsparametern, insofern diese nicht ausschließlich in der Produktionsphase ermittelbar sind. Der in diesem Zusammenhang genutzte Begriff des „Störungsfrei machen“ der Anlage im Sinne der Sicherstellung der zu erreichenden Anlagenverfügbarkeit (gemäß [VDI 3423]) wird als Aufgabe einer der Inbetriebnahme folgenden Anlauf- respektive Produktionsbegleitungsphase gesehen. Dies wird in Abschnitt 2.8.4 näher beschrieben. Der erfolgreiche Abschluss einer Inbetriebnahme lässt daher keine Aussage über die Produktivität einer Anlage zu (in der [NA 35] als „die bewertete Anlage“ beschrieben).

2.8.1 Typischer Ablauf einer Inbetriebnahme

Über das grundsätzliche Vorgehen und die Einordnung der Inbetriebnahme in den Engineering-Prozess wird in der Literatur bereits umfangreich berichtet, so z. B. von [KÖV10] und [KIE07]. Eine umfangreiche Betrachtung unter Einbeziehung der notwendigen expliziten Tätigkeiten des betei-

ligten Personals bietet zudem [ZEU98]. Er sieht als eine der wichtigsten Aufgaben der Inbetriebnahme, neben der systematischen Inbetriebsetzung der Automatisierungstechnischen Hardwarekomponenten und dem Einstellen und Optimieren von Betriebsparametern, das Beseitigen von Fehlern und Mängeln aus den Vorphasen an.

In der Praxis sind systematische Vorgehensweisen während der Inbetriebnahme aufgrund stark schwankender Rahmenbedingungen und nicht nach Plan verlaufendem Montagefortschritt, unvorhersehbaren Schwierigkeiten wie während der Inbetriebnahme neu aufkommender Kundenwünsche (meist handelt es sich um neue Kundenanforderungen, die erst während der Inbetriebnahme ersichtlich werden), erschwert. Die Inbetriebnahme kann in vielen Fällen nur stark fragmentiert vorgenommen werden, so dass an unterschiedlichsten Stellen der Anlage, je nach Montagefortschritt, gearbeitet werden muss, um den eng gesetzten Zeitplan einzuhalten [ZEU98]. Die Plan- und auch Berechenbarkeit einer Inbetriebnahme ist so stark eingeschränkt, die Durchführung konkreter Tätigkeiten erfolgt ad hoc, de facto agil. Erschwerend kommt hinzu, dass die während des Engineering-Prozesses entstandene Software keinen hohen Reifegrad besitzt. Die von [ZWH⁺06] durchgeführte Feldstudie zum Thema Inbetriebnahme ermittelte einen durchschnittlichen Wert von nur 37% korrekt umgesetzten Steuerungsfunktionen (ohne vorherige Simulation). Ergebnis eines fragmentarischen und von hoher Agilität geprägten Vorgehens, kombiniert mit einer unzureichenden Güte des Steuerungscode, ist in Summe die von Kiefer [KIE07] so genannte „Baustellenprogrammierung“. Sie spiegelt sich „[...] aufgrund des großen Zeitdrucks zumeist in suboptimalen Softwarelösungen sowie in unzureichenden Softwaredokumentationen wider, [und führt] letzten Endes zu verzögerten Produktionsanläufen bzw. zu kostspieligen Stillstandszeiten“.

Trotz der bereits beschriebenen, stark von äußeren Faktoren beeinflussten Vorgehensweise bei einer Inbetriebnahme, kann das Vorgehen in zwei grundlegende und sich wiederholende Muster unterteilt werden: Die Inbetriebnahme von Teilfunktionen einzelner Elemente einer Anlage (z. B. eines Förderbands oder auch eines Hebers) und die anschließende Inbetriebnahme von elementübergreifenden Systemfunktionen [ZEU98]. Eine klar definierte Abgrenzung kann nicht getroffen werden, sie unterscheidet sich von Projekt zu Projekt stark [Köv10]. In der Praxis spielen bei der Unterteilung in die Vorgehensmuster Inbetriebnahme von Teilfunktionen/Systemfunktionen die Betriebsarten einer Anlage eine wichtige Rolle. Betriebsarten bieten einen Anhaltspunkt über den jeweiligen Arbeitsstand und werden daher als Indikatoren für den Fortschritt einer Inbetriebnahme insgesamt genutzt. Betriebsarten können als „Einrichten“ (oftmals auch als „Tippbetrieb“ bezeichnet), „Hand/Manuell“ und „Automatik“ definiert werden, wobei es in Deutschland keine bindende normative Festlegung gibt. Bauherrenabhängig sind daher weitere Unterteilungen und auch andere Bezeichnungen möglich. Betriebsarten unterscheiden sich insbesondere hinsichtlich der manuellen Bedieneingriffe in den Prozess sowie hinsichtlich der aktiven Verriegelungen. Abbildung 2-9 zeigt die in den jeweiligen Betriebsarten notwendigen Interaktionen und aktiven Verriegelungen anhand eines einfachen Codebeispiels.

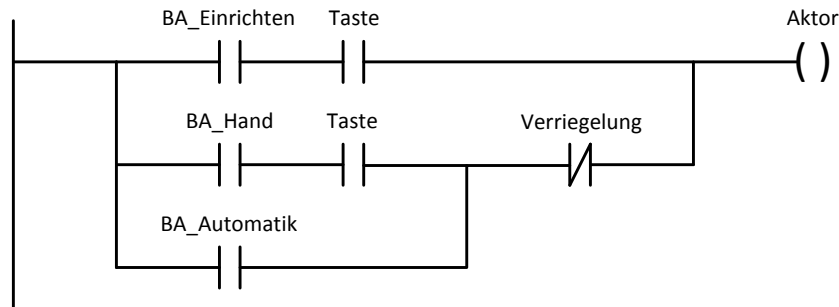


Abbildung 2-9: Stark vereinfachte Übersicht über notwendige Handeingaben und aktiver Verriegelungen über die beschriebenen Betriebsarten

Die Betriebsart *Einrichten* kann als manuelle (An-)Steuerung einzelner Stellgeräte charakterisiert werden, wobei mögliche Verriegelungen (auch jene gegen Beschädigung) nicht beachtet werden („fahren auf Anschlag“). Die Ansteuerung der Stellgeräte erfolgt in jedem Fall bereits über die SPS. Die Betriebsart *Einrichten* eignet sich für grundlegende Tests, ob Stellgeräte (wie z. B. Frequenzumrichter mit angeschlossenem Motor oder auch pneumatische/hydraulische Zylinder) korrekt angesteuert werden und das Zusammenspiel zwischen Aktorik (Wandler, Antriebselemente, etc.) und mechanischer Apparatur (Förderband, Verriegelungsmechanismus, etc.) bereits prinzipiell funktioniert und sich die mechanische Apparatur nicht selbst desintegriert. Diese Tests gelten den grundlegenden Ansteuerungsbausteinen für Aktoren (z. B. über Frequenzumrichter gesteuerte Motoren) und besitzen universellen Charakter, da die Ansteuerung für Geräte gleichen Typs üblicherweise nicht für jedes einzelne Gerät gesondert ausprogrammiert wird. Hierfür werden globale Bausteine genutzt, die anschließend multiinstanziiert werden. Um Beschädigungen an den Aktoren zu vermeiden, sind Geschwindigkeiten wie Motordrehzahlen im Vergleich zu anderen Betriebsarten wegen der nicht aktivierten Schutzvorrichtungen üblicherweise reduziert. Die Betriebsart *Einrichten* kommt als erster Schritt bei der Inbetriebnahme von Teilfunktionen in Betracht. Sie eignet sich nicht oder nur stark eingeschränkt zum Test anlagenübergreifender Funktionen.

Die Betriebsart „*Hand/Manuell*“ kommt üblicherweise als zweiter Schritt bei der Inbetriebnahme von Teilfunktionen in Betracht und ermöglicht das Testen von Verriegelungsmechanismen und somit Teilumfängen des später automatisch ablaufenden Prozesses. In der Betriebsart „*Hand/Manuell*“ werden Bewegungen ebenfalls durch rein manuelle Eingabe ausgelöst. Verriegelungen sind aktiv und die Aktoren bewegen sich üblicherweise mit erhöhter oder bereits voller Geschwindigkeit. Idealerweise sollten Verriegelungen respektive in der SPS programmierte Verriegelungslogiken bereits denjenigen entsprechen, die auch in der Betriebsart „*Automatik*“ verwendet werden. Letztendlich kann diese Betriebsart als manuell angesteuerter Automatikvorgang gesehen werden. Die Betriebsart versteht sich aber nicht als durch einmaliges Auslösen gestartete Semi-Automatik: Die Benutzerinteraktion ist während des kompletten Prozesses notwendig, was jederzeit ein Unterbrechen des Vorgangs ermöglicht. Sie eignet sich ferner zum eingeschränkten Test anlagenübergreifender Funktionen, insofern durch das Benutzerinterface (z. B. durch das Vorhandensein von Gruppentasten, die die Anwahl mehrerer Elemente erlauben) die Möglichkeit dafür geboten wird.

Tests in der Betriebsart „*Automatik*“ werden am Ende einer Inbetriebnahme durchgeführt. Die Prozesse einer Anlage laufen „*vollautomatisch*“ ab, d. h. eine Nutzerinteraktion, die für das Ablufen des Prozesses notwendig ist, wird nicht mehr benötigt. Eine Ausnahme bilden durch den Pro-

zess explizit vorgegebene, manuelle Eingaben (z. B. Drücken eines Freigabetasters durch das Betriebspersonal). Die eingestellten Geschwindigkeiten entsprechen den vom Gewerk Mechanik vorgegebenen Soll-Werten. Die Betriebsart Automatik eignet sich für das Testen von elementübergreifenden Systemfunktionen und, insofern Funktionen soweit vom Gesamtsystem gekapselt sind, dass ein Automatik-Test sinnvoll erscheint, auch für einzelne Teilfunktionen. Erweiterte Funktionen wie Stopp bei Taktende (auch Stopp bei Zyklusende), Leerfahren einer Anlage oder in der SPS implementierte anlagenspezifische Sortierverfahren und ähnliche, automatisch ablaufende Prozesse können nur in der Betriebsart „Automatik“ getestet werden.

Wird jeder prozesstechnische Zustand einer Anlage einzeln ausprogrammiert, wird der Steuerungscode schnell umfangreich und komplex. Dies kann die „Inbetriebnehmbarkeit“ und damit auch den erreichbaren Reifegrad des Steuerungscode stark einschränken. Eine große Herausforderung bei der Programmierung von IEC 61131-3 Steuerungscode ist es, mit möglichst wenig, auf SPS-Zykluszeit optimiertem, allgemeingültigem Code möglichst alle prozesstechnischen Situationen und Zustände einer Anlage abfangen zu können. Der erfolgreiche Abschluss von Automatik-Tests kann in diesem Kontext den Abschluss der Inbetriebnahme hin zur Abnahme einer Anlage implizieren. Er stellt aber nur ein Zwischenergebnis hin zu einer „bewerteten Anlage“ [NA 35] dar. Letztendlich entscheidend für den Reifegrad des SPS-Codes ist, neben der anforderungsgerechten Umsetzung der vom Kunden vorgegebenen Prozesse, die Anlagenverfügbarkeit und somit der störungsfreie Betrieb einer Anlage. Betriebsarten als Indikatoren für den Fortschritt einer Inbetriebnahme erreichen daher ihre Grenzen, sobald die Verfügbarkeit der Anlage eine Rolle spielt. Vielmehr stellt die Fähigkeit einer Anlage, im Automatikbetrieb zu funktionieren, nur den Übergang von der „funktionsfähigen Anlage“ in Richtung der „produktionsfähigen Anlage“ dar. Insbesondere bei größeren Projekten (insbesondere Greenfield Projekten), ist es nicht unüblich, dass zwischen Anlauf (also dem Beginn der Produktion an einer Anlage) und der finalen Abnahme (und damit dem Nachweis der Produktivität) mehrere Jahre liegen können, was nicht nur durch die zum Teil sehr flachen Anlaufkurven bedingt ist. Weitere Gründe sind umfangreiche Mängellisten und zusätzliche Kundenwünsche/-anforderungen, die sich erst im produktiven Einsatz der Anlage ergeben. *Punch Lists* mit mehreren hundert, manchmal tausenden Mängelanzeigen und Änderungswünschen, von denen letztere nicht alle als Änderungsanträge akzeptiert werden, sind keine Seltenheit. Erst durch gesammelte Betriebserfahrung, der Beobachtung der Anlage auch in zum Zeitpunkt der Planung noch nicht bekannten prozesstechnischen Situationen und Zuständen, der damit verbundenen Sicherstellung der vertraglich fixierten Anlagenverfügbarkeit sowie der Beachtung aller durch den Bauherren geforderten Änderungswünsche und deren (nachträgliche) Implementierung und Verifizierung, kann ein Reifegrad des SPS-Codes erreicht werden, der den Übergang einer „produktionsfähigen Anlage“ hin zur „bewerteten Anlage“ und damit die Abnahme einer Anlage ermöglicht.

2.8.2 Testscenarien und Abnahmekriterien

Im Kontext der im vorherigen Abschnitt genannten Rahmenbedingungen und dem typischen Ablauf bei der Errichtung respektive der Inbetriebnahme automatisierter Anlagen fällt es schwer, systematisch durchgeführte Testscenarien/Prüfpläne und konkrete Abnahmekriterien zu identifizieren, die insbesondere für den Steuerungstest fertigungstechnischer Anlagen allgemeingültig sind. Generell sollten *Verhaltenskenngrößen* (→ Abschnitt 2.4) und damit sinngemäß die messbaren Anforderungsspezifikationen einer Anlage eine große Rolle bei der Testplanung und der Testdurchführung während einer Inbetriebnahme spielen. Die konkrete Testplanung findet, insbeson-

dere unter Beachtung der während der Inbetriebnahme im Vordergrund stehenden Validierung des erstellten Steuerungscode, wenn überhaupt, händisch statt. Sie zeigt eine geplante Testreihenfolge, die in den zur Verfügung stehenden „Inbetriebnahme-Zeitslot“ eingepasst wird (vielfach pro forma). In der Literatur konzentriert sich die Betrachtung von Tests bei der Überprüfung des Steuerungscode eher auf die Unterstützung händisch zu definierender Testfälle, ohne im konkreten auf durchzuführende Testszenarien und die Testplanung an sich einzugehen (u. a. [BAR11], [WÜN08], [Köv10]). Aus Anwendungen für die formale Verifizierung von Steuerungscode sind Methoden für die automatisierte Testfallgenerierung bekannt [KWV10], welche im Kontext des eigentlichen Testens an der Maschine/Anlage aber nicht oder nur sehr eingeschränkt nutzbar sind.

Für das Testen wie auch für die Abnahme einer automatisierten, verfahrenstechnischen Anlage haben sich die Begriffe *Factory-Acceptance-Test* (FAT) sowie *Site-Acceptance-Test* (SAT) etabliert [NA 35]. Der FAT liegt zeitlich vor dem SAT und soll die Überprüfung von ausgewählten Teilaspekten einer Anlage, seltener des Gesamtsystems, möglichst noch beim Hersteller/Lieferanten ermöglichen. Der SAT hingegen charakterisiert die Abnahme der kompletten Anlage direkt beim Kunden. Beide Begriffe werden im fertigungstechnischen Anlagenbau nur vereinzelt genutzt, generell werden derartige Tests als Vorabnahme oder Endabnahme beschrieben. Auch wenn der Begriff Vorabnahme eine gewisse Endgültigkeit und Festlegung bei der Abnahme von (Teil-) Projektergebnissen impliziert, ist in der Praxis nur selten die verbindliche Festlegung auf ein erarbeitetes Projektergebnis durch den Bauherren/Auftraggeber anzunehmen. In diesem Fall stellt die Vorabnahme lediglich einen (meist unverbindlichen) Zwischenschritt dar, der mehr der Sicherheit bei der Abstimmung von Projektergebnissen dient als der konkreten Abnahme und damit Festlegung auf eine bestimmte Lösung.

Zur Systematisierung von Testszenarien werden vor allem für die Verwendung bei der Inbetriebnahme technischer Anlagen Checklisten zur Unterstützung vorgeschlagen (u. a. von [KOH93], [AMA94], [WEB02], [WÜN08]). Diese haben mehrere Nachteile:

- Werden Checklisten (sinngemäß in Form von Best-Practices) verwendet, die projektübergreifend zum Einsatz kommen, haben diese nicht immer den notwendigen Detaillierungsgrad und können somit das für jede Anlage charakteristische und einmalige Verhalten nicht ausreichend abdecken. Sie sind in der Regel so allgemein gehalten, dass ihre Aussagekraft leidet und sind daher nur für Routineaufgaben beim Steuerungstest nutzbar (E/A Checks, Checks auf korrekte Kennzeichnung von Operanden, etc.).
- Werden Checklisten eigens für eine bevorstehende Inbetriebnahme vorbereitet, bedeutet dies in vielen Fällen einen sehr hohen Zeitaufwand. Kombiniert mit dem begrenzten Wissen der mit der Umsetzung betrauten Mitarbeiter des Gewerks Automatisierungstechnik werden Checklisten daher entweder gar nicht oder nur unzureichend umgesetzt. Ihr Nutzen für die Inbetriebnahme ist stark eingeschränkt.

Detailliert ausgearbeitete Prüfpläne in Form von Checklisten bieten zwar scheinbare Planungssicherheit und sollen die Berechenbarkeit im Sinne der zeitlichen Planbarkeit einer Inbetriebnahme ermöglichen und unterstützen. Ihr Einsatz ist aber vom Fortschritt der Montage der Anlage *vor Ort* abhängig. Ergeben sich im Verlauf der Montage Verzögerungen, geht dies direkt zulasten der eigentlichen Inbetriebnahme und somit auch der Umsetzung geplanter Testszenarien/Prüfpläne. Konkrete Testszenarien und Testabläufe folgen daher dem bereits beschriebenen grundlegenden Schema *Basistests einzelner Teilsysteme in der Betriebsart Einrichten* → *erweiterte, teils gesamt-*

systemübergreifende Tests in der Betriebsart Hand/Manuell → abschließende gesamtsystemübergreifende Tests in der Betriebsart Automatik, ohne die konkreten Tätigkeiten inhaltlich zu definieren. Von Bedeutung ist der bereits mehrfach erwähnte zielgerichtete Test gegen die während des Engineerings definierten *Verhaltenskenngrößen*, und damit sinngemäß gegen die Anforderungsspezifikationen an die zu automatisierende Anlage. Daneben spielt naturgemäß auch die *Erfahrung* des an der Inbetriebnahme beteiligten Personals des Gewerks Automatisierungstechnik eine große Rolle. Tabelle 1 gibt einen exemplarischen Überblick über in der Praxis während der Inbetriebnahme durchgeführte Tests, abhängig von der Betriebsart.

TestszENARIO:	BA Einrichten	BA Hand/Manuell	BA Automatik
E/A Check	✓	(✓)**	(✓)**
Test von Feldgeräten wie FRQ-Umrichter, ...	✓	✓	(✓)**
Handeingaben (HMI, Knöpfe, Schalter, ...)	✓	✓	(✓)*
Anzeigeelemente (HMI, Lampen, Ampeln, ...)	✓	✓	✓
Sicherstellung Fehlererkennung	(✓)*	✓	✓
Prozesstechnische Verriegelungen	-	✓	✓
Sicherstellung Maschinensicherheit	(✓)*	✓	✓
Sicherstellung Personensicherheit	✓	✓	✓
Anbindung an Leitsystem	-	(✓)*	✓
Verhaltenkenngrößen***:			
Max. Geschwindigkeiten	-	(✓)*	✓
Positioniergenauigkeit	-	(✓)*	✓
Taktzeiten	-	-	✓

* Nur wenn in jeweiliger BA relevant ** Nur bei Änderungen während IBN

*** Auszugsweise

Tabelle 1: Typische Testaspekte während der Inbetriebnahme von Anlagen (Beispiele)

Tests des Steuerungscode von automatisierten Anlagen sind in diesem Kontext nahezu immer als *White-Box Tests* zu klassifizieren. Das Inbetriebnahmepersonal *muss* die innere Funktion eines eingesetzten Bausteins, der für die Steuerung technischer Einrichtungen genutzt wird, nicht nur kennen und verstehen, sondern es muss auch dazu in der Lage sein, den Ablauf innerhalb eines Bausteines beobachten zu können. Nur dadurch ergibt sich die Möglichkeit, das im Feld resultierende Verhalten dahin gehend auf Prozesskonformität hin interpretieren zu können, so dass anschließend, wenn der Test keine Übereinstimmung des Prozesses mit dem geforderten Soll und damit den spezifischen Kundenanforderungen ergibt, auf geeignete Weise und vor allem an der richtigen Stelle korrigierend im Steuerungscode eingegriffen werden kann. Ausdrücklich gilt dies auch für die in Werksstandards zur Steuerung technischer Einrichtungen zur Verfügung gestellten Standardbausteine, insofern die mechanische Apparatur und die Funktion des jeweiligen Standardbausteins nicht synchron sind.

In Hinblick auf mögliche Abnahmekriterien und damit den Abschluss eines Engineering-Projekts ist es gängige Praxis, diese nicht eindeutig und verbindlich bereits im Lastenheft zu regeln. Für die Abnahme von Anlagen wird dort vielfach pauschal auf Richtlinien wie die [VDI 3977] („Empfehlung für die Abnahme von automatischen Lagersystemen“), [VDI 3979] („Abnahmeregelungen für Stückgut-Fördersysteme“) sowie [VDI 3581] („Verfügbarkeit von Transport- und Lageranlagen sowie deren Teilsysteme und Elemente“) verwiesen, ohne auf die konkret für die jeweilige Anlage

geltenden spezifischen funktionalen wie nichtfunktionalen Anforderungen einzugehen. Dies mag auch daran liegen, dass diese erst im Laufe des Engineering-Prozesses von den beteiligten Gewerker definiert werden müssen. Richtlinien können das Ausarbeiten konkreter Abnahmekriterien aber nicht ersetzen, sondern nur einen Handlungsrahmen bieten, der die konkrete Ausführung einer spezifischen durch die Richtlinie abgedeckten Tätigkeit unterstützen und in eine möglichst allgemein akzeptierte Richtung lenken kann. Es gehört zur derzeitigen Praxis, Abnahmekriterien so weit wie möglich im Unklaren zu lassen und die Abnahme durch immer neue Anforderungen hinaus zu zögern. Auftragnehmer werden dadurch so lange wie möglich auf der Baustelle gehalten.

2.8.3 Klassifizierung von typischen Störungen und deren Auswirkungen

Grundsätzliche lassen sich in der Praxis zwei Arten von Störungen identifizieren: *Technische* und *organisatorische Störungen*. Organisatorische Störungen sind in diesem Kontext auf Logistik- und/oder Materialflussfehler zurückzuführen, z. B. mangelhafte Teileversorgung oder ein genereller Abriss des Materialflusses innerhalb einer Anlage (z. B. nach manuell ausgelöstem Montagestopp durch Werker). Organisatorische Störungen können sich aus dem Gut-Prozessablauf heraus ergeben und müssen entsprechend durch die Steuerung erkannt werden. Als Reaktion werden in der Steuerung geeignete Maßnahmen gebildet. Beispiele hierfür können das Unterbrechen des Materialflusses an bestimmten Stellen einer Anlage sein („Aufpuffern“ des Materialstroms), oder auch akustisch-visuelle Warnsignale. Die Erkennung organisatorischer Störungen kann während der realen Inbetriebnahme nicht oder nur eingeschränkt getestet werden, da die Anlage zu diesem Zeitpunkt in der Regel noch nicht vollständig mit Material vollgefahren ist und naturgemäß noch keine Produktion läuft. Tests hinsichtlich der Erkennung und dem Handling von organisatorischen Störungen werden daher während des Produktionsanlaufs oder unter laufender Produktion durchgeführt.

Technische Störungen werden in vielen Fällen in *abschaltende* und *nicht-abschaltende Störungen* unterteilt. Diese Unterteilung bedeutet, dass im Falle einer abschaltenden technischen Störung der automatische Betrieb einer Anlage oder eines Anlagenteils in jedem Fall unterbrochen wird und durch manuellen Eingriff neu initialisiert werden muss. Dies ist immer dann der Fall, wenn erkannt werden kann, dass der Gut-Prozessablauf nicht innerhalb der spezifizierten Parameter läuft und potenziell Beschädigungen an der Anlage oder dem Produkt auftreten können. Nicht-abschaltende Störungen unterbrechen den automatischen Ablauf einer Anlage hingegen nicht. Beispiele hierfür sind Sicherheitsfälle an Anlagenteilen wie HMI oder anderen, am automatisch ablaufenden Prozess nicht unmittelbar Beteiligten. Abschaltende und nicht-abschaltende Störungen treten häufig gemeinsam auf. So kann bei Fall einer Hauptsicherung (abschaltend) die Überwachung anderer Sicherungen (nicht-abschaltend) ansprechen. Die Konsequenz ist das Auflaufen multipler Störmeldungen. Es ist in der Praxis die Aufgabe der Instandhaltung, die genaue Fehlerursache einzugrenzen und die Störungen zu beheben. Steuerungstechnische Maßnahmen in Form von Störmeldungen stellen daher nur einen Hinweis dar, welche Ursache einer Störung zugrunde liegen könnte. Ein wie von [CHR15] vorgeschlagenes wissensbasiertes Konzept, dass es ermöglicht, auflaufende Störmeldungen einzugrenzen, ist derzeit in der Praxis nicht üblich. Abschaltende technische Störungen beschreiben in diesem Zusammenhang immer mechanische oder elektrische Störungen, die mit der Beschädigung einer Anlage verbunden sein können.

Eine Klassifizierung von in einer Anlage potenziell auftretenden *technischen* Störungen wird beispielhaft von Wünsch [WÜN08] gezeigt. Wünsch konzentriert sich vornehmlich auf das Fehlverhalten von Maschinenkomponenten. Diese Störungen werden steuerungstechnisch abgefangen, um einerseits Beschädigungen an der Anlage zu vermeiden, andererseits die Prozessstabilität zu garantieren. Tabelle 2 zeigt diese möglichen Störungen.

Mögliche Störungen	Steuerungstechnische Maßnahmen
Schwergängigkeit Mechanik	Laufzeitüberwachung
Sensordefekt	
Kabelbruch	
Druckabfall	
Aktordefekt	Endlagenüberwachung
Sensordefekt	
Kabelbruch	
Mechanisches Problem	Paarfehlerüberwachung
Sensor defekt	
Sensor fehlerhaft ausgelöst	Druckschalterüberwachung
Medium fehlt	
Sensor defekt	Diskrepanzfehler
Sensor defekt (mehrere zusammenwirkende)	

Tabelle 2: Mögliche Störungen in automatisierten Anlagen in Anlehnung an [WÜN08]

2.8.4 Nur eingeschränkt testbare Szenarien

Nicht alle auszuführenden Prozesse lassen sich während der Inbetriebnahme gleichermaßen überprüfen. Der Soll-Prozessablauf wird zwar umfangreich getestet (nicht unüblich auch noch zur bereits gestarteten Produktion), was aber nicht zwangsläufig auch für spezielle oder wenig genutzte Prozesse, die immer dann zum Einsatz kommen, wenn es organisatorische Maßnahmen oder technische Störungen in der Produktion erforderlich machen, gilt.

Hierfür werden *Notfallstrategien* genannte Prozeduren entworfen, die es ermöglichen sollen, den Produktionsbetrieb (wenn auch eingeschränkt) aufrecht zu erhalten. Aus Kosten- und Zeitgründen werden Tests von Notfallstrategien nicht immer im eigentlich angebrachten Umfang vorgenommen: Einerseits kann das (absichtliche) herbeiführen entsprechender Stöorzustände zu aufwendig sein (ein Hochregallager mit tausenden Stellplätzen kann nicht testweise vollgefahren werden). Dies könnte den Gut-Prozessablauf so weit stören, dass ein Hochfahren auf normale Produktionskapazität zu aufwändig und zeitintensiv ist. Andererseits gibt es auch generell nicht testbare Fehlersituationen, in denen z. B. das Anhalten eines Förderbands mit einer wesentlichen Beschädigung einhergehen kann: So ist es denkbar, bei hoher zu transportierender Last und einer testweise herbeigeführten Notabschaltung Teile der Anlage irreparabel zu beschädigen. Dies wird nur für einen Notfall in Kauf genommen. Der Test beschränkt sich in diesen Fällen auf die reine elektrotechnische Abschaltung der Leistung am (von der Mechanik entkoppelten) Motor des Förderbands, die mittels Messung nachgewiesen wird. Ein anderes Beispiel stellen Scheibenklebeanlagen (insb. im Automobilbau, [DÜRR®]) dar, deren Abschaltung ein Aushärten des Epoxidharzes oder des jeweilig verwendeten Klebers bedingen kann und eine generell lange, im Falle eines Tests ungewollte, Produktionsunterbrechung zur Folge hat.

Stochastisch verteilt auftretende Fehler, deren Auftreten sich auch nachträglich nicht auf Basis eines deterministischen Verhaltens erklären lassen, sind prinzipiell nicht vorab im Sinne der Testbarkeit planbar. Derartige Fehlerzustände lassen sich oft auf unerkannte und unbeabsichtigt herbeigeführte *Deadlocks* oder *Livelocks* innerhalb des Programmcodes zurückführen. Obwohl mit Petri-Netzen die Abbildung derartige Situationen in einem Modell vorstellbar ist und Petri-Netze maßgeblich dazu genutzt werden können, derartige Zustände zu erkennen (für eine genauere Beschreibung von Petri-Netzen und deren mögliche Anwendungsfelder in der Automatisierungstechnik sei auf [SCH99A] verwiesen), wird dies in der Praxis so gut wie nie eingesetzt. Dies ist begründet in der Komplexität und auch Kompliziertheit der auszuführenden Prozesse, ihrer Umsetzung im Steuerungscode, der generellen nicht praktikablen Übertragbarkeit der zu nutzenden, grafischen [IEC 61131-3] Programmiersprachen (KOP/FUP) in ein Stellen/Transitionen-Netz und nicht zuletzt mit dem relativ hohen Anspruch an den Ausbildungshintergrund des die Analyse ausführenden Personals. Der derzeit hohe Implementierungs- wie Experimentieraufwand rechtfertigt nicht eine anlagenweite Untersuchung und die Suche nach derartigen Zuständen. Es ist Aufgabe des Gewerks Automatisierungstechnik, diese möglichen Fehlerzustände durch geeignete und systematische Umsetzung des Programmcodes, unter Einbeziehung des jeweiligen Erfahrungsschatzes aller beteiligten Programmierer, von vornherein so weit wie möglich auszuschließen.

Generell lässt sich folgern, dass Abweichungen vom Sollablauf des Prozesses durch geeignete Programmierung prozesstechnisch sauber auszuschließen und damit die Prozessstabilität sicherzustellen eine der wesentlichen Herausforderung beim Erstellen von Steuerungscode darstellt. In der Regel wird die Stabilität des Programmcodes nicht durch singuläre und zeitlich befristete Tests zur Sicherstellung der Produktionsfähigkeit einer Anlage während einer Inbetriebnahme nachgewiesen, sondern durch die Langzeitbeobachtung der Anlage im Rahmen von Verfügbarkeitstest, die bei Erfolg den Übergang zur bewerteten Anlage und damit zur Abnahme ebnen – wenngleich für Letzteres in der Praxis derzeit keine genauen und allgemeingültigen Kriterien definiert werden können.

2.9 Zusammenfassung der Schwachstellen und Defizite des Engineering-Prozesses

Der Engineering-Prozess zeichnet, betrachtet man ihn gesamtheitlich unter Einbeziehung von Teilaspekten wie Projektorganisation und Projektmanagement, ein ambivalentes Bild. Zwar steht mit der *Digitalen Fabrik* und insbesondere dem *Digitalen Fabrikbetrieb* ein umfangreiches Rahmenwerk für die Durchführung eines *Engineering-Prozesses* zur Verfügung; insbesondere Anlagenbetreiber/Original Equipment Manufacturer (OEM) betreiben individuell hohe Aufwände, das Potenzial der *Digitalen Fabrik* in der Praxis nutzbar zu machen. Letztendlich lebt der Ansatz der *Digitalen Fabrik* aber nicht von der alleinigen bauherrenspezifischen Implementierung desselben, sondern vom Miteinander der einzelnen an einem Engineering-Projekt beteiligten Gewerke. Entscheidend bei der konkreten *Ausführung* sind die an der Planung, Projektierung und Errichtung einer automatisierten Anlage beteiligten Gewerke, meist in Form (hoch-) spezialisierter KMU. Die vorherrschende Heterogenität bei der Umsetzung einzelner Konzepte und Sichtweisen der *Digitalen Fabrik* bedingt ein hohes Maß an Anpassbarkeit und Flexibilität der beteiligten Gewerke.

Die Doktrin der *Digitalen Fabrik* zeigt gerade hier einen systemimmanenten Mangel: Es sind keine verbindlichen, bauherrenübergreifenden Standards verfügbar, stattdessen viele einzelne, teils sogar nur fachabteilungsspezifische Festlegungen und Vorschriften. Nur vereinzelt finden sich

Bestrebungen, einzelne Teileaspekte der Digitalen Fabrik in Fachverbänden bauherrenübergreifend zu standardisieren (→ Abschnitt 2.6). Dies legt die Vermutung nahe, dass *die Digitale Fabrik*, die als eine Zusammenfassung allgemeingültiger Vorschriften, Standards, Vorgehensweisen und Richtlinien verstanden werden soll, bislang nicht realisiert wird. KMU müssen, um am Markt zu bestehen, das Management der vielen unterschiedlichen zu beachtenden Standards, und bauherrenspezifischer Gegebenheiten, leisten. Aufgrund der nur begrenzt zur Verfügung stehenden monetären sowie personellen Mittel werden allein dafür bereits große Ressourcen gebunden. Letztendlich obliegt es so der Verantwortung der am Engineering-Prozess beteiligten Gewerke, die zur Verfügung stehenden Rahmenbedingungen so zu nutzen, dass sich positive Nutzeffekte auf den Engineering-Prozess an sich ergeben. Für KMU kann daher geschlussfolgert werden: Die *Digitale Fabrik kann* einen geeigneten Rahmen für das Engineering schaffen, sollte aber nicht *zwingende Voraussetzung* dafür sein.

Ergänzend zu den vorherrschenden globalen Rahmenbedingungen für den Engineering-Prozess erfolgt die Projektabwicklung an sich derzeit nicht immer optimal. In frühen Projektphasen getroffene Festlegungen und Anforderungen von Auftraggebern/Bauherren, in Form einer Ausschreibung respektive eines Lastenhefts [VDI 3694] festgelegt, sind als zum Teil nicht bekannt oder als nicht stabil einzuordnen (→ Abschnitt 2.4). Ein gesamtheitlicher, über alle mechatronischen Aspekte hinweg stringenter und durchgehender Lösungsansatz scheitert in der Praxis allzu oft an der nur unzureichenden Synchronisation aller mechatronischen Aspekte über Gewerkegrenzen hinweg (→ Abschnitt 2.2). Insbesondere betrifft dies die zu liefernden Leistungen des Gewerks Automatisierungstechnik und damit den zu liefernden Steuerungscode, dessen spezifische Charakteristika in frühen Projektphasen nicht ausreichend gewürdigt werden. Dies erschwert die Programmerstellung insgesamt. Während mechanische Apparaturen generell nur mit hohem Aufwand geändert werden können und so die Aufmerksamkeit der Planer des Bauherren genießen, lässt sich Steuerungscode immer auch nachträglich anpassen – für die hier anfallenden, zum Teil sehr hohen Aufwände wird nicht in gleichem Maße Verständnis aufgebracht.

Es ist in diesem Kontext nur selten der Mangel an spezifischem Fachwissen, das zur (technischen) Umsetzung geeigneter Lösungen erforderlich ist, der Engineering-Projekte erschwert oder scheitern lässt [ENHO07]. Das Wissen um eine bauherrengerechte Umsetzung ist in den Köpfen der beteiligten, in der Regel sehr erfahrenen, Projektleiter des Gewerks Automatisierungstechnik durchaus vorhanden – die Möglichkeit einer (potenziell auch effizienten) Umsetzung ist daher gegeben. Nicht in gleicher Weise gegeben ist die Bereitschaft, dieses Wissen mit den Projektmitarbeitern und damit innerhalb des Projektteams auch zu teilen. Dieses *Herrschaftswissen* genannte Phänomen bezeichnet einen Wissensvorsprung, der hauptsächlich zur Sicherung der eigenen Position dient. Projektmitarbeiter werden so oftmals zu Handlangern degradiert, die dem Projektleiter (in KMU oftmals in Personalunion auch Chefprogrammierer) nur zuzuarbeiten haben und oftmals nur triviale Arbeit zugewiesen bekommen, ohne die Gesamtzusammenhänge der Abläufe einer Anlage verstehen zu können. Bei der Planung und Programmierung automatisierter Anlagen, die generell als komplexe Systeme gemäß der Definition in [HWF*12] bezeichnet werden können und sich dadurch charakterisieren, dass einzelne Systemelemente in starker Interaktion zueinander stehen und voneinander abhängig sind, ist aber gerade das *Systemwissen* ein entscheidender Faktor [LÜPE00]. Werden Projektmitarbeitern in diesem Kontext durch den Projektleiter getroffene Absprachen mit dem Bauherren durch Organisationsmängel oder Unfähigkeit (in der Praxis zu oft beides) vorenthalten, und werden Mitarbeiter an verschiedenen Stellen der Anlage eingesetzt, ohne dass ihnen die Auswirkungen ihres Schaffens auf andere Anlagenteile be-

wusst sind, geht dies naturgemäß zulasten des Reifegrads des jeweilig erstellten Steuerungs-codes-Fragments. Es obliegt dem Projektleiter, diese einzelnen Fragmente zu synchronisieren und in Einklang zu bringen. Dies kann aus Zeitgründen nicht immer bereits vor der realen Inbetriebnahme durchgeführt werden. Werden die Ergebnisse des Gewerks Automatisierungstechnik erst auf der Baustelle validiert, können Verzögerungen bei der Inbetriebnahme und damit auch erhöhte Kosten und im schlimmsten Fall Vertragsstrafen die Folge sein.

Eine entscheidende Phase im Engineering-Prozess ist daher die eigentliche Inbetriebnahme beim Kunden vor Ort. Hier zeigen sich alle Mängel, seien diese inhaltlicher oder organisatorischer Art, aus den zeitlich davor liegenden Projektphasen, das erste Mal in der Realität. Es ist hier aber bereits zu spät für ein systematisches Reagieren auf Planungsunschärfen und -mängel, so dass nur noch (zum Teil unkoordiniert) agiert werden kann. Insbesondere den Teilumfängen des Gewerks Automatisierungstechnik kommt eine maßgebliche Bedeutung zu: Werden über 60% der eigentlichen Inbetriebnahmezeit nur für die Behebung der durch das Gewerk Automatisierungstechnik verursachten (Programmier-)Fehler sowie prozesstechnisch falsch implementierter Steuerungsalgorithmen verwendet, liegt ein maßgeblicher Hebel für die Effizienzsteigerung bei der Überprüfung und somit der Sicherstellung eines möglichst hohen Reifegrads des programmierten Steuerungs-codes noch vor der eigentlichen Inbetriebnahme vor Ort. Gelingt es daher, den Reifegrad des Steuerungs-codes durch Simulation (im Speziellen mittels einer *virtuellen Inbetriebnahme*) zu steigern, sind signifikante Einsparungen in der Inbetriebnahme und damit im Engineering-Prozess möglich (u. a. [Osm98], [ZWH*06], [KÜH06], [BWG09], [VDI 4499-2], [VDI 3693-1]). Selbst wenn die Vision eines zu 100% vorab getestet Steuerungs-codes aufgrund der systemimmanenten Mängel des Engineering-Prozesses (insbesondere wenn im Rahmen der *Digitalen Fabrik* durchgeführt) nicht realistisch erreicht werden kann, sollte der Steuerungscode dennoch einen Reifegrad aufweisen, der zumindest die schnelle Anpassung an neue Gegebenheiten auf der Baustelle ermöglicht oder vereinfacht. Gegebenheiten können in diesem Kontext als neue Anforderungen, Änderungen an der Mechanik im Laufe der Montage und andere verstanden werden. Es sollte daher ein Reifegrad des Steuerungs-codes angestrebt werden, der sicherstellt, dass in jedem Fall alle in der Planungsphase bekannten Funktionen bereits vollständig implementiert sind und, soweit möglich, mit den bis zum Start der realen Inbetriebnahme erstellten und damit möglichst aktuellen Planungsdaten der Gewerke Mechanik und Elektrotechnik synchronisiert sind.

Das Potenzial und vor allem die Frage wie *Simulation* und insbesondere eine *virtuelle Inbetriebnahme* den Engineering-Prozess zielgerichtet unterstützen kann, wird in Kapitel 3 näher betrachtet.

3 Simulation als Methode zur Unterstützung des Engineering-Prozesses

Simulation als den Engineering-Prozess unterstützende Methode kann viele Ausprägungen haben. Insbesondere in der *Digitalen Fabrik* ist Simulation als eine Kerntechnologie zu sehen, die alle Bereiche der *Digitalen Fabrik* von der Kapazitäts- und Montageplanung bis hin zum Test der während eines Engineering-Projekts erstellten Steuerungscode umfassen kann [KÜH06]. Die folgenden Abschnitte geben einen generellen Überblick über die Simulationsmöglichkeiten zur Unterstützung des Engineerings. Insbesondere auf die Methode der *virtuellen Inbetriebnahme* zur frühzeitigen Absicherung des erstellten Steuerungscode wird näher eingegangen.

3.1 Simulationsbasiertes Testen im Engineering fertigungstechnischer Anlagen

Die Sicherstellung einer optimierten, d. h. mit wenig Aufwand verbundenen Validierung von Projektergebnissen, die nicht zu Lasten der angepeilten Produktionsfähigkeit einer neu zu errichtenden Anlage geht, bedingt die Entkopplung der Validierung von der realen Anlage [MAN09]. Simulationsbasiertes Testen stellt in diesem Kontext während des Engineerings fertigungstechnischer Anlagen ein Hilfsmittel „[...] bei der Planung, Realisierung und [dem] Betrieb von technischen Systemen [...]“ dar [VDI 3633-8]. Insbesondere in der *Digitalen Fabrik* schreibt Kühn ([KÜH06]) der Simulation den Status einer „Kerntechnologie“ zu: „Die Simulation von Produktionseinrichtungen und Prozessen kann die Projektierung, Auslegung, Planung und Programmierung von Produktionseinrichtungen wirkungsvoll unterstützen“. Simulation an sich kann allgemein als „[...] die Methode, beliebige reale Vorgänge unter Zuhilfenahme von Computern nachzuahmen“ charakterisiert werden [KRNE98]. Abhängig von der betrachteten Unternehmensebene, dem Einsatzzeitpunkt im Lebenszyklus einer Anlage und der beteiligten Gewerke und Fachabteilungen, die Simulation als unterstützende Methode nutzen, kann in der *Digitalen Fabrik* zwischen verschiedenen Arten von Simulationen differenziert werden, deren Ziele sich beträchtlich voneinander unterscheiden können [KÜH06]. Einzelne Arten von Simulation können sein (Auszug):

- Eine *Ablaufsimulation* zur Absicherung des geplanten Materialflusses oder anderer Untersuchungen mit dem Schwerpunkt Logistik (hauptsächlich, aber nicht nur, in frühen Phasen des Engineerings [KUD00], [KÜH06], [CLHE08]).
- Simulation während der *Planung und Konstruktion von mechanischen Apparaturen* unter Betrachtung der Maschinendynamik zur Absicherung der Projektergebnisse des Gewerks Mechanik (mittels einer Mehrkörpersimulation oder mit Hilfe von Finite-Element-Methoden [WSR92], [CHR03]).
- Simulation mittels einer *virtuellen Inbetriebnahme*, die eine Absicherung des vom Gewerk Automatisierungstechnik erstellten Steuerungscode für das Automatisierungssystem (Robotersteuerungen, SPS) noch vor der eigentlichen Inbetriebnahme vor Ort zum Ziel hat ([OSM98], [SPLA00], [ZÄWÜ05], [SCH07]).

Die hier auszugsweise beschriebenen Simulationsmöglichkeiten werden in dieser Arbeit auf diejenige Simulationsart beschränkt, die das Testen von Projektergebnissen des Gewerks Automatisierungstechnik im Fokus hat: der *virtuellen Inbetriebnahme*. Eine Simulation der Teilefertigung im Sinne der in [DIN 8580] genannten Fertigungsverfahren sowie von nicht-technischen Aspekten wie der Personalsimulation ([BCW03], [HEH11]) wird für die folgenden Betrachtungen ausgeklammert - sie sind nicht Gegenstand der weiteren Untersuchungen. Abbildung 3-1 zeigt im Kontext des *Digitalen Fabrikbetriebs* den Einsatz von virtueller Inbetriebnahme. Virtuelle Inbetriebnahme kann sowohl Phase als auch Methode sein: Als Methode wird virtuelle Inbetriebnahme zielgerichtet in der gleichnamigen Phase eingesetzt. Virtuelle Inbetriebnahme kann aber auch als ergänzende Methode in anderen Phasen des Engineerings zum Einsatz kommen. Ein Beispiel hierfür ist die virtuelle Inbetriebnahme zur Absicherung von vorzunehmenden Umbauarbeiten während der Serienproduktion. Dieser spezielle Anwendungsfall wird in dieser Arbeit nicht betrachtet. Der Fokus liegt auf der Anwendung von virtueller Inbetriebnahme in der gleichnamigen Phase des Engineering-Prozesses.

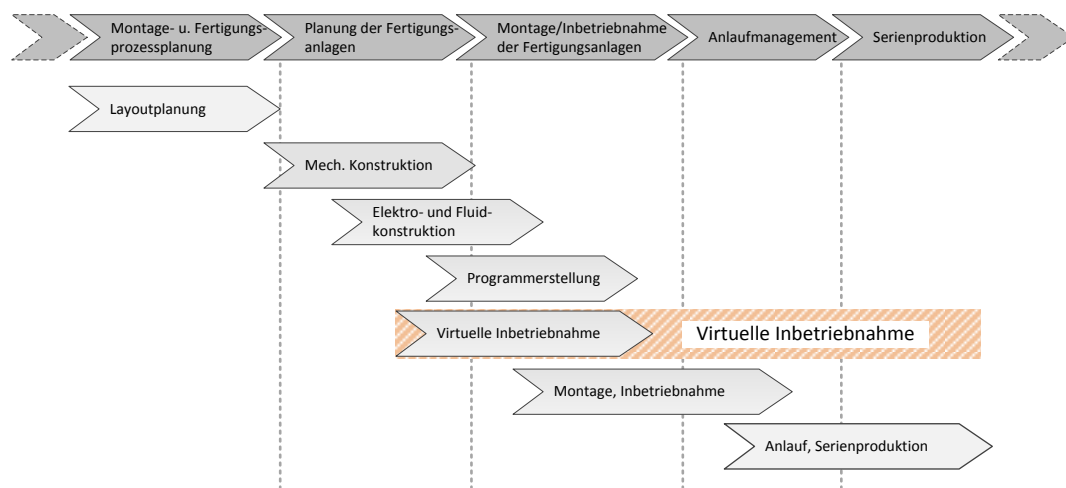


Abbildung 3-1: Virtuelle Inbetriebnahme und ihr Einsatzzeitpunkt während des Digitalen Fabrikbetriebs (schraffiert).

Allen Simulationsarten ist gemein, dass sie auf während des Engineerings erzeugten Daten in Form spezifischer Projektergebnisse angewiesen sind. Simulationsbasiertes Testen kann in für den jeweiligen Anwendungszweck hoch spezialisierten Werkzeugen oder, alternativ, in integrierten Toolsuiten stattfinden.

3.2 Einsatz von Simulation im Rahmen einer virtuellen Inbetriebnahme

Gemäß [VDI 3693-1] kann „[das] [...] Aufdecken von Fehlern aus dem Engineering des Automatisierungssystems“ als Globalziel einer *virtuellen Inbetriebnahme* beschrieben werden. Es ist der Methode der virtuellen Inbetriebnahme daher zu eigen, den Test des erstellten Steuerungs codes und dadurch die Erreichung eines Reifegrads zu ermöglichen, der die Inbetriebnahme von Anlagen vor Ort reibungsloser und schneller ablaufen lässt. Üblicherweise nicht Teil *einer virtuellen Inbetriebnahme* ist die Simulation einzelner, konkreter Fertigungsprozesse wie Bohren, Honen, Sintern und dgl. (wie in der [DIN 8580] genannt, [VDI 3693-1]). Hierfür existieren spezielle Simulationsverfahren und -modelle, die in dieser Arbeit nicht im Fokus stehen. Einzelne Fertigungsprozesse werden für eine virtuelle Inbetriebnahme in der Regel durch einfache Ersatzmo-

delle abstrahiert, die nur die Eingangs- und Ausgangskörper eines Fertigungsschritts nachbilden [VDI 3693-1]. Rein manuell (d. h. durch einen Menschen) durchgeführte Handhabungs- und Montagetätigkeit wird in einer virtuellen Inbetriebnahme ebenso wenig betrachtet [VDI 3693-1]. Die hierfür existierenden Simulationsverfahren und -modelle werden in dieser Arbeit nicht betrachtet.

In der Richtlinie [VDI 3633-8] wird das Ziel einer Simulation als „[...] das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind“ beschrieben. Simulation beachtet immer den inneren Mechanismus eines Systems, ggf. auf verschiedenen Abstraktionsebenen ([EHR84], [KRNE98]). Nach Kramer [KRNE98] ist es für Simulation kennzeichnend, dass „[...] zwischen dem, was simuliert wird, und dem, womit simuliert wird, [...] eine Ähnlichkeitsbeziehung bestehen muss“. Simulationsmodelle der geplanten, dimensionierten und projektierten *mechanischen* und *elektrotechnischen* Ressourcen einer Anlage sind der Kern einer *virtuellen Inbetriebnahme* – die Nachbildung ihres Verhaltens mittels Simulation ist es, welche das charakteristische Anlagenverhalten aufzeigt, das gegen die Anforderungen des Kunden getestet werden kann. Simulationsmodelle können naturgemäß nur Vereinfachungen und Abstraktionen der Realität sein und daher nur Teilaspekte des betrachteten Systems abbilden [HWF⁺12]. Bei der Durchführung von Simulation generell und speziell bei einer *virtuellen Inbetriebnahme* ist daher entscheidend, dass die genutzten Simulationsmodelle im Hinblick auf die (aktuelle) Situation und die Problemstellung genügend aussagefähig sind [HWF⁺12]. Es ist daher „[...] bei allen Überlegungen die Frage nach der Zweckmäßigkeit und Problemrelevanz zu stellen [...]“ ([HWF⁺12], auch [STA73]). Das Anlagensimulationsmodell unterliegt in der *virtuellen Inbetriebnahme* daher Anforderungen hinsichtlich seiner Fähigkeit, die Validierung von IEC 61131-3 Steuerungscode ermöglichen zu können. Bestimmte Teile des Simulationsmodells könnten so einer detaillierten Betrachtung bedürfen als andere. Ein Anlagensimulationsmodell ist in der Regel nicht monolithisch aufgebaut, sondern setzt sich aus vielen verschiedenen Teil-Simulationsmodellen zusammen.

Der Aufwand für eine virtuelle Inbetriebnahme und der erzielte Nutzen müssen stets ausgewogen sein [VDI 4499-2]. Wird mehr Aufwand für eine virtuelle Inbetriebnahme betrieben als sich potenziell bei der realen Inbetriebnahme einsparen lässt, ist die Methode unvereinbar mit den Zeit- wie auch den Kostenzielen des *Digitalen Fabrikbetriebs*. Ziel beim Einsatz einer *virtuellen Inbetriebnahme* sollte daher sein, den für Datenerhebung, Modellerstellung, Testplanung sowie Testdurchführung betriebenen Aufwand durch die *virtuelle Inbetriebnahme* unterstützende Methoden, die Standardisierung der genutzten Datenmodelle und ein durchgängiges Datenmanagement möglichst zu minimieren. Die folgenden Abschnitte ordnen die *virtuelle Inbetriebnahme* in diesem Kontext genauer ein.

3.2.1 Mögliche Anwendungsfelder der virtuellen Inbetriebnahme

Virtuelle Inbetriebnahme kann, abhängig von der betrachteten Unternehmensebene, auf verschiedene Aspekte beim Engineering automatisierter Anlagen abzielen. Das Hauptanwendungsfeld der *virtuellen Inbetriebnahme* wird im Folgenden als das Testen des vom Gewerk Automatisierungstechnik erstellten Steuerungscode gesehen, wie in [VDI 4499-2] und [VDI 3693-1] definiert. In der [VDI 4499-2] sind die möglichen Anwendungsfelder einer *virtuellen Inbetriebnahme* konkret wie folgt umschrieben:

„Mit Hilfe einer virtuellen Inbetriebnahme kann das [bereits vorhandene oder zu erstellende] Anlagen[simulations]modell daraufhin untersucht werden, ob die in der Anlagenplanung gemachten Annahmen mit den tatsächlichen, in der Steuerungstechnik programmierten Abläufen übereinstimmen, z. B. Schrittketten validieren, Spannerpositionen verifizieren.“

Der im Vordergrund stehende Test der im Automatisierungssystem programmierten Abläufe kann in zwei verschiedene Hauptanwendungsfelder differenziert werden: den Test von Robotersteuerungen einerseits und den Test von speicherprogrammierbaren Steuerungen (SPS) andererseits. Auch wenn in beiden Anwendungsfeldern der erstellte Steuerungscode durch Simulation abgesichert werden soll, stehen unterschiedliche Aspekte im Vordergrund: Speicherprogrammierbare Steuerungen können sowohl einzelne Maschinen aber auch ganze Produktionsanlagen, die aus verschiedensten technischen Einrichtungen wie Förderbändern, Hebern, Robotern, etc. zusammengesetzt sein können, steuern. Generell wird diese Heterogenität im Kontext der virtuellen Inbetriebnahme als zu betrachtende Förder-, Montage- und Handhabungstechnik zusammengefasst [VDI 3693-1].

Robotersteuerungen dagegen steuern üblicherweise nur einen oder ggf. mehrere kooperierende Roboter [SMZ07] innerhalb eines genau definierten Bereichs einer Anlage (Zellebene, [ZÄWÜ05]). Sie sind der Montage- und Handhabungstechnik gemäß [VDI 2860] zuzuordnen. Eine SPS ist in dieser Konstellation fast immer mit der Robotersteuerung verbunden und regelt das Zusammenspiel zwischen Roboter und der umgebenden Anlagentechnik. Sie ist der Robotersteuerung also hierarchisch übergeordnet. Je nachdem, ob nun eine Robotersteuerung oder eine SPS betrachtet wird, unterscheiden sich die Anforderungen an das für eine *virtuelle Inbetriebnahme* benötigte Anlagensimulationsmodell ganz erheblich. Beim Test von Steuerungscode für Robotersteuerungen können folgende Aspekte im Vordergrund stehen (Auswahl in Anlehnung an [KÜH06]):

- Sicherstellung von Kollisionsfreiheit zu Störkanten: Die Sicherstellung der Kollisionsfreiheit des Roboters zu Störkanten muss in jedem Fall sichergestellt sein. Störkanten können am Bauteil selber auftreten. Aber auch die Halle oder die Anlage/Zelle, in der sich der Roboter befindet, ist aus Sicht des Roboters eine Störkante.
- Sicherstellung von Kollisionsfreiheit zu andere Robotern: Arbeiten Roboter kollaborativ in einer einzigen Fertigungszelle, wie es z. B. im automobilen Rohbau die Regel ist [BKH*09], ist der Prozess dahin gehend zu überprüfen, dass Kollisionen zwischen Robotern ausgeschlossen werden können.
- Bahnoptimierung, wenn dies nicht schon bei der Erstellung des Steuerungscode in einer Offline-Programmierung beachtet wurde.

Die vorgestellten Aspekte implizieren, dass in der Robotik räumliche oder sphärische Mechanismen im Vordergrund stehen [VDI 2860]. Es wird „[...] eine definierte Lage eines geometrischen definierten Körpers entweder geschaffen oder für eine begrenzte Zeit aufrecht erhalten“ ([VDI 2860], auch [VDI 3693-1]). Dies setzt zwingend ein (dreidimensionales) Geometriemodell für die Simulation in einer *virtuellen Inbetriebnahme* voraus. Eng verknüpft, aber dennoch deutlich abzugrenzen davon ist die in der Robotik etablierte Offline-Programmierung von Robotern [KUG99], welche das Ziel hat, den Robotersteuerungscode (teils automatisiert, [BWG09]) zu erstellen. Bereits hier können grundlegende Tests des Roboterprogramms stattfinden [KÜH06].

Abgegrenzt und mit anderem Fokus steht dazu die virtuelle Inbetriebnahme von (Gesamt-) Anlagen und Maschinen, die mittels SPS gesteuert werden. Die bereits für die Inbetriebnahme von Robotern identifizierten, dort essenziellen Aspekte (räumliche/sphärische Mechanismen und

damit im erweiterten Sinne die Montage- und Handhabungstechnik) *können* auch für einzelne Teile einer Gesamtanlage oder einer Maschine gelten. Sie stehen aber nicht zwingend im Vordergrund und bilden nicht zwangsläufig das charakteristische Merkmal, welches in einer virtuellen Inbetriebnahme beachtet werden sollte. Es sind vielmehr spurgebundene Materialflussapplikationen, die, durch Montage- und Handhabungsgeräte ergänzt, das Gros bei der Betrachtung von Anlagen ausmachen. Diese zumeist spurgebunden Systeme, wie Elektrohängebahnen ([VDI 3643], [VDI 4441-1]), fahrerlose Transportsysteme [VDI 2510], Fördertechnikapplikationen mit Band-, Ketten- und Rollenförderern oder auch Schubplattenanlagen (siehe [MRW04] für eine Übersicht) und dergleichen, schränken die Freiheitsgrade der durch Simulation nachzubildenden Bewegungen ein und erfordern so nicht immer die Verwendung eines dreidimensionalen Geometriemodells als notwendige Basis einer Simulation. Vielmehr sollte im Vordergrund der Planung einer virtuellen Inbetriebnahme stehen, wann welcher Detaillierungsgrad (= *wie ein Modell beschaffen sein sollte*) für die jeweilige Situation und die jeweilig betrachtete Anlage (respektive feingranular für welchen Anlagenteil) notwendig ist, um virtuelle Inbetriebnahme effizient und zielgerichtet einsetzen zu können [FAV11]. Ferner ist ein positiver Aufwand/Nutzen-Effekt durch die Minimierung von Modellierungsaufwänden sicherzustellen.

3.2.2 Einordnung innerhalb der AT-Pyramide

Um den Wirkungsbereich einer virtuellen Inbetriebnahme konkret zu definieren, kann die wohlbekannte Automatisierungspyramide herangezogen werden, die die verschiedenen Ebenen einer industriellen Fertigung darstellt. *Virtuelle Inbetriebnahme* wird im Folgenden als eine Untersuchung auf *Steuerungsebene* definiert: Im Kern wird die eingesetzte Steuerung und im Speziellen der für die Steuerung erstellte Steuerungscode getestet [VDI 3693-1].

Speicherprogrammierbare Steuerungen weisen verschiedene (datentechnische) Schnittstellen zu anderen Ebenen der Automatisierungspyramide auf, die verschiedenartigen Einfluss auf die Anwendung der *virtuellen Inbetriebnahme* haben. Über die in der *Feldebene* vorhandenen Feldgeräte (meist in Form von Ein-/Ausgabebaugruppen, auch Kommunikationsperipherie genannt, hierarchisch unterhalb der Steuerungsebene gelegen) entsteht durch die Bündelung von Sensorsignalen und der Ausgabe von Stellsignalen die signaltechnische Schnittstelle zum eigentlichen Produktionsprozess auf *Prozessebene*. Die über ein Feldbussystem mit der SPS verbundenen Feldgeräte erlauben daher erst das eigentliche Steuern, Regeln und Beobachten des technischen Prozesses. Über den Feldbus werden mit Feldgeräten entweder binäre Signale (überwiegend) oder Prozessdatenwörter (z. B. bei „*intelligenten Feldgeräten*“, [GÜHO10]) bidirektional ausgetauscht, abhängig vom Typus des eingesetzten Feldgeräts.

Speicherprogrammierbare Steuerung besitzen nicht nur Schnittstellen zur *Feld-* respektive (indirekt) zur *Prozessebene*, sondern auch zu übergeordneten Intralogistiksystemen (wie Materialflussrechnern) und weiteren Systemen auf *Prozessleit-/Betriebsleitebene* sowie zu ERP-Systemen auf *Unternehmensebene* wie SAP (z. B. *Automatisches Standard Lager ASL* [BMWASL[®]]). Die Perspektive ist im Vergleich zur bereits beschriebenen Schnittstelle zur *Feld-/Prozessebene* eine diametral andere. Es ist aus diesem Blickwinkel nicht die SPS, die steuert – sie wird gesteuert. Die Kommunikation der SPS zum Leitsystem erfolgt, nicht wie zur und von *Feldebene* üblich, über binäre Signale oder Prozessdatenwörter über einen standardisierten Feldbus, sondern rein telegrammbasiert mittels spezieller Kommunikationsbausteine über das Firmennetzwerk nach dem Client/Server-Prinzip [GIE05]. Diese Kommunikationsbausteine stehen in den meisten Fällen in

Werkstandards zur Verfügung, so dass es die Hauptaufgabe des Gewerks Automatisierungstechnik ist, die Schnittstelle zum Leitsystem in den programmierten Ablauf im Steuerungscode prozessgerecht einzubinden. Während einer virtuellen Inbetriebnahme steht daher weniger die Überprüfung der reinen Kommunikation im Vordergrund, vielmehr ist es das von außen sichtbare Verhalten des Leitsystems, dessen Zusammenspiel mit der SPS getestet werden sollte. Der innere Aufbau des Leitsystems spielt für die SPS keine Rolle. Die SPS muss nicht im Sinne des in der Anlage erfolgenden mechatronischen Zusammenspiels mittels Steuersignalen auf das Leitsystem reagieren.

Grundsätzlich stellt sich die Frage, welche Ebenen der AT-Pyramide in einer virtuellen Inbetriebnahme simuliert, welche emuliert werden sollten. Simulation wird in [IEEE STD. 610.3] als Fähigkeit eines Modells beschrieben, sich wie ein reales System zu verhalten, wenn es von außen mit Eingangsgrößen beschaltet wird. Um sich wie ein reales System verhalten zu können, muss der innere Aufbau des zu simulierenden Systems beachtet und nachgebildet werden (→ Abschnitt 3.1). Dies trifft hier auf die Prozessebene zu: Durch das Setzen von Stellgrößen verhält sich das Modell einer Anlage so, dass Regelgrößen in ihrem Wertebereich variieren. Dadurch wird das Steuern, Regeln und Beobachten eines technischen Prozesses ermöglicht.

Emulation hingegen beschreibt die Fähigkeit eines Modells, dieselben Eingangsgrößen wie ein reales System zu akzeptieren und dieselben Ausgangsgrößen, die vom realen System zu erwarten wären, bereitzustellen [IEEE STD. 610.3]. Der Hauptunterschied zur Simulation liegt darin, dass der innere Aufbau, d. h. wie die Ausgangsgrößen gebildet werden, keine Rolle spielt. Emulation bietet sich daher vor allem immer dann an, wenn rein datentechnische Schnittstellen nachgebildet werden sollen und auf eine definierte Eingangsgröße eine definierte Ausgangsgröße folgen soll.

Im Rahmen einer virtuellen Inbetriebnahme steht auf der Feldebene das signaltechnische Verhalten des Feldgeräts auf dem Feldbus im Fokus [SWD⁺13]. Die Herausforderung liegt daher im Nachbilden der jeweiligen feldbustypischen Signale: Die Feldgeräte müssen die von einer SPS versendeten Signale empfangen und der SPS im Gegenzug Signale, die wiederum die SPS versteht, zur Verfügung stellen. Für das Nachbilden dieses rein datentechnischen Verhaltens kann der innere Aufbau des Feldgeräts (d. h. wie das Feldgerät auf Hardware-Ebene funktioniert) vernachlässigt werden. Die Emulation bietet sich daher für eine Umsetzung des Geräteverhaltens auf Feldebene an. Selbiges gilt für Betrachtungen der Leitsystemebene und darüber liegenden Ebenen wie MES und ERP. Zu beachten ist, dass datentechnische Schnittstellen zwischen der Steuerung und anderen Systemen innerhalb der AT-Pyramide stets so modelliert werden müssen, dass Informationsverlust aufgrund nicht kompatibler Informationsstrukturen ausgeschlossen wird [KHK10].

Abbildung 3-2 zeigt die mögliche Einordnung der *virtuellen Inbetriebnahme* in die Automatisierungspyramide. Die hinter den einzelnen Ebenenbezeichnungen stehenden Identifier (E) und (S) beziehen sich auf die in der virtuellen Inbetriebnahme durchgeführte Umsetzung mittels (E)mulation oder (S)imulation.

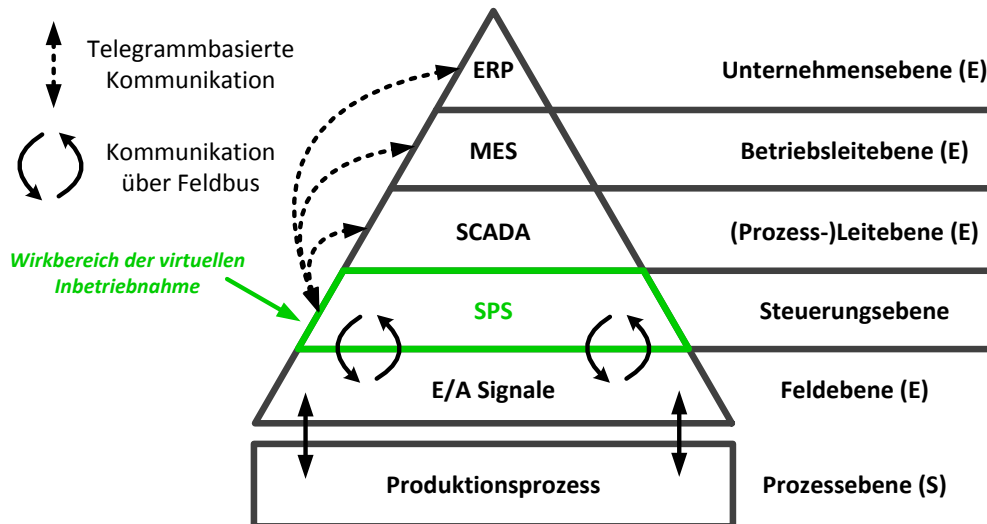


Abbildung 3-2: Automationspyramide mit Wirkungsbereichen der virtuellen Inbetriebnahme

3.2.3 Mögliche Vorteile und Nachteile einer virtuellen Inbetriebnahme

Die Vorteile einer *virtuellen Inbetriebnahme* liegen in der Steigerung der Qualität der Ergebnisse des Gewerks Automatisierungstechnik und damit der Steigerung des Reifegrads von Steuerungscode. Selbst wenn in manchen Projekten das eng mit der virtuellen Inbetriebnahme zusammenhängende *Zeitziel des Digitalen Fabrikbetriebs* nicht erreicht werden kann, liefert die virtuelle Inbetriebnahme potenziell durch die vorgenommene Vorab-Validierung der Steuerungscode zusätzliche Sicherheit. Auf diese Weise kann der Fokus der (realen) Inbetriebnahme auf andere Aspekte, wie der Sicherstellung vorab unklarer Verhaltenseigenschaften der Anlage, gerichtet werden. Virtuelle Inbetriebnahme kann noch mehr leisten: Grundsätzlich steht dem Gewerk Automatisierungstechnik mit der *virtuellen Inbetriebnahme* eine Methode zur Verfügung, die es erlaubt, systematisiert und laufend Steuerungscode zu testen, und parallel zur Erstellung des Steuerungscode bereits an der (in diesem Fall virtuellen) Anlage zu programmieren („*Programming by permutation*“ [Wiki®]). Die folgende Auflistung zeigt die möglichen Vorteile einer virtuellen Inbetriebnahme, wie sie in [VDI 3693-1] zusammenfassend definiert werden:

Virtuelle Inbetriebnahme

- ... kann den Steuerungstest mit Hilfe von Simulationsmodellen in frühe Phasen der Implementierung vorverlagern.
- ... bietet sicherheitstechnische Vorteile: Versagt die AT-Infrastruktur, kann es im simulierten Prozess nicht zu Schäden am Menschen, an der Anlage oder der Umwelt kommen.
- ... bietet die Möglichkeit, den potenziellen Schaden bei Versagen der AT-Infrastruktur gefahrlos zu ermitteln.
- ... kann in einen vorher definierten Anfangszustand (z. B. Lager voll) gebracht werden, um Fehlersituationen nachzustellen
- ... bietet die prinzipielle Möglichkeit, eine Simulation zu einem bestimmten Zeitpunkt abzubrechen, den aktuellen Zustand zu speichern, und später wieder zu initialisieren.
- ... bietet die Möglichkeit, Ein- und Ausgangssignale des Simulationsmodells zu überschreiben (forcen), um fehlerhafte Zustände gezielt zu testen

Die Methode der virtuellen Inbetriebnahme bietet nicht nur Vorteile, auch die Nachteile der Methode sind klar zu benennen. Der größte Nachteil einer *virtuellen Inbetriebnahme* liegt im nach wie vor hohen Implementierungsaufwand, der das zu erreichende, positive Aufwand/Nutzen Verhältnis in der Praxis oft kompromittiert. Durch die Verkürzung der Inbetriebnahme vor Ort sind auf der einen Seite zwar signifikante Einsparungen möglich [ZWH*06], der hohe Aufwand bei der Modellerstellung übertrifft auf der anderen Seite das Einsparpotenzial, das durch *virtuelle Inbetriebnahme* möglich wäre [KHK10]. Acél [Acé96] identifiziert insbesondere die *Datenerhebung* sowie die eigentliche *Modellerstellung* als aufwändige Teilaufgaben bei der Durchführung von Simulationsstudien. Abbildung 3-3 zeigt die von [Acé96] beschriebene zeitliche Aufwandsverteilung bei der Durchführung von Simulationsstudien in Prozent. Die einzelnen Aufwände zur Durchführung spezifischer Tätigkeiten (Abbildung 3-3 links) werden durch die in der Grafik eingezeichneten Balken (Abbildung 3-3, grau hinterlegtes Feld) illustriert. Die Balken sind als Box-Plot [KRO14] dargestellt. Die linke Zahl eines Balkens zeigt das untere Quartil, die mittlere Zahl den Median und die rechte Zahl das obere Quartil der in der Studie aufgenommenen Werte, die in Prozent die Aufwände einzelner Tätigkeiten während einer Simulationsstudie darstellen.

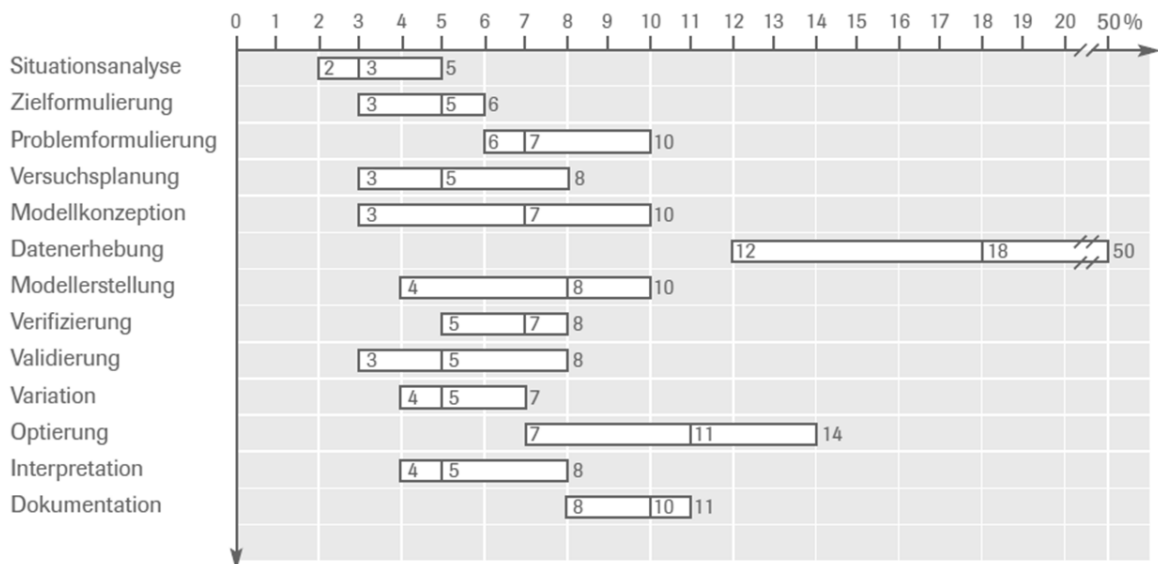


Abbildung 3-3: Aufwände bei der Durchführung einer Simulationsstudie gemäß [Acé96]

Oftmals können oder wollen in der Praxis am Engineering-Prozess beteiligte Gewerke, insbesondere wenn es sich um KMU handelt, eine Modellerstellung von Hand nicht leisten oder verfügen nicht über die notwendige Expertise oder die personellen Ressourcen dafür. Ferner verursacht die Datenerhebung, bedingt durch die vorherrschende heterogene Werkzeuglandschaft und den nicht vorhandenen Schnittstellen zwischen den genutzten Engineering-Werkzeugen (→ Abschnitt 2.6) große Aufwände. Zudem kann die Komplexität wie auch Kompliziertheit eines Modells [HWF*12] unterschätzt werden. Je umfangreicher der Detaillierungsgrad der Validierung (respektive der Simulation) ist, desto beträchtlicher ist der zu leistende Aufwand für die Umsetzung [MAN09]. Allgemein gilt für Simulation: Ist das Modell zu kompliziert, ist die Beschränkung durch die verfügbaren (respektive nicht verfügbaren) personellen Ressourcen ein Hinderungsgrund für das erfolgreiche Umsetzen einer Simulationsstudie [BRTo96].

Wird *virtuelle Inbetriebnahme* zudem nicht mit dem Bauherren und den am Engineering in projektspezifischer Zusammensetzung beteiligten Gewerken abgestimmt und entsprechend in den Engineering-Prozess verankert, besteht die prinzipielle Gefahr, dass sich der Steuerungscode in

eine falsche Richtung entwickeln kann. Der Reifegrad einer steuerungstechnischen Lösung kann überschätzt werden. Potenziell kann so für das jeweilige Projekt großer Schaden entstehen. Das Gewerk Automatisierungstechnik, wie auch die anderen Projektbeteiligten, wiegen sich in falscher Sicherheit (es wurde ja schließlich simuliert). Der eigentlich validierte Steuerungscode kann sich als nicht geeignet und/oder als nicht richtig implementiert herausstellen und die durchgeführte Validierung so nutzlos sein. Der eigentlich im Fokus stehende positive Aufwand-/Nutzeffekt wird dadurch nicht nur ins Negative verkehrt. Der Aufwand auf der Baustelle zur Reparatur der durch falsche Programmierung entstandenen Schäden (immaterieller wie materieller Natur) kann sogar gesteigert werden. Im Fokus einer virtuellen Inbetriebnahme sollte daher nicht nur die reine Ausführung, sondern auch eine mit allen beteiligten Gewerken und insbesondere dem Bauherren definierte Aufgabenspezifikation stehen, „[...] die nach Meinung aller Beteiligten das zu lösende Problem beschreibt und zugleich mit den vorgesehenen Mitteln und im vorgesehenen Zeit- und Kostenrahmen umsetzbar ist“ [RWS06].

Bei nur unzureichendem Projektmanagement und unzureichender Einführung der virtuellen Inbetriebnahme in eine Engineering-Organisation kann die virtuelle Inbetriebnahme daher nicht zur Qualitätssteigerung beitragen, im Gegenteil: Von einem Einsatz sollte in dieser Situation abgesehen werden, da der potenziell entstehende Schaden jeden positiven Nutzeffekt mehr als aufwiegen kann.

3.3 Technische Implementierung von Simulationsmodellen

Die technische Implementierung von Simulationsmodellen, auch als Testkonfiguration bezeichnet [VDI 3693-1], kann auf verschiedene Arten durchgeführt werden. Die virtuelle Inbetriebnahme ist in der Regel als maschinennahe Simulationsmethode zu charakterisieren [KIE07]. Im konkreten wird in der [VDI 3633-8] unter maschinennaher Simulation „[...] die Simulation von Fertigungsmaschinen und ihrer Peripherie“ verstanden. Für die Umsetzung einer derartigen Testkonfiguration können verschiedene Konfigurationen zum Einsatz kommen. Gemäß [VDI 3693-1] unterscheiden sich die eingesetzten Konfigurationen hinsichtlich ...

- ... der Art und dem Umfang der eingesetzten Simulationsmodelle
- ... der Varianten der eingesetzten Simulationswerkzeuge
- ... der Kommunikationsform zwischen Simulation und Automatisierungssystem
- ... Art und Umfang der nachzubildenden Strukturen des Automatisierungssystems

Gebräuchlich für die virtuelle Inbetriebnahme sind gemäß [VDI 3693-1] *Model-in-the-Loop* (MIL), *Software-in-the-Loop* (SIL) und *Hardware-in-the-Loop* (HIL) Konfigurationen.

Eine MIL-Simulation ermöglicht den entwicklungsbegleitenden Test von (prototypisch) implementierten Steuerungsalgorithmen [VDI 3693-1]. Die Steuerungsalgorithmen werden nicht in einer der [IEC 61131-3] Programmiersprachen umgesetzt, sondern z. B. als Automaten dargestellt [VDI 3693-1]. Der Einsatz einer MIL-Simulation ist vorteilhaft, wenn in frühen Projektphasen (vor oder ganz zu Beginn der Programmerstellung) steuerungstechnische Konzepte überprüft werden sollen (Proof-of-Concept). Die (prototypische) Steuerung und die Anlagensimulation werden üblicherweise im gleichen Simulationswerkzeug implementiert [VDI 3693-1]. MIL-Simulation ist unabhängig von der verwendeten Zielhardware und kann immer durchgeführt werden.

Unter einer SIL-Simulation wird gemäß der Definition in [VDI 3693-1] der Test einer *emulierten Steuerung* gegen eine *simulierte Anlage* verstanden. Der Steuerungstest findet in einer SIL-

Simulation daher in einer komplett virtuellen Umgebung statt. In einer SIL-Simulation wird der Seriencode, üblicherweise in einer der [IEC 61131-3] Programmiersprachen implementiert, genutzt [VDI 3693-1]. Eine SIL-Simulation versteht sich als eine auf MIL-Simulation folgende Stufe [VDI 3693-1]. Für die Durchführung einer SIL-Simulation ist das Nachbilden der Zielhardware notwendig [VDI 3693-1]. Die Zielhardware (SPS, Feldgeräte/Kommunikationsperipherie) muss mittels Emulation (→ Abschnitt 3.2.2) nachgebildet werden [VDI 3693-1].

In einer HIL-Simulation wird die *reale Steuerung* gegen eine *simulierte Anlage* getestet [VDI 3693-1]. HIL-Simulation ist als weitergehende Teststufe zu einer SIL-Simulation definiert [VDI 3693-1]. Wie in der SIL-Simulation wird in der HIL-Simulation der Seriencode genutzt. Eine HIL-Simulation eignet sich immer dann, wenn ein Integrationstest oder ein Gesamttest des Automatisierungssystems durchgeführt werden soll [VDI 3693-1]. Bei einer HIL-Simulation darf die eingesetzte Steuerung (die der Zielhardware entsprechen muss) keinen Unterschied zur realen Anlage erkennen [RPK*14[#]]. Das Signalverhalten der Feldgeräte muss daher demjenigen der realen Anlage entsprechen (→ Abschnitt 3.2.2). Im Zuge der Nutzung einer HIL-Simulation müssen daher die projizierten Feldgeräte emuliert werden.

Abbildung 3-4 zeigt die in einer *virtuellen Inbetriebnahme* gebräuchlichen Testkonfigurationen mit Hinblick auf die notwendigen Elemente sowie welche Elemente emuliert oder simuliert werden.

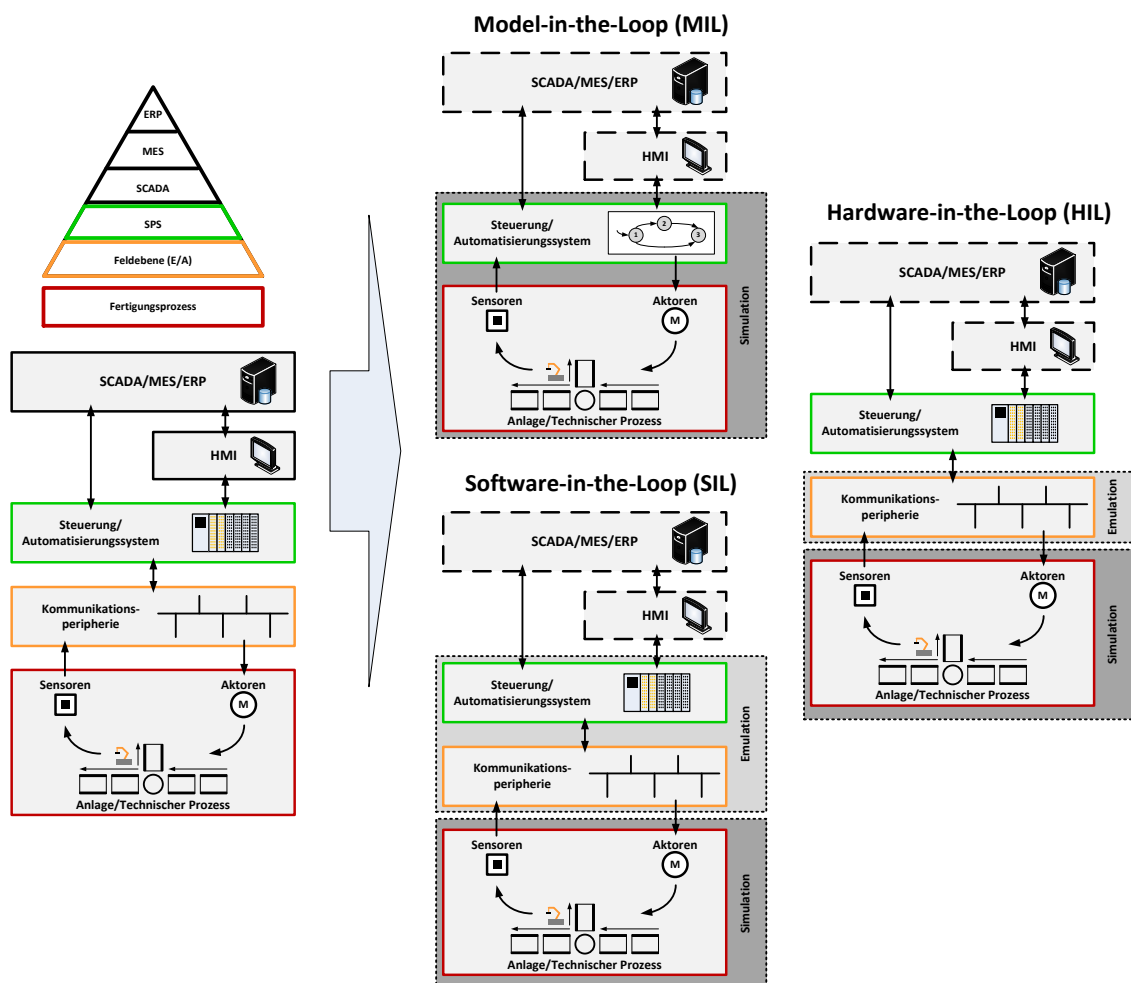


Abbildung 3-4: Übersicht über gebräuchliche Testkonfigurationen der virtuellen Inbetriebnahme in Anlehnung an [VDI 3693-1]

3.4 Typische Simulationswerkzeuge

Die für eine *virtuelle Inbetriebnahme* verwendeten Simulationswerkzeuge sind ebenso mannigfaltig wie deren Einsatzzwecke unterschiedlich sind. Grundsätzlich handelt es sich um hoch spezialisierte Werkzeuge, die sich für einzelne Teilaspekte, wie die Simulation von Robotik (z. B. RF:RobSim [ROBSIM[®]]), Fördertechnik (z. B. WinMOD-SIMLINE [WINMOD[®]]), oder Kombinationen daraus, eignen.

Simulationswerkzeuge für die virtuelle Inbetriebnahme unterscheiden sich oftmals hinsichtlich ihrer zugrunde liegenden Modellierungssprache. Die Modellierung kann als kausal (gerichtete Schnittstellen, die Charakterisierung von Ein- oder Ausgangsgröße ist erforderlich) oder a-kausal (nicht gerichtet Schnittstellen, kann sowohl Eingangs- als Ausgangsgröße oder beides zusammen sein, abhängig von der Gleichung der Simulationsmodelle) beschrieben werden [BRPO09]. Ein Beispiel für einen a-kausalen Modellierungsansatz bietet die Modellierungssprache Modelica [MODELICA[®]], während Matlab-Simulink [MATLAB[®]] einen typischen Vertreter kausaler Modellierungsansätze darstellt. Ferner lassen sich Simulationswerkzeuge, die dem objektorientierten Paradigma folgen und eine entsprechende Gestaltung von Simulationsmodellen erlauben und nicht-objektorientierte Werkzeuge, die eine proprietäre Modellierungssprache bieten, identifizieren.

Kern fast aller Simulationswerkzeuge sind Bibliotheken mit typischerweise in Anlagen eingesetzten Elementen wie Förderbändern, Sensoren und Aktoren, etc. Studien zeigen in diesem Zusammenhang, dass „[...] der Zugriff auf bestehende Modelle als erheblicher Faktor für den Erfolg simulationsbasierter Methoden zu werten ist“ ([ACÉ97], zitiert nach [BAR11]). Ein Bibliothekskonzept ist insbesondere bei der durchzuführenden Modellerstellung hilfreich. Nicht objektorientierte Simulationswerkzeuge stellen in diesem Kontext Bibliotheksmodule unter Anwendung signalorientierter, regelungstechnischer Modellierungstechniken zur Verfügung [ALB99]. Auf die Vorteile der Objektorientiertheit wird im Allgemeinen in Abschnitt 2.3 eingegangen. An dieser Stelle sei zusätzlich vor allem auf [BAR11] hingewiesen, der die Vorteile einer objektorientierten Modellierung eines Simulationsmodells, speziell unter Betrachtung einer virtuellen Inbetriebnahme verfahrenstechnischer Anlagen, ausführlich darlegt.

3.5 Aufkommende Fragestellungen im Kontext der Umsetzung einer virtuellen Inbetriebnahme

Der Nutzen von Simulation und insbesondere einer *virtuellen Inbetriebnahme*, um die Qualität und einen hohen Reifegrad von Projektergebnissen (ergo des projektierten Steuerungscode) sicherstellen zu können, konnte bereits in vielen praktischen Anwendungen gezeigt werden. Erfahrungsberichte, insbesondere aus der Automobilindustrie (u. a. [BKH⁺09]) deuten das große Potenzial der Methode an. Dennoch ist die virtuelle Inbetriebnahme derzeit nicht als Standardmethode im Engineering von automatisierten, fertigungstechnischen Anlagen etabliert [BAR15].

Auch wenn die Automobilindustrie als treibende Kraft bei der Einführung von Simulation gesehen werden kann [BCW03], sind es die am Engineering in projektspezifischer Zusammensetzung beteiligten Gewerke, die die Hauptlast von Simulation – das Gewerk Automatisierungstechnik im Speziellen für die virtuelle Inbetriebnahme – während des Engineerings tragen. Das Gewerk Automatisierungstechnik kann in der Praxis oftmals eine aufwändige Modellerstellung nicht leisten oder besitzen nicht das notwendige Modellierungs-Know-How, um Simulation anforderungsgerecht zu implementieren (→ Abschnitt 3.2.3). Die derzeit hohen Aufwände, im Speziellen für die

Datenerhebung sowie die eigentliche *Modellerstellung*, stehen der Durchführung einer virtuellen Inbetriebnahme entgegen. Oberste Prämisse sollte daher sein, den Aufwand für Simulation so weit zu senken, dass die in Abschnitt 2.2 beschriebenen Zeit-, Qualität- und Kostenziele des *Digitalen Fabrikbetriebs* bei der Durchführung von Simulation erreicht werden können.

Simulation und insbesondere *virtuelle Inbetriebnahme* sollten daher soweit ertüchtigt und für das Engineering automatisierter, fertigungstechnischer Anlagen etabliert werden, dass sie einen sinnvollen Beitrag zur Aufwandsreduzierung im Allgemeinen leisten können und vor allem die reale Inbetriebnahme beim Kunden, als eines maßgeblichen Kosten- und Zeitfaktors in Engineering-Projekten, schneller und reibungsloser ablaufen lassen. Hierfür müssen durch das Gewerk Automatisierungstechnik eingesetzte, die *virtuelle Inbetriebnahme* unterstützende Methoden zur Verfügung stehen. Zudem müssen Modelle und Modellierungsrichtlinien soweit aufeinander abgestimmt werden, so dass *virtuelle Inbetriebnahme* mit wenig Aufwand über bauherrenspezifische Festlegungen und Standards hinweg einsetzbar ist. Aus den in Kapitel 2 gezeigten Rahmenbedingungen des Engineerings und der derzeitigen Durchführung von virtueller Inbetriebnahme ergeben sich verschiedene Fragestellungen, die ein Ansatz zur virtuellen Inbetriebnahme beantworten sollte. Aus den bisherigen Betrachtungen ergeben sich insgesamt 5 Leitfragen:

- ***Leitfrage I: Welche Daten werden für eine virtuelle Inbetriebnahme benötigt und wie sollten diese abgebildet sein?***
- ***Leitfrage II: Wie kann man virtuelle Inbetriebnahme umsetzen?***
- ***Leitfrage III: Wie sollte man virtuelle Inbetriebnahme umsetzen?***
- ***Leitfrage IV: Wie kann man effizient Simulationsmodelle erstellen?***
- ***Leitfrage V: Wie kann man Simulationsmodelle implementieren?***

Diese Leitfragen werden im Folgenden genauer erläutert und mögliche, damit zusammenhängende Fragestellungen, die für eine genauere Untersuchung in Betracht kommen können, identifiziert.

3.5.1 Leitfrage I: Welche Daten werden für eine virtuelle Inbetriebnahme benötigt und wie sollten diese abgebildet sein?

Soll die Methode der virtuellen Inbetriebnahme als Standard etabliert werden, sind herstellerübergreifende Standards zur Abbildung von Anlagenstrukturdaten notwendig, um einen breiten Einsatz der virtuellen Inbetriebnahme zu ermöglichen. Es kann nicht das Ziel sein, auf eine spezifische Technologie oder ein spezifisches Werkzeug zu verweisen – zu heterogen sind die einzelnen Einsatzfelder, Modellierungssprachen, die technischen Implementierungsmöglichkeiten der einzelnen verfügbaren Werkzeuge wie auch die im jeweiligen Einzelfall zu beachtenden Anforderungen an eine *virtuelle Inbetriebnahme*. Die Datenbasis einer *virtuellen Inbetriebnahme* sind immer die Projektergebnisse respektive die erstellten Modelle der betrachteten automatisierten Anlage, erstellt durch die Gewerke Mechanik und Elektrotechnik. Das *Datenmodell der Digitalen Fabrik* kann hierfür einen geeigneten Rahmen bieten, ist aber in der Praxis aufgrund der in Abschnitt 2.6 (letzter Absatz) beschriebenen Defizite nicht immer praktikabel nutzbar. Im Speziellen unter Beachtung der aufgeworfenen Leitfrage I lässt sich daher ableiten:

Frage I.1:

Woher kommen die Daten für die Erstellung von Anlagensimulationsmodellen für eine virtuelle Inbetriebnahme und wie sollten diese strukturiert sein?

Es sind nicht nur die zur Verfügung stehenden Modelle und ihre geeignete Gestaltung, sondern auch deren Güte, die einen Einfluss auf die virtuelle Inbetriebnahme haben. Frühe Projektergebnisse eignen sich unter Umständen nicht oder nur eingeschränkt für eine detaillierte Betrachtung des Anlagenverhaltens; schlicht auch deswegen, weil viele Festlegungen zum Prozessablauf oder konkrete Mechanismen noch nicht auskonstruiert und mit dem Bauherren noch nicht abgesprochen sind, so dass sich im weiteren Projektverlauf potenziell Änderungen ergeben können. Virtuelle Inbetriebnahme ist zu diesem Zeitpunkt im Engineering-Prozess nur eingeschränkt möglich. Die Erstellung und Validierung von Steuerungscode für noch nicht ausgereifte Mechanismen ist mit hohem Risiko verbunden. Um eine anforderungsgerechte Umsetzung zu ermöglichen, ist es daher von größter Wichtigkeit, bestimmen zu können, welche Daten in welcher Güte für eine Simulationsstudie in einer *virtuellen Inbetriebnahme* notwendig sind und vom Gewerk Automatisierungstechnik angefordert werden sollten:

Frage I.2:

Welche Daten in welcher Güte sind für eine virtuelle Inbetriebnahme notwendig?

3.5.2 Leitfrage II: Wie kann man virtuelle Inbetriebnahme umsetzen?

Prinzipiell können fertigungstechnische Anlagen zum Zweck der virtuellen Inbetriebnahme beliebig detailliert simuliert werden. Dies kann eine große Anzahl an unterschiedlichen Simulationsmodellen für unterschiedlichste Anwendungszwecke implizieren. Grundsätzlich lässt sich annehmen, dass eine Simulation im höchsten und kompliziertesten zur Verfügung stehenden Detaillierungsgrad immer die Anforderungen an eine virtuelle Inbetriebnahme erfüllt und das Mittel der Wahl bei der Implementierung einer virtuellen Inbetriebnahme ist. Der Einsatz eines einzigen, sehr detaillierten Simulationsmodells wäre daher denkbar. Bei einem derartigen Vorgehen existieren viele Nachteile: Die Anforderungen an die für die virtuelle Inbetriebnahme genutzte Datenbasis in Form von Modellen sowie an das Simulationssystem an sich sind unter Umständen sehr hoch: detaillierte, kinematische Simulationsmodelle verlangen nach geeigneten 3D-CAD-Daten [ZSL08], das physikalische Verhalten muss von Hand ergänzt werden. Simulationsmodelle von automatisierten Anlagen sind in der Regel nicht monolithisch aufgebaut, sondern setzen sich aus einer großen Anzahl von (Teil-) Simulationsmodellen zusammen. Um die Kompliziertheit von Anlagensimulationsmodellen zu verringern – und damit den Modellierungsaufwand zu senken – sollten die Anzahl der zur Nutzung zur Verfügung stehenden Detaillierungsstufen für diese (Teil-) Simulationsmodelle begrenzt sowie ihre Granularität klar definiert sein. Es muss beachtet werden, dass die (in einer Bibliothek) zur Verfügung stehenden Simulationsmodelle im Hinblick auf die (aktuelle) Situation und die Problemstellung genügend aussagefähig sind [HWF⁺12], was mit Hinblick auf Leitfrage II folgende Fragestellung aufwirft:

Frage II.1:

Welche Detaillierungsstufen sollten für (Teil-)Simulationsmodelle zur Verfügung stehen?

Zusätzlich ist bei der Betrachtung und der Definition von Gestaltungsrichtlinien von Simulationsmodellen nicht nur die mögliche Nachbildung von Gut-Prozessabläufen zu beachten: Fehlerfälle, deren Abfangen im Steuerungscode von größter Wichtigkeit sind, um (mechanische) Beschädigungen an der Anlage zu vermeiden, sollten durchgehend und über alle Modellarten und Detaillierungsstufen hinweg beachtet werden. Auch Fehlersituationen, deren Tests während einer Inbetriebnahme eine große Rolle spielen können, aus organisatorischen Gründen aber nicht durchgeführt werden können, können in einer *virtuellen Inbetriebnahme* beachtet werden. Da naturgemäß nicht jeder Fehlerzustand gleichermaßen in allen Detaillierungsstufen getestet werden kann (alleine schon deswegen nicht, da sich die konkrete Implementierung und die Mächtigkeit im Sinne der Fähigkeit, bestimmte Sachverhalte zu simulieren, unterscheiden kann), sollte festgelegt werden, in welchem Detaillierungsgrad welche Art von Fehlern berücksichtigt werden sollte:

Frage II.2:

In wie weit sollten (technische und organisatorische) Fehler/Störungen der Anlage im Simulationsmodell berücksichtigt werden?

3.5.3 Leitfrage III: Wie sollte man virtuelle Inbetriebnahme umsetzen?

Stehen geeignete, standardisierte und auch konsistente Simulationsmodelle in unterschiedlicher Detaillierung zur Verfügung, ergibt sich die Frage, wann welche Detaillierungsstufe für welchen Teil eines Anlagensimulationsmodells genutzt werden sollte. Nicht alle Teile einer Anlage müssen in gleicher Weise detailliert simuliert werden, es gilt: „[...] *so abstrakt und einfach wie möglich, so detailliert wie nötig*“ [HRD97]. Hierfür wird eine den Modellerstellungsprozess unterstützende Methode benötigt, die es dem Gewerk Automatisierungstechnik erlaubt, mit den zur Verfügung stehenden knappen (zeitlichen, monetären und personellen) Ressourcen interpretieren zu können, welche Detaillierungsstufe in welcher Situation angebracht ist. Zudem sollten Situationen berücksichtigt werden, in denen mehrere Detaillierungsstufen für eine Ressource denkbar könnten. Hierfür ist zu klären, ob es möglich ist, den Detaillierungsgrad zur Laufzeit der Simulation anpassbar zu gestalten.

Frage III.1:

Welcher Detaillierungsgrad ist für welche Situation adäquat? Geht diese Situationsabhängigkeit soweit, dass sich der Detaillierungsgrad dynamisch anpassen sollte?

Abhängig von der verwendeten Modellierungstiefe unterscheidet sich zudem das von außen sichtbare Verhalten eines Simulationsmodells [BRJ05]. Verschiedene Modellarten respektive die sie abbildenden Teil-Simulationsmodelle können daher unterschiedliche Schnittstellen beinhalten, die nicht zwangsläufig miteinander kompatibel sind. Dies kann durch die verwendeten, je nach Modellierungstiefe und Modellart potenziell unterschiedlichen Modellierungsansätze (signalflussbasierte oder objektorientierte Modellierung [BRPO09]) begründet sein. Gemäß des Systems Engineering Ansatzes [HWF*12] können „[...] *im Widerspruch stehende oder fehlende Schnittstellenstandards den Datenaustausch über Systemgrenzen hinweg [hier: Systemgrenzen einzelner Teilmodelle] erschweren*“. Dieses Problem wird durch einzelne Teil-Simulationsmodelle, die in verschiedener Modellierungstiefe abgebildet werden, verschärft. Es ergibt sich daher die Notwendigkeit einer systematisierten Betrachtung, Klassifizierung und Beschreibung von den Mo-

dellbildungsprozess unterstützenden Schnittstellenfunktionen, die die Erstellung eines konsistenten Simulationsmodells im Falle eines heterogenen Detaillierungsgrads erlauben:

Frage III.2:

Wie lässt sich die Konsistenz von (Teil-)Modellen in unterschiedlichem Detaillierungsgrad sicherstellen?

3.5.4 Leitfrage IV: Wie kann man effizient Simulationsmodelle erstellen?

Das positive Aufwand/Nutzen Verhältnis ist eine entscheidende, gemäß der Doktrin der *Digitalen Fabrik* global über alle auftauchenden Fragen hinweg zu beachtende Größe. Im Kontext einer durchzuführenden Simulation ergibt sich eine grundlegende Frage, die ein Ansatz für die *virtuelle Inbetriebnahme* beantworten sollte:

Frage IV.1:

Wie kann der Aufwand für eine virtuelle Inbetriebnahme verringert werden, um Simulation (insbesondere bei KMU) als Standardmethode zu etablieren?

3.5.5 Leitfrage V: Wie kann man Simulationsmodelle implementieren?

Neben der Identifizierung und Bereitstellung geeigneter Datenquellen für das Simulationsmodell kommt der Implementierung desselben in ein geeignetes Simulationsframework eine hohe Bedeutung zu. Diese Implementierung kann von einem Model-in-the-Loop bis hin zu einem Hardware-in-the-Loop Ansatz reichen, die sich im dafür zu betreibenden Aufwand ganz erheblich voneinander unterscheiden können.

Frage V.1:

Wie sollten Simulationsmodelle implementiert werden?

Im folgenden Kapitel 4 wird analysiert, inwieweit die hier aufgeworfenen Fragestellungen im gegenwärtigen Stand der Forschung bereits (und falls ja, in welcher Ausprägung) beantwortet wurden. Abgeleitet daraus ergibt sich der in Kapitel 5 beschriebene Handlungsbedarf.

4 Stand der Forschung im Bereich virtuelle Inbetriebnahme

Aufbauend auf den in Abschnitt 3.5 identifizierten Fragestellungen, die für die Durchführung einer virtuellen Inbetriebnahme beantwortet werden sollten, erfolgt im folgenden Kapitel eine Analyse des derzeitigen Stands der Technik und Wissenschaft zu diesem Thema. Hierfür erfolgt zunächst in Abschnitt 4.1 eine grundlegende Auseinandersetzung und Abgrenzung des Engineering-Vorgehens gemäß [VDI 4499-2] sowie der in der [VDI 2206-1] beschriebenen *Entwicklungsmethodik mechatronischer Systeme*. Dies ist von Bedeutung, da in beiden Ansätzen Simulation eine wichtige Rolle spielt, die grundsätzlichen Sichtweisen und Implementierungen aber grundverschieden sind.

Die Betrachtung von konkreten wissenschaftlichen wie industriellen Ansätzen zur virtuellen Inbetriebnahme, die sich an der *Digitalen Fabrik* orientieren, sowie Ansätzen, die außerhalb derselben stehen, bilden die thematische Basis für die in Abschnitt 4.2 vorgenommene Untersuchung. Die dort beschriebenen *konventionellen Ansätze* für eine virtuelle Inbetriebnahme sind Vorgehen oder Methoden zuzuordnen, bei denen sich die Autoren nicht zu einer (teil-)automatischen Modellgenerierung geäußert haben.

Ansätze für die *virtuelle Inbetriebnahme*, die dem Paradigma einer (teil-)automatischen Modellgenerierung folgen, werden ab Abschnitt 4.3 betrachtet. Aufgrund der geringen Anzahl derartiger Ansätze erfolgt keine explizite Unterscheidung zwischen auf der *Digitalen Fabrik* aufbauenden Ansätzen sowie Ansätzen außerhalb derselben. Eine vergleichende Bewertung wird in Abschnitt 4.5, Tabelle 4, gegeben.

4.1 Ein Vergleich von Engineering-Vorgehen und Produktentwicklungsmethoden

Neben dem Vorgehen des *Digitalen Fabrikbetriebs* existieren vielfach Vorgehen aus der „klassischen“ Produktentwicklung, die für eine Verwendung im Anlagenbau postuliert werden. Das Vorgehen des *Digitalen Fabrikbetriebs* und Vorgehen aus der Produktentwicklung weisen Gemeinsamkeiten, aber auch viele Unterschiede auf, so dass zunächst ein Vergleich beider Vorgehen zweckmäßig erscheint. Während die klassische Produktentwicklung bereits dem Namen nach die Entwicklung von Produkten, die in großer Stückzahl produziert werden, im Fokus hat, hat das Anlagenengineering zum Ziel, Anlagen, die als Unikate gebaut werden, möglichst reibungslos zu errichten [FAY09].

4.1.1 Das Vorgehen des Digitalen Fabrikbetriebs gemäß VDI 4499 und der Entwicklung mechatronischer Systeme gemäß VDI 2206

Das am häufigsten während des Engineerings genutzte Vorgehen lässt sich weitgehend dem sequenziellen Vorgehensmodell des *Digitalen Fabrikbetriebs* gemäß [VDI 4499-2] zuordnen. Für eine Beschreibung des Engineering-Vorgehens gemäß [VDI 4499-2] wird auf Abschnitt 2.2 verwiesen, in welchem der *Digitale Fabrikbetrieb* kurz vorgestellt wird. Neben diesem als De-facto-

Standard etablierten Vorgehen stehen Ansätze aus der (Produkt-)Entwicklung respektive der Entwicklung mechatronischer Systeme. Derartige Ansätze versuchen, in der Produktentwicklung üblicherweise angewendeten Lösungsmechanismen und Methoden, mit denen technische Fragestellungen gelöst werden, auf den Anlagenbau zu übertragen und ggf. an die spezifischen Besonderheiten des Anlagenbaus zu adaptieren. Grundsätzlich ist die Anzahl unterschiedlicher Vorgehensmodelle wie auch Methoden für die Produktentwicklung als sehr groß zu bezeichnen ([EHR03], auch [HWF⁺12]). Viele Ansätze, die Methoden und Vorgehensmodelle der Produktentwicklung auf den Anlagenbau übertragen wollen (z. B. [ENTIME[®]], [KIE07], [KUF11]), haben ihre Ursprünge in der *Entwicklungsmethodik mechatronischer Systeme* gemäß [VDI 2206-1]. Im Folgenden wird der Lösungsraum derartiger Methoden daher auf die in der [VDI 2206-1] vorgestellte *Entwicklungsmethodik für mechatronische Systeme* eingegrenzt, die hier beispielhaft für derartige Ansätze stehen soll.

Das Zusammenspiel zwischen allen am mechatronischen Systementwurf beteiligten Domänen wird als oberste Prämisse bei der Entwicklung mechatronischer Systeme gemäß [VDI 2206-1] gesehen. Leitgedanke des Vorgehens auf Makroebene ist, dass alle am Entwicklungsprozess beteiligten Domänen/Gewerke zunächst *gemeinsam* und unter *Einbringung der jeweiligen gewerkespezifischen Expertise* einen *gesamtheitlichen Systementwurf* erstellen. Der Systementwurf wird anhand der vom Kunden spezifizierten Anforderungen an das Gesamtsystem mit zunehmendem Projektfortschritt immer detaillierter und feingranularer ausgearbeitet. Der gewerkeübergreifenden Kommunikation und Zusammenarbeit kommt in diesem Zusammenhang eine hohe Bedeutung zu, was sich u. a. in interdisziplinären Projektteams widerspiegelt. Eine hohe Bedeutung kann zudem dem Zusammenspiel von Kunden und Auftragnehmern beigemessen werden.

Dem gemeinsamen Systementwurf folgt der *domänenspezifische Entwurf*, in dem die einzelnen geplanten Systemaspekte von den beteiligten Gewerken ausgearbeitet werden und der sich nach den Regeln des Simultaneous Engineering [EHR03] richtet. Simultaneous Engineering bedeutet, dass Arbeitsabläufe, wo immer es möglich und sinnvoll erscheint, parallel stattfinden. Dies kann zwar einerseits zu Mehrarbeit führen, da durch die Überlappung von Arbeitsabläufen in der Regel nicht mit den finalen Planungsdaten gearbeitet wird [HWF⁺12]. Andererseits ist es möglich, Planungsfehler frühzeitig zu erkennen. Die Behebung von Planungsfehler kann in späteren Projektphasen große Aufwände (und damit hohe Kosten) verursachen [HWF⁺12]. Eine enge organisatorische wie auch inhaltliche Zusammenarbeit der einzelnen am Entwurf beteiligten Gewerke ist eine Voraussetzung für das Gelingen des Systementwurfs.

Dem domänenspezifischen Entwurf folgt die *Systemintegration*, bei der ausgehend vom Detail (einzelne Teillösungen) hin zum Groben (der finalen Gesamtlösung) das zu entwerfende System zu einer Gesamtlösung integriert wird. Je nach Projektfortschritt werden einzelne Zwischenergebnisse oder das Gesamtsystem gegen den Systementwurf validiert. Für die Mikro-Vorgehensebene (Vorgehen innerhalb einer Makrophase, [HWF⁺12]) wird in der VDI 2206 der Problemlösungszyklus gemäß [HWF⁺12] vorgeschlagen. Alternative (Mikro-) Vorgehensmodelle sind denkbar und von den jeweiligen Projektteams und ihrer methodischen Expertise oder Best-Practices abhängig.

Nicht immer ist davon auszugehen, dass beim erstmaligen Abarbeiten von Projektinhalten bereits die optimale Lösung erarbeitet wird. Iterationen im Entwicklungsprozess sind daher ausdrücklich vorgesehen: Entsprechen das integrierte Gesamtsystem oder einzelne Teilergebnisse nicht den Anforderungen des ursprünglich erstellten Systementwurfs (was gemäß [VDI 2206-1] die Regel ist), soll der gesamte Makro-Entwicklungszyklus nochmals durchlaufen werden. Mit jedem Itera-

tionsschritt erhöht sich die Produktreife, in der VDI 2206 [VDI 2206-1] beispielhaft beschrieben mit der Entwicklung vom Labormuster über das Vorserienmuster hin zum fertigen Produkt. Simulation ist als Modellbildung und -analyse ein integraler Bestandteil des domänenspezifischen Entwurfs, und „[...] [wird] flankiert durch die Abbildung und Untersuchung der Systemeigenschaften mit Hilfe von Modellen und rechnerunterstützten Werkzeugen [...]“ [VDI 2206-1].

Abbildung 4-1 zeigt die in [VDI 2206-1] dargestellte Entwicklungsmethodik mechatronischer Systeme anhand des typisch in Entwicklungsprozessen verwendeten V-Modells. Es sei angemerkt, dass die Verwendung des V-Modells nur eine mögliche Ausprägung hinsichtlich des Vorgehensmodells darstellt. Andere Vorgehensmodelle, wie das Vorgehensmodell nach Hall-BWI (auch bekannt als Vorgehensmodell des Systems-Engineering) oder das Spiralmodell sind denkbar. Eine generelle Übersicht dazu bietet [HWF⁺12], im Speziellen auf verschiedene Implementierungsmöglichkeiten des V-Modells geht [PHK⁺13] näher ein.

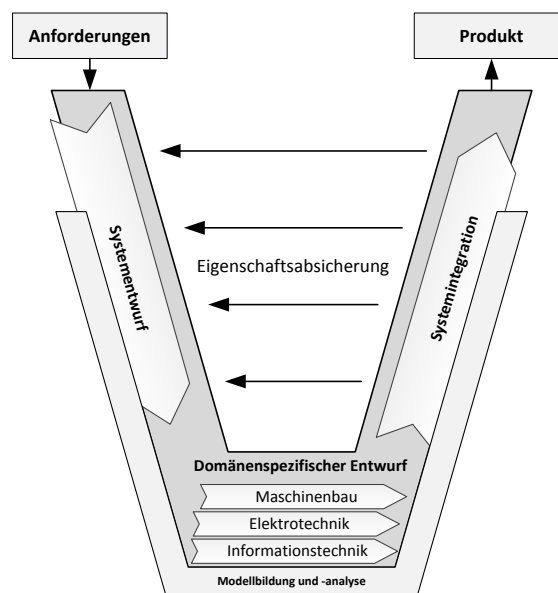


Abbildung 4-1: V-Modell als typisches Vorgehensmodell bei der Produktentwicklung (angelehnt an [VDI 2206-1])

4.1.2 Eine Gegenüberstellung von VDI 2206 und VDI 4499

Ein Vergleich der unterschiedlichen Vorgehensweisen des *Digitalen Fabrikbetriebs* gemäß [VDI 4499-2] sowie der *Entwicklungsmethodik mechatronischer Systeme* wie in [VDI 2206-1] beschrieben zeigt Gemeinsamkeiten, aber auch Unterschiede. Als größte Gemeinsamkeit lassen sich die an der Entwicklung/am Engineering beteiligten Gewerke identifizieren. Gemeinsamkeiten lassen sich ferner bei der grundlegenden Sichtweise auf das zu entwerfende System identifizieren: In beiden Entwurfsdomänen ist die mechatronische Sichtweise, die das Zusammenspiel aus Mechanik, Elektrotechnik und Informationstechnik beschreibt, von Bedeutung, wenn auch in unterschiedlicher Gewichtung. Im Anlagenbau wird das mechatronische Zusammenwirken aufgrund des sequenziellen Systementwurfs nicht immer konsequent und von Anfang an beachtet. Die Situation in Bezug auf vorhandene, genutzte und zwischen den Gewerken auszutauschenden Daten ist ähnlich einzustufen: In beiden Ansätze liegt eine weitgehend heterogene Werkzeuglandschaft vor. Die Sicherstellung der durchgängigen Verfügbarkeit und Auswertbarkeit von Daten

bietet in beiden Ansätzen einen Vorteil und kann zu einer Effizienzsteigerung des Engineerings oder der Entwicklung beitragen.

Neben den Gemeinsamkeiten existiert eine Vielzahl von Unterschieden. Insbesondere ergeben sich bei der eigentlichen Durchführung und Organisation von Projekten starke Abweichungen. Während der mechatronische Systementwurf von allen Gewerken gemeinsam am Anfang des Entwicklungsprozesses erstellt wird, erfolgt im *Digitalen Fabrikbetrieb* de facto eine sequenzielle, zeitlich gestaffelte und nach Gewerken getrennte Abarbeitung der einzelnen umzusetzenden (mechatronischen) Aspekte einer Anlage [VDI 4499-2]. Nur dort, wo es möglich und sinnvoll erscheint (z. B. weil Gewerke nicht auf Ergebnisse der anderen beteiligten Gewerke angewiesen sind), kommt im *Digitalen Fabrikbetrieb* das Prinzip des Simultaneous Engineering zum Tragen.

Gewerkeübergreifende Teams, die gesamtheitlich am Systementwurf arbeiten, sind für den Anlagenbau eher untypisch. Die Interaktion konzentriert sich vornehmlich auf den Austausch von Projektdaten [FAV05]. Kommunikation findet bei Abstimmungsbedarf entweder direkt oder während gemeinsamer Projektbesprechungen statt. Für die Umsetzung der *Entwicklungsmethodik für mechatronische Systeme* sind von den beteiligten Ingenieuren ferner Kenntnisse über alle mechatronischen Domänen notwendig [VDI^B]. Ebenso wird ein hohes Methodenverständnis der beteiligten Ingenieure vorausgesetzt [VDI 6600-2]. Im Gegensatz dazu sind im Anlagenbau hoch spezialisierte Gewerke, die sich auf die jeweilige Domäne konzentrieren, die Regel [FAV05].

Das Vorgehensmodell im Anlagenbau ist dadurch geprägt, dass in fast allen Fällen ein *First-Time-Right* erzwungen ist. Zunächst den Prototyp einer Anlage aufzubauen und diesen in mehreren Iterationen hin zur fertigen Anlage „zu entwickeln“, stellt keinen praktikablen Lösungsansatz dar. Das Wasserfallmodell und ähnlich lineare Vorgehensmodelle kommen daher bevorzugt zum Einsatz, während der mechatronische Systementwurf nicht unbedingt auf ein spezifisches Vorgehensmodell festgelegt ist. Mikrophasen obliegen insbesondere im Anlagenbau allein der Verantwortung des jeweiligen Gewerks. Iterationen, bei der *Entwicklung mechatronischer Systeme* ein fester Bestandteil des Vorgehens, sollen im Anlagenbau aus Zeit- und Kostengründen soweit wie möglich vermieden werden. Simulation spielt in beiden Domänen eine wichtige Rolle, die Implementierungsstrategien unterscheiden sich aber stark. Während des Systementwurfs mechatronischer Systeme und der folgenden domänenspezifischen Implementierung wird ein frühzeitiger Test rein an (Simulations-)Modellen vorgenommen, letztlich auch deswegen, weil insbesondere in der ersten Iterationsschleife noch kein reales System existiert. Dies gilt auch für die Domäne Informationstechnik, da unter Umständen der Controller, auf dem der entworfene Code implementiert und ausgeführt wird, noch nicht festgelegt wurde oder gar erst entworfen werden muss. Modellgetriebene Ansätze, mit dem Ziel der Entkopplung des Softwareentwurfs vom eigentlichen Code und letztendlich der Zielplattform, bieten sich hierfür explizit an.

Auch im *Digitalen Fabrikbetrieb* kann das reine Testen an Modellen von Nutzen sein – vor allem, wenn es gilt, Arbeitsergebnisse der Gewerke Mechanik und Elektrotechnik frühzeitig abzusichern. Werden vom Gewerk Mechanik z. B. Tests für die Kollisionsfreiheit einer mechanischen Apparatur durchgeführt, ist für die Steuerung des Mechanismus nicht zwingend der finale Steuerungscode notwendig. Ein einfaches Automatenmodell kann für diesen Anwendungszweck ausreichend sein. Anders verhält es sich beim Gewerk Automatisierungstechnik, das in dieser Arbeit im Fokus steht. Die Zielplattform (SPS) steht bereits frühzeitig fest, so dass der Steuerungscode direkt in der Zielplattform implementiert wird. Im Vordergrund der Absicherung steht das frühzeitige Validieren des dort erstellten *realen* Steuerungscode, der später an der Anlage direkt eingesetzt wird. Dafür

notwendig ist die Nachbildung der Anlage in einem Simulationsmodell und dort der adäquaten Abbildung der von den Gewerken Mechanik und Elektrotechnik erarbeiteten Ergebnisse.

Tabelle 3 bietet eine Übersicht über die in diesem Abschnitt angesprochenen Aspekte, die für die Entwicklung/das Engineering eine Rolle spielen, und vergleicht diese auf Überschneidungen.

Aspekt	VDI 2206/Mechatr. Entwicklung	VDI 4499/Digitaler Fabrikbetrieb	
Vorgehensmodell (Makroebene)	V-Modell + Abwandlungen, immanentes Simultaneous Engineering, andere Vorgehensmodelle möglich.	Wasserfallartig, stark sequentiell, Simultaneous Engineering wo möglich.	-
Typische Makrophasen	Anforderungsdefinition, Systementwurf, Domänenpezifischer Entwurf, Systemintegration, Abnahme.	Grobplanung, Mechanische Feinplanung, Elektrische Feinplanung, Programmerstellung, Montage und Inbetriebnahme, Anlauf, Serienproduktion.	-
Iterationen	Potentiell über den gesamten Makrozyklus (Erhöhung der Produktreife in jedem Zyklus), ggf. verkürzte Durchführung, je nach notwendigem Umfang der Änderungen.	Wenn möglich Vermeidung angestrebt. Niemals über den vollen Makrozyklus.	-
Vorgehensmodell (Mikroebene)	Problemlösungszyklus gemäß [HWF ⁺ 12], weitere möglich.	Nicht näher definiert, in der Verantwortung der jeweiligen Gewerke.	-
Beteiligte Gewerke	Mechanik, Elektrotechnik, Informationstechnik (+ Auftraggeber).	Mechanik, Elektrotechnik, Automatisierungstechnik (+ Auftraggeber).	+
Teamzusammensetzung	Gewerkeübergreifende Teams, starke interdisziplinäre Prägung.	Meist gewerkespezifische Teams.	-
Systementwurf	Domänenübergreifender Systementwurf, von allen an der Entwicklung beteiligten Gewerken gemeinsam erstellt, parallele Ausarbeitung der einzelnen mechatronischen Aspekte.	Weitgehend sequentieller Systementwurf, zeitlich gestaffelt (Mechanik → Elektrotechnik → Automatisierungstechnik → Implementierung Gesamtsystem).	-
Systemsichtweise	Mechatronische Sichtweise, Unterteilung des Systems in für die Entwicklung sinnvolle, feingranulare Elemente (immanente Modularisierung).	Bewusstsein für mechatronische Sichtweise vorhanden, jedoch keine hierarchische Unterteilung vorgegeben, stark abhängig vom Projekt.	O
Einsatz von Simulation	Entwicklungsbegleitend möglich, nicht genau differenziert, oftmals im Rahmen von Modellgetriebenen Ansätzen.	De Facto: In der Verantwortung der jeweiligen Gewerke. Gewollt: Simulation, insb. virtuelle Inbetriebnahme, als fester Bestandteil des Engineering-Vorgehens.	-
Datenbasis	De facto: Heterogene, werkzeugspezifische Datenbasis. Gewollt: Durchgängiger Datenaustausch in der genutzten (integrierten) Entwicklungsumgebung.	De facto: heterogene, gewerkespezifische Datenbasis. Gewollt: Durchgängiger Datenaustausch in heterogener Werkzeuglandschaft.	+
Interaktion der beteiligten Gewerke	Integrierte Entwicklung, starke Interaktion der beteiligten Gewerke, Produkt wird gemeinsam und schrittweise implementiert.	Weitgehend unabhängige Bearbeitung der einzelnen Themenkomplexe; sequentielle Bearbeitung, wo Gewerke auf den Ergebnissen zeitlich vorher arbeitender Gewerke aufbauen; teils parallele Planung einzelner Teilaspekte, wo möglich. Keine gemeinsame Implementierung der Anlage, stattdessen sequentielles, sehr heterogenes Vorgehen.	-
Notwendiger Ausbildungshintergrund	Kenntnisse über alle (mechatronischen) Bereiche notwendig; hohes Methodenverständnis der beteiligten Ingenieure notwendig.	Fachkenntnisse für den jeweiligen gewerkespezifischen Themenkomplex notwendig.	-

- : Unterschiedliche Sichtweise / O: Keine genaue Unterscheidung möglich / + Gleiche Sichtweise

Tabelle 3: Vergleich von Entwicklung gemäß VDI 2206 und Engineering gemäß VDI 4499

4.1.3 Fazit zum Vergleich von Vorgehensmethoden gemäß VDI 2206 und VDI 4499

Der direkte Vergleich beider Methoden lässt die Schlussfolgerung zu, dass eine Anwendung von Methoden der Produktentwicklung, projiziert auf den Anlagenbau, nur partiell realisierbar erscheint. Nicht vielversprechend oder nur mit großem Aufwand realisierbar ist die Übernahme von organisatorischen Aspekten oder auch von typischen Vorgehenscharakteristiken. Die im Anlagenbau übliche Durchführung eines Projekts (erzwungenes First-Time-Right, sequenzieller Projektlauf, üblicher Organisationsaufbau, etc.) verhindert dies. Es kann in diesem Zusammenhang

durchaus von einem Paradigmenwechsel gesprochen werden [Kuf11] – (etablierte) Vorgehensweisen müssten aufgegeben und neue übernommen werden. Im Speziellen gilt dies auch für Methoden der Simulation und ihrer Durchführung: Wird in einem Model-Driven-Ansatz der finale Steuerungscode aus einem Modell abgeleitet, ergibt der anschließende Test des finalen Steuerungscode gegen das zur Entwicklung genutzte Modell keinen Sinn. Zusammenfassend lässt sich folgern, dass Ansätze, die sich insbesondere mit der Entwicklung von automatisierten Anlagen gemäß der in VDI 2206 beschriebenen oder damit verwandten Methoden befassen, nur eingeschränkt für den Anlagenbau nutzbar sind. Bei der in Abschnitt 4.2 folgenden Untersuchung derartiger Ansätze wird daher insbesondere auf die in Abschnitt 4.1.2 identifizierten Unterschiede, vor allem auch ob und wie diese gelöst werden, geachtet.

4.2 Konventionelle Ansätze zur virtuellen Inbetriebnahme (ohne automatische Modellgenerierung)

4.2.1 Ansätze im Rahmen der Digitalen Fabrik

4.2.1.1 Ansatz von Gulan et al. im Rahmen des Projekts „AVANTI“ (vgl. [GOB⁺14])

Das Forschungsprojekt „AVANTI“ [AVANTI[®]] (*Test methodology for virtual commissioning based on behaviour simulation of production systems*) hatte zum Ziel, die Effizienz und auch den Reifegrad bei der Systementwicklung durch die Erstellung eines virtuellen Produktionssystems zu erhöhen. Wesentliche Teilaspekte des Projekts „AVANTI“ beschäftigten sich mit physikbasierter Simulation, verbesserten Verhaltensmodellen mechatronischer Komponenten und der automatischen Ableitung von Testszenarien, um das Engineering von Produktionsprozessen zu beschleunigen bzw. zu verbessern. Vom Projekt „AVANTI“ wird an dieser Stelle auf folgenden Teilaspekt fokussiert: In diesem Abschnitt auf die physikbasierte Simulation, wie sie von [GOB⁺14] konzeptionell vorgeschlagen wird. Im Abschnitt 4.2.1.2 wird auf den von Süß et al. [SSD15] vorgeschlagenen Ansatz für die Verhaltenssimulation mechatronischer Komponenten zur Nutzung in einer virtuellen Inbetriebnahme eingegangen.

Grundlage der von Gulan et al. [GOB⁺14] beschriebene physikbasierte Simulation „[...] ist ein geometrisches Ersatzmodell [einer] Anlage, welches aus vereinfachten 3D CAD Daten der einzelnen Bauteile aufgebaut ist. Es folgt eine Klassifizierung der Bauteile und Komponenten in stationäre und bewegliche Elemente sowie die Zuordnung verschiedener physikalischer Eigenschaften“. In diesem Kontext spielen Kenngrößen wie Masse, Schwerpunkt und weitere eine große Rolle.

Die physikbasierte Modellierung wird im Rahmen eines dualen Ansatzes durchgeführt. Dual bedeutet, dass die Beschreibung hinsichtlich der Geometrie und der Physik individueller Bauteile und Komponenten in XML-basierten Sprachen (hier: AutomationML) implementiert wird. Die Durchführung der Simulation wird dann in einer separaten Umgebung durchgeführt.

Für die Simulationsumgebung kommen Physik-Engines zum Einsatz. Physik-Engines können die Gesetze der Physik hinsichtlich starrer und deformierbarer Körper (inklusive Kollisionsdetektion) sowie der Fluidmechanik abbilden. Hinsichtlich der Echtzeitfähigkeit wird zusammengefasst, dass „viele der heute erhältlichen Physik-Engines nahezu uneingeschränkt echtzeitfähig sind oder gewisse Beschleuniger anbieten“. Die Physik-Engine muss dazu in der Lage sein, die in XML abgespeicherten Informationen importieren und interpretieren zu können.

*Grenzen des Ansatzes von [GOB*14]*

Der von Gulan et al. beschriebene Ansatz zeigt konzeptionell das Vorgehen einer physikbasierten virtuellen Inbetriebnahme. Die Grenzen des Ansatzes liegen in den für die Durchführung benötigten Daten. So bedingt der Ansatz das Vorhandensein von 3D-CAD Daten. Die Autoren äußern sich nicht zu Fällen, in denen keine geeignete Datenbasis vorliegt. Zu Fällen, in denen eine physikbasierte Simulation nicht notwendig ist, äußern sich die Autoren nicht.

4.2.1.2 Ansatz von Süß et al. im Rahmen des Projekts „AVANTI“ (vgl. [SSD15])

Der Ansatz von Süß et al. zeigt ein Konzept, wie Verhaltensmodelle von Geräteherstellern für Anwender zur Verfügung gestellt werden können. Ferner wird ein Konzept gezeigt, wie diese Verhaltensmodelle für eine virtuelle Inbetriebnahme nutzbar gemacht werden können. Süß et al. [SSD15] nennen als Gründe für das Bereitstellen von Verhaltensmodellen durch den Hersteller, dass der Hersteller einerseits der Einzige ist, der das Wissen besitzt, wie sich eine Komponente im Detail verhält. Andererseits ist der Aufwand für den Anwender, das Verhalten von Komponenten nachzuvollziehen und entsprechende Verhaltensmodelle zu erzeugen, überproportional hoch. Generell werden in diesem Kontext verschiedene Typen von Modellen definiert:

- MBM: Manufacturer Behaviour Model: Verhaltensmodell einer Komponente, die vom Hersteller der Komponente bereitgestellt wird und von ihm gepflegt und erweitert wird. Diese Modelle werden dem Anwender zur Verfügung gestellt.
- UBM: User Behaviour Models. Ein UBM enthält n MBMs und zusätzliche Funktionalitäten. Die Ergänzung zusätzlicher Funktionalitäten ist notwendig, um MBMs gemäß der Anforderungen und Standards der jeweiligen Anwender anpassen zu können. UBMs werden vom Anwender erzeugt.
- PBM: Plant Behaviour Model. Das PBM enthält n UBMs und stellt die zu testende Anlage dar. Das PBM wird vom Anwender erzeugt.

Um die Verhaltensmodelle zwischen Hersteller und Anwender austauschen zu können, ist ein Datenaustauschformat notwendig. Süß et al. [SSD15] untersuchen hierfür die Nutzung von PLCopenXML sowie die Nutzung von FMI/FMU [FMI®]. Süß et al. folgern, dass die Nutzung von FMI/FMU den vielversprechendsten Lösungsweg darstellt. Der Grund hierfür ist die breite Werkzeugunterstützung von FMI/FMU. Als Modellierungswerkzeug zur Erstellung der FMUs soll Matlab/Simulink genutzt werden.

Das Konzept von Süß et al. [SSD15] sieht vor, unterschiedliche FMUs im Rahmen eines Co-Simulationsansatzes zu einem Gesamtanlagenmodell miteinander zu verknüpfen. Das hierfür notwendige Co-Simulations-Framework wird in [SST16] beschrieben. Im Gegensatz zu den Äußerungen in [SSD15] kommt als Modellierungswerkzeug Dymola (und damit die Modellierungssprache Modelica) zum Einsatz. Bei der Beschreibung des Co-Simulationsframeworks wird nicht auf die Synchronisierung der Kommunikation der einzelnen FMUs untereinander eingegangen.

Grenzen des Ansatzes von Süß et al.

Der Ansatz von Süß et al. beschreibt ein Konzept, das den Austausch von Verhaltensmodellen, die für eine virtuelle Inbetriebnahme genutzt werden können, ermöglicht. Aufwände zur Erstellung von Verhaltensmodellen einer spezifischen Komponente durch den Anwender können mit dem beschriebenen Konzept stark gesenkt oder sogar ganz vermieden werden. Voraussetzung dafür ist die herstellerübergreifende Unterstützung von FMU als Austauschformat sowie das Vorhanden-

sein eines zur Implementierung von FMUs geeigneten Co-Simulation-Frameworks. Der vorgestellte Ansatz hat seine Grenzen bei den für eine Implementierung der Verhaltensmodelle benötigten Aufwänden. So ist einerseits Aufwand notwendig, um aus herstellerspezifischen Verhaltensmodellen (MBM) für die virtuelle Inbetriebnahme nutzbare UBM's zu erzeugen. Dies muss vom Anwender durchgeführt werden. Zu den Aufwänden hinsichtlich der Verknüpfung einzelner FMUs zu einem Gesamtanlagenmodell mittels Co-Simulation äußern sich die Autoren nicht.

4.2.1.3 Ansatz von Kiefer, 2007 (vgl. [KIE07])

Kiefer stellt einen stark vom mechatronischen Systementwurf beeinflussten Ansatz zur Planung automatisierter Fertigungszellen im Bereich Karosserierohbau vor. Der Ansatz ist als Hybrid zwischen *Digitaler Fabrik* einerseits und *mechatronischer Entwicklungsmethodik* andererseits zu sehen, wobei der Schwerpunkt auf der Implementierung des postulierten Vorgehens innerhalb der *Digitalen Fabrik* zu sehen ist. Kiefer identifiziert Defizite in Form einer heterogenen, von Insellösungen geprägten Umsetzung der *Digitalen Fabrik*, die insbesondere bei der Planung von automatisierten Anlagen auf Zellebene [ZÄWÜ05], sowie im Speziellen bei der Umsetzung von Simulation in einer virtuellen Inbetriebnahme zum Tragen kommen. Diese Defizite verhindern eine effiziente Durchführung des Entwicklungsprozesses. Die virtuelle Inbetriebnahme ist im Zusammenhang mit der angewendeten mechatronischen Entwicklungsmethodik als Übergang von der virtuellen zur realen Fabrik zu verstehen. Entsprechend wird der virtuellen Inbetriebnahme eine hohe Bedeutung beigemessen.

Ziel des Ansatzes ist es, eine „[...] frühzeitige und integrierte Betrachtung mechanischer, elektrischer, pneumatischer, hydraulischer und steuerungstechnischer Ressourcendaten im Sinne eines mechatronischen Entwicklungsvorgehens“ sicherzustellen. Der Schwerpunkt liegt daher auf der Bündelung von Daten wie Geometriemodellen, Daten aus der Elektroplanung, SPS-Codefragmenten und weiteren zu sogenannten „mechatronischen Ressourcenkomponenten“. Eine derartige Komponente kann einen einzelnen Roboter, aber auch eine einzelne Halterung beschreiben. Um eine zentrale Planungs- und Kommunikationsplattform zu etablieren, reichen mechatronische Ressourcenmodelle laut Kiefer [KIE07] nicht aus. Diese werden daher mit relevanten Produkt- und Prozessdaten zu einem sogenannten „mechatronischen PPR-Datenmodell“ kombiniert. Dieses entspricht weitgehend dem aus der *Digitalen Fabrik* bekannten PPR-Konzept, erweitert um die von Kiefer definierten Ressourcenkomponenten. Für die Schnittstellen (Datenimport wie -export) wird die Verwendung von XML-basierten Austauschformaten angeregt. Die Möglichkeit zur Simulation (insbesondere für die virtuelle Inbetriebnahme) ergibt sich laut Kiefer direkt aus dem PPR-Datenmodell heraus – es enthält alle für eine Simulation notwendigen Daten. Der prinzipielle Simulationsaufbau entspricht dem einer HIL-Simulation. Die Simulation enthält zudem in jedem Fall eine 3D-Visualisierung. Der eigentliche Aufbau des Simulationsmodells sowie Aspekte wie mögliche unterschiedliche Detaillierungsstufen oder unterstützende Methoden für die Durchführung einer Simulation werden nicht erörtert – die Granularität wie auch Aussagefähigkeit der Simulation ist rein abhängig vom Datenmodell. Die Simulation von Störungen soll durch interaktiven Eingriff gewährleistet sein, das Störverhalten muss entsprechend abgebildet sein. Ein praktischer Einsatz des von Kiefer beschriebenen Ansatzes erfolgt in [BKH*09].

Grenzen des Ansatzes von Kiefer

Die Grenzen des Ansatzes liegen in der Anwendbarkeit auch für große, deutlich komplexere Anlagen, die aus mehreren Fertigungszellen/Funktionsgruppen zusammengesetzt sein können. Die

von Kiefer betrachtete Zellebene besteht nur aus verhältnismäßig wenigen Komponenten. Wenngleich der Ansatz als Hybrid zwischen mechatronischer Entwicklungsmethodik und Digitaler Fabrik anzusehen ist, geht Kiefer nicht im Speziellen auf die Unterschiede und die Besonderheiten der beiden Methoden ein. Er differenziert nicht bei der Übertragung von Sichtweisen zwischen Entwicklungs- und Engineering-Ansätzen. Kiefer räumt in diesem Zusammenhang selbst ein, dass der Ansatz nur bei erfolgreichem organisatorischem Zusammenwirken von OEM und ihrer Zulieferern funktionieren kann. Die Schnittstellenproblematik zwischen verschiedenen Engineering-Werkzeugen wird zwar erkannt, aber nur im Speziellen für die im Ansatz verwendete Werkzeugkette betrachtet. Es bleibt unklar, wie KMU, die in der Regel nicht alle möglichen Engineering-Werkzeuge vorhalten können, auf die Daten zugreifen sollen. Wenngleich der Schwerpunkt der Arbeit auf der Definition eines Datenmodells liegt, werden Kriterien für die Güte der Daten (wie Vollständigkeit, Konsistenz, etc.), insbesondere im Hinblick auf die Verwendung für eine virtuelle Inbetriebnahme, nicht weiter betrachtet. Auf die eigentliche Simulation in Form einer virtuellen Inbetriebnahme geht Kiefer nur am Rande ein. Kiefer erkennt die Notwendigkeit eines adäquaten Detaillierungsgrads, klammert dies aber aus seiner Arbeit aus.

4.2.1.4 Ansatz von Wünsch, 2008 (vgl. [Wün08])

Wünsch stellt eine Sammlung unterschiedlichster Methoden für die virtuelle Inbetriebnahme vor. Er verfolgt insgesamt drei Ziele: Die Einführung von virtueller Inbetriebnahme in eine Organisation, die möglichst wirtschaftliche Skalierung einer virtuellen Inbetriebnahme sowie zuletzt die Vorstellung von Methoden zur Modellbildung. Im Folgenden wird auf die dargestellten Technologien für die Modellbildung fokussiert. Ziel ist es, einen Technologiebalken zu entwerfen, „[...] mit dessen Hilfe eine VIBN-Abteilung auf das Projektgeschäft vorbereitet werden kann“. Im Fokus steht nicht die Empfehlung einer konkreten Technologie oder die Etablierung eines bestimmten Ansatzes, sondern vielmehr Methoden oder besser Vorschläge, mit der aus verfügbaren Technologien zielgerichtet die richtige ausgewählt werden kann. Beispielhaft wird die Notwendigkeit eines skalierbaren Detaillierungsgrads („Konzept des Vergrößerungsglases“, [ZÄW05]) dargestellt. Die Aussagen beschränken sich darauf, dass z. B. CAD-Konstruktionsdaten die Basis für eine Visualisierung des Simulationsmodells sein können. Diese Daten müssen entsprechend in ein Simulationswerkzeug importiert werden. Fehlerzustände einer Anlage sollen abgebildet werden und werden von Wünsch in auftretende (technische) Störungen und dadurch bedingte steuerungstechnische Maßnahmen kategorisiert. Wie diese in der Simulation konkret umgesetzt werden können, beschreibt Wünsch allerdings nicht. Für den Entwurf von Simulationsmodellen werden Gestaltungsrichtlinien vorgestellt, ohne Simulationsmodelle gleichartiger Objekte in heterogenem Detaillierungsgrad, wie sie zuvor angesprochen wurden, zu beachten. Die konkrete Modellierung von Verhalten von mechatronischen Objekten oder die Definition eines Datenformats zur Ablage dieser Modelle werden von Wünsch nicht betrachtet.

Grenzen des Ansatzes von Wünsch

Wünsch bietet einen Überblick über unterschiedliche Methoden, die bei der Planung sowie Ausführung einer virtuellen Inbetriebnahme beachtet und durchgeführt werden können. Konkrete Lösungen oder unterstützende Methoden, um z. B. Schnittstellenprobleme zwischen verschiedenen Engineering-Werkzeugen generisch lösen zu können, werden nicht adressiert. Generell liegen das Datenmodell einer Anlage und die Modellierung desselben nicht im Fokus der Arbeit. So bleibt u. a. unklar, woher die Teilmodelle für die virtuelle Inbetriebnahme kommen sollen. Ein skalierbarer Detaillierungsgrad wird erwähnt, aber nicht bis zu den Auswirkungen auf die eigentli-

chen Simulationsmodelle und damit die Aussagekraft einer Simulation untersucht. Im Fokus steht der Umgang mit verschiedenen zur Verfügung stehenden Engineering-Artefakten auf abstrakt-organisatorischer Ebene.

4.2.2 Ansätze außerhalb der Digitalen Fabrik

4.2.2.1 Ansatz von Frieben/Schmüdderich et al., 2013 (vgl. [LSB12], [STB*13], [ScTR14] und weitere)

Im Rahmen des DFG Forschungsprojekts „*Methodik zur virtuellen Inbetriebnahme auf Basis von objektorientierten Verhaltensmodellen mit wählbarer Modellierungstiefe*“ [MVIBN[®]] entstand eine „*neue Methodik zur virtuellen Inbetriebnahme von maschinenbaulichen Anlagen auf Basis von objektorientierten Verhaltensmodellen*“. Der Schwerpunkt des Forschungsprojekts liegt in der Reduzierung des Modellierungsaufwands, was durch drei Maßnahmen erreicht werden soll [FSG*15]: einer Entwicklungsmethodik zur Integration der virtuellen Inbetriebnahme in den Prozess der Anlagenentstehung, einer Systematik zur Unterstützung des Entwicklers bei der Wahl der Modellierungstiefe für Verhaltensmodelle und eine durchgängige Entwicklungsumgebung zur Umsetzung der Methodik. Das grundlegende Vorgehen bei der Anlagenentwicklung ist nicht an die für die *Digitale Fabrik* und insbesondere der *Digitalen Fabrikplanung* vorgestellten Anforderungen an das Engineering von (fertigungstechnischen) Anlagen angelehnt. Vielmehr orientiert es sich an dem in [GSA*11] vorgestellten Vorgehen beim (modellbasierten) Entwurf von Produktionssystemen, der wiederum stark von der *ENTIME* genannten Entwicklungsmethodik für „*intelligente mechatronische Systeme*“ [ENTIME[®]] sowie generell der [VDI 2206-1] beeinflusst ist.

Virtuelle Inbetriebnahme wird als integraler Bestandteil des Anlagenentwicklungsprozesses gesehen und wird parallel zur beschriebenen Entwicklung der automatisierten Anlage durchgeführt [STB*13]. Der Lebenszyklus von fertigungstechnischen Anlagen wird im Vergleich zum Lebenszyklusmodell der *Digitalen Fabrik* gemäß [VDI 4499-1] vereinfacht als Konzeptionierung, Design, Montage/Inbetriebnahme, Betrieb und abschließender „*Redistribution [sic!]*“ gesehen [GSA*11]. Auf die Ansätze der *Digitalen Fabrik* gemäß [VDI 4499-1] wird kein Bezug genommen. Unterschiede in den Methoden, wie sie in Abschnitt 4.1.2 dargestellt wurden, werden daher in [GSA*11] nicht betrachtet.

Kern dieses Vorgehens gemäß [GSA*11] ist ein modellbasierter und über alle mechatronischen Domänen hinweg integrierter Entwicklungsansatz, der stark von der Wiederverwendbarkeit einzelner Teile („*Module*“) des Systems geprägt ist. Für den vorgestellten modellbasierten Systementwurf ist es zwingend notwendig, Modelle vorzusehen, die während des gesamten Systementwurfs zum Einsatz kommen. Die während des Anlagenentwicklungsprozesses erstellten Modelle werden mit laufender Projektdauer so weit verfeinert, bis am Ende das Produkt „*Anlage*“ entsteht. Die Wiederverwendbarkeit der Modelle wird zunächst durch eine anfangs durchgeführte Zerlegung der zu entwickelnden Anlage in Module („*modularization*“) erreicht. Diese Modularisierung ist hinsichtlich des Aspekts „*[...] der modularen Produktkonzeption im Sinne der möglichen Wiederverwendbarkeit*“ vorzunehmen [STB*13]. Auf die Granularität der einzelnen Module und damit abhängig der erstellten Modelle wird nicht genau eingegangen.

Das anschließende Zuweisen von (im Idealfall) bereits existierenden *Lösungsmustern* und abschließend *Lösungselementen* zu diesen Modulen sowie einer laufende Synthese/Analyse des Gesamtsystems komplettiert den Entwicklungsprozess. *Lösungsmuster* definieren sich als abstrak-

te Repräsentation einer Klasse von *Lösungselementen*, die bei der Entwicklung zum Einsatz kommen können oder bereits in vorhergehenden Projekten zum Einsatz gekommen sind. *Lösungselemente* entsprechen „[...] [bereits] realisierten und bewährten Lösungen“ [GSA⁺11], die sich für eine Wiederverwendung innerhalb des Anlagenentstehungsprozesses eignen. Existieren keine passenden *Lösungsmuster* oder *Lösungselemente*, müssen diese neu erstellt werden. Als Konsequenz aus dem vorgeschlagenen, modellbasierten Vorgehen liegt der Steuerungscode ebenfalls als Modell vor [ScTR14]. Es ist anzunehmen, dass auch der Steuerungscode oder besser das Modell des Steuerungscode einer Evolution während des Entwicklungsprozesses unterliegt, im Einzelnen wird darauf nicht eingegangen.

Für die verwendeten Modelle mechatronischer Objekte werden insgesamt vier Modellierungstiefen mit unterschiedlicher Aussagekraft definiert. Je nach fortschreitendem Entwicklungsprozess soll eine immer detailliertere Betrachtung und Analyse des Systems und im Speziellen des Systemverhaltens ermöglicht werden. Diese Modellierungstiefen werden wie folgt beschrieben [LSB12]: 1) *Idealisierte Funktionen*: Diese Modelle bilden die Funktionalität des Systems auf logische Weise ab und beinhalten lediglich die wichtigsten Zustände. Es wird kein Zeitverhalten abgebildet. 2) *Prinzipielle Machbarkeit*: Zusätzliches Abbilden des Zeitverhaltens. Modelle in dieser Modellierungstiefe können zusätzliches physikalisches Verhalten aufweisen. 3) *Systemspezifisches Verhalten*: Physikalisch/dynamische Modellierung. Nebeneffekte wie Reibung oder Verlustströme werden auf „einfache Weise“ berücksichtigt. 4) *Optimierung von Komponenten*: Physikalisch/dynamische Modellierung. Nebeneffekte wie Reibung oder Verlustströme werden „so detailliert wie möglich“ betrachtet. Die hinter den Modellierungstiefen stehenden sog. Modellklassen (wie Mehrkörpersimulationsmodelle, Finite-Element-Modelle, Zustandsautomaten, mathematische Modelle des Steuerungscode, etc.) werden in [SLT13] diesen vier Modellierungstiefen zugewiesen. Die Zuweisung erfolgt nicht eindeutig: So können z. B. Mehrkörpersimulationsmodelle in den Modellierungstiefen 2) bis 4) eingesetzt werden. Es erfolgt keine Definition, was eine Mehrkörpersimulation in welcher Modellierungstiefe im Konkreten leisten muss. Der Aufbau und die Einsatzmöglichkeiten bleiben im Unklaren. Eine Einordnung in Bezug auf betrachtete Freiheitsgrade oder mögliche Testszenarien wird nicht gegeben.

Lochbichler et al. präsentieren in [LSB12] eine Methode zur Identifizierung der notwendigen Modellierungstiefe. Kernaussage von Lochbichler et al. ist, dass, je weiter fortgeschritten der Entwicklungsprozess mechatronischer Komponenten ist, desto detaillierter müssen Modelle für das anforderungsgerechte Testen sein. Die benötigte Modellierungstiefe leitet sich aus den Anforderungen an die mechatronischen Objekte ab, die in jeder Phase des Entwicklungsprozesses neu zu definieren sind. Jede der vier postulierten Modellierungstiefen hat ihre eigenen Anforderungen, die mit den Anforderungen an das zu entwickelnde, mechatronische Objekt/System kompatibel sein muss. Die Auswirkungen nicht kompatibler Anforderungen werden nicht weiter betrachtet. Es ist anzunehmen, dass in diesem Fall neu modelliert werden muss.

Im Rahmen des vorgestellten Ansatzes wird die *virtuelle Inbetriebnahme*, wie bereits eingangs beschrieben, als entwicklungsbegleitende Methode gesehen, die von der Konzeptionierung über das Detail Design bis hin zum Start der eigentlichen Inbetriebnahme beim Auftraggeber zum Einsatz kommt [STB⁺13]. Grundlegend kann die virtuelle Inbetriebnahme in drei Phasen aufgeteilt werden: *Modelling*, *Realization*, *Analysis*. *Modelling* findet weitgehend während des eigentlichen Entwicklungsprozesses der Anlage statt, während *Realization* und schlussendlich *Analysis* erst unter der bereits laufenden Realisierung der Anlage vor Ort und unmittelbar vor der eigentlichen Inbetriebnahme vor Ort stattfinden. Die virtuelle Inbetriebnahme ist eng mit dem eigentlichen

Entwicklungsvorgang verzahnt; es findet eine laufende Synchronisierung beider Vorgehen (inkl. Iterationen, Rücksprüngen, etc.) statt.

Wie bei jeder virtuellen Inbetriebnahme sind Modelle der Anlage der Kern des Ansatzes. Zum Einsatz kommen hier Verhaltensmodelle, die aus den während des modellbasierten Entwicklungsvorgehens bereits erstellten Modellen abgeleitet werden [FSG⁺15]. Besteht das mechatronische Modul aus bereits existierenden *Lösungsmustern* oder *Lösungselementen*, können diese dort ohnehin bereits vorhandenen Verhaltensmodelle für die virtuelle Inbetriebnahme genutzt werden. Andernfalls sind diese neu zu erstellen. Diese Verhaltensmodelle können in unterschiedlicher Modellierungstiefe dargestellt werden – je nachdem, in welcher Phase man sich während des Entwicklungsprozesses befindet. Im konkreten basieren diese auf den bereits genannten vier unterschiedlichen Modellierungstiefen, die beim Systementwurf zum Einsatz kommen ([LSB12], [SLT13]). Für eine virtuelle Inbetriebnahme sind „[...] nur Modelle in den Tiefen idealisierte Funktion, prinzipielle Machbarkeit und systemspezifisches Verhalten notwendig“ [STB⁺13]. Die in [LSB12] definierte und mit dem höchsten Detaillierungsgrad verbundene Modellierungstiefe 4 kommt nicht zum Einsatz.

In [FSG⁺15] wird ferner eine Methode vorgestellt, wie sich die Dauer der Berechnung eines Simulationslaufs mithilfe einer „*adaptierbaren Modellierungstiefe*“ reduzieren lässt. Die als Modelica-Modelle vorliegenden Modelle von mechatronischen Modulen in unterschiedlicher Modellierungstiefe werden zu sogenannten *Einheiten* zusammengefasst. Ein dynamisches Umschalten der Modellierungstiefen basiert auf vorher definierten und in den Modellen hinterlegten „*Umschaltbedingungen*“. Mit Hilfe des Ansatzes konnte die Berechnungszeit eines (in Realzeit) 25 Sekunden langen Prozessablaufs von 78 Minuten Simulationszeit auf 36 Minuten Simulationszeit gesenkt werden. Diese virtuelle Inbetriebnahme soll gemäß [SGS⁺14] in einem geeigneten Simulationsverfahren (HIL, SIL, MIL, etc.) implementiert werden. Auf die hervorstechende Inkonsistenz, dass insbesondere während eines HIL-Ansatzes echtzeitfähige Modelle benötigt werden (wobei sich Echtzeit mit wenigen Millisekunden Berechnungsdauer pro Simulationsschritt definiert), was durch die verwendeten Modelle nicht geleistet werden kann [FSG⁺15], wird nicht weiter eingegangen.

Grenzen des Ansatzes von Friebe/Schmüdderich et al.

Der hier vorgestellte Ansatz beschreibt eine mögliche Vorgehensweise bei der Realisierung (fertigungstechnischer) Anlagen, die virtuelle Inbetriebnahme als entwicklungsbegleitenden Aspekt enthält. Sie bedingen eine hoch integrierte Entwicklungsumgebung, wie sie für mechatronische Systeme, die aus einer Hand geliefert werden, durchaus üblich ist. Im Gegenzug ist dies im Anlagen- und Maschinenbau aufgrund der Vielzahl an beteiligten Gewerken, die sich zudem aus verschiedenen beteiligten Unternehmen „*in projektspezifischer Zusammensetzung*“ [FAY09] zusammensetzen, nur schwer umsetzbar. Zusätzlich verlangt der beschriebene Ansatz nach hoch qualifiziertem Personal, welches in der Lage ist, den vorgestellten modellbasierten Entwicklungsansatz überhaupt umsetzen zu können.

Die definierten und während des modellbasierten Entwicklungsansatzes genutzten Modelle in verschiedenen Modellierungstiefen sind in ihrem Aufbau dem während Entwicklungsprozessen üblichen Schema „Vom Groben ins Detail“ geschuldet und nicht im Speziellen auf die Anforderungen einer virtuellen Inbetriebnahme zugeschnitten. Die Beschreibung der einzelnen Modellierungstiefen erfolgt eher unkonkret: Unklar bleibt, was genau mit einer Modellierung physikalischer Nebeneffekte auf „*einfache Weise*“ oder „*so detailliert wie möglich*“ gemeint ist. Zudem

können einzelne Modellarten (wie Mehrkörpersimulationsmodelle oder auch Finite-Element-Modelle) über verschiedene Modellierungstiefen hinweg zum Einsatz kommen. Es findet keine klare Abgrenzung der Möglichkeiten und Grenzen dieser Modelle im Sinne möglicher Testszenarien oder betrachteter Freiheitsgrade statt. Ebenso wenig wird über mögliche Funktionsumfänge der Modelle in den einzelnen Modellierungstiefen berichtet.

Störszenarien sollen zwar explizit betrachtet werden, und es wird eine Methode vorgestellt, wie sich nachzubildende Störszenarien für eine virtuelle Inbetriebnahme identifizieren lassen. Die Abbildung einzelner Störungen in den beschriebenen Modellen bleibt im Unklaren. Dies trifft insbesondere auf die „*einfachen*“ Modellierungstiefen zu.

Bei der Zuweisung von Modell zu Modellierungstiefen ergeben sich weitere Inkonsistenzen. Beispielsweise werden in [SLT13] der Modellierungstiefe 1 sog. „Gestaltmodelle“ (= detaillierte Konstruktionsmodelle) zugewiesen. Gemäß den Aussagen der Autoren folgend werden diese Modelle erst sehr spät bei der Entwicklung mechatronischer Systeme eingesetzt [FSG⁺15]. Die Wahl der Modellierungstiefe ist abhängig vom Fortschreiten des Entwicklungsprozesses [LSB12]. Anfangs stehen noch keine detaillierten Modelle zur Verfügung, die für die virtuelle Inbetriebnahme genutzt werden können – insofern ergibt sich die Inkonsistenz, in dem Sinn, dass nicht klar wird, wie Fortschritt im Entwicklungsprozess und Verwendung der dort zugewiesenen Modelle für die virtuelle Inbetriebnahme in Einklang zu bringen sind ([STB⁺13], [FSG⁺15]).

Die *Modelling* genannte Phase der Modellerstellung eines Anlagensimulationsmodells findet parallel zum *gesamten* Entwicklungsprozess statt [STB⁺13], wobei unklar ist, welcher Aufwand für die Erstellung des Anlagensimulationsmodells letztendlich notwendig ist. Ob diese Modelle bereits in frühen Phasen und in einer Art virtuellen Inbetriebnahme zum Einsatz kommen können, bleibt offen. Eine Aussage über das Aufwand/Nutzen Verhältnis einer virtuellen Inbetriebnahme kann daher nicht getroffen werden.

Virtuelle Inbetriebnahme im Sinne des Testens von Steuerungsfunktionen kann unter den beschriebenen Rahmenbedingungen daher nur in einer sehr späten Phase des Entwicklungsprozesses eingesetzt werden, in der Modelle in entsprechender Modellierungstiefe zur Verfügung stehen. Konkret soll die *Realisierung* und die *Analyse* einer virtuellen Inbetriebnahme während der bereits begonnenen Realisierung der Anlage vor Ort durchgeführt werden [STB⁺13]. Eine unter diesen Rahmenbedingungen durchgeführte virtuelle Inbetriebnahme erlaubt zwar das Auffinden von Fehlern im Modell des Steuerungscode, deckt aber nicht mehr die allgemein beschriebenen und anerkannten Ziele einer virtuellen Inbetriebnahme ([VDI 3693-1], [VDI 4499-2]) ab. Für Iterationen im Sinne der Behebung von Inkonsistenzen an der Mechanik und Elektrotechnik ist es bereits zu spät, da der Engineering-Prozess bereits zu weit fortgeschritten ist. Der als Modell vorliegende Steuerungscode erlaubt grundsätzlich nur eine virtuelle Inbetriebnahme unter einem *Model-in-the-Loop* Ansatz. Die Ableitung des Steuerungscode in Form von lauffähigem IEC 61131-3 Code aus dem Modell ist nicht Gegenstand der Betrachtung. Für Ansätze in einer HIL- oder auch SIL-Simulation, in [VDI 3693-1] als für die virtuelle Inbetriebnahme naheliegend definiert, sind die beschriebenen Modelle aufgrund ihrer zugrunde liegenden hohen Berechnungsdauer [FSG⁺15] und des lediglich als Modell vorliegenden Steuerungscode prinzipiell nicht geeignet.

Abschließend betrachtet liegen die Grenzen des Ansatzes in der praktischen Anwendbarkeit im derzeit üblichen Vorgehen bei der Planung und Realisierung einer Anlage innerhalb des *Digitalen Fabrikbetriebs*. Dies gilt im Speziellen unter Betrachtung eines heterogenen Projektumfelds mit vielen beteiligten Unternehmen/Gewerken. Es wird nicht zwischen im Anlagenbau üblichen Vor-

gehen und dem vorgestellten mechatronischen Entwicklungsprozess differenziert, die in Abschnitt 4.1 gezeigten prinzipiellen Unterschiede der Methoden werden weder angesprochen noch beachtet. Anforderungen im Speziellen an Modelle für eine virtuelle Inbetriebnahme werden nicht im Detail betrachtet. Die Definition der Modellierungstiefen weist zudem einige Inkonsistenzen und offene Punkte auf. Der Nutzen der definierten Modellierungstiefen außerhalb des vorgestellten mechatronischen Entwicklungsansatzes für Anlagen ist daher eher als eingeschränkt anzusehen.

4.2.2.2 Ansatz von Dominika und Heuschmann, 2007 (vgl. [DOHE07])

Dominika und Heuschmann stellen ein Konzept zur Prozessgutsimulation sowie zur Anlagenteilsimulation vor. Im Unterschied zu Ansätzen der virtuellen Inbetriebnahme, die die Simulation des Anlagenverhaltens zum Ziel haben, fokussiert der Ansatz der Prozessgutsimulation auf die ausschließliche Simulation des zu transportierenden Prozessguts auf der realen Anlage: Die Anlage wird nicht mit dem realen Prozessgut in Betrieb genommen, sondern mit rein Virtuellem. Technische Grundlage des Ansatzes stellt eine HIL-Simulationsumgebung dar, die die Abbildung des vollständigen und identischen peripheren Verhaltens (womit das Verhalten der projektierten steuerungstechnischen Elemente wie E/A-Baugruppen einer Anlage gemeint ist) notwendig macht. Den Aussagen der Autoren nach sind verschiedene Detaillierungsstufen für die Prozessgutsimulation denkbar. Um mit Situationen umgehen zu können, in denen die Anlage noch nicht in ihrer Gänze zur Verfügung steht, schlagen die Autoren die Umsetzung einer sog. Anlagenteilsimulation vor. Letztlich werden damit die noch fehlenden Teile einer Anlage mittels Simulation ergänzt.

Grenzen des Ansatzes von Dominika und Heuschmann

Der erarbeitete Ansatz ist als Alternative zum derzeit üblichen Vorgehen für die virtuelle Inbetriebnahme zu sehen. Die von den Autoren beschriebene Prozessgutsimulation kann nur an der realen Anlage durchgeführt werden: Es bleibt unklar, in welchen Fällen während der Errichtung einer Anlage diese bereits vorhanden, aber exklusiv für den Test mit virtuellem Prozessgut zur Verfügung steht. Auf Themenkomplexe wie notwendige Echtzeitfähigkeit, und damit eng zusammenhängender Fragestellungen wie z. B. der Genauigkeit einer Simulation, wird nicht eingegangen. Eine Methode, wie sich die von den Autoren als sinnvoll erachteten, unterschiedlichen Detaillierungsstufen für eine Simulation identifizieren lassen, wird nicht vorgestellt. Die von den Autoren beschriebene Anlagenteilsimulation soll die Anwendung des Ansatzes in Situationen, in denen die Anlage noch nicht vollständig zur Verfügung steht, ermöglichen. Vertieft dargestellt wird dies nicht.

Es erfolgt keine Erklärung der verwendeten Simulationsmodelle oder etwaig verwendeter Simulationsmodellbibliotheken, noch in welcher Granularität oder in welchem Detaillierungsgrad Simulationsmodelle vorliegen sollten. Woher die eingesetzten Modelle kommen bleibt daher im Unklaren. Fehlersituationen sollen einfach nachgestellt werden können. Die konkrete Umsetzung wie auch der Einfluss stochastischer sowie deterministischer Einflussgrößen auf Prozess- sowie auf technische Fehler wird nicht betrachtet.

4.2.2.3 Ansatz von Hoher und Verl, 2012 (vgl. [HOVE12])

Hoher und Verl stellen ein Konzept einer Multi-Simulator-Materialflusssimulation mit dem Schwerpunkt *virtuelle Inbetriebnahme* vor. Im Fokus steht die Sicherstellung der Simulation einer Materialflussapplikation im adäquaten Detaillierungsgrad in harter Echtzeit (definiert mit 16ms

Zykluszeit). Kern des Ansatzes und der ausgeführten Überlegungen ist es, die durchgeführte Simulation situationsbedingt im adäquaten Detaillierungsgrad durchzuführen. Hierfür wird zunächst in zwei grundlegende Detaillierungsstufen, mit denen ein Materialflusssystem abgebildet werden kann, differenziert: A) Einer rein „kinematischen Simulation“, bei der die der Bewegung zugrunde liegenden Kräfte nicht betrachtet werden sowie B) einer von den Autoren so genannten „physikalischen Simulation“, die die physikalischen Eigenschaften zur Berechnung der wirkenden Kräfte auf ein zu transportierendes Stückgut betrachtet. Entsprechende Simulationsmodelle werden in einer kinematischen und einer physikalischen, modularisiert aufgebauten Simulationsmodellbibliothek vorgehalten. Wesentlicher Bestandteil des Ansatzes ist es, Möglichkeiten aufzuzeigen, beide Simulationsansätze dahin gehend zu koppeln, so dass die harte Echtzeitfähigkeit der Simulation jederzeit gewährleistet werden kann. Ein dynamischer Modellwechsel zur Laufzeit ist nicht möglich, aber von den Autoren angedacht. Die dem Ansatz zugrunde liegende Simulationsplattform entspricht in ihrem Aufbau einer HIL-Plattform. Die Implementierung des Ansatzes erfolgt in einem hauseigenen, proprietären Simulationswerkzeug.

Grenzen des Ansatzes von Hoher und Verl

Der Ansatz von Hoher und Verl beschreibt keinen Ansatz im Sinne einer allgemein anwendbaren Methode. Vielmehr steht die technische Umsetzung der Kopplung verschiedener Simulationsmodelle, die in unterschiedlichem Detaillierungsgrad vorliegen, unter den Aspekten numerische Stabilität, optimiertem Rechenaufwand und Genauigkeit der Simulationsergebnisse im Vordergrund. Das Konzept weist bei näherer Betrachtung einige Restriktionen und offene Punkte auf. So muss die beschriebene Simulatorkopplung derzeit vollständig von Hand erstellt werden. Wie von den Autoren selbst eingeräumt, ist der Modellierungs- wie auch Simulationsaufwand daher als sehr hoch einzuschätzen. Die Granularität der verwendeten Simulationsmodelle und mögliche Unterschiede werden nicht näher betrachtet. Eine Betrachtung alternativer Detaillierungsstufen, außer den zwei genannten, wird nicht vorgenommen. Zudem findet keine Ausarbeitung statt, wann welcher Detaillierungsgrad angemessen ist. Der dadurch entstehende zusätzliche Aufwand, von den Autoren selbst wegen des zwangsweise durchgeführten Try-and-Error-Prinzips als sehr hoch beschrieben, wird als mögliche zukünftige Erweiterung des Ansatzes erkannt.

4.2.2.4 Ansatz von Spitzweg, 2009 (vgl. [Spi09])

Das Ziel des Ansatzes von Spitzweg ist die Entwicklung einer Methode für die Erstellung von Physikmodellen für einen integrierten Entwicklungs- und Konstruktionsprozess. Im Fokus der Arbeit stehen Anlagen und Maschinen, die mittels SPS gesteuert werden. Voraussetzung für den Erfolg beim Einsatz ist die schnelle und aufwandsarme Erzeugung der zum Einsatz kommenden Simulationsmodelle. Die eigentliche Simulation basiert auf einem Physikmodell und einer Visualisierung sowie der Koppelung derselben an kommerziell verfügbare Simulationswerkzeuge wie [WINMOD®]. Die als Datenbasis genutzten 3D-CAD Daten werden zunächst „simplifiziert“, um die Anzahl der Polygone zu reduzieren, und damit die Performance insgesamt zu erhöhen. Das entstandene sog. simplifizierte Modell wird mit physikalischen Eigenschaften angereichert, wodurch letztendlich ein (dreidimensionales) Physikmodell entsteht. Dieses Modell erlaubt die Betrachtung von Kollisionen zwischen Stückgütern. Implementiert wird das Simulationsmodell in eine proprietäre und eigens dafür entwickelte Virtual-Reality-Plattform, die mit handelsüblichen Simulationswerkzeugen (hier [WINMOD®]) verbunden ist. Der Aufbau entspricht einer typischen HIL-Plattform.

Grenzen des Ansatzes von Spitzweg

Zwingende Voraussetzung für den Ansatz von Spitzweg ist das Vorhandensein von 3D-CAD Daten. Woher die Informationen zur Anreicherung desselben zu einem Physikmodell kommen sollen, bleibt im Unklaren. Unterschiedliche Modellierungstiefen werden nicht betrachtet, der Aufwand zur Modellerstellung ist generell als hoch einzuschätzen. Die von Spitzweg beworbene teilautomatische Generierung beschränkt sich auf den Hinweis, dass in bestimmten Situationen teilautomatisiert Modelle generiert werden könnten.

4.2.2.5 Ansatz von Lacour, 2012 ([LAC12])

Lacour stellt einen Ansatz für die physikbasierte Simulation materialflussintensiver Produktionsanlagen vor. Schwerpunkt der Arbeit liegt auf der möglichst realistischen Darstellung des (physikalischen) Verhaltens von Produktionsanlagen, was mögliche Kollisionen von Stückgütern mit einschließt. Der Ansatz weist einige Gemeinsamkeiten mit dem Ansatz von Spitzweg [SPI09] auf. Im Grunde werden Methoden und Ansätze von [SPI09] detaillierter sowie der Modellerstellungsprozess insgesamt genauer betrachtet. Das Vorgehen zur Erstellung eines derartigen Simulationsmodells gliedert sich in insgesamt 5 Makrophasen: Die postulierte physikbasierte Simulation besteht aus mehreren miteinander interagierenden Bestandteilen und wird Schritt für Schritt mit Informationen angereichert.

Kern ist immer ein 3D-Modell, das aus den dafür zwingend vorhandenen 3D-CAD Daten abgeleitet werden muss. Die Simulationsmodelle sind in ihrer Detaillierung skalierbar: Die Darstellung von detaillierten 3D-CAD Modellen geht zulasten der Performance einer Simulation, so dass vereinfachte Geometrien zum Einsatz kommen. Das 3D-CAD Modell wird dazu in der ersten Phase in mehreren Schritten trianguliert, d. h. vereinfacht. In einem zweiten Schritt wird das triangulierte Modell in mehreren durchzuführenden Schritten zerlegt und ein sog. Kollisionsmodell abgeleitet. Dieses Kollisionsmodell wird wiederum mit physikalischen Parametern angereichert (Masse, Maße, Schwerpunkt, etc.). Das so entstandene Physikmodell muss in der letzten Phase noch kinematisiert werden, d. h. es werden Gelenke (Einschränkung der Bewegungsfreiheit) sowie die steuerungstechnische Einbindung von Motoren und Sensoren definiert. Als Simulationsplattform kommt ein eigens entwickeltes, proprietäres System zum Einsatz. Die eigentliche Umsetzung der Physik erfolgt mittels kommerziell verfügbarer Physik-Engines, die als Module in das verwendete Simulationswerkzeug eingebunden werden. Der Aufbau der Simulationsplattform entspricht einer HIL-Architektur.

Grenzen des Ansatzes von Lacour

Das von Lacour postulierte, auf dem Ansatz von Spitzweg [SPI09] basierende, mehrphasige Vorgehen bei der Erstellung physikbasierter Simulationsmodelle ist zum einen mit hohem Aufwand verbunden. Zum anderen erfordert das Vorgehen umfangreiches Wissen beim Umgang mit 3D-CAD Daten. Unterschiedliche Detaillierungsstufen werden von Lacour nicht betrachtet und beziehen sich rein auf die Abstrahierung von 3D-Geometrien. Unklar ist, wie Verhalten von z. B. Automatisierungstechnischen Geräten wie Umrichtern oder intelligenten Sensoren eingebunden wird und woher dieses Verhalten kommen soll. Der Import und Export von Daten in die verwendeten Engineering-Werkzeuge wird nicht im Speziellen betrachtet, für die Abbildung der 3D-Daten kommt das Datenformat [COLLADA[®]] zum Einsatz. Die Novität und Originalität des Ansatzes be-

steht in erster Linie darin, den Modellbildungsprozess im Vergleich zu [SPI09] deutlich detaillierter zu beschreiben. Zusammenfassend lässt sich folgern, dass die Grenzen des Ansatzes von Lacour in der Anwendbarkeit auch für kleinere und mittlere Unternehmen liegen.

4.2.2.6 Ansatz von Harrison et al., 2011 [HTY11]

Der Ansatz von Harrison et al. unterscheidet sich in seiner Sichtweise auf die Dinge deutlich von „klassischen“ Ansätzen zur virtuellen Inbetriebnahme. Steht bei der „deutschen“ Sichtweise auf die virtuelle Inbetriebnahme in den meisten Fällen – unabhängig davon, ob inner- oder außerhalb der *Digitalen Fabrik* – die Simulation der gesamten Anlage und der anschließende Transfer des vorab getesteten Steuerungscode auf die reale Anlage im Vordergrund (unabhängig von der dahinterstehenden Methodik), schlagen Harrison et al. eine „Hybride-Prozess-Simulation“ [HTY11] vor.

Kern des Ansatzes ist die Definition einer HIL-Testumgebung, die aus mehreren Komponenten besteht. Diese Komponenten können typische automatisierungstechnische Entitäten wie z. B. Förderbänder, Sensoren und Aktoren oder auch Roboter sein, die in ihrer Gesamtheit die betrachtete Anlage darstellen. Diese einzelnen simulierten Komponenten können nun, je nach Fortschritt der Montage vor Ort oder anderen Kriterien wie speziell definierter Testfälle (zu denen auch ein „Try-and-Buy“ und damit vertriebliche Aspekte gehören können), durch reale Komponenten ersetzt werden. Der Ansatz ist nicht nur auf den frühzeitigen Test des Steuerungscode zugeschnitten, sondern soll auch die Überprüfung der Leistungsfähigkeit des Systems (Taktzeiten, Optimierungspotenziale) ermöglichen. Ferner lassen sich Szenarien, in denen ein nachzurüstender Anlagenteil „on-the-fly“ während der laufenden Produktion in die bestehende Anlagenstruktur eingebettet werden soll, umsetzen. Denkbar ist es daher, dass der geänderte Steuerungscode an der realen Anlage in Betrieb genommen wird, wobei die neu hinzuzufügenden Anlagenteile zunächst simuliert werden. Zu den beispielhaft genannten Testszenarien gehört ferner ein sogenannter „dry-run“, bei der der automatisierte Ablauf einer realen Anlage ohne Stückgut durchgeführt wird. Die Sensorsignale werden mittels Simulation erstellt und an die Steuerung transferiert.

Explizit wird die Herausforderung der Stückgutübergabe zwischen simulierten und realen Anlagenteilen adressiert. Gelöst wird dies durch den Einsatz sogenannter „dualer“ Sensoren: Der gleiche Sensor ist in der Realität und in der Simulation vorhanden. Mittels Mapping des Sensorwertes von Simulation auf Realität und umgekehrt soll eine signaltechnisch saubere Implementierung gewährleistet sein. Welches Signal Priorität hat, wird nicht genau betrachtet. Dieser Teilaspekt weist von der grundlegenden Denkweise (Vermischung realer und simulierter Anlagenteile, Übergabe des Stückguts von simulierten auf reale Anlagenteile und zurück) Ähnlichkeiten zu dem in Abschnitt 4.2.2.3 behandelten Ansatz von [HOVE12] auf, ist in seiner Gesamtheit aber deutlich ausführlicher und konsistenter beschrieben. Der Ansatz von Harrison et al. stellt ferner eine umfassende und in sich konsistente Ontologie bereit, die den Einsatz und die Einführung der Methode unterstützt.

Der Ansatz stellt bezüglich des Umfangs „nur“ eine formalisierte Methode zur Durchführung einer Hybriden Prozess-Simulation vor. Hinweise oder Erläuterungen hinsichtlich des für die Durchführung benötigten Aufwands (Implementierung, Modellerstellung) sowie der für eine Umsetzung notwendigen Datenquellen und Informationen anderer am Engineering beteiligter Gewerke werden nicht gegeben, die Notwendigkeit dafür aber erkannt.

Grenzen des Ansatzes von Harrison

Der Ansatz von Harrison et al. kann eine Möglichkeit darstellen, virtuelle Inbetriebnahme, wie sie im deutschsprachigen Raum verstanden wird, durchzuführen. Er ermöglicht ferner das Durchführen multipler Anwendungsfälle, die über eine „im deutschsprachigen Raum übliche“ virtuelle Inbetriebnahme, wie sie in Kapitel 3 und bisher in Kapitel 4 beschrieben wurden, hinausgehen. Die Durchführung dieser zusätzlich möglichen Anwendungsfälle (try-and-buy, dry-run, etc.) bedingt naturgemäß ein hohes Maß an Einbindung und vor allem auch Akzeptanz nicht nur des Auftraggebers, sondern aller am Engineering beteiligten Gewerke. Hier liegt das Potenzial, gleichzeitig aber auch die größte Schwäche der vorgestellten Methode: Eine enge Zusammenarbeit der am Engineering beteiligten Gewerke ist de facto nicht üblich, vielmehr steht das parallele Bearbeiten gewerkespezifischer Themenkomplexe im Vordergrund. Die Durchführung einer virtuellen Inbetriebnahme erfolgt im deutschsprachigen Raum de facto alleinig getrieben durch das Gewerk Automatisierungstechnik, mit *möglicherweise* stattfindender Unterstützung der anderen am Engineering beteiligten Gewerke und des Auftraggebers. Die hier vorgestellte Sichtweise widerspricht dieser Vorgehensweise, so dass ohne grundlegende Änderungen am Engineering insgesamt eine Anwendung nicht realistisch erscheint. Bezüglich des aufzuwendenden Aufwands bei der Implementierung der Simulation sind die Aufwände als sehr hoch einzuschätzen – die potenziell möglichen nutzbringenden Effekte werden so relativiert.

4.3 Ansätze mit teil-automatischer Modellgenerierung

Ansätze mit teil-automatischer Modellgenerierung beschreiben Vorgehen, bei denen das Simulationsmodell nicht vollständig im Sinne von *ohne Nacharbeit sofort im Simulationswerkzeug einsetzbar* ist. Stattdessen werden Teilaspekte des Simulationsmodells automatisch generiert, die manueller Nacharbeit und Ergänzung bedürfen. Damit nicht gemeint sind ggf. zusätzlich notwendige, automatisiert ablaufende Schritte wie die Überführung in ein Runtime-fähiges Format innerhalb des Simulationswerkzeugs.

4.3.1 Ansatz von Lindworsky, 2011 (vgl. [LIN11])

Um den hohen Aufwand bei der Erstellung von Simulationsmodellen für die virtuelle Inbetriebnahme zu reduzieren, schlägt Lindworsky eine Reihe von Maßnahmen vor, die den Entwicklungsprozess insgesamt effizienter gestalten sollen. Die Modellbildung und Simulation sollen zum einen möglichst früh in den Entwicklungsprozess von automatisierten Anlagen integriert und entwicklungsbegleitend durchgeführt werden. Andererseits sollen die für die virtuelle Inbetriebnahme genutzten Modelle teilautomatisiert abgeleitet werden und relevante Störszenarien integriert werden.

Die datentechnische Basis zur Ableitung von Simulationsmodellen für die virtuelle Inbetriebnahme stellt ein sogenanntes Funktionsmodell dar, das während der Planungs- und Projektierungsphase erstellt wird. Es enthält sowohl Schnittstelleninformationen als auch das Maschinenverhalten. Ferner können bereits im Funktionsmodell Störszenarien spezifiziert werden. Das Funktionsmodell soll darüber hinaus die Eingangsinformation für die „disziplinspezifischen Entwicklungsstränge“ bilden, mit dem Ziel, entwicklungsbegleitend als Abstimmungs- und Kommunikationsmittel zu dienen. Aus dem Funktionsmodell werden folglich sog. Mechanik-CAD Modelle, Elektro-CAD Modelle und letztlich auch das Modell für die virtuelle Inbetriebnahme abgeleitet. Technische Grundlage des Funktionsmodells ist das im (stark von Sichtweisen der Informatik geprägten)

Forschungsprojekt *AUTOVIBN* [BHH⁺10] entwickelte Metamodell für die Modellierung von Anlagenstrukturen. Für die genaue Betrachtung der dort beschriebenen Abbildungsregeln sei auf den zitierten Projektbericht verwiesen.

Das Prinzip der (teil-)automatischen Modellgenerierung lässt sich (stark vereinfacht) wie folgt zusammenfassen: Aus einem Plattform-Independent-Model (PIM, sinngemäß dem Funktionsmodell) wird das Virtuelle-Inbetriebnahme-Modell implementiert (als lauffähiger C++ Code). Das Simulationssystem setzt sich letztlich aus den Teilen Simulationsmodell (C++ Code), Visualisierung (abgeleitet aus dem Funktionsmodell) sowie Bedienung zusammen. Die Implementierung erfolgt in einer proprietären Simulationsplattform, die als HIL- oder SIL-Simulation für die Durchführung einer virtuellen Inbetriebnahme genutzt werden kann. Im Ansatz werden unterschiedliche Detaillierungsstufen für Simulationsmodelle betrachtet – in Hinblick auf die zur Verfügung stehende Güte der Daten im Funktionsmodell. Am Anfang des Entwicklungsprozesses stehen entsprechend grob-granulare Daten zur Verfügung, so dass nicht sehr detailliert simuliert werden kann. Je mehr und genauere Daten (von der Grobplanung hin zur Feinplanung) zur Verfügung stehen, desto genauer werden das Funktionsmodell und folglich auch die Simulation. Ziel ist immer die möglichst genaue und exakte Simulation.

Wie das physikalisch/dynamische Verhalten im konkreten abgebildet wird, beschreibt Lindworsky nicht. Die Grenzen von teilautomatischer zu vollautomatischer Simulationsmodellerstellung liegen in den manuellen Aufwänden begründet, die für die Erstellung des Funktionsmodells zu erbringen sind. Eine weitere Einschränkung sind einzuhaltende Konstruktionsrichtlinien, um einen automatisierten Import von CAD-Daten in das Funktionsmodell zu ermöglichen. Die Erstellung des eigentlichen Simulationsmodells respektive des generierten C++ Codes, der für die Durchführung einer virtuellen Inbetriebnahme genutzt wird, erfolgt stets vollautomatisch.

Grenzen des Ansatzes von Lindworsky

Die Grenzen der Anwendbarkeit des Ansatzes liegen bei Szenarien, in denen eine andere Datenbasis als das vorgestellte Funktionsmodell zum Einsatz kommt. Der Ansatz ist im Speziellen auf die Denkweise *PIM* → *Implementierung in ausführbarem Code* ausgelegt und daher wenig generisch. Es gibt keinen eigentlichen Modellgenerierungsalgorithmus, der für die Verwendung mit multiplen Datenformaten adaptiert werden könnte. Vielmehr ist die Modellgenerierung eine Konsequenz aus dem (dem AUTOVIBN-Ansatz zugrunde liegenden) Modellierungsgedanken für Anlagenstrukturen. Detaillierungsstufen werden von Lindworsky mit Hinblick auf die Simulation nicht im Speziellen betrachtet, das Simulationsmodell wird immer so genau wie möglich aus der vorhandenen Datenbasis abgeleitet – die am Anfang naturgemäß weniger detailliert sein kann als zu einem späten Zeitpunkt des Entwicklungsprozesses. Wie vom Autor selbst eingeräumt, ist ein Nachvollziehen der einzelnen Bestandteile des Simulationsmodells innerhalb des generierten C++ Codes kaum möglich. Bei gewünschten Änderungen im Simulationsmodell müssen diese folglich im Funktionsmodell vorgenommen werden, aus dem dann wiederum ein neues Simulationsmodell generiert wird. Ein Debugging des Simulationsmodells ist kaum möglich, was die Lesbarkeit und auch die Validierungsmöglichkeiten des Simulationsmodells stark einschränkt.

4.3.2 Ansatz von Böhme et al., 2009 (vgl. [BKS⁺09])

Der Ansatz von Böhme et al. beschreibt ein Baukastensystem für die virtuelle Inbetriebnahme, der stark von einem „Virtual Engineering“ Ansatz und der Anwendung von „Virtual Reality“ geprägt ist. Der Weg des hohen Modellierungsaufwands der zum Einsatz kommenden Simulations-

modelle soll durch Modellgenerierung mittels Modellübersetzung gelöst werden. Hierfür werden kinematische Modelle wie auch Daten aus elektrischen Schaltplänen in die Modellierungssprache [MODELICA[®]] überführt.

Ziel des Ansatzes ist es, Mehrkörpersimulationsmodelle (MKS-Modelle) aus CAD-Zeichnungen für das virtuelle Engineering von Sondermaschinen (im Konkreten CNC-Universalbearbeitungszentren für Großteile) herzuleiten. Als ein Problem wird die Ergänzung der CAD-Daten um Informationen, die dort üblicherweise nicht vorliegen, identifiziert – als Beispiele werden für die Simulation benötigte Motorenmodelle genannt. Der Übersetzungsprozess besteht daher aus zwei Teilen: 1) Der Erzeugung eines kinematisch strukturierten, visualisierungsfähigen Zwischenformats direkt aus den CAD-Daten heraus. 2) Dem Hinzufügen und Parametrieren von Antriebselementen in das in Schritt 1) erzeugte 3D-Modell. Dies muss in einem geeigneten Simulationswerkzeug stattfinden. Anschließend erfolgt die Generierung eines Simulationsmodells. Um die Durchgängigkeit der entwickelten Lösung sicherzustellen, wird ein proprietäres XML-basiertes Zwischendatenformat definiert (XWS genannt). Verwendete Engineering-Werkzeuge (3D-CAD Tools, Simulationswerkzeuge) müssen dieses Datenformat unterstützen. Eine Umsetzung von Schritt 1) wird beispielhaft mittels eigens für den Ansatz entwickelter Plug-Ins für [PROENGINEER@] gezeigt. Das verwendete, proprietäre Visualisierungswerkzeug RobotMax unterstützt XWS nativ. Weitere Informationen zum Zwischenformat oder Aussagen, welche Simulationswerkzeuge XWS noch unterstützen, werden nicht gegeben.

Daten über das Verhalten der elektrischen Komponenten werden aus EPLAN extrahiert und enthalten Parametersätze wie auch die Verbindungen zwischen den einzelnen Komponenten. Eine automatisierte Ableitung ist nur dann möglich, wenn alle in EPLAN abgebildeten Bauteile als Modelica-Modelle in einer Modellbibliothek verfügbar sind. Die Simulationsumgebung besteht aus verschiedenen Werkzeugen, die zur Kommunikation untereinander mittels RTI-Interfaces verbunden sind. Die Durchführung der virtuellen Inbetriebnahme erfolgt im Kontext von Virtual Reality Anwendungen, mit dem Ziel, dem Bediener möglichst frühzeitig die Einweisung an einer Maschine zu ermöglichen.

Grenzen des Ansatzes von Böhme et al.

Der Fokus der Arbeit liegt auf einer möglichst realistischen Darstellung der erzeugten Visualisierung zur Bedienschulung. Weniger betrachtet wird die Anwendung typischer Testszenarien für die virtuelle Inbetriebnahme, wie sie in Abschnitt 3.2.1 beschrieben wurden. Die möglichen positiven Nutzeffekte einer virtuellen Inbetriebnahme im von den Autoren gegebenen Kontext sind daher unklar. Auf die eigentliche Simulation und verwendete Simulationsmodellbibliotheken wird nicht näher eingegangen. Die mögliche Nachbildung von Fehler- und Störszenarien wird nicht betrachtet. Unterschiedliche Detaillierungsstufen werden nicht vorgestellt, vielmehr wird so detailliert wie möglich simuliert. Die Kopplung vieler unterschiedlicher Werkzeuge zur Integration von Visualisierung, Simulation, Steuerung und anderen impliziert einen hohen Synchronisations- wie Erstellungsaufwand, der nicht angesprochen wird. Die Grenzen des Ansatzes liegen daher einerseits in der Anwendbarkeit der Methode, andererseits im Nutzen auch für kleinere und mittlere Unternehmen.

4.3.3 Ansatz von Meyer, 2014 (vgl. [MEY14])

Meyer stellt ein Rahmenwerk für die wirtschaftliche Erstellung von Emulationsmodellen für die virtuelle Inbetriebnahme vor. Meyer spricht bei einer virtuellen Inbetriebnahme stets von „Emulation“ statt „Simulation“ – wie schon der Titel seiner Arbeit zeigt. Mit Emulation ist die Gesamtheit eines Tests bei einer virtuellen Inbetriebnahme gemeint, der Simulation als *einen* Bestandteil beinhalten kann. Die Verwendung der Begriffe Emulation und Simulation kann in der vorgestellten Arbeit als heterogen bezeichnet werden. Globales Ziel des Ansatzes ist es, die aufwändige Modellerstellungsphase durch eine (teil-) automatische Generierung des zur virtuellen Inbetriebnahme genutzten Emulationsmodells gering zu halten, und so die Wirtschaftlichkeit einer virtuellen Inbetriebnahme zu optimieren.

Ein wesentlicher Aspekt ist hierfür die Wiederverwendung bereits existierender Simulationsmodelle, vor allem der in frühen Projektphasen durchgeführten Materialflusssimulation. Durch die Kombination einer Anlagensimulation mit einer Materialflusssimulation entsteht eine sog. „Integrierte virtuelle Inbetriebnahme“, deren Kernidee die Verknüpfung unterschiedlicher Abstraktionsniveaus ist. Das Ziel ist nicht, *eine* Simulation in unterschiedlichen Abstraktionsniveaus aufzubauen, die Fragenstellungen in mehreren Detaillierungsstufen beantworten kann. Vielmehr steht die Verknüpfung *unterschiedlicher Simulationen* (Anlagensimulation *und* Materialflusssimulation) im Vordergrund, so dass jede einzelne Simulation die Fragestellungen des jeweiligen Abstraktionsniveaus bedienen kann. Hierfür ist der Einsatz unterschiedlicher Simulatoren notwendig. Begründet wird dies mit Fragestellungen, die auf unterschiedlichen Hierarchiestufen auftreten können, im Konkreten: der Feldebene (SPS, Robotersteuerung) und der Prozessleitebene.

Grenzen des Ansatzes von Meyer

Die Grenzen der Arbeiten von Meyer liegen in der Aussagefähigkeit der genutzten Emulationsmodelle. Die Nachbildung physikalischer Dynamik kann im vom Meyer beschriebenen Rahmenwerk aus Simulationswerkzeug und Aufbau der verwendeten Simulationsmodelle nicht betrachtet werden. Zudem beschränkt sich die verwendete Datenbasis auf die üblicherweise bei der Errichtung von Anlagen in der Automobilindustrie verfügbaren Planungsdaten, im Speziellen auf eine Materialflusssimulation. Der Detaillierungsgrad wird von Meyer zwar betrachtet, allerdings nur zu dem Zweck, Fragestellungen, die für unterschiedliche Abstraktionsniveaus typisch sind (im Konkreten Feldebene/Prozessleitebene), beantworten zu können. Im Vordergrund steht nicht die Aussagefähigkeit der Emulation hinsichtlich statischer/dynamischer Effekte, die beim Betrieb der Anlage auftreten können.

4.4 Ansätze mit voll-automatischer Modellgenerierung

Ansätze mit voll-automatischer Modellgenerierung beschreiben Vorgehen, bei denen das generierte Simulationsmodell *ohne Nacharbeit sofort im Simulationswerkzeug einsatzbereit* ist. Nicht unter diese Konvention fallen ggf. zusätzlich notwendige, automatisiert ablaufende Schritte wie Generierung eines Runtime-fähigen Formats innerhalb des Simulationswerkzeugs.

4.4.1 Ansatz von Kufner, 2011 (vgl. [KUF11])

Der von Kufner vorgestellte Ansatz fokussiert auf die automatisierte Erstellung von Maschinenmodellen für die HIL-Simulation von Montagemaschinen. Kufner orientiert sich beim Vorgehen

zur Entwicklung von Montagemaschinen an der aus der [VDI 2206-1] bekannten und in Abschnitt 4.1 im Kontext des Anlagenbaus kritisch gewürdigten *Entwicklungsmethodik für mechatronische Systeme*.

Eine Definition der Granularität der betrachteten Maschinen wird unter dem Paradigma der Modularisierung (u. a. [BJB*03]) getroffen. Eine Montagemaschine kann aus n Modulen bestehen. Ein Modul wiederum kann aus $n \geq 1$ Komponenten bestehen, die sich wiederum aus $n \geq 0$ Bauteilen oder maximal einer Baueinheit zusammensetzen. Ein Bauteil entspricht laut Definition genau einem Sensor oder Aktor, während eine Baueinheit mehrere Sensoren und Aktoren beinhalten kann. Die hier beschriebenen Objekte müssen sich in den Bibliotheken der jeweilig verwendeten Engineering-Werkzeuge wiederfinden. Bei der Abbildung der Struktur in einem Datenmodell ist ein wichtiger Aspekt die Vollständigkeit des Maschinenmodells, das von Kufner wie folgt definiert wird: „Das Maschinenmodell ist dann vollständig, wenn es aus Sicht der Steuerung nicht von der realen Maschine zu unterscheiden ist“. Dies bedeutet im Konkreten, „[...] dass die gesamte Adressbreite des Eingangs- und des Ausgangsabbildes der Maschine modelliert [werden muss]“. Dies ist von größter Wichtigkeit, da das E/A-Abbild der einzelnen Bauteile/Baueinheiten die Basis des zu erstellenden Maschinenstrukturmodells darstellt, welches später für die automatische Modellgenerierung genutzt wird.

Die eigentliche Modellgenerierung basiert auf einem mehrstufigen Konzept, das hier verkürzt beschrieben werden soll: Zunächst werden Struktur (d. h. Aufbau) einer Maschine und Verhalten von Bauteilen und Baueinheiten voneinander getrennt betrachtet. Die Struktur einer Maschine wird, genauso wie das Verhalten der einzelnen Bauteile, automatisiert generiert. Basis für die Generierung der Struktur der Maschine stellen die Buskonfiguration und die erstellte E/A-Liste dar. 3D-CAD-Daten des Gewerks Mechanik und damit auch die dort hinterlegten Parameter mechanischer Grundsysteme werden nicht betrachtet. Module und Komponenten ergeben sich aus den E/A-Adressbereichen der beinhalteten Bauteile/Baueinheiten. Basis für die Generierung des Verhaltens von Bauteilen stellen wiederum sog. Basisfunktionen dar. Die Datenquelle für diesen Generierungsschritt ist der während des Engineerings erzeugte Stromlaufplan sowie eine Bibliothek mit den Basisfunktionen. Ergebnis sind sog. Blockschaltbilder des Komponentenverhaltens, die das Verhalten in einer definierten Modellierungsstufe wiedergeben (siehe letzter Absatz) und durch ihren E/A-Bereich definiert sind.

Die Zuordnung von Komponenten-Verhaltensmodellen zu in der Maschinenstruktur definierten Komponenten (und damit die eigentliche Simulationsmodellgenerierung) findet in einem Modellierungstool statt, welches als Datenquelle wiederum den Stromlaufplan nutzt. Die konkrete Zuordnung basiert auf identischen E/A-Adressbereichen. Ergebnis ist das sog. Maschinenmodell. Das Maschinenmodell ist nicht immer bereits vollständig parametrisiert. Nicht vorhandene Parameter, wie z. B. Verzögerungen bei Übergangszeiten, müssen anschließend von Hand im Maschinenmodell nachgetragen werden. Streng genommen handelt es sich also um eine teil-automatische Generierung. Das erstellte Maschinenmodell wird, als finaler Schritt, in ein geeignetes Simulationswerkzeug importiert. Als Simulationssoftware wurde hier die Software Virtuos [VIRTUOS®] gewählt. Auch hier ist eine Nachbearbeitung notwendig: Verhaltensmodelle von Komponenten müssen manuell an das im Simulationswerkzeug vorhandene Geometriemodell angebunden werden – eine Konsequenz aus der Nicht-Beachtung der 3D-CAD-Daten. Ob dieser Schritt immer notwendig ist und das Vorhandensein eines Geometriemodells stets Voraussetzung ist, wird nicht weiter betrachtet.

Unabhängig von der eigentlichen Modellgenerierung definiert Kufner fünf verschiedene Detaillierungsstufen, die Komponenten im Maschinenmodell annehmen können. In Modellierungsstufe 1 („Logisches Abbild“) wird nur das logische Verhalten der Maschine beschrieben, zeitbasiertes Verhalten wird nicht betrachtet. Die Modellierungsstufen steigen in ihrer Komplexität und Aussagekraft schrittweise an (erst durch das Hinzufügen von zeitbasiertem Verhalten, dann durch das schrittweise Ergänzen von physikalisch/dynamischem Verhalten). In Modellierungsstufe 5 („Physikalisches Zustandsabbild“) wird schließlich eine exakte Kopie des physikalischen Verhaltens gefordert, bis hin zu Einflüssen, *„welche durch eine fehlerhaft gefertigte Einzelkomponente oder durch eine fehlerbehaftete Montage sowie durch Verschleiß und Verformung entstehen“*. Fehlerzustände sind der höchsten Modellierungsstufe vorbehalten, in anderen Modellierungsstufen findet entsprechend keine Betrachtung derartiger Zustände statt. Kufner schränkt den von ihr aufgespannten Lösungsraum für Modellierungsstufen sogleich bewusst ein: Modellierungsstufe 4 und 5 sind insbesondere bei der Modellierung der Werkstückbearbeitung von Bedeutung, die für die eigentliche Simulation der Montagemaschine aber nur eine untergeordnete Rolle spielen und daher nicht weiter fokussiert werden. Die virtuelle Inbetriebnahme konzentriert sich auf Modellierungsstufe 1, 2 und 3, in (nicht betrachteten) Ausnahmefällen auf Modellierungsstufe 4.

Grenzen des Ansatzes von Kufner

Rein simulationsseitig liegen die Grenzen insbesondere in der Aussagekraft der definierten Modellierungsstufen und damit in den für eine virtuelle Inbetriebnahme wesentlichen Simulationsmodellen. So werden z. B. in Modellierungsstufe 3 Zusammenhänge zwischen physikalischen Größen mittels Definition durch *„Faustformeln“* abgeschätzt. Was unter Faustformeln verstanden wird, bleibt ebenso offen wie die Frage nach den Auswirkungen auf die Aussagefähigkeit der Simulation an sich. Eine Methode oder Systematik, wann welche Modellierungsstufe angemessen erscheint, wird nicht vorgestellt. Eine heterogene Abbildung des Maschinenmodells scheint generell nicht vorgesehen zu sein.

Hinsichtlich der Modellgenerierung werden das zugrunde liegende Datenmodell und zugehörige Modellierungsvorschriften zwar umfangreich beschrieben, es bleibt aber unklar, wie diese im Zusammenhang mit einem derzeit etablierten Engineering-, generell aber auch einem Entwicklungsprozess zu sehen sind. Die Granularität der Daten wie z. B. die Einteilung in Module, Komponenten, Bauteile, Baueinheiten und anderen ist nicht immer eindeutig zuzuordnen. Das Datenmodell und die dort beinhaltete Maschinenstruktur ist im Speziellen für die automatische Modellgenerierung entworfen worden. Letztlich muss sich das gesamte Engineering als auch die Entwicklung an der automatischen Modellgenerierung orientieren, wie von Kufner selbst eingeräumt wird: *„[Der Ansatz] verlangt tief greifende, langfristige Änderungen im Engineeringprozess“*. Alternative, von Kufner als effizient beschriebene Modellierungsmethoden, wie z. B. das baukastenbasierte Erstellen von Simulationsmodellen, werden als nicht wirtschaftlich bezeichnet. Dies widerspricht sich. Der Ansatz ist nicht generisch anwendbar: Simulationsmodellbibliothek und hinterlegte Basisfunktionen im Datenmodell sind identisch und müssen bei Wechsel des Simulationswerkzeugs stets neu aufgebaut werden oder anhand der im (neuen) Simulationswerkzeug vorhandenen Bibliothek synchronisiert werden. Die Auswirkungen unterschiedlicher Granularität werden nicht betrachtet. Die Herausforderung der Sicherstellung der Interoperabilität aufgrund potenziell unterschiedlich auszugebender Datenformate durch das Modellierungstool (je nach verwendetem Simulationswerkzeug) wird nur dahin gehend adressiert, dass bei Inkompatibilitäten schlicht ein anderes Modellierungstool benötigt wird.

4.4.2 *Ansatz von Barth, 2011 (vgl. [BAR11])*

Barth stellt einen umfassenden Ansatz zur vollautomatischen Generierung von Simulationsmodellen für die virtuelle Inbetriebnahme verfahrenstechnischer Anlagen vor. Der Ansatz zielt nicht auf die Nachbildung des genauen Verhaltens einer Anlage ab (es werden z. B. keine chemischen Reaktionen betrachtet), sondern soll das vorzeitige Testen des erstellten Prozessleitsystems (PLS) mittels einer virtuellen „Wasserfahrt“ [WEB02] ermöglichen. Die Modellgenerierung basiert auf zwei potenziell verfügbaren Datenquellen, die beim Engineering verfahrenstechnischer Anlagen zur Verfügung stehen: 1) CAE-Planungsdaten, maßgeblich dem Rohrleitungs- und Instrumentenfließbild (nachfolgend R&I-Fließbild) und 2) PLS-Engineering-Daten. Barth folgert insgesamt drei Konzeptalternativen, die sich aus den zur Verfügung stehenden Datenquellen und der Kombination daraus ergeben. Die Datenkonsistenz und Verfügbarkeit muss je nach Alternative ebenso sichergestellt sein wie der gleiche Bearbeitungsstand beider Datenquellen. Je nach gewählter Konzeptalternative unterscheiden sich zudem die von Barth identifizierten Konzeptkomponenten: Modellierungsebene (Netzwerkebene, Systemebene, [LHM⁺99]), Datenaustausch (z. B. mittels CAEX [IEC PAS 62424]), Modellierungsansatz (kausal/a-kausal) und Visualisierung. Am Beispiel der von Barth postulierten Konzeptalternative K1 sei dies verdeutlicht: Als Modellierungsebene wird dort die Netzwerkebene gewählt. Diese enthält Parameter wie Behälterdurchmesser, Stützenquerschnitt und weitere. Ferner erfolgt der Datenaustausch mittels eines objektorientierten, offenen Datenaustauschformats. Der gewählte Modellierungsansatz ist dazu passend eine objektorientierte Modellierung. Das Ziel ist die Übernahme von Objekten aus dem CAE-System in das Simulationssystem. Die Visualisierung ist an das R&I-Fließbild angelehnt.

Die von Barth für die Simulation eingesetzte Modellierungsumgebung ist [MODELICA[®]] und im Speziellen die Modelica-Fluidstandardbibliothek. Die Modellgenerierung profitiert in Konzeptvariante K1 von der durchgängigen Objektorientiertheit der verwendeten Planungsdaten wie auch der Modellierungsumgebung. Letztendlich erfolgt die Modellgenerierung durch die Transformation von objektorientierten Planungsdaten zum objektorientierten Simulationsmodell. Unterstützt wird dies durch manuell erstellte Mappingtabellen, in denen die in den Datenquellen vorhandenen Parameter auf die in der Simulationsmodellbibliothek enthaltenen Parameter abgebildet sind. Laut Barth ist es für den Steuerungstest nicht notwendig, alle physikalischen Effekte einer Anlage abzubilden. Die verwendeten Simulationsmodelle können daher auf das Wesentliche abstrahiert werden, was an Beispielen gezeigt wird. Eine systematische Einordnung in mehrere Detaillierungsstufen, je nach Testszenario, wird von Barth nicht vorgenommen. In den anderen Konzeptalternativen ist eine automatische Generierung aufgrund fehlender Daten nicht immer möglich.

Grenzen des Ansatzes von Barth

Die Grenzen der Arbeiten von Barth liegen simulationsseitig in der Aussagekraft der abstrahierten Simulationsmodelle: unterschiedliche, systematisierte Detaillierungsstufen werden nicht betrachtet. Eine genaue Definition, wann welcher Detaillierungsgrad angemessen erscheint, wird nicht gegeben. Situationen, in denen dynamisches Verhalten einer Anlage von Interesse sein kann, können durch die Abstrahierung der Simulationsmodelle potenziell nicht immer nachgebildet werden. Die von Barth beschriebene Abstraktion der Simulationsmodelle ermöglicht nicht die Verwendung derartiger Modelle über eine Wasserfahrt hinaus, z. B. für die Bedienschulung.

Die von Barth vorgestellte Methode zur Modellgenerierung, insbesondere die Konzeptalternative K1, stellt einen vielversprechenden und gleichzeitig auch pragmatischen Weg dar: Überträgt man

das Grundprinzip auf die Fertigungstechnik, könnte dies einen möglichen Grundstock für eine automatische Modellgenerierung bilden. Sieht man von den unterschiedlichen Rahmenbedingungen Verfahrenstechnik/Fertigungstechnik ab, liegen die Grenzen des Ansatzes insbesondere bei der verfügbaren Güte der vorhandenen Daten: Die Datenquellen müssen stets vollständig und konsistent sein. Eine mögliche Abstufung, mit Daten geringer Güte nur sehr vereinfachte Modelle automatisch generieren zu können, wird in der Arbeit von Barth nicht vorgenommen. Ferner müssen die von Barth als zwingend notwendig identifizierten Voraussetzungen für die automatische Modellgenerierung stets erfüllt werden. Inwieweit die vorgestellten Ansätze hilfreich sein können, sollten diese Voraussetzungen nicht erfüllt sein, wird nicht betrachtet. Lediglich Nachlässigkeiten im R&I-Fließbild werden angesprochen, so z. B. ein unvollständiger Übergang von der Rohrleitung zum Behälter. Der Ansatz von Barth hat bereits Einzug in die industrielle Praxis gefunden [HFB15].

4.4.3 Ansatz von Oppelt et al., 2014 (vgl. [OWD⁺14])

Oppelt et al. stellen eine Methode zur automatischen Modellgenerierung vor. Basis des Ansatzes sind Planungsdaten, die in Planungswerkzeugen der Firma Siemens vorliegen. Grundlegend wird ein Workflow vorgestellt, wie sich Simulationsmodelle für unterschiedliche Anwendungsfälle erzeugen lassen. Genannt werden die Anwendungsfälle „Wasserfahrt“ und die „Stromlaufplan Logik-Simulation“. Das „etablierte Planungswerkzeug [dient] als alleinige Informationsquelle und wird nur um simulationsrelevante Informationen erweitert“ [OWD⁺14]. Simulationsrelevante Informationen sind Simulationsmodelle, die im Werkzeug SIMIT vorliegen. Der vorgestellte Workflow bezieht sich im Speziellen auf die proprietäre Umsetzung innerhalb einer Siemens-Toolkette. „Mit Hilfe einer COMOS-internen Abfragetechnologie werden alle für die Simulation relevanten Komponenten und Informationen gesammelt und in eine XML-Datei geschrieben“. Diese XML-Datei wird anschließend weiter verwendet und bildet die Datenbasis bei der Erstellung eines Simulationsmodells.

Grenzen des Ansatzes von Oppelt et al.

Die Betrachtung von Simulationsmodellen sowie der Modellierung derartiger Modelle entspricht weitgehend den bereits von [BAR11] postulierten Grundgedanken bei der Simulation verfahrenstechnischer Anlagen zum Zweck der virtuellen Inbetriebnahme. Unterschiede lassen sich in den verwendeten Simulationswerkzeugen und deren spezifischen Eigenheiten feststellen. Während die Lösung von Barth [BAR11] herstellerneutral ist (sowohl bzgl. des PLS Systems als auch bzgl. des CAE-Tools als auch bzgl. des Simulationswerkzeugs), ist die von Oppelt et al. beschriebene Lösung nur im Rahmen der integrierten Siemens-Toolkette möglich.

4.5 Vergleichende Bewertung

Tabelle 4 bietet eine Gesamtübersicht aller betrachteten Ansätze und vergleicht diese auf Basis der in Abschnitt 3.5 aufgeworfenen Fragestellungen, die den Leitfragen I-V zugeordnet werden konnten.

Aus Tabelle 4 ergibt sich, dass die in Abschnitt 3.5 aufgeworfenen Fragestellungen als nur teilweise beantwortet klassifiziert werden können.

Große Heterogenität herrscht bei den für eine virtuelle Inbetriebnahme nutzbaren Datenquellen. In den meisten Fällen wird auf proprietäre Datenquellen zurückgegriffen, die sich für den jeweiligen Einsatzzweck eignen.

Hinsichtlich einer Systematisierung von Detaillierungsstufen für Simulationsmodelle besteht zudem noch großer Handlungsbedarf. Ebenso wird die mit Letzterem eng zusammenhängende Identifikation, welcher Detaillierungsgrad für einzelne Teil-Simulationsmodelle angemessen ist, in keinem der untersuchten Ansätze thematisiert.

In Kapitel 5 werden die Konsequenzen aus den in Tabelle 4 zusammenfassend dargestellten Ergebnissen gezogen.

Ansatz von	Modellerstellung	Methodik (DF Digitale Fabrik, 2206: mechr. Entw.-Methodik, MDA: Model-Driven-Architecture)	Datenquelle (Frage I.1)	Datengüte (Frage I.2)	Simulationsmodellbibliothek und Gestaltung (Frage II.1)	Detaillierungsstufen von Simulationsmodellen (Frage II.1)	Verwendetes Simulationswerkzeug (Frage II.1)	Betrachtung von Fehlerzuständen einer Anlage (Frage II.2)	Identifikation benötigter Detaillierungsgrad (Frage III.1)	Dynamischer Wechsel Detaillierungsgrad möglich (Frage III.1)	Schnittstellenkonsistenz von Simulationsmodellen (Frage III.2)	Aufwand Simulationsmodell-erstellung (Frage IV.1)	Technische Implementierung (Frage V.1)
[GOP*14] (AVANTI-Projekt)	konventionell	konventionell	3D-CAD, proprietär	O	X	O	proprietär	X	X	X	X	X	Physik-Engine
[SSD15] (AVANTI-Projekt)	konventionell	konventionell	Verhaltensmodelle vom KomponentenhHersteller	O	X	O	Modelica	X	X	X	X	mittel	Co-Simulation
[Kie09]	konventionell	DF, 2206	proprietär	O	proprietär	X	Dassault DELMIA	-	O	O	O	hoch	HIL
[Wün08]	konventionell	X	X	O	X	-	X	+	-	O	O	X	HIL/MIL/SIL
[LSB12] ff.	konventionell	2206, MDA	X	O	proprietär	-	Modelica	-	-	-	O	X	HIL/MIL/SIL
[DoHe07]	konventionell	X	proprietär	O	proprietär	-	Sinumerik Mach. Simulator	-	O	O	O	hoch	HIL
[HoVe12]	konventionell	X	proprietär	O	proprietär	-	proprietär	X	-	O	O	hoch	HIL
[Lac12]	konventionell	X	CAD, proprietär	O	proprietär	O	proprietär	X	O	O	O	hoch	HIL
[HTY11]	konventionell	X	X	O	X	O	O	O	O	O	O	X	Hybrid-HIL
[Lin11]	teil-automatisch	2206	proprietär	O	proprietär	X	proprietär	X	O	O	O	hoch	HIL/SIL
[BKS*09]	teil-automatisch	2206	CAD, proprietär	O	proprietär	O	proprietär	X	O	O	O	hoch	HIL
[Mey14]	teil-automatisch	X	Materialfluss	O	proprietär	X	Plant Sim., Winmod	X	O	O	O	X	HIL
[Spi09]	konventionell/ teil-automatisch	X	CAD, proprietär	O	proprietär	O	proprietär, Winmod	X	O	O	O	hoch	HIL
[Kuf11]	teil-automatisch/ voll-automatisch	2206	CAD, proprietär	O	proprietär	-	proprietär, Virtuos	X	O	O	O	hoch	HIL
[Bar11] *	teil-automatisch/ voll-automatisch	~ NA35	R&I, PLS, Excel	O	Modelica-Standard	-	Modelica	+	O	O	O	niedrig	HIL/SIL
[OWD*14] *	teil-automatisch/ voll-automatisch	X	In Siemens Werkzeugen vorhandene Daten	O	proprietär	X	Siemens Werkzeuge	X	O	O	O	X	HIL/SIL

O: wird in dem jeweiligen Ansatz nicht berücksichtigt; +: wird betrachtet, erfüllt Kriterium; -: wird betrachtet, erfüllt Kriterium nicht; X: keine Aussage möglich; *Ansätze aus der Verfahrenstechnik

Tabelle 4: Vergleichende Bewertung aller betrachteten Ansätze

5 Handlungsbedarf und Lösungskonzept

5.1 Identifizierte Defizite

Die in Abschnitt 3.5 identifizierten Leitfragen können Indikatoren für eine effiziente Umsetzung einer virtuellen Inbetriebnahme sein. Aus den in Tabelle 4 dargestellten Untersuchungsergebnissen lässt sich folgern, dass keiner der besprochenen Ansätze in der Lage ist, die in Abschnitt 3.5 aufgeworfenen Fragen hinsichtlich der Praktikabilität, insbesondere für KMU, zu beantworten. Partiiell finden sich Lösungen, die für eine weitergehende Betrachtung geeignet scheinen. Eine Applikation dieser Lösungen für eine *virtuelle Inbetriebnahme* scheitert derzeit an der nicht vorhandenen praktischen Umsetzbarkeit, mangelnder Reflexion der Unterschiede zwischen klassischem Entwicklungs- und Engineeringvorgehen, der Definition proprietärer Datenformate und -modelle als zu verwendende Datenbasis sowie inkonsistenter und partiell unvollständiger Implementierung der verwendeten Simulationsmodelle. Etablierte Ansätze (insb. [BAR11]) stammen aus anderen technischen Domänen, deren Randbedingungen nicht mit dem fertigungstechnischen Anlagenbau vergleichbar sind. Die Übernahme spezifischer Vorgehen und Methoden ist so erschwert oder nicht möglich. Aus den in Kapitel 4 betrachteten Ansätzen lassen sich, in Verbindung mit den in Abschnitt 3.5 aufgestellten Fragen, die sich aus den Betrachtungen in Kapitel 2 und 3 ergeben haben, zusammenfassend fünf grundlegende Defizite ableiten (→ Tabelle 5).

I. Welche Daten werden für eine virtuelle Inbetriebnahme benötigt und wie sollten diese abgebildet sein?	Frage I.1: Woher kommen die Daten für die Erstellung von Anlagensimulationsmodellen für eine virtuelle Inbetriebnahme und wie sollten diese strukturiert sein?	Nicht ausreichend beantwortet (für die Fertigungstechnik)	Defizit #1: Verwendung heterogener Datenquellen und nicht standardisierter Datenmodelle für die virtuelle Inbetriebnahme
	Frage I.2: Welche Daten in welcher Güte sind für eine virtuelle Inbetriebnahme notwendig?	Nicht ausreichend beantwortet	
II. Wie kann man virtuelle Inbetriebnahme umsetzen?	Frage II.1: Welche Detaillierungsstufen sollten für (Teil-)Simulationsmodelle zur Verfügung stehen?	Nicht ausreichend beantwortet	Defizit #2: Unklare Granularität/Detaillierung der verwendeten Simulationsmodelle
	Frage II.2: In wie weit sollten (technische und organisatorische) Fehler/Störungen der Anlage im Simulationsmodell berücksichtigt werden?	Nicht ausreichend beantwortet (für die Fertigungstechnik)	
III. Wie sollte man virtuelle Inbetriebnahme umsetzen?	Frage III.1: Welcher Detaillierungsgrad ist für welche Situation adäquat? Geht diese Situationsabhängigkeit soweit, dass sich der Detaillierungsgrad dynamisch anpassen sollte?	Nicht ausreichend beantwortet	Defizit #3: Unklare Implementierung von Teilsimulationsmodellen hinsichtlich des benötigten Detaillierungsgrads
	Frage III.2: Wie lässt sich die Konsistenz von (Teil-) Modellen in unterschiedlichem Detaillierungsgrad sicherstellen?	Nicht ausreichend beantwortet	
IV. Wie kann man effizient Simulationsmodelle erstellen?	Frage IV.1: Wie kann der Aufwand für eine virtuelle Inbetriebnahme verringert werden, um Simulation (insbesondere bei KMUs) als Standardmethode zu etablieren?	Nicht ausreichend beantwortet	Defizit #4: Hoher Aufwand bei der Modellerstellung
V. Wie kann man Simulationsmodelle implementieren?	Frage V.1: Wie sollten Simulationsmodelle implementiert werden?	Nicht ausreichend beantwortet	Defizit #5: Unklare technische Implementierung von Simulationsmodellen in Simulationsframework

Tabelle 5: Aufgeworfene Fragen, Fazit sowie abgeleitete Defizite

Die abgeleiteten Defizite werden im Folgenden näher beschrieben.

Defizit #1: Verwendung heterogener Datenquellen und nicht standardisierter Datenmodelle für die virtuelle Inbetriebnahme

Das Engineering automatisierter Anlagen ist durch heterogene, nicht standardisierte, geschlossene, proprietäre Datenquellen und -formate geprägt. Hoher Aufwand für den Transfer von Daten ist häufig notwendig. Aus der Analyse des Stands der Forschung und Technik wird ersichtlich, dass die Definition proprietärer, im Speziellen auf die virtuelle Inbetriebnahme zugeschnittener Datenformate und Beschreibungen der Struktur automatisierter Anlagen erfolgt. Derartige Ansätze sind in der Regel wenig generisch und bedingen einen hohen Aufwand bei der Erstellung des Datenmodells.

Generelle Aspekte wie eine durchgehende Objektorientierung können als nicht etabliert beschrieben werden, wenngleich dies in [VDI 4499-1] für den (fertigungstechnischen) Anlagenbau allgemein gefordert ist. Die Verwendung von neutralen Datenformaten kann im Gegensatz dazu zwar als mittlerweile etabliert (z. B. AutomationML, [AUTOMATIONML[®]]), die konkrete Abbildung von Anlagenstrukturen dagegen als nicht standardisiert bezeichnet werden [DFB11]. Obwohl das *Datenmodell der Digitalen Fabrik* in einigen Ansätzen Beachtung findet, steht weniger die pragmatische und systematisierte Umsetzung desselben, sondern vielmehr die Erweiterung um zusätzliche Inhalte im Vordergrund. Die Datengüte und damit die Betrachtung, ob virtuelle Inbetriebnahme überhaupt implementiert werden kann, findet keine Beachtung. Selbiges gilt sinngemäß für die verwendete Simulationsmodellbibliothek.

Es findet zudem keine Entkopplung von verwendeter Simulationsmodellbibliothek (und entsprechend enthaltener Simulationsmodelle) und Datenmodell als Basis für die Modellerstellung statt. Die vorherrschende enge Verknüpfung aus (Engineering-/Entwicklungs-)Vorgehen, darauf abgestimmtem Datenmodell und wiederum darauf eingepasster Simulationsbibliothek nimmt den Ansätzen die Flexibilität, in wechselnden Rahmenbedingungen nutzbringend einsetzbar zu sein. Insbesondere gilt dies im Hinblick auf die im fertigungstechnischen Anlagenbau vorherrschende heterogene und nicht standardisierte Datenbasis sowie für die große Anzahl an eingesetzten Simulationswerkzeugen.

Defizit #2: Unklare Granularität/Detaillierung der verwendeten Simulationsmodelle

Aus den in Kapitel 4 durchgeführten Untersuchungen lässt sich ferner folgern, dass in vielen Ansätzen das möglichst realistische Nachbilden des Verhaltens von automatisierten Anlagen im Vordergrund steht. Derartige Ansätze stellen das Anlagensimulationsmodell in einem homogenen und sehr hohen Detaillierungsgrad dar, der einen entsprechend hohen Aufwand bei der Modellerstellung bedingt. Detaillierte Simulationsmodelle haben zudem naturgemäß einen hohen Rechenaufwand, so dass Kriterien wie die notwendige Echtzeitfähigkeit der Simulation meist nicht mehr erfüllt werden können.

Fast alle der in Kapitel 4 betrachteten Ansätze setzen auf geschlossene, mitunter eigens für den jeweiligen Ansatz entwickelte Simulationswerkzeuge. Diese nutzen zumeist eine proprietäre Simulationsmodellbibliothek, deren Eigenschaften und Mächtigkeit naturgemäß vom verwendeten Simulationswerkzeug abhängig sind. Keine Beachtung finden (generische) Gestaltungsprinzipien, die eine allgemeingültige Anwendbarkeit der Ansätze unter Beachtung wechselnder Rahmenbedingungen und Simulationswerkzeuge ermöglichen würden. Rahmenbedingungen können in diesem Zusammenhang zur Verfügung stehende Simulationsplattform, Budgets für die Durchführung einer virtuellen Inbetriebnahme oder auch der Zeitpunkt im Engineering, an dem virtuelle Inbetriebnahme stattfinden soll, sein. Steht andererseits ein heterogener Detaillierungsgrad im Vor-

dergrund, ergeben sich bei den betrachteten Ansätzen Inkonsistenzen und Einschränkungen in der Aussagefähigkeit der definierten Detaillierungsstufen.

Defizit #3: Unklare Implementierung einer Simulation hinsichtlich des benötigten Detaillierungsgrads

Simulationsmodelle bilden den Schlüssel einer jeden virtuellen Inbetriebnahme. Das möglichst realistische Nachbilden des Verhaltens von automatisierten Anlagen steht in vielen Fällen im Vordergrund. Je nach Anwendungs- und Testfall ist nicht immer eine sehr genaue Simulation notwendig – oftmals reichen bereits sehr einfache Modelle aus, um aussagekräftige Ergebnisse zu erzielen. Derzeit ist keine Methode bekannt, mit der sich bei vorliegenden Simulationsmodellen der notwendige Detaillierungsgrad bestimmen lassen kann.

Zudem wird die Sicherstellung der Konsistenz von (datentechnischen) Schnittstellen zwischen in unterschiedlichen Detaillierungsstufen abgebildeten Simulationsmodellen in heterogen aufgebauten Anlagensimulationsmodellen derzeit nicht betrachtet.

Defizit #4: Hoher Aufwand bei der Modellerstellung

Für die virtuelle Inbetriebnahme ist derzeit ein hoher Aufwand notwendig. Zwar existieren Ansätze, mittels (teil-)automatischer Modellgenerierung, den Aufwand für die Simulationsmodellerstellung zu minimieren. Derartige Ansätze sind aufgrund des verwendeten, mit hohem Aufwand zu erstellenden, proprietären Datenmodells der betrachteten Anlage nicht für einen Einsatz im Rahmen einer virtuellen Inbetriebnahme geeignet und wenig generisch. Unterstützende Methoden, mit denen sich eine manuelle Erstellung von Anlagensimulationsmodellen systematisieren und unterstützen lassen würde, werden nicht oder nur für Entwicklungsprozesse betrachtet, die für das Engineering von automatisierten Anlagen nicht ohne weiteres einsetzbar sind.

Defizit #5: Unklare technische Implementierung einer virtuellen Inbetriebnahme

Die technische Implementierung einer virtuellen Inbetriebnahme wird in den untersuchten Ansätzen nur am Rande untersucht. Dies betrifft einerseits die geeignete Emulation der in der Steuerung projektierten Feldbusgeräte, andererseits die konkrete Implementierung innerhalb einer Hardware-, Software- oder Model-in-the-Loop Plattform. *Anm.: Defizit #5 wird in dieser Arbeit nicht vertieft betrachtet, eine Abgrenzung sowie Begründung hierfür findet sich in Abschnitt 5.3.2.*

5.2 Zielformulierung und Haupttätigkeitsfelder

Um die in Abschnitt 5.1 identifizierten Probleme lösen zu können, ist es notwendig, unterstützende Methoden für die Durchführung einer *virtuellen Inbetriebnahme* zu konzipieren. Im Vordergrund sollte stets die Reduzierung des Aufwands durch die Systematisierung der Modellerstellung stehen. Dies beinhaltet damit zusammenhängende Arbeitsschritte wie die Identifizierung eines adäquaten Detaillierungsgrads für einzelne Teile des Simulationsmodells. Wie von [BAR11] dargelegt, kann insbesondere eine automatische Modellgenerierung einen wesentlichen Beitrag dazu leisten, den aufzubringenden Aufwand für die Modellerstellung und somit für eine virtuelle Inbetriebnahme insgesamt zu senken. Eine automatische Modellgenerierung stellt daher einen er-

strebenswerten Zielzustand dar, der aber von vielen Aspekten, insbesondere hinsichtlich des als Basis der Modellgenerierung verwendeten Datenmodells, abhängig ist. Tabelle 6 zeigt die für eine automatische Modellgenerierung gemäß des Ansatzes von [BAR11] relevanten Aspekte, ergänzt um die Anforderungen an das *Datenmodell der Digitalen Fabrik* (gemäß [VDI 4499-1]) für den geforderten Aspekt, sofern zutreffend:

Aspekt für automatische Modellgenerierung gemäß [BAR11]	Voraussetzung für automatische Modellgenerierung gemäß [BAR11]:	Anforderungen an das Datenmodell der Digitalen Fabrik [VDI 4499-1]:
Datenquellen	Rechnergestützt auswertbare Datenquellen	Rechnergestützt auswertbare Datenquellen
Datenaustauschformat	Offenes Datenformat für den Datenaustausch	Neutrales Datenformat für den Datenaustausch
Systemsichtweise	Objektorientierte Planung	Objektorientierte Darstellung
Anforderungen an das Datenmanagement (Güte der Daten)	Vollständigkeit, Konsistenz	Vollständigkeit, Konsistenz, Aktualität, Validität, Korrektheit, Genauigkeit (=Granularität)
Simulationsmodellbibliothek	Muss vorhanden sein, objektorientierter Aufbau vorteilhaft	(wird nicht betrachtet)
Zuweisung Anlagenobjekt/Simulationsmodell	Mittels manuell zu erstellender Mappingtabellen	(wird nicht betrachtet)
Zugriff auf Konfiguration des Steuerungssystems	Konfiguration muss verfügbar sein zur Kopplung Simulation/Steuerung	Zugriff ist möglich, da die Konfiguration des Steuerungssystems Bestandteil des PPR-Konzepts ist

Tabelle 6: Voraussetzungen für eine automatische Modellgenerierung (angelehnt an [BAR11]) und Vergleich mit den Anforderungen an das Datenmodell der Digitalen Fabrik

Insbesondere in den Aspekten Datenquellen, Systemsichtweise, Anforderungen an das Datenmanagement und Zugriff auf Konfiguration des Steuerungssystems lassen sich Überschneidungen zwischen Anforderungen an die automatische Modellgenerierung und Anforderungen an das *Datenmodell der Digitalen Fabrik* gemäß [VDI 4499-1] (→ Abschnitt 2.3) feststellen. Keiner der beschriebenen Aspekte der Modellgenerierung kann aber für den fertigungstechnischen Anlagenbau als derzeit gegeben vorausgesetzt werden. Eine Beeinflussung von Aspekten wie Datengüte, im Speziellen Vollständigkeit und Konsistenz der zur Verfügung stehenden Daten (erstellt von den Gewerken Mechanik und Elektrotechnik) liegt zudem außerhalb der Möglichkeiten der mit *virtueller Inbetriebnahme* beschäftigten Ingenieure des Gewerks Automatisierungstechnik.

Die von [BAR11] zum Zweck der Modellgenerierung verwendeten Datenquellen *R&I-Fließbild* und *CAE-Planungssysteme* existieren im fertigungstechnischen Anlagenbau zudem nicht in dieser Form, so dass eine Übernahme der von [BAR11] beschriebenen, *konkreten* Vorgehensweisen und Methoden für den fertigungstechnischen Anlagenbau nicht möglich ist. In Abhängigkeit der verfügbaren Daten, ihrer Güte sowie der zur Verfügung stehenden Simulationsumgebung und Simulationsmodellbibliothek müssen bei der Durchführung einer virtuellen Inbetriebnahme im ferti-

gungstechnischen Anlagenbau daher die rein manuelle wie auch die (teil-) automatische Modellerstellung gleichermaßen in Betracht gezogen werden. Aufgrund dessen sollten die die virtuelle Inbetriebnahme unterstützenden Methoden einen gewissen Universalismus aufweisen, der es erlaubt, die Methoden in allen drei prinzipiell möglichen Formen der Modellerstellung (manuell, teilautomatisch, automatisch) nutzbringend einzusetzen.

Aus Tabelle 6 lässt sich folgern, dass das *Datenmodell der Digitalen Fabrik* eine grundlegende Basis für die Erreichung des Zielzustands der automatischen Modellgenerierung bilden kann. Ferner lässt sich aus Tabelle 6 schließen, dass das Datenmodell um für die *virtuelle Inbetriebnahme* relevante, und im Speziellen die automatische Modellgenerierung ermöglichende, Aspekte erweitert werden muss. Zudem wäre es vorteilhaft, Methoden und Abbildungsvorschriften zu ergänzen, die eine Zuweisung von Simulationsmodell zu Anlagenobjekt direkt aus dem Datenformat ermöglichen. Derzeit ist dafür keine Methode bekannt, die ohne einen hohen Aufwand für die Erstellung und die Ertüchtigung des Datenmodells auskommt. Eine zweckdienliche Abbildung des *Datenmodells der Digitalen Fabrik* sowie die Ergänzung desselben um für die virtuelle Inbetriebnahme relevante Aspekte stellt daher das erste Haupttätigkeitsfeld dieser Arbeit dar.

Ferner ist es notwendig, virtuelle Inbetriebnahme unter dem Aspekt der Simulation zu betrachten: Sichergestellt werden sollte, dass die eingesetzten Simulationsmodelle jederzeit genügend aussagefähig sind. Ziel sollte es sein, einzelne Anlagenteile in heterogenen, d. h. unterschiedlichen Detaillierungsstufen darzustellen, je nach Notwendigkeit. Es sollte gelten: Die Simulation sollte so einfach wie möglich und so kompliziert wie nötig sein.

Aus den Untersuchungen in Kapitel 4 geht hervor, dass die Umsetzung eines heterogenen Detaillierungsgrad zwar einerseits prinzipiell möglich ist; andererseits ist die Implementierung aufgrund mangelnder Systematik und Inkonsistenzen bei den Beschreibungen der zum Einsatz kommenden Simulationsmodelle mit hohem Aufwand verbunden. Es gilt daher, unter Vermeidung von hohem Aufwand, einen adäquaten, heterogenen Detaillierungsgrad einzelner Simulationsmodelle zu ermöglichen. Das zweite Haupttätigkeitsfeld dieser Arbeit umfasst daher die Definition von Detaillierungsstufen. Dies beinhaltet, neben der Definition systematisch aufeinander aufbauender Detaillierungsstufen, auch eine Methode, die es ermöglicht, den adäquaten Detaillierungsgrad situationsspezifisch zu identifizieren. Ferner wird eine Methodik benötigt, die die Konsistenz von heterogen aufgebauten Simulationsmodellen sicherstellt.

5.3 Lösungskonzept, Einschränkung des Scopes und technische Rahmenbedingungen

5.3.1 Lösungskonzept

Auf Grundlage der identifizierten Defizite und der beiden beschriebenen Haupttätigkeitsfelder kann ein mehrstufiges Lösungskonzept entwickelt werden. Das Lösungskonzept setzt sich aus mehreren Bausteinen zusammen, die in den folgenden Abschnitten kurz beschrieben werden. Als finaler Zielzustand wird eine automatische Modellgenerierung angestrebt. Die Methoden haben aber den Anspruch, für alle der drei prinzipiell möglichen Modellbildungsarten nutzbringend einsetzbar zu sein (→ Abbildung 5-1). Wie sich eine automatische Modellgenerierung umsetzen lässt und welche Voraussetzungen dafür notwendig sind, wird in Abschnitt 8.4 beschrieben. Die hier

vorgestellten Lösungsbausteine stellen eine Kurzzusammenfassung der in den späteren Kapiteln ausführlich betrachteten konkreten Lösungen dar.

5.3.1.1 Baustein 1: Abbildung des Datenmodells der Digitalen Fabrik (1.1) und Erweiterung des Datenmodells um für die virtuelle Inbetriebnahme relevante Aspekte (1.2) – Defizit #1

Baustein 1.1 beschreibt im Konkreten, wie Engineering-Daten im *Datenmodell der Digitalen Fabrik* abgelegt werden sollten. Von grundlegender Bedeutung sind die Anforderungen an das *Datenmodell der Digitalen Fabrik*. Der Fokus der Betrachtungen wird insbesondere auf die abzubildende Granularität und Hierarchisierung der Anlagenstruktur gelegt. Die derzeit übliche Betrachtung von Anlagen in der automatisierungstechnischen Praxis spielt in diesem Zusammenhang eine große Rolle. Der in *Baustein 1.1* entwickelte Vorschlag zur Anlagenstrukturbeschreibung deckt alle für das PPR-Prinzip relevanten Bereiche ab. *Baustein 1.1* enthält zudem eine Untersuchung, welches konkrete Datenformat für die Abbildung des *Datenmodells der Digitalen Fabrik* geeignet ist – dies insbesondere im Hinblick auf die spätere Verwendung in einer *virtuellen Inbetriebnahme* und darüber hinaus.

Baustein 1.2 beschreibt für die virtuelle Inbetriebnahme relevante Erweiterungen des Datenmodells. Prämisse der Betrachtungen ist die möglichst einfache und gleichzeitig aufwandsarme Identifizierung, welches Anlagenobjekt mit welchem der zur Verfügung stehenden Simulationsmodelle für die virtuelle Inbetriebnahme substituiert werden kann.

Die Bausteine 1.1 und 1.2 stehen unter Leitfrage I: *Welche Daten werden für eine virtuelle Inbetriebnahme benötigt und wie sollten diese abgebildet sein?*

5.3.1.2 Baustein 2: Definition von systematisiert aufeinander aufgebauten Detaillierungsstufen – Defizit #2

In *Baustein 2* wird eine Systematik vorgestellt, mit der sich heterogene Detaillierungsstufen für Anlagensimulationsmodelle implementieren lassen. Im Fokus der Betrachtungen stehen die für den fertigungstechnischen Anlagenbau essenziellen Stückgutbewegungen. Die Betrachtungen reichen von rein zeitbasierten Betrachtungen hin zu möglichem physikalischem Verhalten. Die Simulation von n-dimensionalen Bewegungen wird betrachtet und abhängig vom Detaillierungsgrad eingeordnet.

Ziel ist es, eine systematisierte Einordnung der benötigten Verhaltenscharakteristika zu ermöglichen. Die in *Baustein 2* entwickelten Modellarten bilden die Basis für die in [VDI 3693-1] beschriebenen Modellarten. Im Vergleich zur [VDI 3693-1] ist der betrachtete Lösungsraum in dieser Arbeit kleiner.

Baustein 2 steht unter Leitfrage II: *Wie kann man virtuelle Inbetriebnahme umsetzen?*

5.3.1.3 Baustein 3: Methoden zur systematisierten Identifizierung des adäquaten Detaillierungsgrads (3.1) und Methode zur Sicherstellung der Konsistenz von Anlagensimulationsmodellen in heterogenem Detaillierungsgrad (3.2) – Defizit #3

Die Simulation in einem einzigen, homogenen Detaillierungsgrad bedingt bei sehr detaillierten Simulationsmodellen einen hohen Rechenbedarf. Stehen unterschiedliche Detaillierungsstufen zur Verfügung, stellt sich die Frage, wann und in welcher Situation welcher Detaillierungsgrad

genutzt werden sollte. In *Baustein 3.1* wird zur Beantwortung dieser Frage eine Methode vorgestellt, welche es ermöglicht, aus dem in den verfügbaren Anlagendaten dargestellten Wissen um die (u. a. konstruktive) Gestaltung einzelner Anlagenteile den für den jeweiligen Anlagenteil adäquaten Detaillierungsgrad für eine Simulation systematisiert abzuleiten. Die Betrachtungen fokussieren zunächst auf eine starre Zuweisung der Detaillierungsstufen. Der Baustein enthält darüber hinausgehend eine Untersuchung möglicher nutzbringender Potenziale eines dynamischen, während der Laufzeit der Simulation wechselbaren Detaillierungsgrads.

In Abhängigkeit vom betrachteten Detaillierungsgrad unterscheidet sich das von außen sichtbare Verhalten eines Simulationsmodells. Abhängig vom gewählten Detaillierungsgrad und der damit möglichen Beachtung physikalisch/dynamischer Einflussgrößen kann das beobachtbare Verhalten als kausal oder als a-kausal beschrieben werden. In *Baustein 3.2* werden speziell Situationen betrachtet, in denen Simulationsmodelle mit unterschiedlichem Verhalten miteinander gekoppelt werden. Es wird eine Methode vorgestellt, wie sich potenziell auftretende Inkonsistenzen automatisiert erkennen und beheben lassen.

Die Bausteine 3.1 und 3.2 stehen unter Leitfrage III: *Wie sollte man virtuelle Inbetriebnahme umsetzen?*

5.3.1.4 Baustein 4: Generischer Modellgenerierungsalgorithmus (4) – Defizit #4

Für den Fall, dass sich die zur Verfügung stehenden Daten für eine (teil-) automatische Modellgenerierung eignen, wird ein generischer Modellgenerierungsalgorithmus benötigt. Dieser Modellgenerierungsalgorithmus muss es ermöglichen, die ansonsten manuell durchgeführten Schritte der Modellbildung automatisiert durchführen zu können. Zudem muss der Modellgenerierungsalgorithmus unabhängig von der verwendeten Simulationsumgebung und Simulationsmodellbibliothek funktionieren.

Baustein 4 steht unter Leitfrage IV: *Wie kann man effizient Simulationsmodelle erstellen?*

5.3.1.5 Baustein 5: Implementierungsplattform (5) – Defizit #5 (nicht betrachtet)

Der Begriff Implementierungsplattform bezieht sich auf die verwendete Simulationsumgebung, wie in Abschnitt 3.3 beschrieben. Eine detaillierte Betrachtung der Implementierungsplattform und eine technische Erläuterung der Funktionsweise erfolgen in dieser Arbeit nicht im Detail. Eine Abgrenzung und Begründung findet in Abschnitt 5.3.2 statt. Baustein 5 steht unter der Leitfrage: *Wie kann man Simulationsmodelle implementieren?*

5.3.1.6 Übersicht über Lösungskonzept und enthaltene Bausteine 1.x bis 5

Abbildung 5-1 zeigt das Lösungskonzept und die einzelnen im Lösungskonzept enthaltenen Bausteine in einer Übersicht.

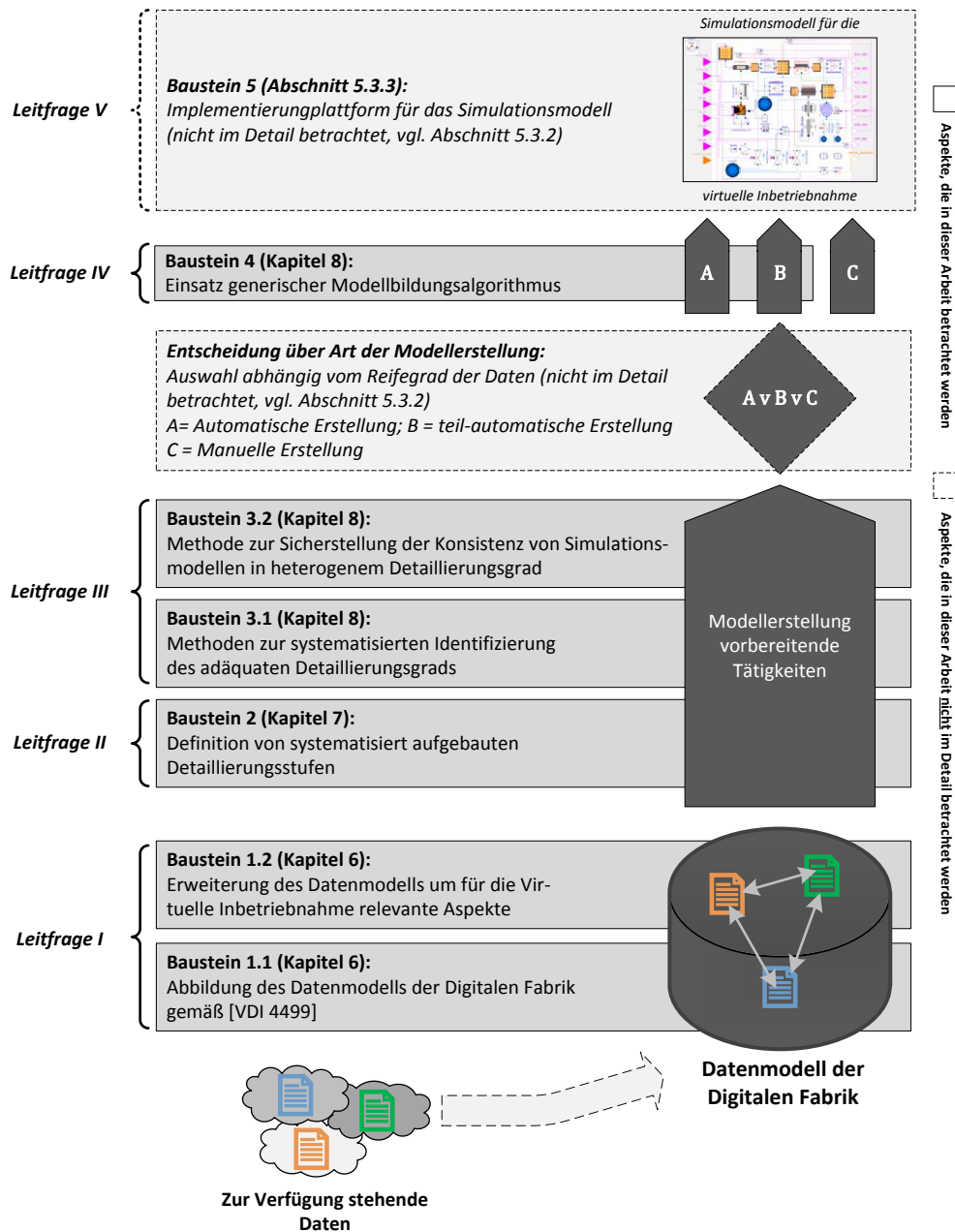


Abbildung 5-1: Lösungskonzept

5.3.2 Einschränkung des Scopes

In den folgenden Abschnitten wird der in dieser Arbeit betrachtete Scope hinsichtlich der identifizierten Defizite (→ Abschnitt 5.1) sowie des abgeleiteten Lösungskonzepts (→ Abschnitt 5.3.1) eingeschränkt.

5.3.2.1 Einschränkung hinsichtlich betrachteter Anlagentypen

Die Arbeit fokussiert auf die virtuelle Inbetriebnahme fertigungstechnischer Anlagen. Nicht im Fokus steht die in der [VDI 3693-1] bereits für die virtuelle Inbetriebnahme ausgeschlossene Simulation von konkreten Fertigungsprozessen wie Bohren, Honen und weiteren (→ Abschnitt 3.3).

Anlagentechnisch wird der Fokus dieser Arbeit auf durch SPS gesteuerte Anlagen gelegt. Dies umfasst fördertechnische Anlagen wie EHB-Systeme [VDI 4441-1], fahrerlose Transportsysteme (sofern durch SPS gesteuert) und andere, konventionelle Fördertechniksysteme [MRW04] genauso wie Bearbeitungsstationen (z. B. Scheibenklebeanlagen in der Automobil-Endmontage).

Nicht im Detail betrachtet wird die virtuelle Inbetriebnahme von Robotersteuerungen und Bearbeitungsmaschinen (CNC), die eine (oder mehrere) der in [DIN 8580] beschriebenen Bearbeitungen ausführen. Wie in Abschnitt 3.3 identifiziert, liegt der Fokus bei Robotersteuerungen auf der Offline-Programmierung und bedingt in fast allen Fällen eine genaue Simulation der Kinematik des Roboters. Dies wird in dieser Arbeit nicht im Detail betrachtet. Der in dieser Arbeit betrachtete Lösungsraum für Simulationsmodelle zum Zweck der virtuellen Inbetriebnahme ist daher generell enger gefasst als der in [VDI 3693-1] definierte.

5.3.2.2 Einschränkungen hinsichtlich der Betrachtungen des Reifegrads von zur Verfügung stehenden Engineering-Daten (Entscheidung: über Art der Modellerstellung)

Eine Betrachtung des Reifegrads und die Ableitung, welche der drei Modellerstellungsarten möglich ist (*Entscheidung über Art der Modellerstellung*, vergleiche Abbildung 5-1 Teil A v B v C) erfolgt in dieser Arbeit nur grundlegend anhand der für eine teil- oder voll-automatische Modellgenerierung notwendigen Voraussetzungen, nicht im Detail (→ Abschnitt 8.4). Im Vordergrund dieser Arbeit stehen Methoden, die für alle drei potenziell möglichen Arten der Modellerstellung zum Einsatz kommen können – unabhängig davon, welche Modellerstellung im konkreten durchgeführt wird. Die Definition eines Rahmenwerks für die Analyse und Einordnung von Daten in ein spezifisches Reifegradmodell ist derzeit Gegenstand der Forschung am Institut für Automatisierungstechnik der Helmut-Schmidt-Universität (Projekt EISIVIO, ZIM Förderkennzeichen ZF4063901SS5). Erste Ergebnisse hierzu wurden in [PGF⁺¹⁶] veröffentlicht.

5.3.2.3 Einschränkungen der Betrachtungen hinsichtlich der für die virtuelle Inbetriebnahme verwendeten Implementierungsplattform (erkanntes Defizit #5)

Eine genauere Betrachtung der als Defizit #5 erkannten Implementierungsplattform findet in dieser Arbeit nicht im Detail statt. Grundsätzlich sind bei der Wahl und dem Aufbau einer geeigneten Implementierungsplattform folgende Herausforderungen zu sehen ([SWD⁺¹³], [RPK⁺¹⁴]):

- Wahl eines geeigneten Simulationswerkzeugs [KPR⁺¹⁴]
- Wahl der Kopplungstechnologie *Implementierungsplattform* ↔ *SPS* [RPK⁺¹⁴]
- Sicherstellung der Echtzeitfähigkeit der Lösung ([HOVE12], [LIFR09]), insb. bei Verwendung eines echtzeitfähigen Feldbusses zur Kopplung *Implementierungsplattform* ↔ *SPS* [RPK⁺¹⁴]
- Technische Umsetzung der Emulation der in der SPS genutzten Feldbusgeräte [RPK⁺¹⁴]
- Auswahl, welche Funktionen des jeweiligen Feldgeräts nachgebildet werden müssen [SWD⁺¹³]
- Geringe Hardwarekosten [RPK⁺¹⁴]

Letztendlich handelt es sich bei den meisten dieser Punkte in erster Linie um technische Problemstellungen. Die Frage, welche Funktionen des jeweiligen Feldgeräts in einer virtuellen Inbetriebnahme und darüber hinaus in anderen Projektphasen beachtet werden müssen, wird in [SWD⁺¹³] detailliert betrachtet.

Diese Arbeit beschränkt sich daher darauf, die im Projekt „HIL-IMA“ (ZIM Förderkennzeichen KF2625802KM2) gewonnenen Erkenntnisse über die Emulation von Feldgeräten innerhalb einer für die virtuelle Inbetriebnahme geeigneten HIL-Experimentierplattform darzustellen. Diese beinhalten Vorschläge zur Emulation von Feldbusgeräten, deren Anbindung an die eigentliche Simulation innerhalb einer Hardware-in-the-Loop Plattform sowie die Betrachtung einer möglichen Experimentiersteuerung. Letzteres bezeichnet die Bedienungsmöglichkeiten der Simulation. Die detaillierten Ergebnisse der Untersuchungen sind in [RPK⁺14[#]] zu finden. Diese werden im nächsten Abschnitt 5.3.3 kurz vorgestellt und dienen für die folgenden Untersuchungen als gesetzte Implementierungsplattform, ohne im Detail die technische Umsetzung zu betrachten. *Anm.: In Abschnitt 5.3.3 erfolgen teils wörtliche Wiedergaben aus [RPK⁺14[#]].*

5.3.3 Framework für die Durchführung einer virtuellen Inbetriebnahme

Virtuelle Inbetriebnahme basiert technisch gesehen auf zwei Systemen, die miteinander verbunden werden müssen. Auf der einen Seite steht der zu testende Steuerungscode, auf der anderen Seite das Simulationsmodell der Anlage, gegen das der Steuerungscode getestet werden kann. Beide Elemente müssen in eine entsprechende Plattform integriert werden, welche die Ausführung des Steuerungscode und der Simulation ermöglicht. Diese Plattformen können real sein (d. h. mittels für den Anwendungszweck geeigneter Hardware umgesetzt), aber auch rein virtuelle Umgebungen sind denkbar.

Aus Sicht der AT-Pyramide ist die SPS als den Steuerungscode ausführende Instanz der Steuerungsebene zugeordnet, während eine Simulationsplattform für die virtuelle Inbetriebnahme der Feld- sowie Prozessebene zugewiesen werden kann (→ Abschnitt 3.2.2 Abbildung 3-2). Beide Ebenen müssen untereinander Daten austauschen (im konkreten das Prozessabbild), um eine virtuelle Inbetriebnahme zu ermöglichen. Die Gesamtheit beider Plattformen wird in [VDI 3693-1] als sog. *Testkonfiguration* bezeichnet. Für eine grundlegende Übersicht über die für *virtuelle Inbetriebnahme* infrage kommenden Testkonfigurationen (*HIL*, *MIL*, *SIL*) sei auf Abschnitt 3.3 verwiesen. Die Entscheidung, welche der möglichen Testkonfiguration zum Einsatz kommen sollte, ist von vielen Faktoren abhängig. In der [VDI 3693-1] werden hierfür Art und Umfang der eingesetzten Modelle, Art und Struktur des nachzubildenden Automatisierungssystems, aber auch die zur Durchführung einer virtuellen Inbetriebnahme verfügbaren Simulationswerkzeuge genannt.

Eine Ausnahme hiervon können Situationen darstellen, in denen eine virtuelle Inbetriebnahme neu in ein Unternehmen eingeführt werden soll. Konkrete Handlungsempfehlungen, wie dies geschehen kann, werden derzeit vom VDI/GMA Fachausschuss 6.11 [GMA FA 6.11[®]] erarbeitet; diese werden als Blatt 2 der Richtlinie VDI 3693 veröffentlicht werden. Für die folgenden Betrachtungen wird hinsichtlich der technischen Implementierung auf die im Projekt „HIL-IMA“ entwickelte HIL-Plattform zurückgegriffen. Diese Plattform wird in den folgenden Abschnitten kurz vorgestellt. Die in dieser Arbeit vorgestellten Lösungsbausteine sind technologieunabhängig, die Implementierung erfolgt daher rein beispielhaft anhand der hier beschriebenen Lösung.

5.3.3.1 Synchronisation zwischen HIL-Plattform und SPS

Grundsätzlich muss der Simulationsablauf und der in der SPS ausgeführte Programmablauf zeitlich synchronisiert werden [WAFR07]. In diesem Zusammenhang spielen die Begriffe Echtzeit und Realzeit eine wichtige Rolle. Unter Echtzeit versteht man den Betrieb eines Rechensystems, so dass „[...] Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind“

[ISO/IEC 2382]. Eine genaue Zeitdefinition lässt sich daraus nicht ableiten. Die Zeit kann, je nach Anwendung, prinzipiell beliebig groß oder klein gewählt werden. Die Echtzeitfähigkeit spielt für eine HIL-Simulation immer dann eine Rolle, wenn mit einer festen Abtastrate (z. B. 1 kHz) Signale aufgenommen werden sollen. Dies ist immer dann der Fall, wenn ein Bussystem emuliert wird, bei dem das Prozessabbild mit einer festen Rate aktualisiert wird (→ Abschnitt 5.3.3.3 Alternative 1). Als Realzeitsysteme werden dagegen Systeme bezeichnet, in denen Zeit schlicht kontinuierlich fortschreitet.

Die im Projekt „HIL-IMA“ erarbeitete HIL-Plattform wurde dahin gehend entworfen, dass eine harte Echtzeitfähigkeit sichergestellt wird [RPK*14[#]]. Eine harte Echtzeitfähigkeit bedeutet, dass die HIL-Plattform *immer* innerhalb einer fest definierten Zeitspanne alle relevanten Berechnungen, die für die Funktionsfähigkeit wesentlich sind, durchführen muss. Wesentliche ist die Emulation der Feldbusteilnehmer. Die HIL-Plattform muss der SPS die Informationen (E/A-Signale sowie feldbustypische Diagnoseinformationen) so zur Verfügung stellen, wie es die realen Feldbusgeräte auch tun würden (→ Abschnitt 3.2.2). Abhängig vom verwendeten Feldbussystem kann die Takt- rate der Informationsbereitstellung und -verarbeitung variieren. Im Falle von PROFINET IO kann die Taktrate zwischen 1 bis 10ms liegen [PROFINET[®]]. Im Projekt „HIL-IMA“ kamen Feldgeräte zum Einsatz, die mit einer Taktrate von 1ms arbeiten. Entsprechend ergibt sich für die HIL-Plattform eine harte Echtzeitfähigkeit von 1ms.

Um Signalverfälschungen auszuschließen muss die harte Echtzeitfähigkeit im selben Maße für die eigentliche Simulation gelten. Das bedeutet, dass die durchgeführte Simulation innerhalb einer fest definierten Zeitspanne Berechnungsergebnisse zur Verfügung stellen muss. Im Falle der im Projekt „HIL-IMA“ erstellten HIL-Plattform bedeutet dies, dass das verwendete Simulationsmodell mit einer festen Schrittweite von 1ms simuliert werden muss. Hierfür sind Anpassungen an den Simulationsmodellen notwendig. Dies wird gesondert in Abschnitt 7.4.2.2 erläutert.

5.3.3.2 Beispielhafte Umsetzung einer HIL-Simulationsumgebung

Wie Eingangs von Abschnitt 5.3.3 beschrieben, besteht eine Testumgebung für die virtuelle Inbetriebnahme immer aus mindestens zwei Elementen. Bei der hier betrachteten HIL-Plattform wird für die Implementierung des Steuerungscode die später auch in der Anlage verwendete Steuerung (SPS), inkl. der enthaltenen Hardwarekonfiguration, verwendet. Dem gegenüber steht die Simulationsplattform. Das Simulationsmodell einer Anlage kann aus Sicht der Simulationsplattform zunächst als Black-Box beschrieben werden. Für die Simulationsplattform spielt der innere Aufbau und die Art der Modellierung des Simulationsmodells zunächst keine Rolle. Entscheidend sind vielmehr die Schnittstellen des Simulationsmodells nach außen, d. h. zu einzelnen Teilen der Simulationsplattform und, im erweiterten Sinn, zu dessen Steuerung. Im Sinne der generischen Austauschbarkeit der Simulationsmodelle, und um damit die Nutzung verschiedener Simulationswerkzeuge zu ermöglichen, ist es vorteilhaft, hierfür eine standardisierte Schnittstelle zu nutzen.

Das FMI-Framework [FMI[®]] stellt eine standardisierte Schnittstelle dar, über die sich der Datenaustausch eines Simulationsmodells nach außen parametrieren lässt. Das FMI lässt sich dazu nutzen, ein einziges Simulationsmodell, z. B. einer automatisierten Anlage, zu implementieren. Der FMI-Standard ermöglicht es ferner auch, unterschiedlichste Simulationsmodelle, die in unterschiedlichen Simulationswerkzeugen erzeugt wurden, miteinander koppeln zu können. Zu diesem

Zweck müssen die Simulationsmodelle in ein spezielles Format gebracht werden, die sog. Functional-Mockup-Unit (FMU) [FMI®].

Die FMU ist als Archivdatei beschreibbar und beinhaltet zum einen das Simulationsmodell als Bibliotheksdatei (als *.dll-Datei) und/oder aus dem ursprünglichen Simulationsmodell generierten C-Code. Zudem enthält sie eine XML-Datei, welche alle Zugriffsmöglichkeiten und die Parametrierung des Simulationsmodells beinhaltet, um dieses gemäß FMI-Spezifikation simulieren zu können [RPK+14[#]]. Im Rahmen der von [RPK+14[#]] beschriebenen HIL-Plattform wird die FMI-Spezifikation 1.0 in der Betriebsart „Model-Exchange“ [FMI®] verwendet. Im Gegensatz zur Betriebsart „Co-Simulation“ [FMI®] besteht hier die Möglichkeit, eigene Integrationsverfahren und Simulationsalgorithmen zu implementieren. Eine FMU in der Betriebsart „Co-Simulation“ enthält bereits den zur Simulation benötigten Solver und wäre alleine lauffähig. Die Nutzung spezieller Solver ist immer dann notwendig, wenn eine harte Echtzeitfähigkeit sichergestellt werden muss. Abbildung 5-2 zeigt die von einer FMU bereitgestellten Schnittstellen und Funktionen in Kombination mit einem externen Solver, wie in [RPK+14[#]] dargestellt. Eingangs- sowie Ausgangsvektor der FMU (Abbildung 5-2 links) werden von einem Signalmanager gehandhabt (Abbildung 5-3).

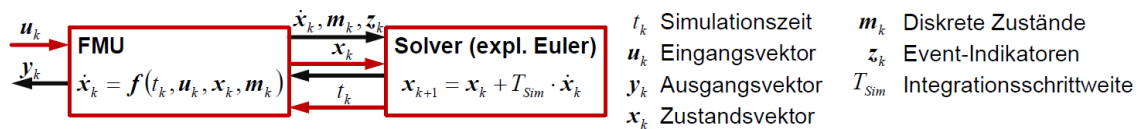


Abbildung 5-2: Signale und Funktionen bei Nutzung einer FMU mit externen Integrationsverfahren gemäß [RPK+14[#]]

Die eigentliche Simulation erfolgt mittels eines Simulationskerns, der die Verwendung unterschiedlicher Simulationsverfahren (Solver) ermöglicht. Eine FMU unterstützt nur explizite Berechnungen von Systemfunktionen. Daher können für die Echtzeitsimulation, bei der konstante Zeitschrittweiten notwendig sind, prinzipiell nur explizite Integrationsverfahren wie z. B. das explizite Euler-Verfahren eingesetzt werden [RPK+14[#]]. Wegen des im Vergleich zu impliziten Integrationsverfahren deutlich kleineren numerischen Stabilitätsbereichs erhöhen sich die Anforderungen an die Modellierung der eigentlichen Simulationsmodelle. Mit Simulationsmodellen sind in diesem Zusammenhang die einzelnen Teil-Simulationsmodelle gemeint, aus denen sich ein gesamtheitliches Anlagensimulationsmodell zusammensetzt [RPK+14[#]]. Die Auswirkungen auf die Modellierung dieser Teil-Simulationsmodelle werden gesondert in Abschnitt 7.4.2.2 betrachtet.

Abbildung 5-3 zeigt eine Gesamtübersicht über die im Projekt „HIL-IMA“ erstellte Simulationsplattform. Für eine genauere Beschreibung, insb. der HIL-Experimentsteuerung und der damit verbundenen Möglichkeiten, sei auf [RPK+14[#]] verwiesen.

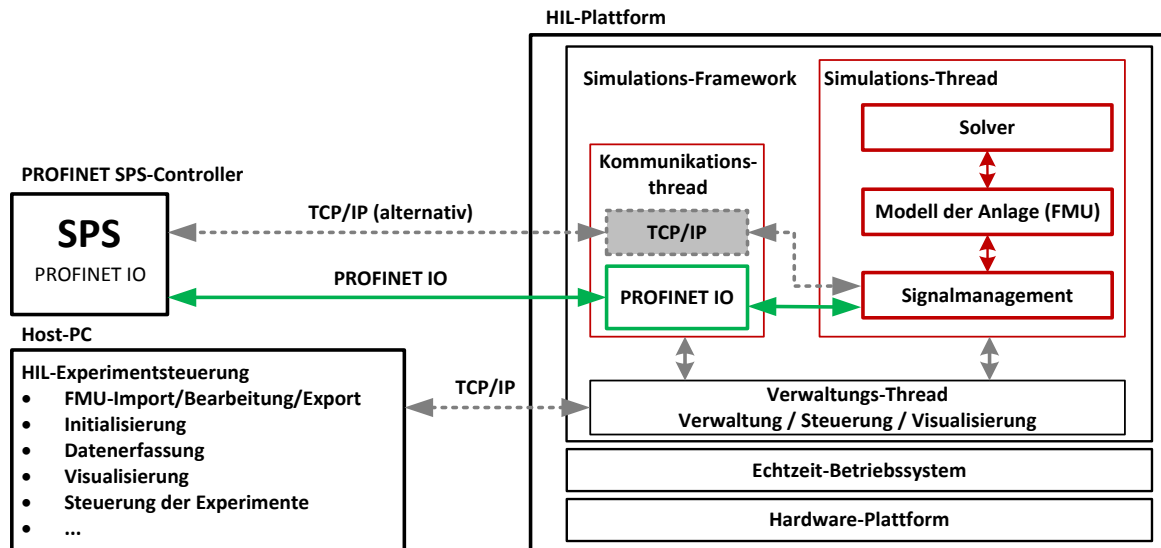


Abbildung 5-3: Aufbau der im Projekt „HIL-IMA“ entwickelten HIL-Simulationsumgebung gemäß [RPK¹⁴]

Aus Abbildung 5-3 wird ersichtlich, dass die über das FMI bereitgestellten Signale (hier: das Prozessabbild) von und zur SPS übertragen werden müssen. Dies wird mittels eines von der eigentlichen Simulation unabhängigen Kommunikations-Threads sichergestellt. Eine Erklärung dazu findet sich im folgenden Abschnitt.

5.3.3.3 Kommunikation zwischen SPS und Simulationsframework

Die Anbindung der SPS an die Simulation ist bei einer HIL-Simulation zwingend notwendig (→ Abbildung 5-2). Unter der Prämisse, dass die SPS nicht den Unterschied zwischen realer Anlage und Simulationsumgebung „bemerken“ darf, bedingt dies in einer HIL-Umgebung eine Emulation der in der SPS projektierten Feldgeräte. In der Praxis ist dieses Szenario derzeit nicht immer durchführbar: Situationen, in denen das Prozessabbild bereits festgelegt wurde und bereits einen hohen Reifegrad aufweist, die Projektierung und Konfiguration der Feldbusteilnehmer aber noch nicht abgeschlossen ist, können nicht ausgeschlossen werden. Bei vorhandenem Prozessabbild könnte so zwar bereits prinzipiell eine virtuelle Inbetriebnahme durchgeführt werden, andererseits wäre die Verwendung einer HIL-Plattform nicht möglich.

Um die Verwendung der hier beschriebenen HIL-Plattform auch in Fällen zu gewährleisten, in denen eine Emulation des Feldbusses nicht möglich ist, lassen sich mehrere Alternativen identifizieren, mit denen eine SPS unter Verwendung der in Abbildung 5-3 beschriebenen HIL-Simulationsumgebung gekoppelt werden kann. Im Projekt „HIL-IMA“ wurden die folgenden drei Möglichkeiten für eine Kopplung des Simulationsmodells an eine SPS in Betracht gezogen:

- *Alternative 1: Emulation des Feldbusses.* Alle Feldbusteilnehmer werden in Alternative 1 mittels Emulation nachgebildet. Voraussetzung für dieses Verfahren ist einerseits das Vorhandensein einer konsistenten und auswertbaren SPS-Hardwarekonfiguration. Andererseits muss eine geeignete Implementierungsplattform zur Verfügung stehen, die es erlaubt, die Feldgeräte mittels Emulation nachzubilden. Die Emulation muss den technischen Spezifikationen des eingesetzten Feldbusses genügen, was für PROFINET IO z. B. eine harte Echtzeitfähigkeit von 1 bis 10ms Taktrate bedeutet, je nach verlangter Echtzeitklasse [PROFINET[®]].

- Alternative 2: Anbindung mittels TCP/IP.* Steht die finale Hardwarekonfiguration noch nicht zur Verfügung, ist eine Emulation des Feldbusses nicht möglich. Um die HIL-Simulationsplattform bei bereits vorhandenem und weitgehend stabilem Prozessabbild dennoch mit der SPS verbinden zu können, kann die SPS mittels TCP/IP-Protokoll gekoppelt werden. Dies entspricht streng genommen keinem echten HIL Einsatzszenario, ermöglicht aber einen einfachen Austausch des Prozessabbildes. TCP/IP ist nicht hart echtzeitfähig [PAU12]. Änderungen am Steuerungscode sind zudem notwendig: Zur Implementierung einer derartigen Verbindung müssen direkt nach dem Einlesen des Eingangs-Prozessabbildes sowie unmittelbar vor dem Schreiben des Ausgangs-Prozessabbildes SPS-Bausteine ergänzt werden. Diese Änderungen betreffen aber nicht den zu testenden Teil des Steuerungscode. Eine Untersuchung hinsichtlich der entstehenden Zykluszeitverfälschung ergab eine Abweichung im Bereich 1 ms . Ein möglicher Vorteil ist, dass die TCP/IP-Kommunikation deutlich ressourcenschonender als eine Emulation von Busteilnehmern ist. Sie ist daher vor allem bei größeren, rechenintensiveren Anlagenmodellen eine mögliche Alternative, wenn das Busverhalten hinsichtlich Echtzeitfähigkeit und Nachbildung der Busteilnehmer keine Rolle spielt. Abbildung 5-4 zeigt das grundlegende Schema des Datenaustausches bei einer TCP-Kommunikation am Beispiel eines 200 Byte langen Prozessabbildes. Hinsichtlich der Verwendung von TCP/IP und der in Abschnitt 5.3.3.2 beschriebenen notwendigen Synchronisation zwischen Steuerung und Simulationssystem sei angemerkt, dass die Schrittweite der Simulation t_s an die maximale Zyklusdauer der TCP/IP-Kommunikation angepasst werden muss. Schwankt der TCP/IP Kommunikationsthread-Zyklus zwischen t_{min} und t_{max} , muss die Geschwindigkeit der Simulation auf eine Schrittweite von t_{max} eingestellt werden. Der Zyklus des TCP/IP-Kommunikationsthreads korreliert mit der Größe des zu übertragenen Prozessabbildes. Können bei der Übertragung kleiner Prozessabbilder (z. B. 200 Byte) noch Zykluszeiten kleiner oder gleich 1 ms erreicht werden, steigt die Zykluszeit bei Verwendung großer Prozessabbilder im Kilobyte-Bereich schnell auf mehrere Millisekunden an.

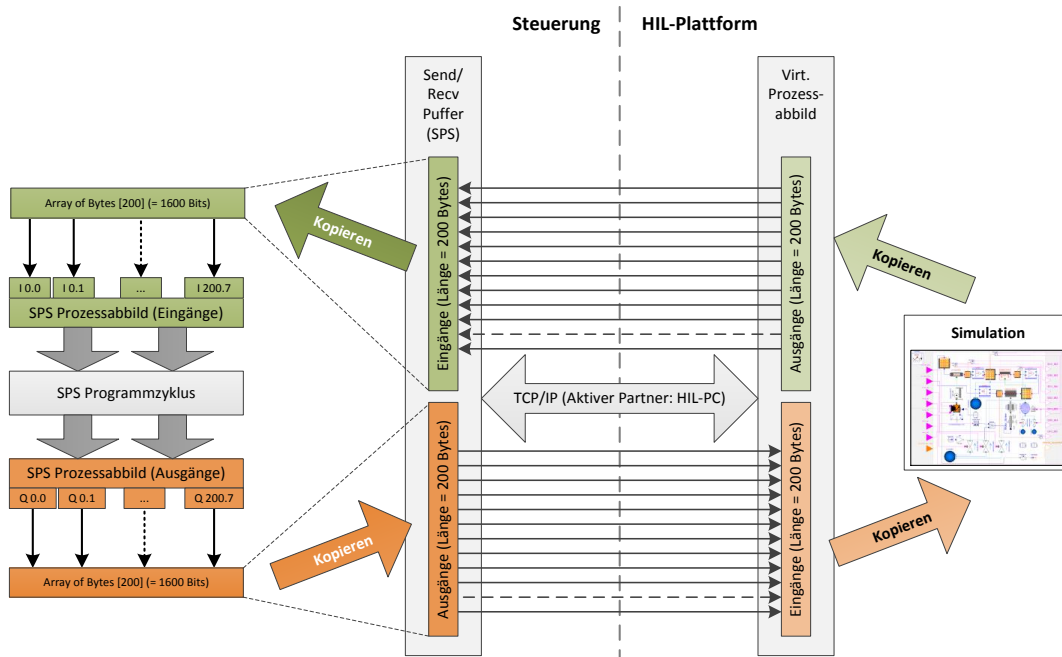


Abbildung 5-4: Schematische Darstellung der TCP/IP Kopplung HIL-Simulationsplattform/SPS

- *Alternative 3: Anbindung mittels OPC UA.* Über einen OPC-Server können Prozessabbild-daten zwischen einer Steuerung und einem beliebigen anderen Teilnehmer ausgetauscht werden. OPC UA ist nicht für echtzeitfähige Anwendungen geeignet [OPCUATSN[®]]. Derzeit befindet sich mit OPC UA TSN eine Lösung in der Entwicklung, die es ermöglichen soll, mittels OPC UA echtzeitfähig kommunizieren zu können [OPCUATSN[®]]. Eine Aussage, ob OPC UA TSN eine harte Echtzeitfähigkeit von z. B. 1 ms erreichen wird, kann derzeit aber nicht getroffen werden. Eine Anwendung von OPC UA im Kontext der virtuellen Inbetriebnahme wurde bereits von [BAR11] vorgestellt und wird in dieser Arbeit nicht weiter vertieft.

Für die folgenden Betrachtungen wird die als Alternative 1 genannte Lösung, entsprechend der Ziele des Projekts „HIL-IMA“, als gegeben vorausgesetzt. Alternative 2 wurde prototypisch implementiert und hat sich für den Einsatz mit kleinen Prozessabbildern bewährt.

6 Datenmodell für eine virtuelle Inbetriebnahme

In diesem Kapitel werden die Rahmenbedingungen einer virtuellen Inbetriebnahme hinsichtlich eines als Basis der Modellerstellung nutzbaren Datenmodells näher beschrieben. Hierzu werden in Abschnitt 6.1 bzw. 6.2 die als *Bausteine 1.1 (Abbildung des Datenmodells der Digitalen Fabrik)* und *1.2 (Erweiterung des Datenmodells um für die virtuelle Inbetriebnahme relevante Aspekte)* erarbeiteten Lösungen im Detail beschrieben. Das nachfolgende Kapitel steht unter Leitfrage I: *Welche Daten werden für eine virtuelle Inbetriebnahme benötigt und wie sollten diese abgebildet sein?*

6.1 Durchgängiges Datenmanagement im Kontext einer virtuellen Inbetriebnahme

Eine Kernforderung des *Digitalen Fabrikbetriebs* ist ein *durchgängiges Datenmanagement*, das einen Austausch von Engineering-Daten zwischen den am Engineering-Prozess beteiligten Gewerken ermöglicht. Die folgenden Betrachtungen konzentrieren sich in diesem Kontext insbesondere auf die Abbildung einer *Anlagenstruktur* innerhalb des *Datenmodells der Digitalen Fabrik*. Vor der Definition spezifischer Anforderungen und Abbildungsvorschriften, die als Konkretisierung der in [VDI 4499-1] definierten Anforderungen zu verstehen sind, wird zunächst die Bedeutung des im *Datenmodell* enthaltenen *Anlagenstrukturmodells* für die Erstellung eines Simulationsmodells herausgearbeitet (→ Abschnitt 6.1.1). Die Abschnitte 6.1.2 und nachfolgende widmen sich im Speziellen der Darstellung dieser Anlagenstruktur und der systematischen und strukturierten Einbettung selbiger in den Kontext *Datenmodell der Digitalen Fabrik*. In den folgenden Abschnitten wird daher die Frage beantwortet, *was* für eine Simulation im Kontext der virtuellen Inbetriebnahme *notwendig ist*.

6.1.1 Bedeutung des Anlagenstrukturmodells für die Simulationsmodellerstellung

Allgemein beschreibt der Begriff der Modellerstellung (auch: Modellbildung) im Kontext einer Simulationsstudie den Vorgang, dass für eine Untersuchung genutzte Simulationsmodell zu erzeugen [KRNE98]. Modelle im Allgemeinen (und Simulationsmodelle im Speziellen) können Repräsentationen natürlicher und künstlicher Originale sein, d. h. ihrerseits auf Modellen basieren ([STA73], zitiert nach [DIMÜ11]). Es ist in diesem Zusammenhang zunächst unerheblich, wie die Modellerstellung im konkreten (manuell, automatisiert) vorgenommen wird. Kennzeichnend für die Simulation technischer Systeme ist es, dass die Struktur des Simulationsmodells zumeist der des realen, zu simulierenden Systems entspricht. Dies ist in Zusammenhang mit der Aussage zu verstehen, nach der Simulation auch immer den inneren Mechanismus eines Systems betrachtet ([EHR84], [KRNE98]). In diesem Zusammenhang formuliert [GLÖ14]: „Bei der strukturtreuen Nachbildung werden die grundlegenden mathematischen Zusammenhänge des realen Objekts im Modell implementiert und bilden damit das prinzipielle Verhalten im Rahmen des erforderlichen Detaillierungsgrades [...] ab“. Es ist daher die Systemstruktur, die (kombiniert mit dem gerätespezifischen Verhalten einzelner Objekte innerhalb der Struktur) das Verhalten eines zu untersuchen-

den, technischen Systems (wenn gesamtheitlich betrachtet) bestimmt [Bos04]. Die Zusammenhänge zwischen einzelnen, in einer Simulation nachgebildeten Objekten werden in diesem Kontext als Wirkbeziehungen bezeichnet [Bos04]. Die Objekte eines Systems und deren Wirkbeziehungen untereinander müssen für eine effiziente weil aufwandsarme Modellerstellung in geeigneter Art vorhanden sein. Das bedeutet: Sie sollten nicht aus einem vorhandenen System vollständig manuell ermittelt werden müssen. Mögliche Szenarien, in denen lediglich das Verhalten des realen Objekts mittels Gleichungen abgebildet wird, ohne die Struktur des nachzubildenden technischen Systems zu beachten, beschränken sich auf wenige Ausnahmefälle. Dies ist immer dann der Fall, wenn die Struktur des realen Objekts entweder nicht bekannt ist oder der Aufwand für die Nachbildung der Struktur zu groß wäre [Glö14]. Derartige, rein gleichungsbasierte Modelle basieren auf Verhaltensbeobachtungen, z. B. in Form von aufgenommenen Messreihen [Glö14]. Da eine virtuelle Inbetriebnahme in der Praxis in den meisten Fällen vor oder unmittelbar während der (physischen) Realisierung einer Anlage stattfindet, kann die reale Anlage und ihr spezifisches Verhalten folglich nicht als Grundlage für die Modellerstellung durch Verhaltensbeobachtungen genutzt werden. Derartig umgesetzte Modelle wären reine Annahmen über die Funktionsweise einer Gesamtanlage, das Ergebnis rein spekulativ [Glö14] und damit nicht für eine virtuelle Inbetriebnahme geeignet. Davon unberührt kann ein derartiges Konzept immer dann zur Anwendung kommen, wenn es um die konkrete Modellierung einzelner, in der Anlage vorhandener Geräte geht, die bereits außerhalb dieser Anlage physikalisch existieren und deren Verhalten bekannt ist. Diese Gerätemodelle werden dann in einem Anlagensimulationsmodell in größerem Zusammenhang verwendet.

Aus diesen Aussagen lässt sich zunächst folgern, dass eine notwendige Voraussetzung für eine Modellerstellung eines Anlagensimulationsmodells eine für diesen Zweck geeignete Abbildung der Struktur und der Relationen der einzelnen in der Struktur vorhandenen Objekte (im Sinne der Wirkbeziehungen untereinander) des zu simulierenden Systems ist. Von Vorteil ist in diesem Kontext die Zusammenführung aller vorhandenen Modelle, die während des Engineering-Prozesses entstehen, in ein gesamtheitliches Datenmodell, damit nur eine Datenquelle genutzt werden muss (für die Vorteile eines gesamtheitlichen Datenmodells sei auf Abschnitt 2.3.2 verwiesen). Weiter lässt sich folgern, dass speziell in einer virtuellen Inbetriebnahme der eigentliche Vorgang der Modellerstellung im Wesentlichen die Transformation der Struktur eines Anlagenmodells hin zu einem neuen Modell bedeutet: dem Simulationsmodell der Anlage. Die in Modellen (= Projektergebnisse in Form von Anlagenplanungsdaten) projizierten Objekte wie Sensoren, Aktoren, Förderbänder, Roboter und weiteren werden mit für die Simulation dieser Objekte geeigneten Modellen substituiert und gemäß der im Anlagenstrukturmodell abgebildeten Wirkbeziehungen miteinander verbunden. Ergänzend sei hinzugefügt, dass die bisherige Verwendung des Begriffs „Objekte“ nicht impliziert, dass die Darstellung stets nach einem objektorientierten Schema erfolgt – auch wenn dies für das Engineering gefordert wird [VDI 4499-1].

Wird die Simulationsmodellerstellung rein manuell vorgenommen, kann neben der aufwändigen, rein manuellen Übertragung der Anlagenstruktur sowie der manuell vorzunehmenden Identifizierung passender Zuordnungen (Anlagenstrukturobjekt \leftrightarrow Simulationsmodell) möglicherweise zusätzlicher Aufwand nötig sein, um die Wirkbeziehungen zwischen einzelnen Objekten einer Anlage zu identifizieren. Gelingt es in diesem Zusammenhang, die einzelnen Anlagenplanungsdaten in einem rechnerauswertbaren Datenmodell abzubilden (das sowohl die Struktur als auch die Wirkbeziehungen von in der Struktur enthaltenen Elementen abbilden kann), würde die Modellerstellung, unabhängig von der Art ihrer tatsächlichen Ausführung, insgesamt profitieren.

Vorteilhaft wäre zudem eine das Datenmodell ergänzende und nutzende Methodik, mit der sich die Zuordnung „Anlagenstrukturobjekt \leftrightarrow Simulationsmodell“ unterstützen ließe. Im Kontext einer möglichen automatischen Modellgenerierung kann beides als eine Grundvoraussetzung gesehen werden. Hierfür ist zunächst das Zusammenführen der einzelnen Anlagenplanungsdaten in ein gesamtheitliches *Datenmodell der Digitalen Fabrik* sowie eine strukturell einheitliche und vor allem standardisierte Abbildung der Daten notwendig.

6.1.2 Mögliche Datenquellen für das Datenmodell der Digitalen Fabrik hinsichtlich einer durchzuführenden virtuellen Inbetriebnahme

Um ein durchgängiges Datenmanagement zu gewährleisten und damit die Darstellung einzelner Anlagenplanungsdaten in einem Anlagenstrukturmodell zu ermöglichen, sind mehrere Faktoren von Bedeutung. In [VDI 3695-4] werden in diesem Kontext vier Zielzustände der *Interoperabilität* genannten Fähigkeit, Daten zwischen Engineering-Werkzeugen effizient (d. h. ohne aufwändige, manuelle Nachbearbeitung) austauschen zu können, definiert [DFB11]:

- *Zielzustand A*: Planungsdaten einer Anlage oder Aspekte davon liegen im Rechner vor, sind aber nicht rechnergestützt auswertbar. (Beispiel: PDF-Dateien).
- *Zielzustand B*: Die Planungsdaten einer Anlage liegen als rechnergestützt auswertbare Daten vor. Die jeweiligen Formate sind von Projekt zu Projekt gleich. (Beispiel: Verwendung eines standardisierten Datenaustauschformats)
- *Zielzustand C*: Die Planungsdaten einer Anlage können direkt zwischen den Engineering-Werkzeugen ausgetauscht werden. (Beispiel: Die in Zielzustand 2 definierten Daten lassen sich in ein Engineering-Werkzeug importieren, d. h. das „importierende“ Engineering-Werkzeug ist in der Lage, die hinterlegten Daten nicht nur zu lesen, sondern durch standardisierte Semantik auch „richtig“ interpretieren zu können).
- *Zielzustand D*: Die Planungsdaten einer Anlage werden bei Änderungen automatisch zwischen den beteiligten Werkzeugen ausgetauscht.

Nach [DFB11] können die Zielzustände A und B heute als erreicht angesehen werden. Zielzustand D wird nur in integrierten, in sich geschlossenen Toolsuiten erreicht. In einer heterogenen Werkzeuglandschaft, wie sie im Anlagenbau vorherrscht, scheitert die Erreichung von Zielzustand C derzeit an der „[...] Vielfalt der Dateiformate mit individuellen (sich teilweise fortwährend ändernden) Datenmodellen“ [DFB11]. Eine besondere Bedeutung bei einem durchgängigen Datenmanagement kommt daher der im Datenmodell – *im Konkreten* – ausgeführten Abbildung von Anlagenplanungsdaten hinsichtlich Struktur, Hierarchie, Granularität und einer geeigneten Abbildung von Wirkbeziehungen zu. Tabelle 7 zeigt in diesem Zusammenhang, welche Daten von welchem Gewerk für die Erstellung eines gesamtheitlichen Datenmodells infrage kommen könnten. Die Betrachtung in Tabelle 7 erfolgt im Speziellen im Hinblick auf die für eine virtuelle Inbetriebnahme benötigten Anlagenplanungsdaten und stellt daher nur einen Ausschnitt aus möglichen Anwendungsfeldern der *Digitalen Fabrik* dar.

Aspekt:	Auftraggeber		Gewerk Mechanik				Gewerk Elektrotechnik		Gewerk Automatisierungstechnik	
	Groblayout (vgl. Abschnitt 2.5.1)	Layouts (vgl. Abschnitt 2.5.1)	Konstruktionsdaten (vgl. Abschnitt 2.5.1)	"Andere Daten" wie z.B. Prozessbeschreibungen (vgl. Abschnitt 2.5.1)	EPLAN (vgl. Abschnitt 2.5.2)	IEC 61131-3 Code (vgl. Abschnitt 2.5.3)	Visualisierungsdaten (vgl. Abschnitt 2.5.3)			
Anlagenelemente I: Aktive Elemente: Sensoren, Aktoren	I.d.R. nicht vorhanden, ggf. als Mengengerüst vorgegeben	- Als Entitäten teilweise vorhanden, grobgranular	O Als Entitäten vorhanden, feingranular	O I.d.R. nicht vorhanden, BMKs* können Teil der Beschreibung sein	- BMK* vorhanden, inkl. Adressinformationen	O Wird aus EPLAN abgeleitet, daher identisch mit Gewerk Elektrotechnik	X	Nicht relevant, da nur grafische Darstellung	X	
Anlagenelemente II: Passive Elemente: Mechanische Apparaturen	Als Entitäten vorhanden	O Als Entitäten teilweise vorhanden, grobgranular	O Als Entitäten vorhanden, feingranular	O I.d.R. nicht vorhanden, BMKs* können Teil der Beschreibung sein	- Nicht vorhanden, nur indirekt erkennbar**	- Nicht vorhanden, nur indirekt erkennbar**	-	Nicht relevant, da nur grafische Darstellung	X	
Anlagenelemente III: Ortsangabe der Anlagen- elemente	I.d.R. nicht vorhanden	- Als BMK* teilweise vorhanden (Aktoren, mechanische Apparaturen)	O Als BMK* vorhanden (Sensoren, Aktoren, mechanische Apparaturen)	+ I.d.R. nicht vorhanden, BMKs* können Teil der Beschreibung sein	- Als BMK* vorhanden (Sensoren, Aktoren, indirekt daraus auch mechanische Apparaturen erkennbar**)	O Wird aus EPLAN abgeleitet, daher identisch mit Gewerk Elektrotechnik	X	Nicht relevant, da nur grafische Darstellung	X	
Anlagenelemente IV: Objekte aus Bibliothek instanzierbar	I.d.R. nicht gegeben, Werkzeugabhängig, Vorgangabhängig***	- I.d.R. nicht gegeben, Werkzeugabhängig, Vorgangabhängig***	- I.d.R. nicht gegeben, Werkzeugabhängig, Vorgangabhängig***	- Nicht relevant	X I.d.R. nicht gegeben, Werkzeugabhängig	- Nicht relevant	X	Nicht relevant, da nur grafische Darstellung	X	
Prozessablauf I: Materialfluss (Anlagenweit)	Ergibt sich implizit aus der Struktur der Anlage, explizit meist nur in Prosaform	O Ergibt sich implizit aus der Struktur der Anlage, Verbindungen i.d.R. nicht explizit abgebildet	O Ergibt sich implizit aus der Struktur der Anlage, Verbindungen i.d.R. nicht explizit abgebildet	O Ggf. als Excel-Tabellen, Zeichnungen (z.B. Visio), Videos oder in Prosaform vorhanden	O Nicht relevant	X Implizit im Code vorhanden	-	Nicht relevant, da nur grafische Darstellung	X	
Prozessablauf II: Mechanische Apparaturen (Anlagenteile, z.B. Schraub- stationen)	Meist nur in Prosaform, kaum als formales Modell	O Nicht vorhanden	- Nicht vorhanden	- Ggf. als Excel-Tabellen, Videos oder in Prosaform vorhanden	O Nicht vorhanden	- Implizit im Code vorhanden	-	Nicht relevant, da nur grafische Darstellung	X	
Parameter I: Physik, Geometrie	Nicht oder nur teilweise vorhanden (insb. geometrische Angabe wie Länge, Breite, Höhe)	- Teilweise vorhanden (insb. geometrische Angabe wie Länge, Breite, Höhe)	O Teilweise vorhanden (insb. geometrische Angabe wie Länge, Breite, Höhe)	O Können, insofern als Prozessparameter relevant, z.B. in Exceltabellen vorhanden sein (insb. Geschwindigkeiten, Beschleunigungen, etc.)	O Nicht relevant	X Nicht vorhanden	-	Nicht vorhanden	-	
Parameter II: Geräteparameter	Nicht relevant	X Nicht relevant	X Nicht relevant	X Nicht vorhanden	- Nicht vorhanden	- Im Code und Konfiguration hinterlegt, basiert auf Annahmen	-	Nicht relevant, da nur grafische Darstellung bzw. Editiermaske	-	
Verhalten (nur für VIBN): Verhalten von Geräten (z.B. Frequenzrichtern, intelligente Feldgeräte, weitere)	Nicht relevant	X Nicht relevant	X Nicht relevant	X Nicht vorhanden	- Nicht vorhanden	- Ggf. aus Ansteuerungsbaustein interpolierbar, ansonsten: Nicht vorhanden	-	Nicht vorhanden	-	

O: Als Datenquelle nur in Verbindung mit anderen Aspekten geeignet; +: Als Datenquelle geeignet; -: als Datenquellen nicht geeignet; X: Nicht relevant

*BMK: Betriebsmittelkennzeichen; ** Indirekt erkennbar bedeutet, dass ein Ingenieur nur durch die Anwendung von (Erfahrungs-)Wissen darauf schließen kann, dass es sich bei einem abgebildeten Element um z.B. ein Förderband handelt;

*** Vorgangabhängig bedeutet, dass die Konstruktion sowohl objektorientiert, aber auch rein feature-basiert erfolgen kann (vgl. Abschnitt 2.5.1)

Tabelle 7: Nutzbare Anlagenplanungsdaten für die Abbildung eines Anlagenstrukturmodells

In Bezug auf Tabelle 7 sei angemerkt, dass die dort genannten „Entitäten“ bedeuten, dass die Abbildung dieser Anlagenelemente in den Projektergebnissen nicht zwangsläufig als Objekte hinsichtlich einer objektorientierten Sichtweise erfolgt (→ Abschnitt 2.5). Es kann sich um Objekte, aber auch nur um räumlich oder funktionell zusammenhängende Linien oder Geometrien handeln, die ein Anlagenelement (z. B. ein Förderband) in abstrakter Form repräsentieren sollen.

Aus den Darstellungen in Tabelle 7 lässt sich zunächst folgern, dass es keine singuläre Datenquelle gibt, die sich als substanzielle Datenquelle für die Abbildung einer Anlagenstruktur und der dort beinhalteten Wirkbeziehungen eignet. Vielmehr müssen unterschiedliche Daten/Artefakte der am Engineering beteiligten Gewerke zusammengeführt werden [VDI 4499-1]. Ein möglicher Workflow dazu wird in [LSJ16] beschrieben. Erst dies ermöglicht die Nutzung des Datenmodells als Datenquelle für eine virtuelle Inbetriebnahme. Hieraus lassen sich drei Teilaufgaben ableiten:

1. Die erste Teilaufgabe sollte die Frage, *woher* die Anlagenplanungsdaten kommen, beantworten.
2. Die zweite Teilaufgabe besteht darin, die vorliegenden Planungsdaten so zu verarbeiten, dass sie in einem (objektorientierten) *Datenmodell der Digitalen Fabrik* abgebildet werden können. Hierfür können den Datentransfer unterstützende *Interoperabilitätsfunktionen* definiert werden [DFB11], um einen effizienten, d. h. aufwandsreduzierten Datentransfer zu gewährleisten. Teilaufgabe 2 beantwortet die Frage nach dem *womit* Daten interpretiert werden können, um im *Datenmodell der Digitalen Fabrik* abgebildet zu werden.
3. Die dritte Teilaufgabe besteht darin, geeignete Abbildungsvorschriften zu definieren, die beschreiben, wie die Daten *im konkreten* im Datenmodell abgelegt werden sollten. Diese Abbildungsvorschriften sollten sich als *Konkretisierung* des Konzepts des *Datenmodells der Digitalen Fabrik* verstehen und nicht als alternativer Vorschlag. Im Wesentlichen beantwortet dies die Frage, *wie* Daten hinsichtlich Struktur, Granularität und Wirkbeziehungen abgebildet werden können. Dies versteht sich losgelöst von der Frage, welches Datenformat im konkreten zum Einsatz kommen sollte.

In dieser Arbeit wird, in Übereinstimmung mit dem definierten Lösungsbaustein 1.1, auf die in Teilaufgabe 3 beschriebenen Abbildungsvorschriften, die das *Datenmodell der Digitalen Fabrik* konkretisieren sollen, fokussiert. Grundsätzlich stellt sich in diesem Kontext die Frage nach der Bearbeitungsreihenfolge hinsichtlich der beschriebenen Teilaufgaben. Im Konkreten, ob Anlagenplanungsdaten überhaupt verfügbar sind und dahin gehend interpretiert werden können, um in das *Datenmodell* transferiert werden zu können. Hierin liegt ein allgemeines Hauptproblem der *Digitalen Fabrik*. *Interoperabilität*, die der Lösung von Teilaufgabe 1 und 2 dient, kann im Kontext der *Digitalen Fabrik* nur durch die eindeutige Definition der Datenstrukturen von Attributen, Objekten und Relationen erfolgreich sein [DFB11] (ausdrücklich nicht gemeint ist in diesem Kontext die konkrete Abbildung in einem spezifischen Datenformat, z. B. mittels einer Meta-Sprache). Hierfür ist das Vorhandensein von gemeinsamen Standards notwendig ([DLH11], [VDI 4499-2]). Daraus kann gefolgert werden, dass für den jeweiligen Anwendungsfall geeignete Abbildungsvorschriften eine Grundvoraussetzung für das Etablieren von *Interoperabilitätsfunktionen* sind. Im Kontext einer virtuellen Inbetriebnahme kommt insbesondere der Abbildung der Anlagenstruktur eine entscheidende Rolle zu, wie in Abschnitt 6.1.1 dargelegt wurde. Im folgenden Abschnitt werden daher die Teilaufgabe 3 thematisiert, konkrete Abbildungsvorschläge beispielhaft beschrieben und mittels generischer Abbildungsvorschriften fixiert. Ferner wird ein Beitrag zur Erreichung von *Zielzustand C* der in [VDI 3695-4] beschriebenen Zielzustände der *Interoperabilität* geleistet.

6.1.3 Anforderungen an das Datenmodell der zu simulierenden Anlage

Um das *Datenmodell der Digitalen Fabrik* abbilden zu können, ist die Abbildung von *Produkten*, *Prozessen* und *Ressourcen* (PPR) erforderlich (→ Abschnitt 2.3.2). Hierzu zählen auch die Verbindungen und Abhängigkeiten zwischen diesen Objekten [VDI 4499-1]. Die drei Aspekte *Produkte*, *Prozesse* und *Ressourcen* können innerhalb des Datenmodells zunächst als wesentliche Knoten identifiziert werden [VDI 4499-1]. „Knoten“ definieren sich in diesem Zusammenhang als ausgezeichnete Punkte innerhalb einer Hierarchie, die ein System gesamtheitlich abbilden. Das theoretische Konzept des *PPR-Prinzips* sowie grundlegende Anforderungen an das *Datenmodell der Digitalen Fabrik* wurden bereits in Abschnitt 2.3.2 vorgestellt, auf welches an dieser Stelle verwiesen sei. Im Folgenden konzentrieren sich die weiteren Ausführungen auf Anforderungen an das Datenmodell hinsichtlich der konkreten Abbildung eines Anlagenstrukturmodells.

6.1.3.1 Anforderung 1: Zweckdienliche Hierarchisierung der Anlagenstruktur im Knoten „Ressourcen“

Ein wichtiger Aspekt der Darstellung der Anlagenstruktur ist die zielgerichtete *Hierarchisierung* und damit letztendlich auch die Bestimmung der *Granularität* der abgebildeten Anlage und der in der Anlage beinhalteten Objekte sowie der Relationen dieser Objekte untereinander – unabhängig von der konkreten Ausführung der Modellerstellung. Die in der Anlagenstruktur beinhalteten Objekte sowie ihre Beziehungen untereinander beschreiben in diesem Kontext das eigentliche Gefüge innerhalb einer Anlage. Dieses *muss* eine Ordnung aufweisen, um die Struktur einer Anlage beschreiben zu können [HWF⁺12].

Eine hierarchisch-strukturierte Sichtweise ist in der Automatisierungstechnik kein neues oder ungewöhnliches Konzept. Im Speziellen für die Prozesstechnik wird eine Hierarchie in [IEC 61512-1] beschrieben. Grundsätzlich unterliegt *jede* automatisierte Anlage einer häufig schlicht implizit vorhandenen Hierarchie. Diese beschreibt den Aufbau des Systems „Anlage“. Aus dem *Systems Engineering* sind bereits seit Längerem mehrere Möglichkeiten zur strukturell-hierarchischen Darstellung technischer Systeme bekannt. Sie werden zumeist unter dem Begriff „*System of Systems*“ zusammengefasst [SHE94]. „*System of Systems*“ bedeutet vereinfacht, dass grundsätzlich jedes Element innerhalb eines Systems selber ein System sein kann [HWF⁺12]. Aus dieser Sichtweise ergeben sich verschiedenste Möglichkeiten, ein technisches System hierarchisch nachzubilden. Haskins et al. [HAS11] beschreiben in diesem Kontext den Aufbau einer Hierarchie, die die Unterteilung eines Systems in *Elemente*, *Subsysteme*, *Module*, *Komponenten* und *Teile* beinhalten kann (Abbildung 6-1).

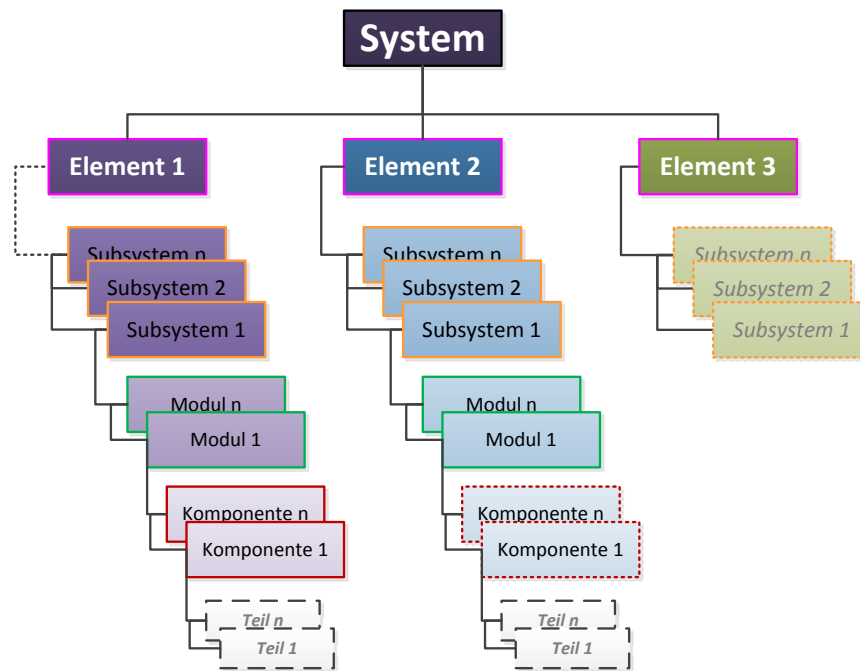


Abbildung 6-1: Hierarchie technischer Systeme in Anlehnung an [HAS11]

Die von [HAS11] beschriebene Hierarchie eignet sich sehr gut für die Darstellung der Struktur einer Anlage, wie nachfolgend gezeigt wird. So ist es möglich, unter Beachtung der mechatronischen Sicht auf eine Anlage, die wesentlichen Anlagenteile wie mechanische Grundsysteme und zugehörige Sensoren und Aktoren direkt einzuordnen. Ferner ermöglicht sie, einzelne Anlagenteile in einen größeren Kontext zu stellen – so, wie es in einer Anlage in der Praxis durch das Zusammenfassen einzelner mechatronischer Systeme zu z. B. Funktionsgruppen üblich ist. Im Folgenden wird die Einordnung einer Anlagenstruktur in das von [HAS11] vorgestellte Schema detailliert beschrieben.

Im Zuge einer hierarchischen Darstellung einer Anlagenstruktur spielt zunächst der Knoten *Ressourcen* eine wichtige Rolle. Grundlegend beinhaltet dieser alle für die Anlage wesentlichen Objekte einer Anlage. Dies können Betriebsmittel und Anlagenkomponenten sein [GÜFA08]. Aus Sicht der virtuellen Inbetriebnahme kann der Knoten *Ressourcen* und alle möglichen ihm untergeordneten Sub-Knoten daher das zu untersuchende System beschreiben. Der Knoten *Ressourcen* wird in diesem Kontext daher synonym mit dem zu untersuchenden System gesetzt. Sub-Knoten und damit einzelne Elemente des Systems können gemäß der Definition von [GÜFA08] *Anlagenstruktur*, *Steuerungsstruktur* und *Busstruktur* sein. Sie bezeichnen die wesentlichen, strukturbildenden Elemente einer automatisierten Anlage. *Anlagenstruktur* beschreibt in diesem Zusammenhang den Aufbau und das Gefüge der Anlage an sich – so, wie sie im Feld stehen soll und in den Anlagenplanungsdaten im Entwurf erarbeitet worden ist. Der bisher genutzte Begriff *Anlagenstrukturmodell* ist synonym mit dem Begriff *Anlagenstruktur* zu sehen. Typische Objekte wie einzelne Sensoren, Aktoren, mechanische Apparaturen sowie mögliche logische Zusammenfassungen dieser Objekte sind dort abgebildet. Dazu gehören auch die für die Objekte relevanten Parameter. Auf die Definition des Begriffs Parameter sowie der Abbildung dieser Parameter im Anlagenstrukturmodell wird in Abschnitt 6.2.2 gesondert eingegangen. *Steuerungsstruktur* beschreibt die in der SPS hinterlegte Hardwarekonfiguration (Geräte wie Buskoppler und andere), *Busstruktur* die Topologie des verwendeten Feldbusses. Die Steuerungsstruktur ist immer dann von Bedeutung, wenn das jeweilige Bussystem mittels Emulation nachgebildet wird

(→ Abschnitt 3.2.2). Im Folgenden wird die Steuerungsstruktur nur dahin gehend betrachtet, als dass sie ein Hilfsmittel für die Anbindung des Prozessabbildes an die in der Anlagenstruktur vorhandenen Sensoren und Aktoren darstellt. Die Busstruktur kann für eine virtuelle Inbetriebnahme vernachlässigt werden, solange keine Untersuchungen über das Busverhalten zu den möglichen Testscenarien gehören. In dieser Arbeit spielen diese keine Rolle, die Bustopologie wird daher nicht weiter betrachtet.

Der Sub-Knoten *Anlagenstruktur* kann gemäß der in Abschnitt 6.1.1 entwickelten Sichtweise als substantielle Quelle für die in der virtuellen Inbetriebnahme zu identifizierenden Simulationsobjekte genutzt werden. Das Objekt *Anlagenstruktur* kann aus hierarchischer Sicht zunächst in weitere Objekte (*Subsysteme*) zerlegt werden. Diese lassen sich in der Praxis als sogenannte *Funktionsgruppen* oder auch *Lastkreise* identifizieren und finden sich in fast jeder Anlage so wieder. Je nach internem Firmenstandard/angewandten Best-Practices sind weitere Bezeichnungen und/oder weitere (Zwischen-) Hierarchiestufen denkbar. Funktionsgruppen können „[...] als synergetische Integration verschiedener mechatronischer Module gesehen werden“ [VDI 2206-1]. Sie sind daher als logische Zusammenfassungen von hierarchisch tiefer liegenden (mechatronischen) Modulen zu (in der Praxis mal mehr, mal weniger) sinnvollen Gruppen zu verstehen. Eine Anlage besteht immer aus $n \geq 1$ Subsystemen, ein Subsystem immer aus $n \geq 1$ Modulen.

Module, als den Subsystemen untergeordnete Objekte, setzen sich aus Komponenten zusammen. Diese Objekte repräsentieren die Domänen Elektrotechnik und Mechanik eines mechatronischen Systems. Im Zusammenspiel mit der (im Falle der virtuellen Inbetriebnahme zu testenden) Informationsverarbeitung ergeben diese eine Einheit, die einen bestimmten Teilprozess innerhalb des Gesamtsystems ausführt. Module können daher als die wesentlichen Funktionsträger innerhalb einer Anlage gesehen werden. Komponenten bilden in diesem Zusammenhang die kleinsten Elemente, die für die sinnvolle Betrachtung der Granularität von Anlagenstrukturen bei der Durchführung einer virtuellen Inbetriebnahme infrage kommen. Eine Abbildung von Bauteilen kann dann sinnvoll sein, wenn diese Information für das Vorhalten von Ersatzteilen oder generell Verschleißteilen benötigt wird, und als Datenquelle das Anlagenstrukturmodell dienen soll. Ein Modul besteht gemäß der mechatronischen Sichtweise innerhalb der Anlagenstruktur immer aus mindestens $n \geq 3$ Komponenten (ein mechanisches Grundsystem, ein Sensor, ein Aktor), eine Komponente aus $n \geq 1$ Teilen.

Abbildung 6-2 zeigt die hier beschriebene hierarchische Darstellung einer Anlage an einem Beispiel. Die Anlagenstruktur besteht darin aus zwei Subsystemen (Funktionsgruppen =B1, =Bx), sowie insgesamt 3 Förderbändern als mechatronische Module (=B1+SC01, =B1+SC02, =B1+SC03). Das Förderband =B1+SC01 enthält in diesem Beispiel insgesamt drei Komponenten (mechanisches Grundsystem =B1+SC01-CV; Aktor =B1+SC01-A01; Sensor =B1+SC01-S01). Die Steuerungsstruktur und die Busstruktur sind in diesem Kontext als ergänzend zu betrachten und werden nicht weiter vertieft.

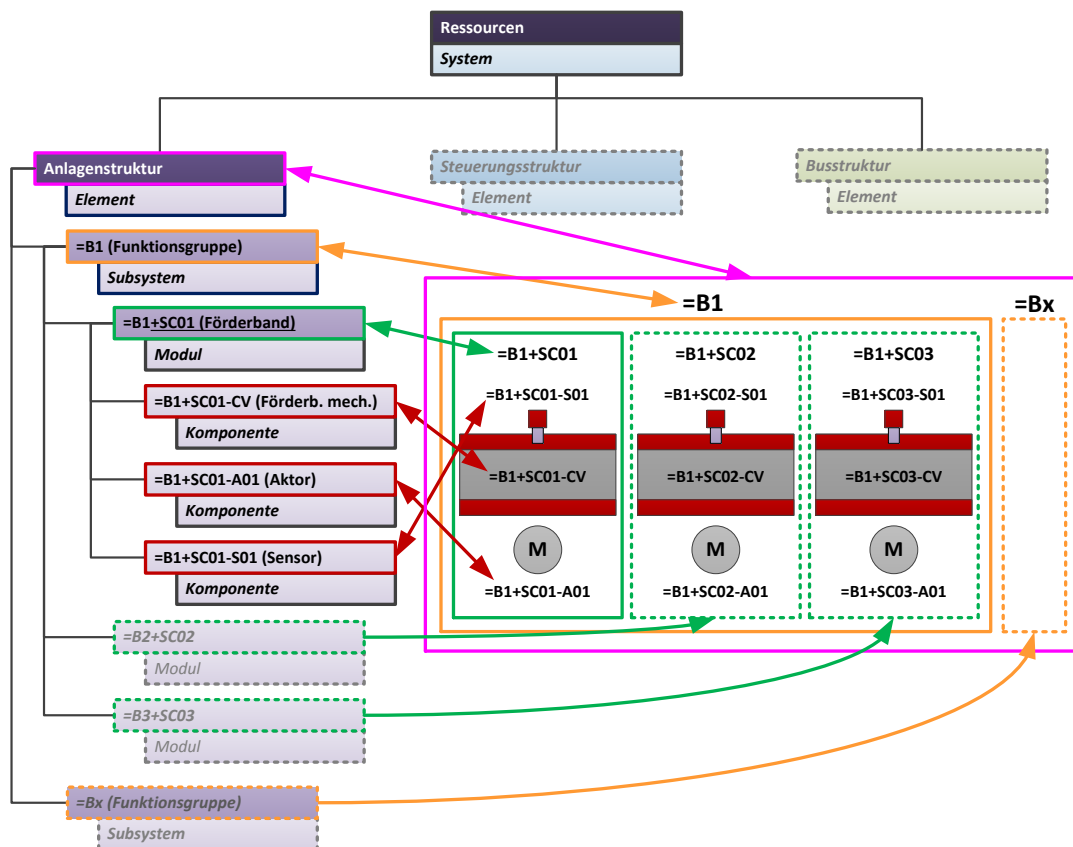


Abbildung 6-2: Beispielhafte Darstellung einer Anlagenhierarchie

Es sei angemerkt, dass eine klare hierarchische Definition einer Anlage, wie hier idealtypisch dargestellt, weder immer möglich noch immer sinnvoll ist. Einzelne mechatronische Module an sich könnten so komplex sein, dass sie gemäß der „System Of Systems“ Sichtweise selber als komplette Systeme klassifiziert werden können. Ein typisches Beispiel aus der Automatisierungstechnischen Praxis wären z. B. Elektro-Hängebahnen [VDI 3643], oder auch Fahrerlose Transportsysteme [VDI 2510]. Einzelne Gehänge oder Transportfahrzeuge können bei diesen Anlagentypen so komplex sein, dass sie selber aus vielen einzelnen, mechatronischen Modulen bestehen. Nicht ungewöhnlich ist es, dass Gehänge oder Transportfahrzeuge aufgrund der Vielzahl an verbauter Sensorik und Aktorik über eine eigene Steuerung verfügen. Aus Gesamtsystemsicht wären einzelne Gehänge oder Transportfahrzeuge daher eigenständige, in sich geschlossene Systeme mit definierten Schnittstellen zum Gesamtsystem „Anlage“. Als Konsequenz muss die hier beschriebene Hierarchie so angepasst werden, dass das Gefüge des Gesamtsystems für den Anwendungsfall geeignet strukturiert wird. Ein Beispiel hierfür findet sich in Kapitel 9.

6.1.3.2 Anforderung 2: Geeignete Abbildung von fertigungstechnischen Prozessen

Der Abbildung von fertigungstechnischen Prozessen wird in der Praxis keine hohe Bedeutung zugemessen (→ Abschnitt 2.5.1, letzter Absatz). Prozessbeschreibungen beziehen sich in der Fertigungstechnik zumeist auf Materialflussapplikationen, Handhabungs- und Montagetätigkeiten oder auf die Bearbeitung eines Werkstücks [MBS⁺11]. Ein Vergleich mit der Prozesstechnik zeigt die bedeutenden Unterschiede bei der Betrachtung von Prozessen: Dort kommt der strukturell formalisierten Abbildung von Prozessen eine wichtige Rolle zu [UFF⁺08]. Verfahrenstechnische Prozesse laufen in der Regel selbstständig nach physikalischen und chemischen Gesetzen ab und können üblicherweise nicht beliebig an- und abgeschaltet werden [MBS⁺11]. Der verfahrenstech-

nische Prozess wird mit Hilfe von Sensorik und Aktorik dahin gehend geregelt, dass die definierten Grenzen (z. B. eine definierte Maximaltemperatur) des (verfahrenstechnischen) Prozesses eingehalten werden. Verfahrenstechnische Prozesse laufen in der Regel langsamer ab als fertigungstechnische Prozesse. Die Temperaturänderung eines Stoffs kann mitunter im Minutenbereich liegen. Die Steuerung kann in diesem Fall mit mehreren Sekunden Abweichung auf die Temperaturänderung reagieren, ohne den Prozess an sich zu gefährden. Chargenprozesse können mit Hilfe einer Rezeptursteuerung derart angepasst werden, dass unterschiedliche Produkte auf einer Anlage erzeugt werden können.

Im Gegensatz dazu sind Prozesse in fertigungstechnischen Anlagen von mechatronischen Gesichtspunkten geprägt (→ Abschnitt 2.2.1, erster Absatz, vgl. auch Abschnitt 2.4.1). Prozesse in einer fertigungstechnischen Anlage laufen in der Regel nicht selbstständig, sondern nur bei direkter Ansteuerung ab. Diskrete Prozesse können daher im Unterschied zu verfahrenstechnischen Prozessen prinzipiell an beliebiger Stelle unterbrochen werden. Wird im Steuerungscode die Ansteuerung von Aktoren aufgrund festgelegter und fest programmierter Kriterien beendet, wird auch der durchgeführte diskrete Prozess unterbrochen. Die Reaktionszeit des Systems muss im Vergleich zu einer verfahrenstechnischen Anwendung daher deutlich niedriger liegen (in der Praxis meist im Millisekunden-Bereich).

Der Prozesstechnik und der Fertigungstechnik ist gemeinsam, dass sich Prozesse, die in einer Anlage durchgeführt werden, meist in viele einzelne Teilprozesse unterteilen lassen. Um die Beschreibung von Prozessen einer fertigungstechnischen Anlage genauer untersuchen zu können, ist es vorteilhaft, zu bestimmen, welche Teilprozesse welcher Hierarchieebene einer Anlage zugewiesen werden können. Eine Möglichkeit zur hierarchischen Einordnung von Prozessen bietet das Prozessmodell der [IEC 61512-1]. Das Prozessmodell der [IEC 61512-1] ermöglicht die Dekomposition eines Anlagen-Gesamtprozesses nach folgendem Schema: Ein *Gesamtprozess* besteht aus n *Prozessabschnitten*. Ein einzelner *Prozessabschnitt* kann aus n *Prozessoperationen* bestehen. Eine *Prozessoperation* kann wiederum aus n *Prozessschritten* bestehen.

Das in der [IEC 61512-1] definierte Prozessmodell kann prinzipiell mit der in Abschnitt 6.1.3.1 definierten Hierarchie fertigungstechnischer Anlagen verknüpft werden. Abbildung 6-3 zeigt das dahinter stehende Prinzip am Beispiel der „Hochzeit“ im Automobilbau. Die gesamte Anlage führt den Prozess „Hochzeit“ aus. Der Prozess „Hochzeit“ kann in einzelne Prozessabschnitte unterteilt werden, z. B. „Zuführung von Karosse/Antriebsstrang“ oder „Fügen Karosse und Antriebsstrang“. Diese Prozessabschnitte können einzelnen Funktionsgruppen einer Anlage zugeordnet werden. Prozessoperationen wie z. B. „Transport einer Karosse“ können wiederum Modulen innerhalb der Funktionsgruppen zugewiesen werden, während die in einem Modul beinhalteten Komponenten Prozessschritte wie „Motor dreht links“, „Motor dreht rechts“ oder „Sensorüberwachung“ ausführen können.

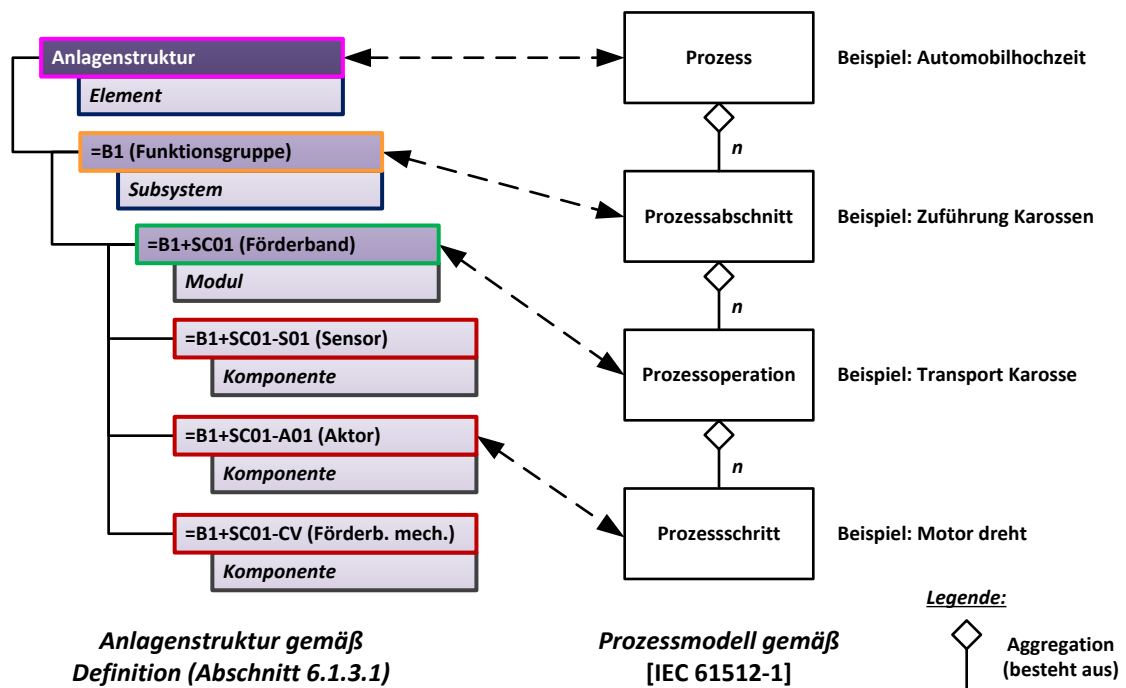


Abbildung 6-3: Zusammenhang zwischen Prozessen (gemäß [IEC 61512-1]) und Hierarchieebenen (wie in Abschnitt 6.1.3.1 definiert)

Prozessbeschreibungen, die für die Erstellung eines Anlagensimulationsmodells im Datenmodell benötigt werden könnten, können in folgende Typen klassifiziert werden:

- *Prozessbeschreibungen hinsichtlich der Abläufe einer Komponente (Beschreibung eines Prozessschritts)*

Fertigungstechnische Anlagen unterliegen dem mechatronischen Prinzip (→ Abschnitt 2.2.1, erster Absatz, vgl. ferner Abschnitt 2.4.1). Ein Prozessschritt ergibt sich in diesem Kontext aus dem spezifischen Verhalten einer Komponente, kombiniert mit der Ansteuerung dieser Komponente [STR13]. Die Ansteuerung erfolgt stets innerhalb des zu programmierenden [IEC 61131-3] Steuerungscode, z. B. in Form von Schrittketten oder mittels vorgegebener Standardbausteine. Einzelne Prozessschritt-Beschreibungen müssen im Datenmodell daher nicht explizit abgebildet werden – sie sind in der Steuerung als [IEC 61131-3] Code hinterlegt.

- *Prozessbeschreibungen hinsichtlich der Abläufe innerhalb eines Moduls (Beschreibung einer Prozessoperation)*

Die durch ein Modul ausgeführte Prozessoperation ergibt sich aus dem Zusammenwirken einzelner Prozessschritte. Wesentlich hierfür ist die Darstellung von Wirkbeziehungen zwischen Komponenten innerhalb eines Moduls. Nur dadurch lassen sich einzelne Prozessschritte zu einer Kausalkette kombinieren, die die Prozessoperation eines Moduls repräsentiert (z. B. Motor dreht → Stückgut bewegt sich auf Förderband → Sensor löst aus). Die Abbildung der Wirkbeziehungen zwischen einzelnen Komponenten ist im Datenmodell daher notwendig. Die Abbildung einer Beschreibung der eigentlichen Prozessoperation über die Wirkbeziehungen einzelner Prozessschritte hinaus ist für eine virtuelle Inbetriebnahme nicht notwendig – eine Beschreibung der Prozessoperation findet sich in Form einzelner Prozessschritte im [IEC 61131-3] Steuerungscode.

- *Prozessbeschreibungen hinsichtlich des Materialflusses (Beschreibung modulübergreifender Prozesse)*

Prozessbeschreibungen hinsichtlich modulübergreifender Prozesse und daher Prozesse auf Prozessabschnitts-/Gesamtanlagen-Prozessebene beschreiben in der Regel den Materialfluss innerhalb einer Anlage. Bestimmte Teilaspekte dieser Prozesse, wie die Abbildung von Vorgänger-Nachfolger Beziehungen, welche die Verbindungen zwischen Modulen repräsentieren, müssen im Datenmodell berücksichtigt werden. Dies liegt darin begründet, dass insbesondere Vorgänger-Nachfolger-Relationen als Bestandteil der Wirkbeziehungen betrachtet werden können, deren Abbildung für ein Simulationsmodell essenziell ist. In diesem Kontext ist vor allem die Abbildung des Materialflusses von Relevanz, da bei fertigungstechnischen Anlagen (im Gegensatz zu einer prozesstechnischen Anlage) der Produktfluss nicht aus einer ununterbrochenen Kette von Komponenten nachvollzogen werden kann [GÜFA08], und so eine Beschreibung von Produktflüssen ggf. notwendig sein kann (→ Abschnitt 2.5.1). Dies trifft insbesondere auch auf Quellen und Senken zu, die im Anlagenstrukturmodell explizit abgebildet werden sollten. Quellen und Senken stellen die Verbindungen der betrachteten Anlage zur Umgebung (andere Anlagen) und daher die Schnittstellen an den Systemgrenzen dar ([HWF*12], [CHF14A]).

- *Prozessbeschreibungen hinsichtlich des Steuerungsablaufs (innerhalb der Steuerung)*

Die Beschreibung des eigentlichen Steuerungsablaufs liegt in der Praxis, wenn überhaupt, informell vor (z. B. in Prosaform oder mittels Excel-Tabellen, → Tabelle 7). Eine Abbildung von Prozessen im Datenmodell würde in erster Linie ein Hilfsmittel für die Erstellung des Steuerungscode sein. Für die Erstellung eines Anlagensimulationsmodells aus Anlagenstrukturdaten hat dies keine Bedeutung. Zum Zeitpunkt einer virtuellen Inbetriebnahme sind einzelne Prozessschritte bereits im Steuerungscode umgesetzt und sollen validiert werden.

Zusammenfassend lässt sich festhalten, dass einzelne Prozessschritte in der Fertigungstechnik durch das Zusammenspiel einer Steuerung mit den Komponenten einer Anlage (Sensoren, Aktoren, mechanische Grundsysteme) entstehen. In einer virtuellen Inbetriebnahme werden die Steuerung einzelner Prozessschritte sowie das Zusammenspiel von Prozessschritten getestet. Um den Test des programmierten Steuerungscode einer Anlage zu ermöglichen, müssen daher das spezifische Verhalten und die Interaktion von Komponenten in einem Anlagensimulationsmodell nachgebildet werden. Prozessbeschreibungen in Form eines innerhalb des *Datenmodells der Digitalen Fabrik* abzubildenden Prozessmodells können für die Erstellung eines Anlagensimulationsmodells daher größtenteils vernachlässigt werden. Ein Prozessmodell wäre ein Hilfsmittel für die Programmierung des während einer virtuellen Inbetriebnahme zu testenden, für die Errichtung einer Anlage neu zu entwickelnden Steuerungscode.

Die Abbildung einzelner Wirkbeziehungen, im Konkreten zwischen Komponenten (*Interaktion einzelner Prozessschritte*) und Modulen (*Beziehungen zur Beschreibung des Materialflusses*), ist im Datenmodell jedoch notwendig. Die Abbildung von Wirkbeziehungen kann für die virtuelle Inbetriebnahme direkt im Knoten *Ressourcen* der Anlagenstruktur (→ Abschnitt 6.1.3.1) geschehen, ohne diese Prozesse als eigenständige Objekte abbilden zu müssen. Dies ist von Vorteil, da somit nur noch das Anlagenstrukturmodell bei der Simulationsmodellerstellung untersucht werden muss. Es müssen keine weiteren Datenquellen hinzugezogen werden. Ferner muss kein Aufwand betrieben werden, ein Prozessmodell zu erstellen, das für die Erstellung eines Anlagensimulationsmodells vernachlässigt werden kann. Die Darstellung von im Datenmodell abzubildender Wirkbeziehungen wird in Abschnitt 6.1.3.4 beschrieben, ein konkretes Beispiel zeigt Abschnitt 6.1.5.

6.1.3.3 Anforderung 3: Geeignete Abbildung der zu fertigenden Produkte

Fertigungstechnische Prozesse zeichnen sich dadurch aus, dass ein Stückgut oder generell ein Produkt durch einen konzentrierten, räumlichen Prozesseingriff verändert oder in sonstiger Art bearbeitet wird [MBS⁺11]. Auch wenn der eigentliche Eingriff für eine virtuelle Inbetriebnahme nur eine untergeordnete oder gar keine Rolle spielt, kann die Veränderung des Produkts hinsichtlich seiner physikalischen Parameter beachtenswert sein. Ein Beispiel hierfür ist die „Hochzeit“ im Automobilbau, bei der das Produkt „Karosserie“ und das Produkt „Antriebsstrang“ zum Produkt „Gesamtfahrzeug“ integriert werden. Die durch Förderbänder vor und nach der Hochzeit zu transportierende Masse kann so erheblich voneinander abweichen. Hinsichtlich einer virtuellen Inbetriebnahme kann dies, abhängig vom gewählten Detaillierungsgrad der Simulation, von großer Bedeutung sein. Produkte und *relevante* Zwischenzustände müssen daher im *Datenmodell* unter Beachtung der für eine virtuelle Inbetriebnahme relevanten Parameter im Knoten *Produkte* abgebildet werden (Abbildung 6-4).

Die Abbildung aller Zwischenzustände ist aber weder immer möglich noch immer sinnvoll. Mitunter bestehen Produkte aus vielen tausend Einzelteilen, die im Laufe eines komplexen Montageprozesses zusammengefügt werden. Nicht jedes an diesem Prozess beteiligte Einzelteil spielt für die Automatisierung einer Anlage eine Rolle. Für eine *virtuelle Inbetriebnahme* sollten Einzelteile immer nur dann abgebildet werden, wenn ...

- ... sich physikalische Parameter eines Einzelteils (Maße, Masse, Änderungen der Geometrie, etc.) wesentlich ändern.
- ... das Einzelteil im automatisch ablaufenden Fertigungsprozess eine Rolle spielt, d. h. das Einzelteil durch Sensoren erkannt/überwacht werden soll und/oder mittels Aktoren manipuliert wird.

Als Beispiel kann die bereits erwähnte Hochzeit im Automobilbau dienen. Ergänzend während oder nach der vollautomatisch ablaufenden Hochzeit könnten Tätigkeiten, wie das manuelle Anbringen von Kunststoffverblendungen, notwendig sein. Die dadurch verursachten Massenänderungen am Produkt Gesamtfahrzeug sind einerseits so gering, dass diese vernachlässigt werden können. Andererseits spielen diese Tätigkeiten keine Rolle für den automatisch ablaufenden Prozess „Hochzeit“.

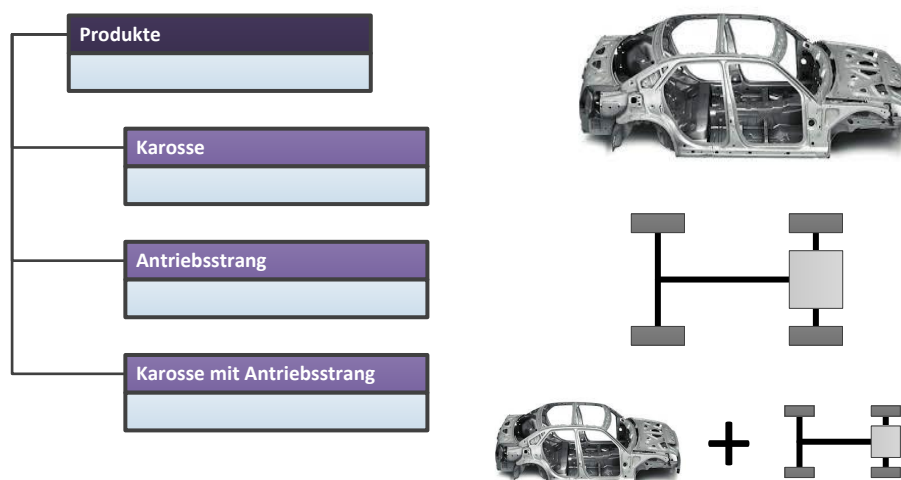


Abbildung 6-4: Abbildung von Produkten im Knoten *Produkte* in Anlehnung an [ScDr09]

6.1.3.4 Anforderung 4: Relationen innerhalb und zwischen den Knoten Ressourcen, Prozessen und Produkten

Relationen zwischen den Objekten einer Anlage sind für das *Datenmodell der Digitalen Fabrik* essenziell. Nur so lassen sich Wirkbeziehungen identifizieren, die insbesondere für eine Simulation von Bedeutung sein können (→ Abschnitt 6.1.3.2). Schleipen und Drath [SCDR09] stellen in diesem Kontext ein Konzept vor, wie die Relationen zwischen *Produkten*, *Prozessen* und *Ressourcen* auf einer vorwiegend globalen Betrachtungsebene abgebildet werden können. Das Konzept sei am Beispiel einer Panoramadachinstallation in der Automobilmontage kurz erklärt. Stark vereinfacht sind bei einem derartigen Montageprozess insgesamt drei *Produkte* (die Rohkarosse, das Panoramadach und letztlich die Karosse inklusive Panoramadach) während des zeitlichen Verlaufs beteiligt. Der Prozess selber wird von einer *Ressource* ausgeführt (hier ein Industrieroboter). Das hinter der Darstellung der Verbindungen dieser Elemente stehende Konzept lässt sich in einem Satz zusammenfassen: *Ein Prozess definiert, wie eine Ressource ein Produkt bearbeiten soll – in Abhängigkeit vom Produkt*. Abbildung 6-5 zeigt die grundlegenden Zusammenhänge zwischen *Produkten*, *Ressourcen* und den resultierenden Wirkbeziehungen in diesem Kontext.

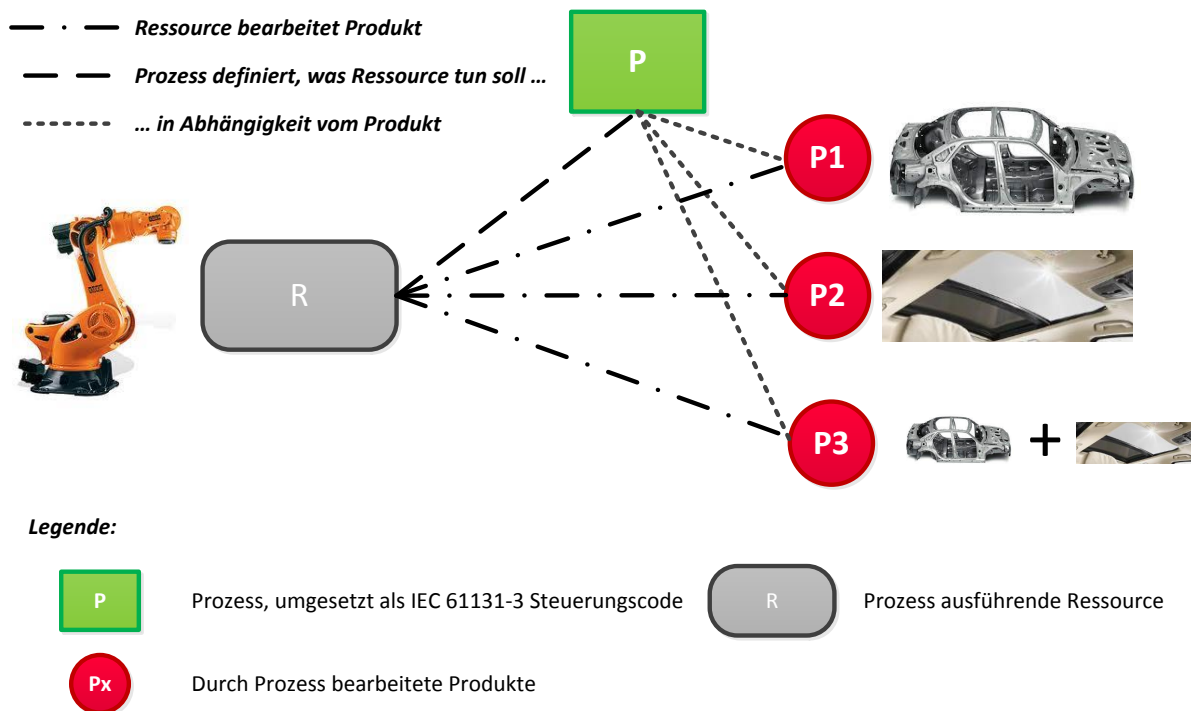


Abbildung 6-5: Verbindungen zwischen Produkten, Prozessen und Ressourcen in Anlehnung an [SCDR09]

Diese sehr grundlegende Sichtweise auf die Dinge ist für eine Darstellung der prinzipiellen Zusammenhänge zwischen *Produkten*, *Prozessen* und *Ressourcen* nutzbar, wie unter anderem in [DRA10] gezeigt wird. Die einzelnen zu betrachtenden Wirkbeziehungen *innerhalb* einer Anlagenstruktur, abgebildet im Knoten *Ressourcen*, sind im Gegensatz dazu hochgradig abhängig von der dargestellten Granularität und der, z. B. in einer *virtuellen Inbetriebnahme*, potenziell notwendigen Untersuchung der Anlagenstruktur auf verschiedenen Hierarchieebenen. Abhängig von der betrachteten Hierarchieebene kann die Art der Wirkbeziehungen erheblich voneinander abweichen. So können nicht nur die Materialflüsse und Zuordnungen einzelner *Produkte* zu einzelnen *Ressourcen* von Bedeutung sein, sondern auch die Wirkbeziehungen im Sinne des mechatroni-

schen Zusammenwirkens einzelner Komponenten (\rightarrow Abschnitt 6.1.3.2) sowie die Interaktionen zwischen einzelnen Modulen, insofern es um den Austausch physikalischer Größen geht. Im Folgenden werden mögliche Wirkbeziehungen innerhalb der Anlagenstruktur untersucht.

Die Wirkbeziehungen zwischen einzelnen mechatronischen Modulen (*Modul* \leftrightarrow *Modul* Beziehungen) können in den meisten Fällen auf den Materialfluss, also die Übergabe eines Produkts, beschränkt sein, bei der keine oder nur vernachlässigbare physikalische Größen ausgetauscht werden. Aber auch ein direkter Austausch physikalischer Größen, wie z. B. ein Volumenstrom \dot{V} , ist denkbar: Ein Beispiel wäre die Übergabe von Schüttgut zwischen verschiedenen Modulen oder auch der Transport von Flüssigkeiten, so z. B. der Transport von Kleber einer Pumpstation zu einem Applikationskopf an einem Industrieroboter. Hier gilt: Wird eine Druckänderung (Δp) zwischen zwei Potentialen (Tank, Applikationskopf) erzeugt, ist der dadurch resultierende Volumenstrom \dot{V} entsprechend als Materialstrom identifizierbar.

Physikalische Größen können zudem über das am Prozess beteiligte Produkt *indirekt* zwischen Modulen und den beinhalteten Komponenten ausgetauscht werden. Dies sei an einem einfachen Beispiel erklärt: Ein Werkstück wird in eine Bearbeitungsstation geliefert. Die Bearbeitungsstation besteht aus zwei Modulen. Ein Modul 1 enthält ein Förderband mit elektro-mechanischem Spanner, der das zu bearbeitende Werkstück (z. B. ein Gehäuse) fixiert. Unabhängig davon existiert ein Modul 2 Bohrapparatur, das Löcher in das Werkstück bohrt. Abhängig vom verwendeten Material des Werkstücks (z. B. Holz, Metall) ist der Druck des Spanners so einzustellen, dass einerseits eine Beschädigung des Werkstücks (z. B. durch Verbiegen) durch den Spanner ausgeschlossen wird, andererseits das Werkstück noch sicher im Spanner gehalten wird. Physikalisch betrachtet wirkt durch das Bohren ein Drehmoment \vec{M}_{Bohrer} auf das Werkstück (der spanende Bohrvorgang sei vernachlässigt). Dies würde eine Bewegung des Werkstücks um die Bohrachse verursachen – das Drehmoment \vec{M}_{Bohrer} wird auf das Werkstück partiell übertragen. Dem dadurch resultierenden Drehmoment $\vec{M}_{Werkstück}$ wird durch den Spanner entgegengewirkt. Daraus lassen sich die Wirkbeziehungen *Ressource (Modul 1) \leftrightarrow Produkt (Werkstück) \leftrightarrow Ressource (Modul 2)* ableiten. Modul 1 wirkt in dieser Sichtweise *indirekt* (über das Werkstück) auf Modul 2.

Ähnlich verhält es sich bei der Betrachtung auf Komponentenebene innerhalb eines Moduls. Die Komponenten wirken hier derart zusammen, dass das Modul seine eigentliche Funktion ausführt. Diese Wirkbeziehungen sind im Gegensatz zu *Modul \leftrightarrow Produkt \leftrightarrow Modul* Beziehungen in der Regel *direkt* gerichtet, also als *Komponente \leftrightarrow Komponente* Beziehung beschreibbar. Diese Beziehung kann durch Potential- und Flussgrößen beschrieben werden, z. B. als ein durch eine Druckänderung Δp hervorgerufener Volumenstrom \dot{V} , oder ein durch Änderung einer Rotation ω bewirktes Drehmoment \vec{M} .

Die Wirkbeziehungen zwischen Komponenten sind nicht nur auf rein physikalische Größen beschränkt. Die mögliche Wirkbeziehung zwischen einem Sensor und einem Förderband kann auch als Zuordnung verstanden werden: Es gibt keine physikalische Größe, die zwischen dem mechanischen Grundsystem und Sensor ausgetauscht wird. Diese Art von Wirkbeziehung sagt vielmehr aus, dass am mechanischen Grundsystem Förderband ein Sensor vorhanden ist, der, abhängig von der Position des Produkts auf dem Förderband, ein Signal auslöst. Die eigentliche Position kann z. B. durch einen Parameter bestimmt werden. Die Nachbildung von Zuordnungsbeziehungen ist notwendig, um beim Vorhandensein mehrerer gleichartiger Objekte (z. B. mehrerer Sensoren) eine eindeutige Zuweisung sicherzustellen. Tabelle 8 zeigt die hier beschriebenen Wirkbeziehungen in einer Übersicht anhand der hier beschriebenen Beispiele.

Wirkbeziehungstyp:	Beschreibung:
	<p>Wirkbeziehungen Modul ↔ Modul: Logistische Beziehungen im Sinne des Materialflusses (Abbildung z. B. mittels des Ansatzes von [ScDR09]); Austausch physikalischer Größen wie z. B. \dot{V}.</p>
	<p>Wirkbeziehungen Modul auf Modul unter Beteiligung eines Produkts (Modul ↔ Werkstück ↔ Modul): „Indirekter“ Austausch physikalischer Größen.</p>
	<p>Wirkbeziehungen zwischen Komponenten innerhalb eines Moduls (Komponente ↔ Komponente): Direkter Austausch physikalischer Größen (z. B. \dot{V}, \vec{F}, \vec{M}).</p>
	<p>Wirkbeziehungen zwischen Komponenten innerhalb eines Moduls (Komponente ↔ Komponente): Zuordnungsbeziehung von Sensor zu Förderband.</p>

Tabelle 8: Auswahl möglicher Wirkbeziehungen im Datenmodell der Digitalen Fabrik

Diese verschiedenen Wirkbeziehungen sollten innerhalb des Datenmodells abgebildet werden können. Insbesondere, aber nicht nur, für eine virtuelle Inbetriebnahme sind diese Wirkbeziehungen für das grundlegende Verständnis des entworfenen mechatronischen Zusammenwirkens von hoher Bedeutung. Sie können während der Planungsphase den beteiligten Ingenieuren helfen, die Gesamtzusammenhänge der Abläufe einer Anlage schneller begreifbar zu machen. Die Identifikation der zu beachtenden Schnittstellen eines Simulationsmodells (interne, d. h. im Simulationsmodell vorhandene sowie externe, d. h. Schnittstellen des Simulationsmodells zur Simulationsumgebung gemäß der Ausführungen in Abschnitt 5.3.3) kann sich zudem vereinfachen. Die

Auswirkungen dieser unterschiedlichen Wirkbeziehungen auf die Simulation und die daraus resultierenden Konsequenzen für die Gestaltung einzelner Teil-Simulationsmodelle werden in Abschnitt 8.3 ausführlich betrachtet. Ein konkretes Beispiel, wie sich derartige Wirkbeziehungen abbilden lassen, wird in Abschnitt 6.1.5 gegeben.

6.1.4 Datenformat zur Abbildung des Datenmodells der Digitalen Fabrik

In den letzten Abschnitten wurde die Abbildung von Informationen, insbesondere hinsichtlich der Strukturierung und der Darstellung von Wirkbeziehungen, betrachtet. Gemäß dem Sender-Empfänger Modell [SHA48] müssen diese Informationen in geeigneter Art und Weise codiert werden, um von einem Empfänger richtig interpretiert werden zu können [DiMÜ11]. In den ingenieurwissenschaftlichen Disziplinen kommen hierfür verschiedenste Datenformate und darin enthaltene Datenmodelle zum Einsatz. Für eine weitergehende Erklärung, insbesondere der damit im Zusammenhang stehenden Aspekte wie Syntax, Semantik und Pragmatik von Modellbeschreibungen, sei auf [DiMÜ11] verwiesen.

Für die Abbildung des *Datenmodells der Digitalen Fabrik* stehen verschiedene Datenformate zur Verfügung. Eine umfassende Übersicht über in der Praxis zum Einsatz kommende Datenformate wird in [BWG09] gegeben. Die von [BWG09] beschriebenen Datenformate stehen beispielhaft für die in den Abschnitten 2.6 und 5.1 erwähnte große Heterogenität bei der Nutzung von Datenformaten im Anlagenbau. Auf eine detaillierte Auflistung aller dort erwähnten Datenformate wird aus Gründen der Übersichtlichkeit verzichtet. Generell kommen gemäß [BWG09] rein hersteller-spezifische, proprietäre und geschlossene Datenformate ebenso wie offene, frei verfügbare, auf Metamodellen basierende Formate zum Einsatz. Der sehr große Lösungsraum an möglichen Datenformaten kann hinsichtlich der allgemeinen Anforderungen an das *Datenmodell der Digitalen Fabrik*, der in Abschnitt 6.1.2 beschriebenen Anforderungen hinsichtlich *Interoperabilität*, sowie der Modellbildung im Kontext der *Digitalen Fabrik* allgemein [DiMÜ11] stark eingeschränkt werden.

Insbesondere der Verwendung von *Metamodellen* kann, im Rahmen eines geforderten hohen Formalisierungsgrads für den Datenaustausch im Kontext der *Digitalen Fabrik*, eine wichtige Bedeutung zukommen [DiMÜ11]. Metamodelle können u. a. eine fundamentale Bedeutung für die *Digitale Fabrik* hinsichtlich der Wiederverwendbarkeit von Modellierungsaspekten haben [DiMÜ11]. Metamodelle können als Sprachmittel identifiziert werden, mit denen der Aufbau von Modellen explizit beschrieben werden kann [EPP03]. Sie ermöglichen ein „[...] *modellübergreifendes gemeinsames Verständnis der Bedeutung der übertragenen Daten*“ [EPP03] und können so den Datenaustausch maßgeblich erleichtern. Metamodelle ermöglichen es, die in diesem Abschnitt 6.1.3 bisher beschriebene Struktur einer Anlage mittels einer Sprache zu formulieren, die durch einen Empfänger (sofern ihm die Sprache geläufig ist) identifiziert werden kann.

Ein Datenformat, das die Ansprüche des *Datenmodells der Digitalen Fabrik* sowie die hier ergänzend beschriebenen Anforderungen erfüllen kann, ist das Datenaustauschformat *AutomationML* [AUTOMATIONML[®]]. *AutomationML*, standardisiert in [IEC 62714-1], ist als herstellernerutrales, offenes Datenaustauschformat definiert. Der Einsatz von *AutomationML* im fertigungstechnischen Anlagenbau ist bereits vielfach beschrieben worden (beispielhaft sei auf [DLP⁺08] und [DRA10] verwiesen, für eine gesamtheitliche Architekturübersicht siehe [DRA10A[®]]) und wird daher hier nur stark verkürzt wiedergegeben. *AutomationML* besteht im Kern aus drei Elementen, deren Gesamtheit das Datenaustauschformat bildet:

1. *CAEX*, normiert in [DIN EN 62424], bildet das Dachformat von *AutomationML*. *CAEX* ist eine XML-basierende Metasprache zur Beschreibung von Anlagentopologien (Strukturmodellen). Grundsätzlich ist der hinter *CAEX* stehende Kerngedanke, herstellerbezogene Objektbibliotheken (z. B. in Form von Produktkatalogen) als *SystemUnitClassLibraries* (SUCL) abzubilden und dann in einer Instanzhierarchie (IH) als *InternalElements* (IE), abhängig von den Anforderungen an das einzusetzende Objekt, zu instanzieren, und ggf. neu zu parametrieren [DRA10]. Die Struktur kann auch zunächst mittels abstrakten IE aufgebaut werden. Auf diese IE werden anschließend (herstellerspezifische) Klassen aus einer SUCL instanziiert. In beiden Fällen wird die Beschreibung von Anforderungen an das einzusetzende herstellerbezogene Objekt benötigt. Das in *CAEX* nativ vorhandene Rollenkonzept ist dafür grundsätzlich geeignet. Es beschreibt Anforderungen an abstrakte Objekte und damit, mit welchen (*SystemUnitClass*-)Objekten diese grundsätzlich erfüllt werden können [DRA10]. Rollen werden in einer *RoleClassLibrary* (RCL) vorgehalten. Die Verbindungen zwischen Objekten (die den in Abschnitt 6.1.3.4 beschriebenen Wirkbeziehungen entsprechen) erfolgt über miteinander zu verbindende Konnektoren, die in einer *InterfaceClassLibrary* (ICL) beschrieben werden können. Die eigentlichen Verbindungen zwischen diesen Konnektoren werden mittels *InternalLinks* (IL) konkret umgesetzt. Allen in *CAEX* beschreibbaren Objekten können Attribute zugewiesen werden, um beispielsweise Parameter eines Geräts abbilden zu können.
2. Das Datenformat [*COLLADA*[®]] kommt in *AutomationML* für den Austausch von Geometrie- und Kinematikbeschreibung zum Einsatz. *COLLADA* selbst ist ein XML-basiertes, offenes Datenformat für den Austausch von 3D-Daten. Geometrien und Kinematiken, die in *COLLADA* beschrieben sind, werden über wohl definierte Schnittstellen (ICL) in *CAEX* referenziert [DRA10]. Sie sind nicht direkt in *CAEX* enthalten.
3. Die Einbindung diskreter Verhaltensbeschreibungen wird mittels *PLCopen XML* [*PLCOPEN XML*[®]] ermöglicht. *PLCopen XML* ermöglicht die Abbildung von [IEC 61131-3] Code in XML. Wie bei *COLLADA* erfolgt die Einbindung derartiger Verhaltensbeschreibungen mittels Referenzieren in *CAEX* [DRA10].

AutomationML wurde mit der Intention geschaffen, das *Datenmodell der Digitalen Fabrik* abbilden zu können [HIR12]. Die in *AutomationML* enthaltenen Abbildungskonzepte auf Meta-Ebene ermöglichen zudem die Darstellung der hier ergänzend dargestellten Anforderungen hinsichtlich der konkreten Abbildung einer Anlagenstruktur. Die folgenden Betrachtungen erfolgen daher unter Verwendung von *AutomationML* als Datenformat. Angemerkt sei, dass grundsätzlich auch jedes andere Datenformat, das die genannten Anforderungen erfüllt, einsetzbar wäre. Die Verwendung von *AutomationML* versteht sich daher als rein exemplarische Anwendung, die stellvertretend für derartige Datenformate steht.

6.1.5 Mögliche Darstellung einer Anlagenstruktur

Nach Bestimmung des zu verwendenden Datenformats ist es möglich, die Anlagenstruktur konkret abzubilden. Hierfür wird das in Abschnitt 6.1.3.1 genutzte Beispiel einer Fördertechnik mit drei Förderbändern herangezogen (→ Abbildung 6-2). Für die Darstellung der Anlagenstruktur wurden eine Reihe von exemplarischen SUC- und IC-Bibliotheken entworfen, da bisher keine frei verfügbaren Herstellerkataloge und dazu passenden Schnittstellendefinitionen für *AutomationML* existieren. Die SUC-Bibliotheken enthalten Objekte einer Anlage, wie mechanische Grundsysteme, Sensoren und Aktoren, und stellen eine beispielhafte, in keinem Fall aber vollständige Auswahl

typischer verwendeter Komponenten in einer automatisierten Anlage dar. Die entworfene SUC-Bibliothek steht hier stellvertretend für eine herstellerspezifische Bibliothek. Die Granularität der Bibliotheken wurde so gewählt, dass eine Abbildung einer Anlagenstruktur auf Komponentenebene möglich ist. Die in dieser Bibliothek enthaltenen Objekte enthalten eine Vielzahl von Attributen (→ Abschnitt 6.2.2). Die IC-Bibliothek wurde im Hinblick auf die in Abschnitt 6.1.3.4 beschriebenen abzubildenden Wirkbeziehungen entworfen. Sie enthält eine Reihe von allgemeinen Schnittstellen wie z. B. mechanische Flansche, Produktflussschnittstellen, Sensor-Zuweisungen und weitere, für die Anbindung an das Prozessabbild der SPS relevante Schnittstellen.

Abbildung 6-6 zeigt einen Ausschnitt aus den erstellten SystemUnitClass- und InterfaceClass-Bibliotheken. Die ebenfalls partiell zum Einsatz kommenden AutomationML-Standardbibliotheken werden hier nicht betrachtet; für eine Übersicht sei auf [DRA10] verwiesen.

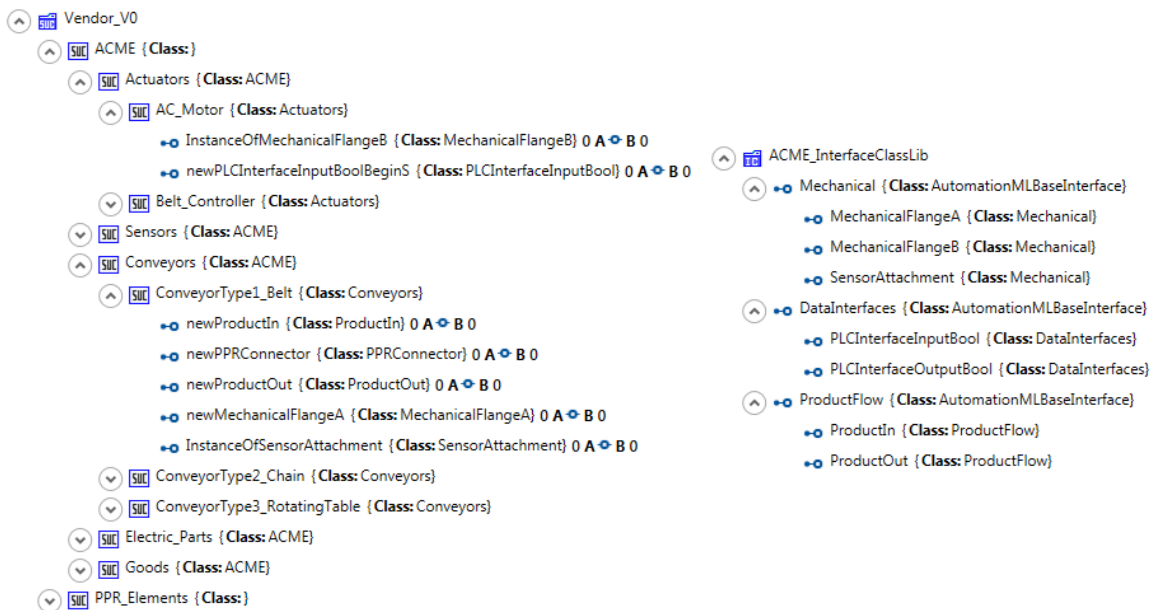


Abbildung 6-6: Ausschnitt der erstellten Bibliotheken SUCL (links) und ICL (rechts)

Abbildung 6-7 zeigt, unter Zuhilfenahme der entworfenen SUC- und IC-Bibliotheken, die Anlagenstruktur in der Granularität Komponentenebene am erwähnten Anlagenbeispiel. Die explizit dargestellten Wirkbeziehungen sind im Datenformat mittels IL abgebildet (hier nicht als solche dargestellt). Die Anlagenstruktur inkl. der abgebildeten Wirkbeziehungen bildet exemplarisch die Funktion des mechatronischen Moduls Förderband =B1+SC01 nach. Sie wird im Folgenden kurz beschrieben. Die Komponente Motor (=B1+SC01-A01) wirkt auf das mechanische Grundsystem Förderband (=B1+SC01-CV) über eine Schnittstelle *MechanicalFlange* durch die Erzeugung einer rotatorischen Bewegung ein. Die damit dargestellte physikalische Wirkbeziehung entspricht der Übertragung eines durch den Motor erzeugten Drehmoments an die mechanische Apparatur Förderband. Das Produkt interagiert mit dem mechanischen Förderband dahin gehend, dass es diesem zugewiesen ist (Schnittstelle *PPR_Interface*). Die Wirkbeziehungen im Sinne des Materialflusses werden mittels *Product_In/Product_Out* Schnittstellen direkt an der Komponente Förderband dargestellt. Diese Schnittstelle entspricht der globalen Ein- und Ausgangsschnittstelle des gesamten Moduls Förderband und wird der entsprechenden Komponente, die diese Wirkbeziehung handhabt, zugewiesen. Der im Modul vorhandene Sensor ist mittels *SensorAttachment*-Schnittstelle an das Förderband angebunden. Wie in Abschnitt 6.1.3.4 beschrieben, handelt es

sich um eine Schnittstelle zur reinen Zuordnung Förderband \leftrightarrow Sensor. Ergänzend dargestellt ist die Anbindung des Sensors und des Aktors an die Steuerungstechnikstruktur.

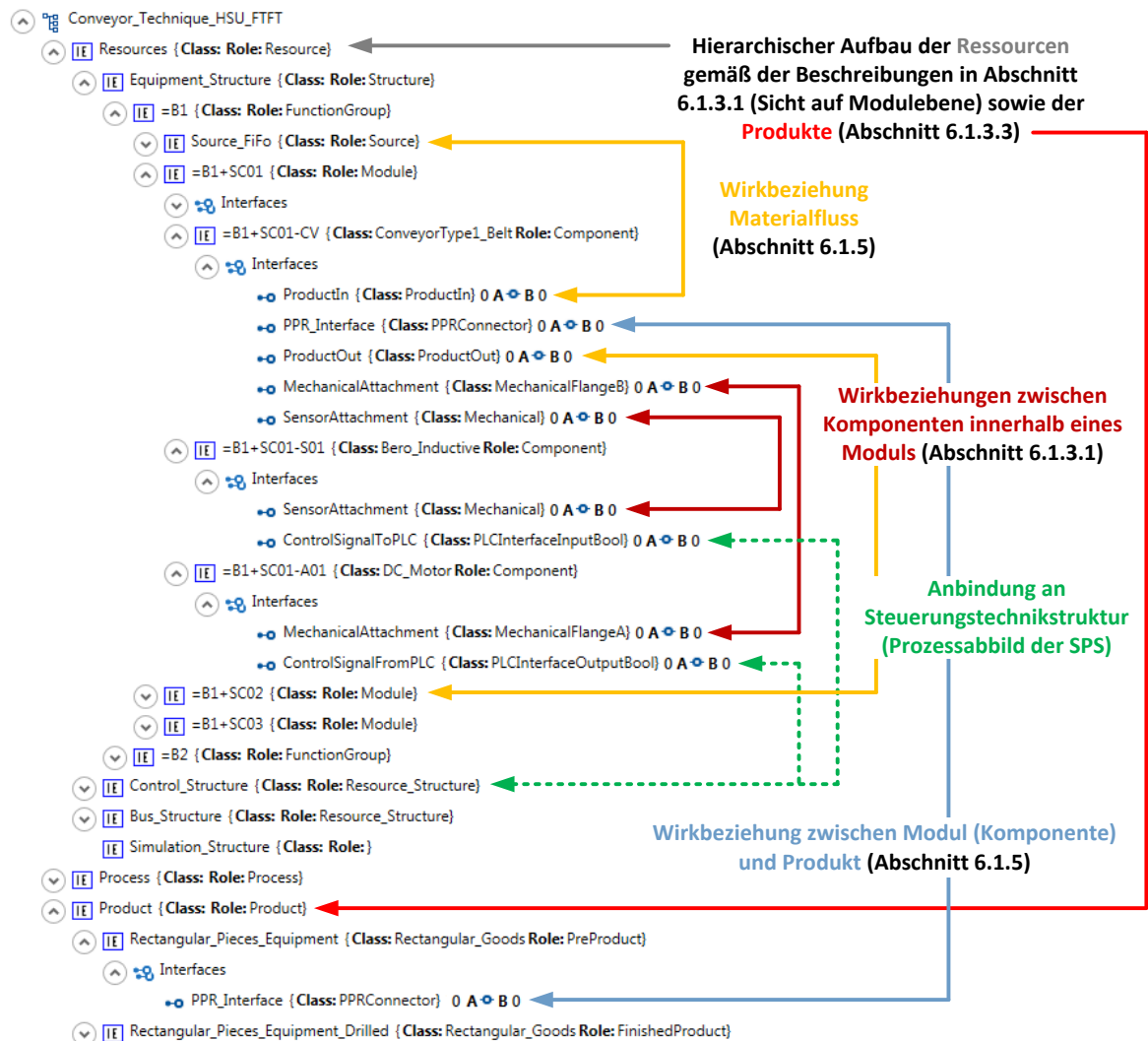


Abbildung 6-7: Anlagenstruktur auf Modulebene

6.2 Ergänzende Bibliotheken und Methoden für die Modellerstellung

Die bisher beschriebenen Ansätze können helfen, die Darstellung einer Anlagenstruktur im *Datenmodell der Digitalen Fabrik* zu konkretisieren. Um die Modellerstellung im Rahmen einer virtuellen Inbetriebnahme zu unterstützen (\rightarrow Abschnitt 5.3.1 *Baustein 1.2*), kann das Datenmodell der Digitalen Fabrik zusätzlich um bestimmte Aspekte erweitert werden.

6.2.1 Identifikation von zu simulierenden Elementen einer Anlage

Wie in Abschnitt 6.1.1 dargelegt, besteht die für eine *virtuelle Inbetriebnahme* durchgeführte Modellerstellung im Wesentlichen aus der Transformation der Anlagenstruktur in ein Simulationsmodell. Hierfür ist ein Hilfsmittel zur Identifikation der zur Substitution realer Objekte geeigneten Simulationsmodelle direkt aus dem Anlagenstrukturmodell heraus vorteilhaft (\rightarrow Abschnitt 6.1.1). Für die Ableitung eines Simulationsmodells direkt aus der Instanzhierarchie

und dem dort abgebildeten Anlagenstrukturmodell muss dieses es ermöglichen, aus dem Blickwinkel der virtuellen Inbetriebnahme auf das Anlagenstrukturmodell blicken zu können. Durch gezielte Erweiterungen am Anlagenstrukturmodell, wie z. B. durch das *Gruppenkonzept* [DRA10], das mit dieser Intention eingeführt wurde, kann dies ermöglicht werden. Das Gruppenkonzept ermöglicht eine Separierung von Instanz- und Strukturinformationen. Dies bedeutet, dass identische Objekte in unterschiedlichen Hierarchien angeordnet werden können. Es wäre daher möglich, neben der eigentlichen Anlagenstruktur eine reine Simulationsstruktur aufzubauen. Diese würde nur die für eine Simulation relevanten Objekte und Schnittstellen beinhalten. Es ist aber unklar, welche Datenquelle ein um das Gruppenkonzept erweitertes Anlagenstrukturmodell haben könnte (→Tabelle 7). Eine manuelle Implementierung der für eine Simulationsmodellerstellung notwendigen Ergänzungen (konkret: die Möglichkeit zur Identifizierung, welche Simulationsmodelle auf welche realen Komponenten einer Anlage passen) wäre notwendig. Ein (wissensbasierter) Algorithmus, der die notwendigen Informationen systematisch ergänzt, wäre hierfür denkbar [CHF14B]. In diesem Fall wäre aber fallspezifisches Wissen für jedes vorhandene Objekt nötig. Zudem wäre auch ein Ansatz denkbar, der nur rollenspezifisches oder klassenspezifisches Wissen benötigt. Der Einsatz von wissensbasierten Methoden wäre aber mit zusätzlichem Aufwand zur Implementierung und Ausführung verbunden. Vorteilhaft wäre es daher, das Simulationsmodell direkt aus dem die reale Anlage darstellenden Anlagenstrukturmodell abzuleiten, und dieses möglichst unverändert zu lassen. Das in CAEX enthaltene Rollenkonzept kann hierfür eine mögliche Grundlage sein.

Um die Ableitung eines Simulationsmodells direkt aus der Instanzhierarchie zu ermöglichen, können die in einer Simulationsbibliothek vorliegenden Modelle in einem ersten Schritt zunächst direkt als für sich eigenständige SUCL in AutomationML integriert werden (als Objekte mit Parametern und Schnittstellen, aber ohne Logik- oder Verhaltensbeschreibungen – diese sind nur in der Simulationsmodellbibliothek zu finden → Abschnitt 6.1.3.2). Die in der Simulationsmodellbibliothek hinterlegten Schnittstellen zwischen den Modellen werden entsprechend diesem Muster in eine simulationsspezifische ICL abgebildet. Eine Übernahme der Modellbibliothek kann automatisiert durch ein Export/Import-Tool erfolgen, das zunächst die Simulationsmodellbibliothek inkl. relevanter Daten wie Parameter, deren Datentypen, Default-Parameterwerte und Schnittstellen ausliest und diese anschließend in AutomationML importiert. Ggf. ist diese Übernahme manuell vorzunehmen, wenn kein Zugriff auf die Simulationsmodellbibliothek im Sinne der Rechnerauswertbarkeit besteht. Dies bedeutet zunächst zusätzlichen Aufwand, der in der Regel als einmalig klassifiziert werden kann. Abbildung 6-8 zeigt einen Ausschnitt aus einer derartigen, in AutomationML als SUC importierten Bibliothek, inkl. der relevanten, ebenfalls zu hinterlegenden Simulationsmodellparameter am Beispiel des Simulationsmodells „Conveying_Belt_Bus“.

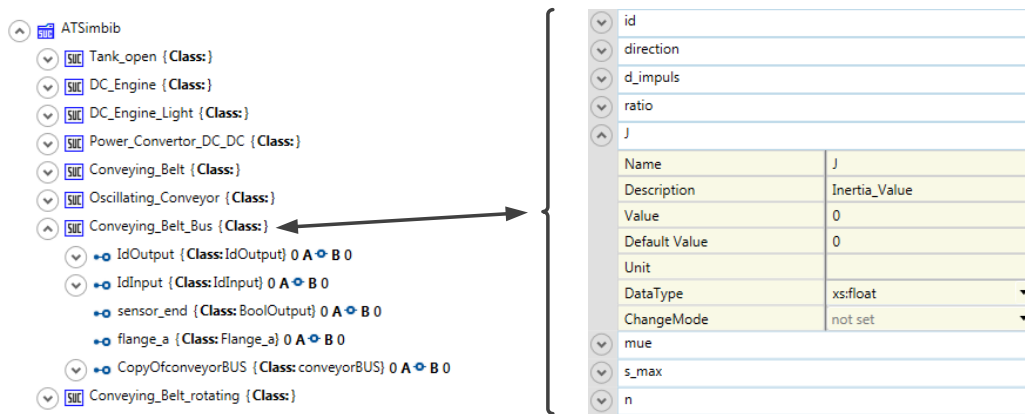


Abbildung 6-8: Abbildung einer Simulationsmodellbibliothek als SystemUnitClassLibrary

Die korrekte Identifizierung der für die Simulation relevanten Objekte, deren Verbindungen untereinander sowie mit den Parametern und Parameterwerten innerhalb des Anlagenstrukturmodells ist hilfreich, um eine manuelle Modellerstellung zu unterstützen und eine automatische Ableitung eines Simulationsmodell grundsätzlich zu ermöglichen. Potenziell kann dies durch eine geeignete Rollenklassenbibliothek umgesetzt werden. Vorzugsweise können die dort enthaltenen Rollen sowohl typische Automatisierungskomponenten aus Herstellerproduktkatalogen identifizieren wie auch das Simulationswerkzeug respektive die im AutomationML-Format abgebildete Simulationsbibliothek gleichermaßen unterstützen. Rollen sollen im konkreten helfen, die für die Simulation notwendigen Objekte innerhalb des in der Instanzhierarchie abgebildeten Anlagenstrukturmodells zu erkennen. Eine Rollenbibliothek, die die Vorzüge herstellerunabhängiger Merkmale vorweist, ist von Vorteil, da das notwendige Mapping der Parameterwerte und Verbindungen der einzelnen Objekte zwischen dem realen Anlagenstrukturmodell und dem entstehenden Simulationsmodell durch den Einsatz von global gültigen Merkmalen standardisiert und somit vereinfacht werden kann.

Für die Umsetzung einer derartigen Rollenbibliothek gibt es viele Ansätze. AutomationML enthält von sich aus bereits einige Standardrollenbibliotheken [DRA10], die aber sehr allgemein gehalten sind und eher als Basisklassen für darauf aufbauende, erweiterte Rollenbibliotheken geeignet sind. Eine mögliche Umsetzung, die den Anforderungen der industriellen Praxis genügen und die dort vorherrschende Heterogenität bei der Verwendung unterschiedlicher Komponenten wiedergeben kann, wird durch das Klassifizierungssystem *eCl@ss* ermöglicht. *eCl@ss* ist ein branchenübergreifender Produktdatenstandard für die Klassifizierung und eindeutige Beschreibung von Produkten und Dienstleistungen [ECL@SSA[®]] und wird von vielen Herstellern bereits unterstützt. *eCl@ss* ist in vier Ebenen eingeteilt, die, je tiefer in der Hierarchie liegend, Produkte und Dienstleistungen mit feinerer Granularität auflisten. Diese Ebenen sind (in hierarchischer Reihenfolge) *Sachgebiet*, *Hauptgruppe*, *Obergruppe* und *Untergruppe*. Die Untergruppen ausgewählter Sachgebiete können, gemäß der Definitionen in Abschnitt 6.1.3.1, als Komponenten einer typischen automatisierten Anlage angesehen werden. Beispiele hierfür wären *27-02-21-01 NS-Drehstrom-Asynchronmotor* oder auch *27-27-01-01 Induktiver Näherungsschalter*.

Eine *eCl@ss*-Rollenbibliothek auf dieser Hierarchiestufe ermöglicht durch die konkrete Rollenzuweisung einzelner Objekte in AutomationML in Form eines globalen Identifiers die Verknüpfung von bestehendem Anlagenstrukturmodell und neu entstehendem Simulationsmodell. Abbildung 6-9 zeigt das zugrunde liegende Prinzip der Substitution von realen in der Anlage vorhanden

Objekten mit Simulationsmodellen. Die in Abbildung 6-9 gezeigten Schritte können wie folgt beschrieben werden:

1. Im ersten Schritt wird eine *eCl@ss*-Rollenklasse aus der *eCl@ss*-Rollenbibliothek einem abstrakten Objekt (IE) in der Instanzhierarchie (IH) sowie jeweils einem Objekt in der herstellerepezifischen SUC-Bibliothek als auch der SUC-Simulationsmodellbibliothek zugewiesen (Abbildung 6-9 Schritt 1). Die Zuweisung einer Rolle zu Objekten in der IH wie auch den beiden SUCL (Herstellerkatalog, Simulationsmodellbibliothek) erfolgt mittels *SupportedRole-Class-Element* [DRA10]. Dies ermöglicht zunächst die semantische Zuordnung von nutzerdefinierten Klassen (hier: Herstellerkatalog, Simulationsmodellbibliothek) zu einer *eCl@ss*-Rolle.
2. Im zweiten Schritt (Abbildung 6-9 Schritt 2) kann durch die in Schritt 1 erzeugte semantische Zuordnung ein Objekt aus der herstellerepezifischen SUCL auf das abstrakte IE in der IH instanziiert werden. Aus dem abstrakten IE, das zunächst nur eine Rollenzuweisung enthalten hat, wird so ein „herstellerepezifisches“ IE, das auf eine konkrete, herstellerepezifische Klasse verweist.
3. Im dritten Schritt kann nun ein Simulationsmodell, das ein herstellerepezifisches IE substituieren kann, identifiziert werden (Abbildung 6-9 Schritt 3): Ist die gleiche Rollenklasse am betrachteten IE, in der herstellerepezifischen SUC und der Simulations-SUC an jeweils einer Klasse vorhanden ...
4. ... ermöglicht dies die Identifikation von zur Substitution realer Objekte geeigneter Simulationsmodelle (Abbildung 6-9 Schritt 4).

Das hier beschriebene und in Abbildung 6-9 skizzierte Prinzip kann einerseits als Vorgehensschema bei der vollständig manuellen Modellerstellung dienen und diese unterstützen, andererseits aber auch eine Blaupause für einen möglichen Modellgenerierungsalgorithmus und damit den Grundstock einer automatischen Modellgenerierung bilden (→ Abschnitt 8.4).

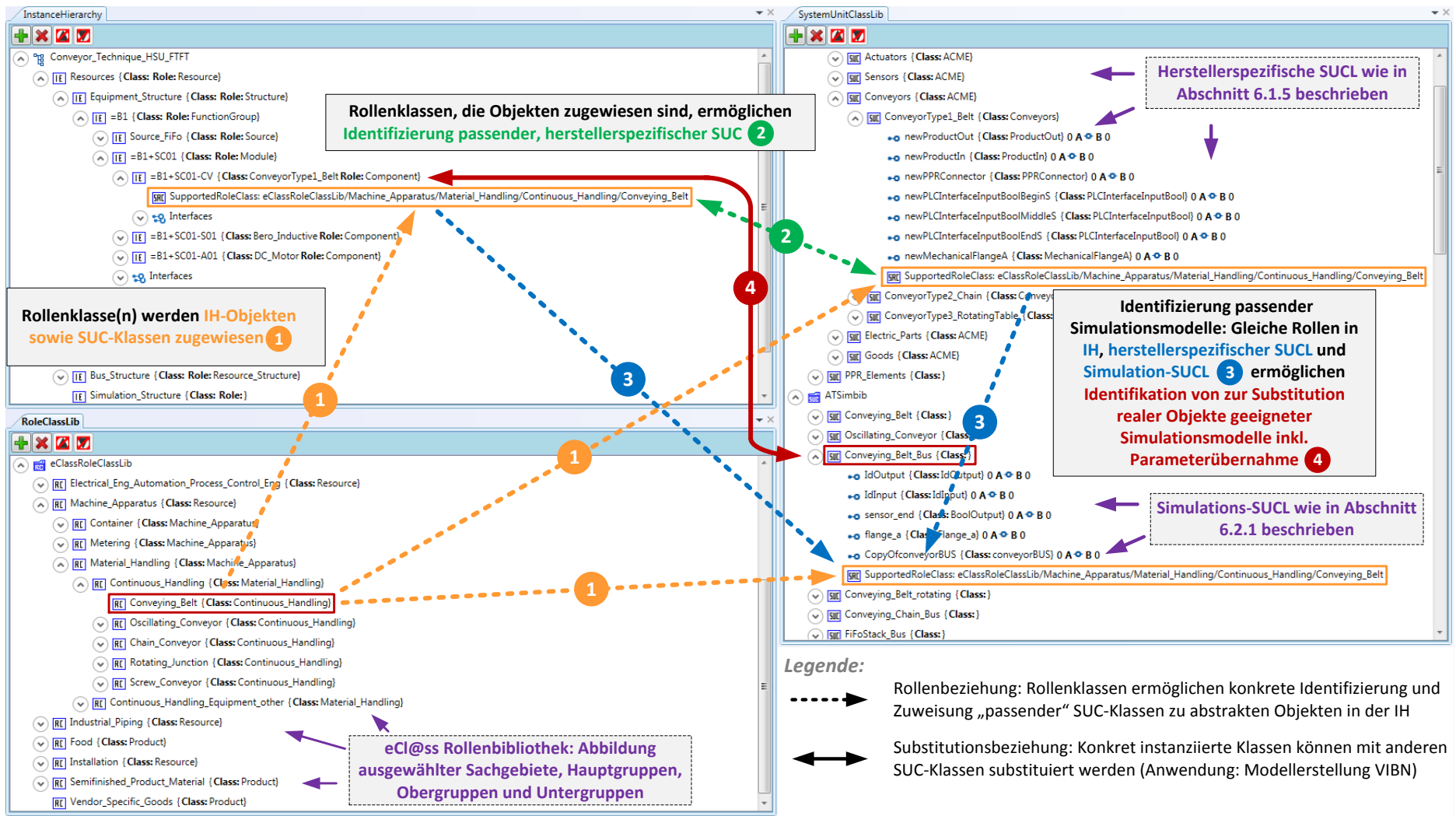


Abbildung 6-9: Identifizierung von Simulationsmodellen mit Hilfe von eCl@ss und beschriebenem Konzept

6.2.2 Merkmalleisten als Basis des Parametermappings und ergänzendes Interfacemapping

Während der Modellerstellung ist nicht nur die Abbildung der Struktur und der Wirkbeziehungen einzelner in der Struktur enthaltener Objekte relevant. Auch die Übernahme von in der Anlagenstruktur in den einzelnen Objekten hinterlegten Parametern in das Simulationsmodell muss gewährleistet sein. Quelle dieser Parameter können die in einer SUCL enthaltenen herstellerspezifischen Klassen von Anlagenkomponenten sein. Die Abbildung von Parametern kann dort vollständig individuell, d. h. nach herstellerspezifischen Vorgaben, erfolgen. Eine mögliche Grundlage für eine herstellerübergreifende Abbildung von Parametern können aber auch standardisierte Merkmalleisten sein (Anm.: Der Begriff Merkmal und Parameter sind in diesem Kontext als synonym zu sehen). Merkmalleisten stellen Zusammenfassungen einzelner Merkmale dar und dienen dem Zweck, Anlagenkomponenten hinsichtlich ihrer Eigenschaften zu beschreiben [RSF08]. Merkmalleisten sind aus vielen Domänen bekannt und werden z. B. durch *eCl@ss* zur Verfügung gestellt. Jedes *eCl@ss*-Bibliothekselement einer Untergruppe enthält neben seiner eindeutigen Bezeichnung in Form einer Nummer eine große Anzahl an Merkmalen. Merkmalleisten können als Hilfsmittel zur herstellerübergreifenden Standardisierung hinsichtlich der Eigenschaften von (technischen) Komponenten angesehen werden. Das dahinterstehende Konzept ist in Abbildung 6-10 dargestellt.

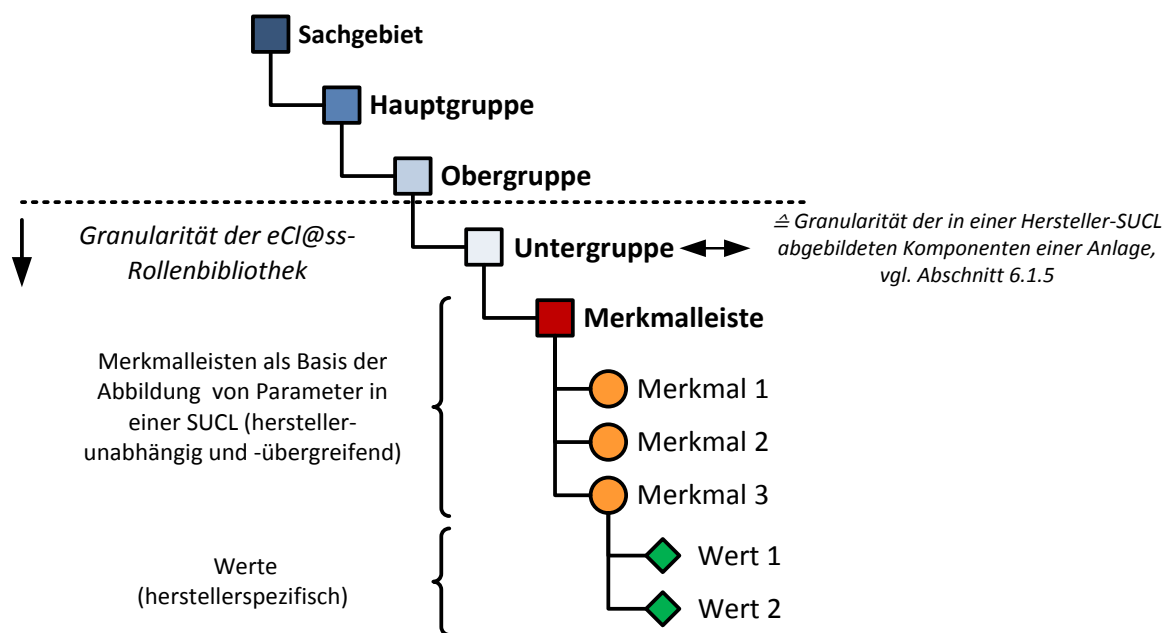


Abbildung 6-10: Merkmalleisten in *eCl@ss* in Anlehnung an [*eCl@ss*[®]] und ihre mögliche Verwendung

Die in *eCl@ss* enthaltenen Merkmalleisten können einer *eCl@ss*-Rollenbibliothek hinzugefügt werden. Sie können genutzt werden, um mittels Mapping eine Verbindung zwischen Parametern, die reale Objekte beschreiben, und Parametern, die für die Simulation benötigt werden, zu schaffen (Abbildung 6-11). Ein Vorteil bei der Verwendung von *eCl@ss* ist, dass die Merkmalleisten auf viele Herstellerkataloge bezogen werden können und ein Mapping zu Objekten in AutomationML (wie z. B. den Objekten in der Simulations-SUCL) daher nur initial erstellt werden muss. *eCl@ss*-Merkmalleisten sind für die spezielle Domäne Automatisierungstechnik nicht immer vollständig und müssen entsprechend ergänzt werden, sollten wesentliche Merkmale nicht vorhanden sein. Die in der AutomationML-Datei abbildbaren AttributeNameMapping-Objekte, die die technische

Realisierung des Mappings darstellen, sind für jedes Merkmal einmalig anzulegen. Das hier vorgestellte Konzept ist in Abbildung 6-11 dargestellt.

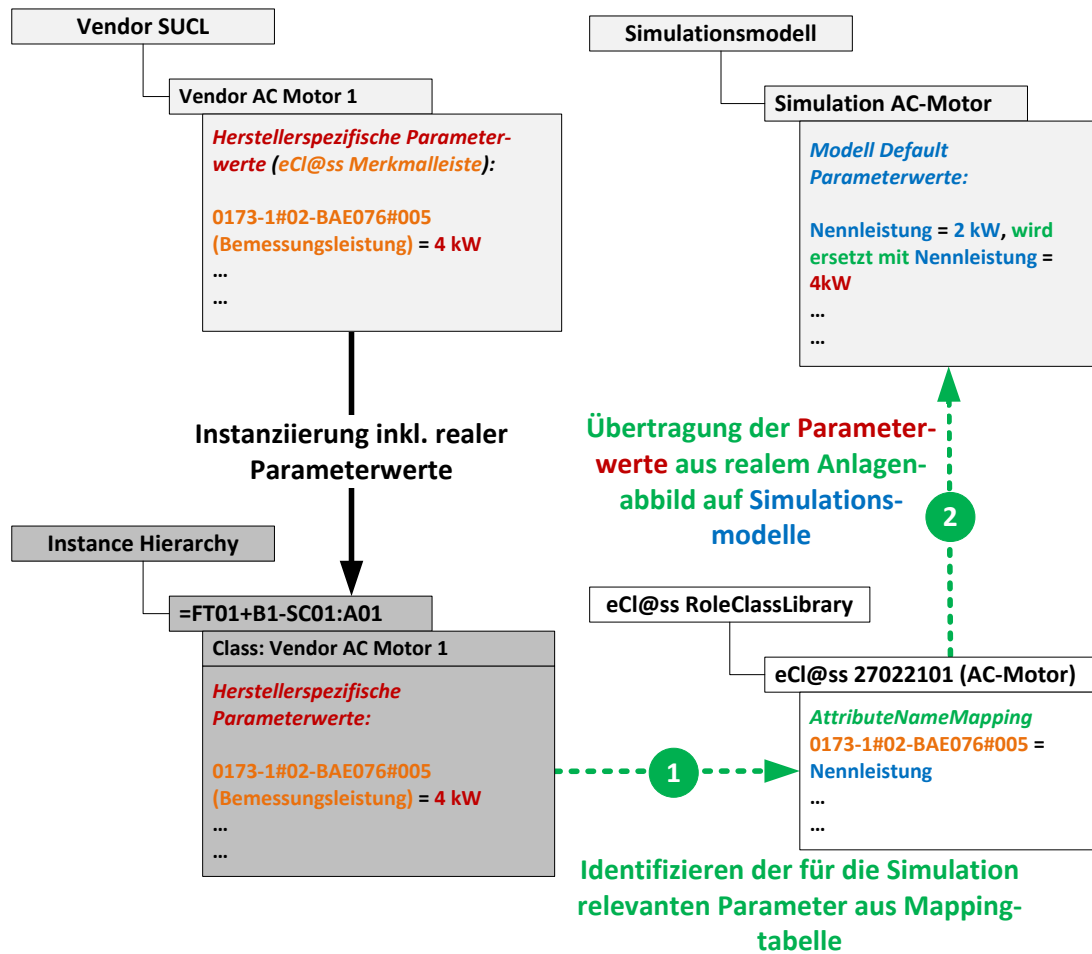


Abbildung 6-11: Konzept für das Mapping von Parametern mit Hilfe von eCl@ss

Neben der Zuordnung von herstellere-spezifischen Parametern auf Simulationsparameter ist zudem die Zuordnung der in der Instanzhierarchie abgebildeten Schnittstellen und ihrer Verbindungen untereinander auf die spezifischen Schnittstellen der Simulationsmodellbibliothek vorteilhaft. Dadurch kann ein direktes Übertragen der in der Anlagenstruktur abgebildeten Wirkbeziehungen (→ Abschnitt 6.1.3.4) in das Simulationsmodell erreicht werden. Dies kann prinzipiell nach demselben Schema, wie es bereits für Merkmale beschrieben wurde, erfolgen. Die technische Realisierung kann mittels in AutomationML abbildbaren *InterfaceNameMapping*-Objekten erfolgen, um eine semantische Zuordnung von herstellere-spezifischen Schnittstellen zu Schnittstellen in einer Simulationsmodellbibliothek zu erreichen. Zusätzlich müssen die Verbindungen zwischen den herstellere-spezifischen Schnittstellen innerhalb des Anlagenstrukturmodells (in Form von InternalLinks) in das Simulationsmodell übertragen werden. Die Verwendung vordefinierter Merkmalleisten und die Nutzung damit verbundener Synergieeffekte wie herstellere-übergreifende Standardisierung und hohe Wiederverwendbarkeit stoßen hierbei aber an Grenzen. Derzeit sind keine standardisierten „Schnittstellenleisten“ oder Ansätze bekannt, die auf eine Definition selbiger zielen. Die Definition der Schnittstellen wie auch der entsprechenden Mapping-Objekte muss derzeit komplett manuell vorgenommen werden. Auch hier kann wiederum von Einmalaufwand gesprochen werden, insofern die gleiche Simulationsbibliothek und eine herstellere-übergreifende ICL zum Einsatz kommen. Abbildung 6-12 zeigt den Ablauf des Mappings von herstellere-spezifischen

schen Schnittstellen zu simulationsspezifischen Schnittstellen sowie das Übertragen der Verbindungsinformation aus Anlagenstrukturmodell in Simulationsmodell.

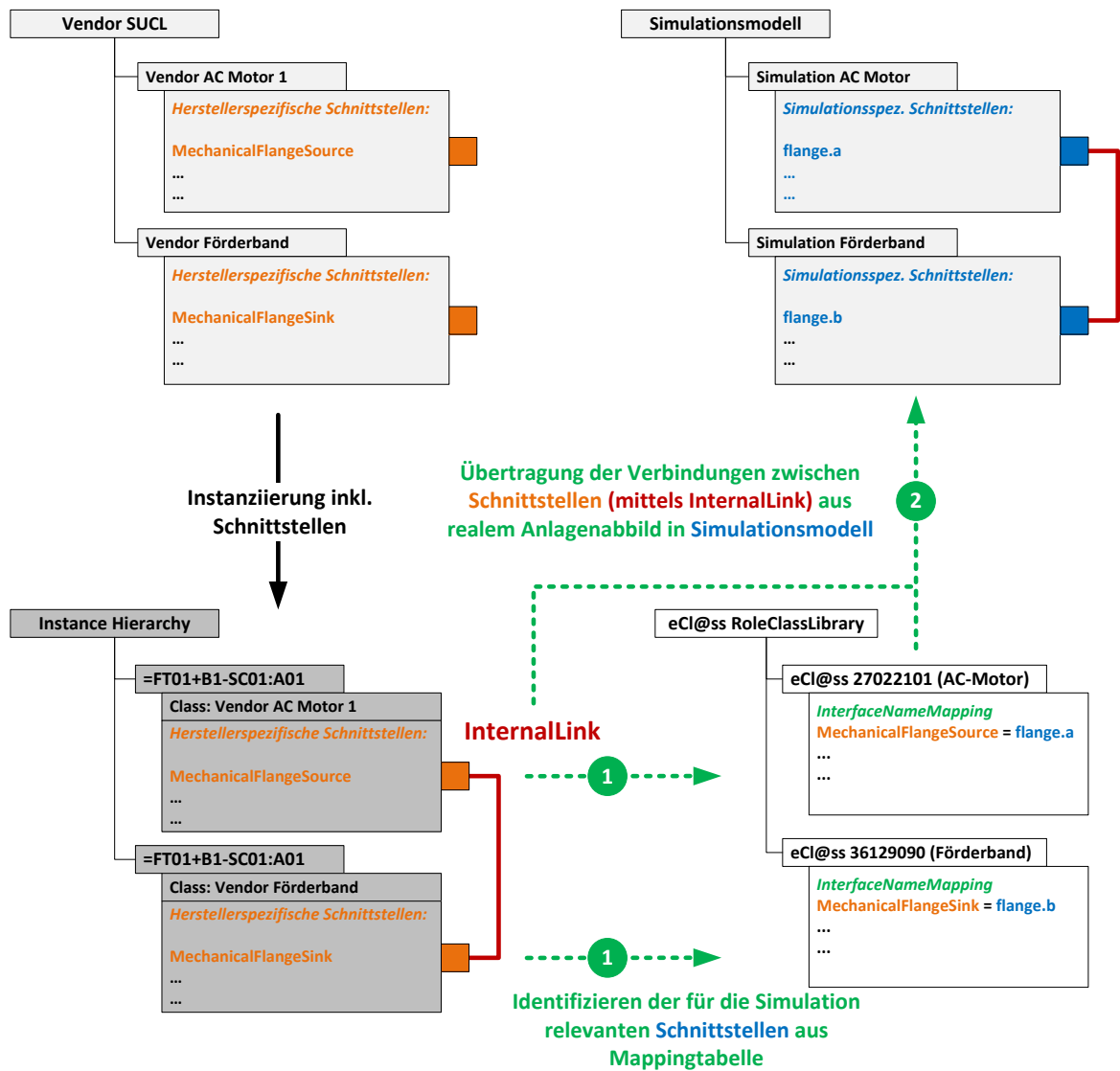


Abbildung 6-12: Konzept für das Mapping von Schnittstellen mit Hilfe von eCl@ss

7 Definition von Modellarten und Detaillierungsstufen

Nach der Beschreibung der für eine virtuelle Inbetriebnahme nutzbaren Datenquellen werden in den folgenden Abschnitten die zum Einsatz kommenden Simulationsmodelle näher betrachtet. Gemäß *Lösungsbaustein 2* wird eine Systematik vorgestellt, mit der sich heterogene Detaillierungsstufen für Anlagensimulationsmodelle im Konkreten implementieren lassen. Im Fokus der Betrachtungen stehen die für den fertigungstechnischen Anlagenbau im Vordergrund stehenden Stückgutbewegungen. Die Betrachtungen reichen von der Nachbildung eines einfachen, zeitbasierten Verhaltens hin zu möglichem physikalischem Verhalten. Das nachfolgende Kapitel steht unter Leitfrage II: *Wie kann man virtuelle Inbetriebnahme umsetzen?*

7.1 Grundlegende Betrachtung der Granularität und Detaillierung von Simulationsmodellen

7.1.1 Zusammenhang zwischen Granularität und Detaillierung

Die Granularität und die Detaillierung eines Anlagensimulationsmodells spielen für die virtuelle Inbetriebnahme eine wichtige Rolle. Granularität beschreibt in diesem Zusammenhang die Auflösung des für eine virtuelle Inbetriebnahme genutzten Anlagensimulationsmodells und damit, welche Hierarchieebenen einer Anlage als (Teil-)Simulationsmodelle nachgebildet werden sollen. Ein grobgranulares Anlagensimulationsmodell würde aus einem einzigen, monolithischen Simulationsmodell bestehen – die betrachtete Hierarchieebene ist in diesem Fall die Gesamtanlagenebene oder die Funktionsgruppenebene. Ein grobgranulares Anlagensimulationsmodell enthält daher alle Prozessschritte einer Anlage (deren Nachbildung für die virtuelle Inbetriebnahme wesentlich sind, Abschnitt 6.1.3.2) in *einem* Simulationsmodell. Ein mittel- bzw. feingranulares Anlagensimulationsmodell würde dagegen aus (vielen) einzelnen Teil-Simulationsmodellen und damit aus der Nachbildung von Modulen und Komponenten einer Anlage bestehen. Teil-Simulationsmodelle auf Komponenten oder Modulebene bilden daher nur wenige Prozessschritte (je nach spezifischem Verhalten der Komponenten) nach. Teil-Simulationsmodelle auf Modul- oder Komponentenebene ermöglichen den Aufbau bzw. den Einsatz einer Simulationsmodellbibliothek. Für die automatische Modellgenerierung ist dies eine notwendige Voraussetzung (→ Abschnitt 5.2). Beim Einsatz grobgranularer Simulationsmodelle ist der Einsatz einer Simulationsmodellbibliothek nicht möglich – die Simulationsmodellbibliothek müsste in diesem Fall ganze Anlagen vorhalten.

Detaillierung dagegen beschreibt ein Maß, wie genau Simulationsmodelle (Teil-)Prozesse einer Anlage nachbilden. Simulationsmodelle mit niedriger Detaillierung bilden nur einzelne Teilaspekte eines realen Prozesses nach – z. B. durch eine rein zeitbasierte Betrachtung. Hohe Detaillierung bedeutet dagegen, dass ein möglichst „realistisches“ Abbild der Anlage erreicht werden soll – dies wird in der Regel durch die Nachbildung physikalischen Verhaltens ermöglicht. Generell gilt: Mit steigender Detaillierung von Simulationsmodellen ...

- ... steigt der Modellierungsaufwand
- ... steigt die Kompliziertheit
- ... steigt der Aufwand für die Parametrierung
- ... steigt die Anzahl an Simulations-Events, so dass der Rechenaufwand für die Durchführung einer Simulation steigt

In Bezug auf den Zusammenhang zwischen Granularität und der möglichen Detaillierung einer Simulation ergeben sich verschiedene Kombinationsmöglichkeiten. Diese Kombinationsmöglichkeiten werden im Nachfolgenden kurz beschrieben und mit Hinsicht auf ihre Eignung für eine virtuelle Inbetriebnahme bewertet.

Für grobgranulare Simulationsmodelle (Simulation auf Anlagenebene, Funktionsgruppenebene) gilt:

- Bei *hoher Detaillierung* werden alle in einem Gesamtprozess beinhalteten Prozessschritte in einem einzigen, *monolithischen* Anlagensimulationsmodell sehr genau abgebildet. Das Anlagensimulationsmodell wird in diesem Fall naturgemäß sehr kompliziert. Der Aufwand zur Modellierung eines derartigen Anlagensimulationsmodells ist als erheblich einzuschätzen. Das Vorhalten derartiger Modelle in einer Simulationsmodellbibliothek ist naturgemäß nicht möglich – die Simulationsmodellbibliothek müsste ganze Anlagen enthalten.
- Bei *niedriger Detaillierung* werden einzelne Prozessschritte nur sehr vereinfacht, möglicherweise gar nicht betrachtet. Ein Beispiel hierfür kann die Abbildung einer kompletten Anlage als FIFO-Speicher sein. Die Aussagefähigkeit des Anlagensimulationsmodells kann in diesem Fall eingeschränkt, eine Nutzung für eine virtuelle Inbetriebnahme nicht möglich sein. Eine Ausnahme hiervon sind Anlagenmodelle, die als Quelle oder Senke für andere Anlagensimulationsmodelle dienen.

Fazit: Grobgranulare Anlagensimulationsmodelle auf Anlagen- oder Funktionsgruppenebene eignen sich nicht für eine virtuelle Inbetriebnahme. Eine Ausnahme hiervon besteht, wenn diese Anlagensimulationsmodelle die Quellen oder Senken für andere Anlagensimulationsmodelle darstellen und die Quellen bzw. Senken in niedriger Detaillierung modelliert sind.

Für mittelgranulare Modelle (Simulation auf Modulebene) gilt:

- Bei *hoher Detaillierung* werden alle in einer Prozessoperation beinhalteten Prozessschritte in einem Simulationsmodell eines Moduls sehr genau modelliert. Das Anlagensimulationsmodell besteht in diesem Fall aus Modulen. In Abhängigkeit der Anzahl von nachzubildenden Prozessschritten ist die Situation als ähnlich wie bei grobgranularen Simulationsmodellen einzuschätzen: Das Simulationsmodell eines Moduls kann in Abhängigkeit der nachzubildenden Prozessschritte sehr kompliziert werden. Das Vorhalten derartiger Modelle in einer Simulationsmodellbibliothek ist erschwert, da Prozessschritte prinzipiell beliebig miteinander kombiniert werden können. Dies impliziert eine große Anzahl an Modellen, die in einer Simulationsmodellbibliothek vorgehalten werden müssten.
- Bei *niedriger Detaillierung* werden einzelne Prozessschritte in einer Prozessoperation, die dann gesamtheitlich für alle Prozessschritte in einem Modul steht, zusammengefasst. Voraussetzung dafür ist, dass die zu testenden Prozessschritte dadurch hinreichend für den Test der Steuerung beschrieben werden können. Dies könnte dann der Fall sein, wenn es sich um die Abbildung von „Standard-Prozessoperationen“ wie dem Transport von Stück-

gütern auf einer Zuführ-Förderstrecke handelt. Die Aussagefähigkeit niedrig detaillierter Simulationsmodelle auf Modulebene ist eingeschränkt, sie könnten jedoch ein grundlegendes Testen einfacher Steuerungsfunktionen ermöglichen.

Fazit: Mittelgranulare Modelle auf Modulebene können sich in niedriger Detaillierung für eine virtuelle Inbetriebnahme eignen.

Für feingranulare Modelle (Simulation auf Komponentenebene) gilt:

- Bei *hoher Detaillierung* wird das Verhalten einer Komponente sehr genau abgebildet. Die Betrachtung bezieht sich immer nur auf eine einzelne Komponente. Das Anlagensimulationsmodell besteht daher aus vielen einzelnen Komponenten, die miteinander kombiniert werden. Die Kompliziertheit der Simulationsmodelle ist auf wenige Prozessschritte und damit das spezifischen Verhalten einer Komponente beschränkt. Werden Komponentenmodelle in einer Simulationsmodellbibliothek vorgehalten, konzentriert sich die Modellierung auf das Parametrieren und Verknüpfen einzelner Komponentenmodelle. Dadurch lassen sich beliebige Prozessoperationen nachbilden.
- Bei *niedriger Detaillierung* gilt sinngemäß dasselbe wie bei hoher Detaillierung. Die Aussagefähigkeit der Simulationsmodelle ist im Vergleich zur detaillierten Betrachtung jedoch eingeschränkt. Im Vergleich zur detaillierten Modellierung ist daher nur ein grundlegendes Testen (d. h. eine Überprüfung ohne genaue Beachtung einzelner Parameterwerte, die einen Einfluss auf das Verhalten einer Komponente haben könnten) von Steuerungsfunktionen möglich.

Fazit: Feingranulare Modelle auf Komponentenebene können sich für eine virtuelle Inbetriebnahme in hoher und niedriger Detaillierung eignen.

Die Zusammenhänge zwischen Granularität und Detaillierung eines Simulationsmodells sind zusammenfassend in Tabelle 9 dargestellt.

Granularität des Anlagensimulationsmodells	Hierarchieebene (vgl. Abschnitt 6.1.3.1)	Detaillierung Simulationsmodell(e)	Beispielmodelle	Bewertung	Für VIBN geeignet
"Grob" (→ monolithisches Anlagensimulationsmodell)	Gesamtanlagenebene, Funktionsgruppenebene	Hoch ↑	Detailliertes Modell einer kompletten Anlage in einem monolithischen Simulationsmodell	Kompliziertheit des Simulationsmodells sehr hoch, sehr hoher Modellierungsaufwand, sehr hoher Rechenaufwand. Kann nicht in Simulationsmodellbibliothek vorgehalten werden.	-
		Niedrig ↓	Wenig detailliertes, monolithisches Anlagen-Simulationsmodell, z.B. Anlage als FIFO-Speicher	Keine Aussage über spezifisches Verhalten einzelner Anlagenteile möglich	-
"Mittel" (→ Anlagensimulationsmodell besteht aus Teilmodellen)	Modulebene	Hoch ↑	Detailliertes Simulationsmodell eines kompletten Moduls	Kompliziertheit des Teilsimulationsmodells kann hoch sein, je nach Prozessschritten, die nachgebildet werden müssen	-
		Niedrig ↓	Einfaches Modell, das die Prozessoperation eines Moduls (z.B. Transport) nachbildet	Kompliziertheit durch Fokussierung auf Prozessoperation des Moduls verringert, Aussagefähigkeit aber eingeschränkt	+
"Fein" (→ Anlagensimulationsmodell besteht aus vielen Teilmodellen)	Komponentenebene	Hoch ↑	Detailliertes Motormodell, z.B. zum Test von Steuerungsfunktionen und zur genauen Bestimmung von Parameterwerten wie Rampenvorgaben	Kompliziertheit durch Fokussierung auf wenige Prozessschritte verringert	+
		Niedrig ↓	Einfaches Motormodell, zum grundlegenden Test von Steuerungsfunktionen (z.B. Ansteuerung des Motors)	Kompliziertheit durch Fokussierung auf wenige Prozessschritte verringert, Aussagefähigkeit aber eingeschränkt	+

Tabelle 9: Zusammenhang zwischen Granularität und Detaillierung eines Simulationsmodells

7.1.2 Relevante Hierarchieebenen

Aus den bisherigen Beschreibungen ergibt sich, dass für eine virtuelle Inbetriebnahme nur mittel- bzw. feingranular aufgelöste Simulationsmodelle in Modul- oder Komponentenebene in Betracht kommen. Module innerhalb einer fertigungstechnischen Anlage führen meist gekapselte Prozessoperationen durch und besitzen daher in der Regel klare Systemgrenzen, die sie von anderen Modulen innerhalb einer Anlage abgrenzen. Module können daher als eine grundlegend zu betrachtende Hierarchieebene in Betracht kommen.

Einzelne Prozessschritte werden durch die Ansteuerung von Komponenten umgesetzt (→ Abschnitt 6.1.3.2). Im gemeinsamen Verbund ergeben Komponenten eine Einheit, die eine bestimmte Prozessoperation innerhalb des Gesamtsystems umsetzt. Abbildung 7-1 zeigt diese Sichtweise unter Zuhilfenahme der in Abschnitt 6.1.3.1 definierten Anlagenhierarchie.

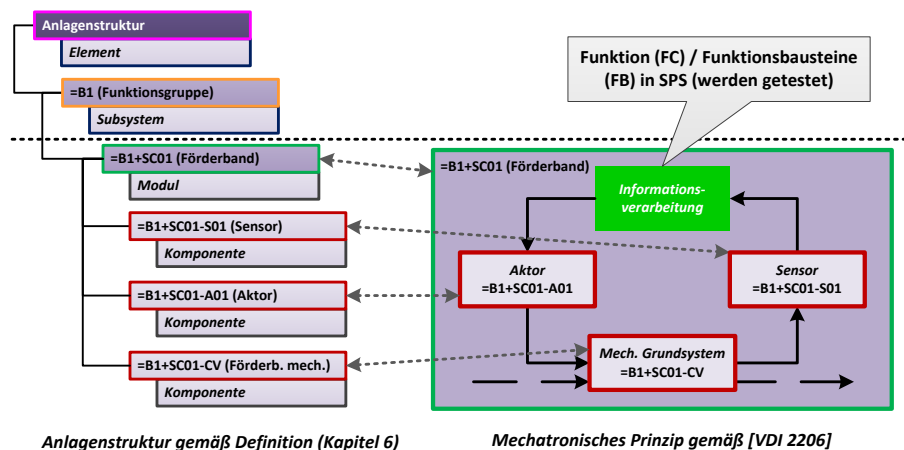


Abbildung 7-1: Für eine virtuelle Inbetriebnahme relevante Hierarchieebenen in Bezug auf die Anlagenstruktur

7.1.3 Ableitung grundlegender Modellarten

Aus der entwickelten Sichtweise lassen sich zunächst zwei grundlegende Modellarten ableiten: Ein Modell des kompletten Moduls als *integriertes Modell der Domänen Mechanik und Elektrotechnik* (Abbildung 7-2 links), welches eine spezifische Prozessoperation innerhalb eines Systems gesamtheitlich nachbildet. Ferner können Modelle auf *Komponentenebene*, bei denen Komponentenmodelle einzelne Prozessschritte ausführen und im Verbund die Prozessoperation eines Moduls nachbilden (Abbildung 7-2 rechts), identifiziert werden.

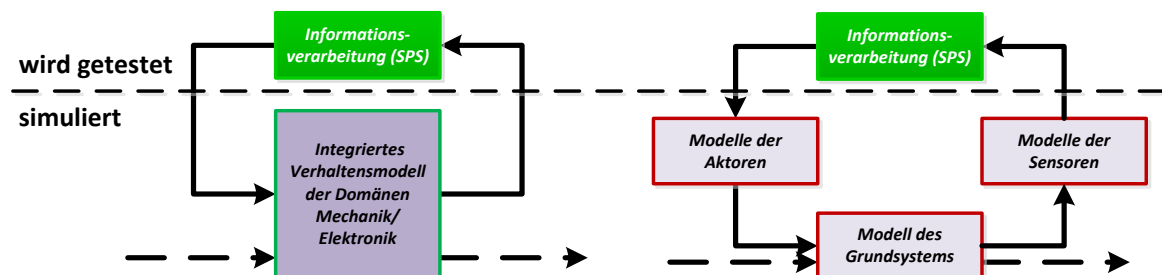


Abbildung 7-2: Grundlegende Modellarten

Diese beiden grundlegenden Modellarten bilden die Basis für die weiteren Überlegungen hinsichtlich für die virtuelle Inbetriebnahme geeigneter Modellarten.

7.2 Modellartendefinition und deren wesentliche Aspekte

Basierend auf den bisher identifizierten grundlegenden Modellarten können spezifische Modellarten definiert werden, die sich für einen Einsatz in einer *virtuellen Inbetriebnahme* eignen können. Die hier vorgestellten Modellarten basieren auf wissenschaftlichen Vorarbeiten, die vom Autor während der Erstellung der VDI-Richtlinie [VDI 3693-1] geleistet wurden. Die hier vorgestellten Modellarten finden sich daher in ähnlicher Form auch dort. Entsprechend des in dieser Arbeit im Vergleich zur [VDI 3693-1] verkleinerten Lösungsraums (→ Abschnitt 5.3.2.1) bei der Betrachtung automatisierter Anlagen umfassen die hier betrachteten Modellarten nur einen Teil der in der Richtlinie beschriebenen und potenziell zum Einsatz während einer *virtuellen Inbetriebnahme* infrage kommenden Modellarten. Die Vorstellung der einzelnen Modellarten erfolgt in den nächsten Abschnitten gemäß des Prinzips *vom Groben ins Detail* und daher mit steigender Detaillierung.

Die hier vorgestellten Modellarten beziehen sich hinsichtlich der Granularität immer auf ein komplettes Modul. Für Modelle auf Komponentenebene wird folgende Festlegung getroffen: Alle nachzubildenden Prozessschritte, die gesamtheitlich eine Prozessoperation bilden, sind im selben Detaillierungsgrad nachzubilden. Durch diese Festlegung werden konsistente Schnittstellen zwischen Komponentenmodellen sichergestellt. Dies senkt den Modellierungsaufwand hinsichtlich der Sicherstellung konsistenter Schnittstellen erheblich. Fälle, in denen sich unterschiedliche Schnittstellen zwischen Modulen aufgrund des gewollten heterogenen Detaillierungsgrads eines Anlagensimulationsmodells (→ Abschnitt 5.2) nicht vermeiden lassen, werden in Abschnitt 8.3 gesondert behandelt.

7.2.1 Einfache Betrachtung von Abläufen und Bewegungen

Bei Betrachtung des Detaillierungsgrads von Simulationsmodellen der Domänen Mechanik und Elektrotechnik kann zunächst eine grundlegende Modellart vorgesehen werden: Die Simulation beider Domänen zusammen in nur einem Modell. Dies entspricht einer Simulation auf Modulebene. Das Verhalten des Moduls wird auf dieser Hierarchieebene als Ganzes nachgebildet. Aus Gesamtsystemsicht entspricht dies einer mittel-granularen Modellierung. Das Modell auf Modulebene kann aus der Perspektive des Gesamtsystems als Black-Box gesehen werden, es handelt sich also um ein geschlossenes System unter Vernachlässigung des inneren Aufbaus. Da nur das Verhalten eines kompletten Moduls als ein konkretes Simulationsmodell abgebildet wird, kann dies durch eine einfache, durch eine Totzeit verzögerte Transformation von Stell- und/oder Steuersignalen in Rückmeldesignale geschehen („*Einfache Totzeitmodelle*“). *Einfache Totzeitmodelle* auf Modulebene können als signalflussbasierte Modelle umgesetzt werden, da ihr Verhalten keinem physikalischen Prinzip folgt.

Totzeitmodelle benötigen nur wenige Parameter: Wesentlich ist die Totzeit T_t selber. Der Parameter Totzeit kann wiederum aus anderen Parametern, wie z. B. der Position eines Sensors (bezogen auf die Länge eines Förderbands) und der Fördergeschwindigkeit ermittelt werden. Die Berechnung der Totzeit (= Zeit, die vergeht, bis bei konstanter Fördergeschwindigkeit v eine Position x erreicht wurde) kann mittels einfacher Formel geschehen ($T_t = x/v$). Liegen keine Parameterwerte vor, aus denen sich die Totzeit ermitteln ließe, muss die Totzeit geschätzt werden. Dies hat zur Folge, dass bestimmte steuerungstechnische Funktionen nur grundlegend auf Funktionsfähigkeit getestet werden können. So ist beispielsweise eine Bestimmung des für eine Laufzeitüberwachung notwendigen (und im Steuerungscode hinterlegten) Parameters „Laufzeit“ nicht möglich.

Die Abbildung von Störungen (technische wie organisatorische, Abschnitt 2.8.3) wie die bereits erwähnte Laufzeitüberwachung oder das Erzeugen eines Paarfehlers (= Situationen, in denen mehrere Sensoren aktiv sind, es aber laut Ablauf nicht sein dürften) und andere (für eine Übersicht siehe Abschnitt 2.8.3) sind im Modell vorzusehen und müssen für deterministische sowie stochastische Untersuchungen geeignet sein. Deterministisch bedeutet in diesem Zusammenhang, dass eine Störung gezielt herbeigeführt werden kann, d. h. durch manuellen Eingriff zur Laufzeit der Simulation. Stochastisch bedeutet, dass die Störung mit einer zu definierenden Wahrscheinlichkeit während eines Simulationslaufs eintritt. Die Bildung von Störungen wird bei Totzeitmodellen entweder durch die Beeinflussung der an die SPS gesendeten Sensorsignale oder durch eine Beeinflussung des Parameters T_t ermöglicht. Eine Betrachtung von Stückgütern erfolgt in dieser Modellierungsstufe nicht.

Abbildung 7-3 zeigt das hier beschriebene Totzeitmodell am Beispiel eines einfachen Förderbands mit zwei Sensoren. Als Parameter sind lediglich die Totzeiten T_{t1} für Sensor 1 und T_{t2} für Sensor 2 vorzusehen.

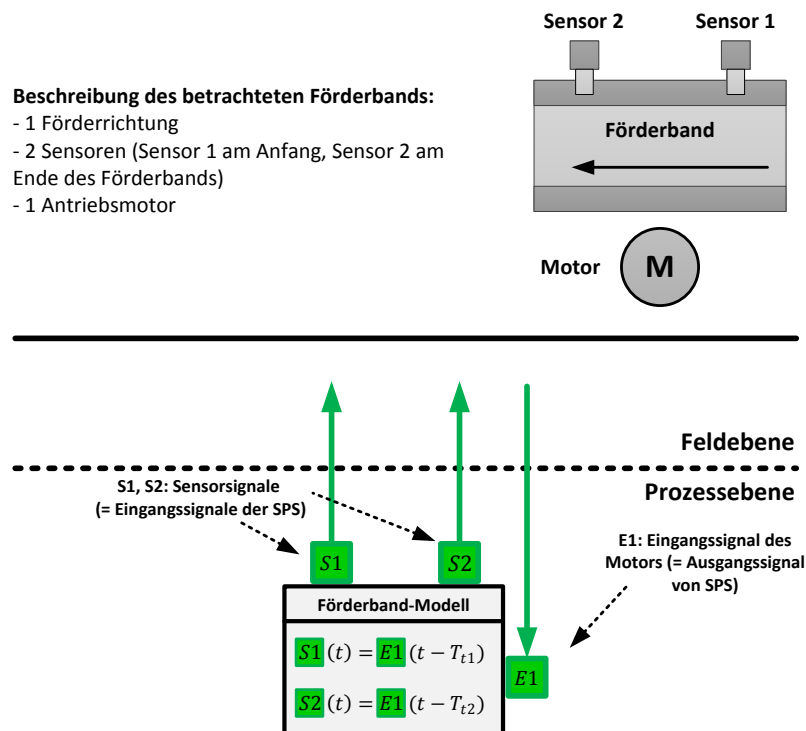


Abbildung 7-3: Einfache Betrachtung von Abläufen und Bewegungen mittels Totzeitmodellen

7.2.2 Betrachtung beschleunigungsfreier Bewegungen

Oftmals ist eine Betrachtung auf Modulebene nicht ausreichend. Dies kann immer dann der Fall sein, wenn ...

- ... die Auswirkung unterschiedlicher Parameterwerte auf das Verhalten einzelner Komponenten untersucht werden soll.
- ... das Zusammenwirken von einzelnen Komponenten anders ist, als im „Normalfall“ zu erwarten wäre. Ein „Normalfall“ kann z. B. die Verwendung von standardisierten Modulen, die gemäß vorgegebener „Best-Practices“ (→ Abschnitt 2.5.1) umgesetzt wurden, sein.

- ... zu viele Prozessschritte innerhalb des Moduls betrachtet werden müssen, und diese Prozessschritte nur mit hohem Modellierungsaufwand zu einer Prozessoperation zusammengefasst werden können.

In diesen Fällen ist ein Betrachten des Verhaltens der einzelnen Komponenten der Domänen Elektrotechnik und Mechanik erforderlich. Dies kann auf der hierarchisch tiefer liegenden Komponentenebene geschehen, bei der durch das Zusammenwirken der einzelnen Komponenten die Prozessoperation eines Moduls entsteht und so dem Modul seine individuelle Charakteristik verliehen wird.

In der aus Systemsicht fein-granular aufgelösten Komponentenebene wird daher das konkrete Verhalten der beteiligten Komponenten simuliert (= Nachbilden einzelner Prozessschritte). Aus Gesamtsystemsicht ist das mechatronische Modul in diesem Fall eine White-Box. Innerhalb der einzelnen Komponenten kann dies in der Domäne Elektrotechnik durch eine einfache Transformation von Stell- und/oder Steuersignale in Rückmeldesignale umgesetzt werden (als *Einfache Gerätemodelle*). *Einfache Gerätemodelle* bilden das Verhalten des jeweiligen Geräts parametrierbar ab, betrachten aber kein physikalisches Verhalten. *Einfache Kinematikmodelle*, die die Bewegung eines Stückguts unter Vernachlässigung der einwirkenden Kräfte betrachten, bilden die Modelle des Grundsystems. Die beschleunigungslose Bewegung erfolgt auf einem festgelegten Pfad, was eine Translation oder Rotation oder auch Kombinationen daraus sein können. Bei *einfachen Geräte- und Kinematikmodellen* ist eine signalflussbasierte Modellierung ausreichend, da kein physikalisches Verhalten abgebildet wird.

Fehlerfälle sind vorzusehen und müssen für deterministische sowie stochastische Untersuchungen geeignet sein. Grundsätzlich ist hierfür die Beeinflussung der zur SPS gesendeten Sensorsignale (wie in Abschnitt 7.2.1 beschrieben) möglich. Die Umsetzung kann, ergänzend dazu, durch eine Beeinflussung von Parameterwerten geschehen. Wird beispielsweise der Parameter „Motordrehzahl“ bewusst gesenkt, würde sich die Fördergeschwindigkeit verringern. Damit könnte eine Laufzeitüberwachung ausgelöst werden. Eine Überprüfung von Kausalketten (d. h. das Nachvollziehen, welche Ursache eine Störung haben könnte) ist damit jedoch nicht möglich, da unklar bleibt, was die Motordrehzahl beeinflusst hat.

Abbildung 7-4 zeigt die hier vorgestellte Modellart, analog zum Beispiel aus Abschnitt 7.2.1. Als Parameter werden benötigt:

Teilmodell Motor:

- Motordrehzahl n .

Teilmodell Förderband:

- Durchmesser der Antriebsrolle $d_{\text{Antriebsrolle}}$.

Teilmodell Sensor 1:

- Position des Sensors am Förderband ($Pos1_{FB}$); Schaltabstand (= Bereich, in dem der Sensor bei vorhandenem Stückgut ein Signal auslöst) des Sensors SB_1 .

Teilmodell Sensor 2:

- Position des Sensors am Förderband ($Pos2_{FB}$); Schaltabstand des Sensors SB_2 .

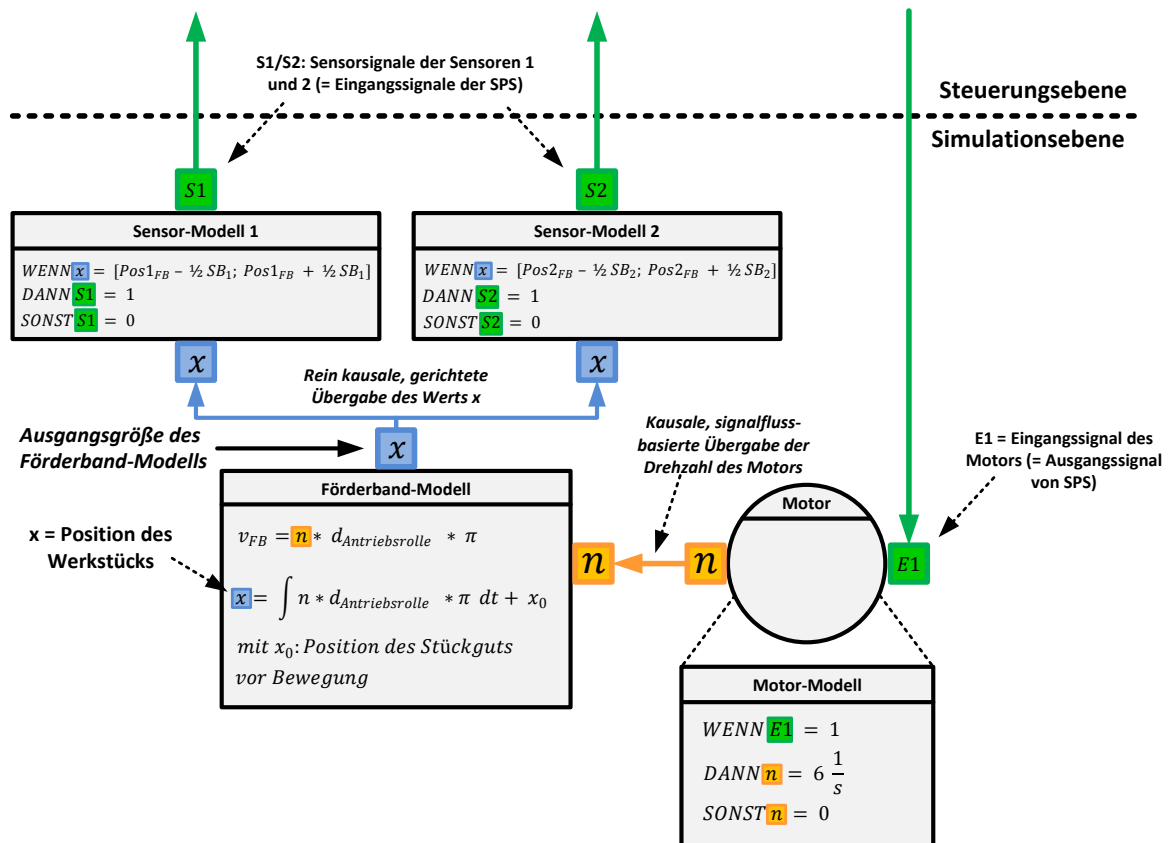


Abbildung 7-4: Betrachtung einer beschleunigungsfreien Bewegung auf definiertem Pfad

7.2.3 Betrachtung von mehrdimensionalen Bewegungen auf definierten Pfaden

Wenn Parameterwerte mit großer Genauigkeit bestimmt werden müssen, wenn physikalische Effekte wie z. B. Reibung oder Schlupf einen direkten Einfluss auf die umzusetzenden Steuerungsfunktionen haben, oder wenn sich Stückgüter in mehreren Dimensionen auf festgelegten Pfaden bewegen können, ist eine detaillierte Betrachtung der einzelnen Prozessschritte notwendig. Muss die maximale Drehzahl einer mechanischen Apparatur vorab genau definiert werden, um Beschädigungen bei der realen Inbetriebnahme zu verhindern, ist es beispielsweise notwendig, auftretende Kräfte- und Momentengleichgewichte zu betrachten.

In der Domäne Elektrotechnik kann dies durch Komponentenmodelle umgesetzt werden, die Potential- und Flussgrößen physikalischer Systeme betrachten (als *Gerätemodelle*). Gerätemodelle müssen spezifisches Verhalten wie z. B. Rampenfunktionen (= langsames Hochfahren eines Motors bis auf Solldrehzahl/langsames Herunterregeln der Solldrehzahl bis zum Stillstand) nachbilden können. Simulationsmodelle, bei denen die Kräfte, die eine Bewegung eines Stückguts hervorrufen, betrachtet werden, bilden die Modelle des Grundsystems (*einfache Dynamikmodelle*). Die Bewegung von Stückgütern erfolgt auf festgelegten Pfaden, was eine Translation oder Rotation oder auch Kombinationen daraus sein können. Eine freie Bewegung im Raum ist jedoch ausgeschlossen. Die Abbildung mehrerer Stückgüter auf einem Förderband sollte möglich sein. In diesem Fall summieren sich die wirkenden Kräfte und Momente.

Fehlerfälle sind vorzusehen und müssen für deterministische sowie stochastische Untersuchungen geeignet sein. Fehlerfälle können durch die Beeinflussung von Sensordaten oder durch die

Änderung von Parametern der einzelnen Komponentenmodelle hervorgerufen werden (→ Abschnitte 7.2.1, 7.2.2). Fehlerzustände können in der hier beschriebenen Modellart zudem Kausalketten aufzeigen: Die Erhöhung des Werts des Parameters „Reibung“ an einem Förderband (die eine mechanische Störung wie die Schwergängigkeit einer mechanischen Apparatur nachbilden könnte) würde zu einem erhöhten Drehmoment am Motor führen. Dies kann eine Störung des Motors verursachen. Das gezielte Testen einzelner Fehlerzustände ist daher nach dem Ursache-Wirkungsprinzip möglich. Ferner sollten Kollisionen zwischen Stückgütern in den betrachteten Freiheitsgraden möglich sein. In der hier beschriebenen Modellart ist das Aufzeigen daraus resultierender Kausalketten jedoch nur sehr eingeschränkt möglich, da die Bewegung nur auf festgelegten Pfaden erfolgt. Eine Kollision könnte daher „nur“ eine (ungewollte) Positionsänderung in einem der betrachteten Freiheitsgrade auslösen oder eine Erhöhung der Reibung bewirken.

Die Modellierung der hier beschriebenen Modellart ist vorzugsweise in einer objektorientierten, a-kausalen Modellierungssprache vorzunehmen, die sich für die Darstellung physikalischen Verhaltens eignet. Der Vorteil bei der Verwendung einer a-kausalen Modellierungssprache liegt darin, dass es keine explizite Festlegung auf Ein- und Ausgangsgrößen gibt. Einzelne Teil-Simulationsmodelle werden vielmehr über physikalische Verbindungen miteinander verbunden. Die Wirkrichtung einer physikalischen Größe (z. B. das durch einen Motor erzeugte Drehmoment, das auf ein Förderband übertragen wird) ergibt sich aus den in den Simulationsmodellen hinterlegten Gleichungen. Wird beispielsweise ein Förderband durch erhöhte Stückgut-Masse gehemmt, übt dies ein hohes Drehmoment auf den Motor aus.

Abbildung 7-5 zeigt die hier vorgestellte Modellart anhand eines Beispiels. Betrachtet wird ein Förderband, das Stückgüter in x-Richtung transportiert. Mittig des Förderbands befindet sich ein pneumatischer Zylinder, der es erlaubt, Stückgüter in y-Richtung vom Förderband in eine hier nicht näher definierte Senke zu verschieben. Enthaltene Simulationsmodelle sind ein Motormodell (= Aktor 1), ein Modell eines pneumatischen Zylinders (= Aktor 2), ein Förderbandmodell (= mechanisches Grundsystem) sowie zwei Sensormodelle (= Sensor 1 und Sensor 2).

Für die Parametrierung der einzelnen Teil-Simulationsmodelle werden folgende Parameter benötigt (Anm.: Reibung innerhalb des mechanischen Grundsystems und sich daraus ergebende Kräfte/Momente werden der Übersichtlichkeit halber vernachlässigt).

Teilmodell Motor inkl. idealem Getriebe (= Aktor 1):

- Masse des Rotors des Motors m_{RotMot} ; Durchmesser des Rotors des Motors d_{RotMot} ; Daraus ergibt sich das Trägheitsmoment $J_A = (m_{RotMot}/2) * (d_{UR1}/2)^2$ des Antriebs.
- Drehzahl-Kennlinie n_{Mot} ;
- Übersetzungsverhältnis des idealen Getriebes i .

Teilmodell Pneumatik Zylinder (= Aktor 2):

- Druck des Zylinders p_{Zyl} ; Durchmesser des Zylinders d_{Zyl} .

Teilmodell Sensor 1:

- Position des Stückguts auf Förderband $PosX_{SG}$. Dies entspricht der Position, wenn das Stückgut am pneumatischen Zylinder ist.

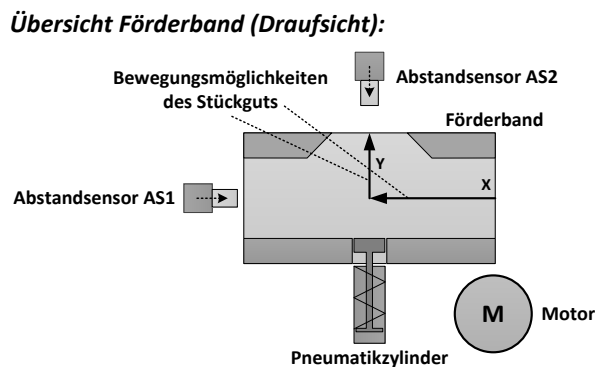
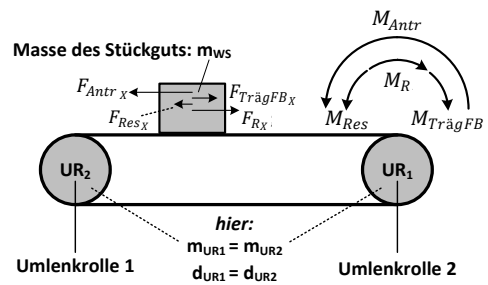
Teilmodell Sensor 2:

- Position des Stückguts nach verschieben durch pneumatischen Zylinder $PosY_{SG}$. Dies entspricht der Position, wenn das Stückgut mittig auf der Senke ist.

Teilmodell Förderband (= mechanisches Grundsystem):

- Massen der Umlenkrolle 1 und 2: m_{UR1} und m_{UR2} ; Durchmesser der Umlenkrolle 1 und 2: d_{UR1} und d_{UR2} ; Hier wird vereinfachend angenommen: $m_{UR1} = m_{UR2}$ sowie $d_{UR1} = d_{UR2}$. Daraus ergibt sich das Trägheitsmoment $J_{FB} = m_{UR1} * (d_{UR1}/2)^2$ des Förderbands.
- Masse des Stückguts m_{WS} ;
- Reibwert zwischen Förderband und Stückgut in x-Richtung: μ_X .
- Reibwert zwischen Förderband und Stückgut in y-Richtung: μ_Y ; Hier wird vereinfachend angenommen: $\mu_X = \mu_Y = \mu$.

Abbildung 7-5 zeigt eine Übersicht des Modells inkl. der wirkende Kräfte/Momente, die nachgebildet werden.

**Förderband (Seitenansicht) inkl. wirkenden Kräfte in x-Richtung und rel. Momenten:**

Massen der Umlenkrollen: m_{UR1}, m_{UR2}
Durchmesser der Umlenkrollen: d_{UR1}, d_{UR2}

Bezeichnungen:

F_{Resx}	Resultierende Kraft, die die Bewegung des Stückguts in x-Richtung erzeugt
$F_{TrägFBx}$	Gegen die Bewegung wirkende Kraft, die durch die Trägheit der Umlenkrollen 1 und 2 verursacht wird
F_{Antrx}	Durch Antrieb erzeugte Kraft, die in x-Richtung auf das Stückgut wirkt
F_{Rx}	Reibkraft, die entgegengesetzt zur Bewegung des Stückguts in x-Richtung wirkt
M_{Res}	Resultierendes Drehmoment
M_{Antr}	Durch Antrieb erzeugtes Drehmoment
M_R	Drehmoment, das durch die Reibkraft des Stückguts verursacht wird
$M_{TrägFB}$	Drehmoment, das durch die Trägheit der Umlenkrollen 1 und 2 verursacht wird
F_{Resy}	Resultierende Kraft, die die Bewegung des Stückguts in y-Richtung ermöglicht
F_{Zyl}	Durch Antrieb erzeugte Kraft, die in y-Richtung auf Stückgut wirkt
F_{Ry}	Reibkraft, die entgegengesetzt zur Bewegung des Stückguts in y-Richtung wirkt

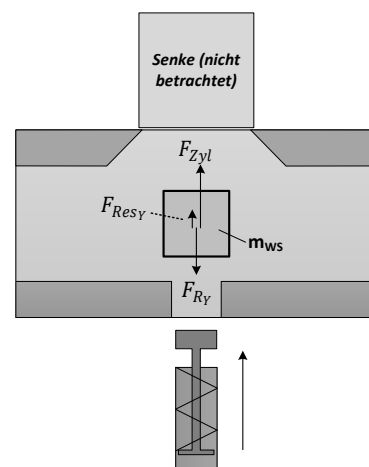
Förderband (Draufsicht) inkl. wirkenden Kräften in y-Richtung:

Abbildung 7-5: Modellbeschreibung und wirkende Kräfte/Momente

Abbildung 7-6 zeigt die einzelnen Teil-Simulationsmodelle, ihren inneren Aufbau (vereinfachte Darstellung) sowie die Schnittstellen zwischen den einzelnen Teil-Simulationsmodellen.

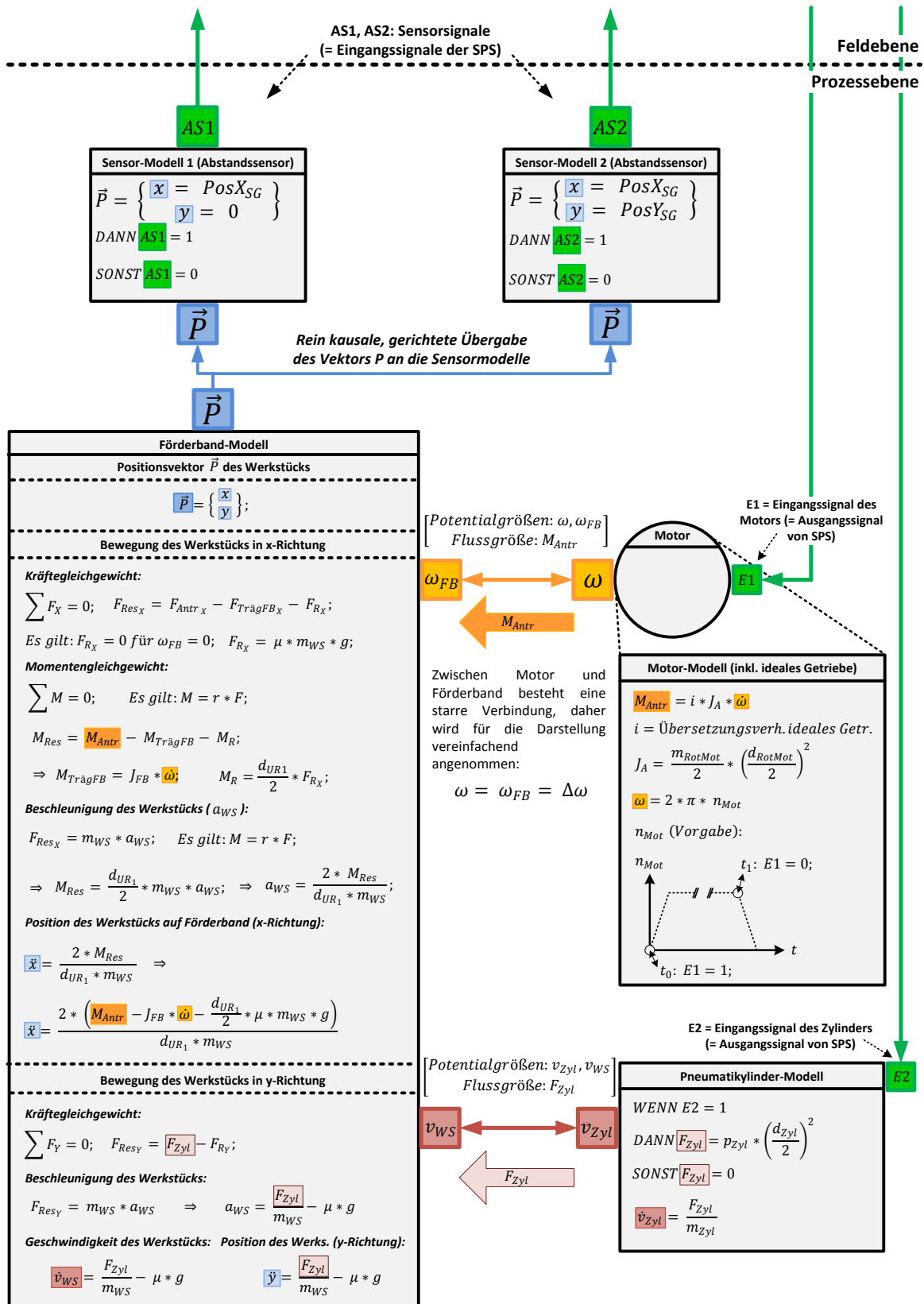


Abbildung 7-6: Physikbasierte Betrachtung von Bewegungen auf definierten Pfaden

7.2.4 Betrachtung von mehrdimensionalen Bewegungen im freien Raum

Muss für die Betrachtung eines Prozesses die freie Bewegung des Stückguts oder des Anlagenteils im Raum betrachtet werden, um die softwaretechnische Umsetzung der Automatisierungstechnischen Funktion hinreichend prüfen zu können, reicht die Betrachtung der Bewegung auf festgelegten Pfaden nicht mehr aus. Simulationsmodelle müssen in der hier beschriebenen Modellart vielmehr in der Lage sein, die Realität in einem Maße abbilden zu können, welches Anforderungen wie z. B. die Kollision mit Störkanten oder auch mögliche Verklemmungen an Engpassstellen nachbilden kann. Zusätzlich müssen Kollisionen zwischen Stückgütern und die Auswirkungen der Kollision abgebildet werden können. Simulationsmodelle des Grundsystems basieren in diesen Fällen zwingend auf geometrischen Konstruktionsdaten [RELA10].

Die Betrachtung der freien Bewegung von Stückgütern im Raum kann immer dann notwendig sein, wenn räumliche oder sphärische Vorgänge bei der Nachbildung einzelner Prozessschritte eine Rolle spielen. Diese Vorgänge können Teil der nachzubildenden Prozessschritte sein: Durch die gewollte Veränderung eines Stückguts hinsichtlich seiner Ausrichtung bzw. Lage im Raum. Dies kann z. B. beim Einsatz eines Portalroboters der Fall sein. Räumliche oder sphärische Vorgänge können sich aber auch aus der (fehlerhaften) Ausführung einzelner Prozessschritte ergeben. Simulationsmodelle in der hier beschriebenen Modellart müssen daher Kausalketten, die zur (ungewollten) Bewegung eines Stückguts führen können, sehr genau aufzeigen können. Kollidieren zwei Stückgüter miteinander, könnte dies eine Bewegung des Stückguts bedeuten – in Abhängigkeit der einwirkenden Kräfte könnte diese Bewegung ein (ungewolltes) Verdrehen oder auch ein Anheben des Stückguts sein. Die Kausalkette muss sich daher über alle Komponenten innerhalb eines Moduls erstrecken: Verklemmt ein Stückgut aufgrund zu hoher Fördergeschwindigkeit und daraus resultierender ungewollter Lageänderung an einer Engpassstelle, kann erhöhte Reibung die Folge sein. Die erhöhte Reibung könnte zu einem höheren Lastmoment führen, welches gegen das vom Antriebsmotor erzeugte Drehmoment wirkt. Mögliche Auswirkungen wären das Abschalten des Motors aufgrund einer Überlast oder auch eine Beschädigung an der mechanischen Apparatur oder auch die ungewollte Bewegung des Stückguts in einem (prinzipiell beliebigen) Freiheitsgrad.

Die Bewegung der Stückgüter unterliegt in allen Freiheitsgraden den Regeln der Physik („*Komplexe Dynamikmodelle*“). Im Vergleich zu der im Abschnitt 7.2.3 beschriebenen Modellart sind sechs Freiheitsgrade für die Bewegung eines Stückguts vorzusehen (drei translatorische, drei rotatorische). Die betrachteten Stückgüter können sich prinzipiell frei in diesen sechs Freiheitsgraden bewegen – abhängig von den auf das Stückgut wirkenden Kräften. Um eine physikalisch korrekte Bewegung simulieren zu können, muss das Verhalten der Gerätemodelle in gleichem Maße detailliert sein. (Einfache) Gerätemodelle, wie sie in den Abschnitten 7.2.2 und 7.2.3 definiert wurden, reichen hierfür nicht aus, da unrealistisches Verhalten der *komplexen Dynamikmodelle* die Folge sein kann. Unrealistisches Verhalten bedeutet, dass die sich über alle Komponenten hinweg ergebende Kausalkette nicht hinreichend abgebildet wird. Dies ist z. B. dann der Fall, wenn ein einfaches Motormodell ein konstantes Drehmoment, unabhängig vom vorhandenen Lastmoment am sehr detailliert modellierten Förderband, liefern würde. In diesem Fall würde bei einer Kollision oder auch Verklemmung eines Stückguts immer ein konstantes Drehmoment auf das Förderband wirken. Die Auswirkungen der durch eine Kollision oder Verklemmung hervorgerufenen Kräfte, die eine Bewegung des Stückguts bewirken können, können daher nicht nachvollzogen werden. Vielmehr muss das Verhalten des Motors bei erhöhtem Lastmoment abgebildet werden, so dass die Auswirkungen auf die sich ergebende Bewegung des Stückguts in allen Freiheitsgraden nachvoll-

zogen werden können. Gegebenenfalls relevante Gerätefunktionen wie z. B. das Verschleifen von Rampen bei Frequenzumrichtern zur sanfteren positiven wie negativen Beschleunigung sollten abgebildet werden können, und ihre Auswirkungen sollten im Simulationsergebnis sichtbar sein.

Die Modellierung derartiger Modelle ist, wie bereits die Modellierung von *Gerätemodellen* und *einfachen Dynamikmodellen* (→ Abschnitte 7.2.3), in einer objektorientierten, a-kausalen Modellierungssprache möglich, die sich für die Darstellung von physikalischem Verhalten eignet. Notwendig ist in jedem Fall die Verknüpfung mit einem Geometriemodell der Anlage, um zu betrachtende räumliche oder sphärische Vorgänge abbilden zu können. In Kombination damit kann auch die Nutzung einer Physik-Engine erwogen werden, wie sie z. B. im Projekt „AVANTI“ genutzt wird (→ Abschnitt 4.2.1.1).

7.3 Ableitung der für die virtuelle Inbetriebnahme wesentlichen Detaillierungsstufen

7.3.1 Definition von Detaillierungsstufen unter Einbeziehung der definierten Modellarten

Eine Betrachtungsweise von Detaillierungsstufen für die virtuelle Inbetriebnahme, die in einem leicht verständlichen Schema eingeordnet sind, ermöglicht ein Blick in die Verkehrssimulation. Dort sind drei Detaillierungsstufen etabliert: Die *makroskopische*, *mikroskopische* und *mesoskopische* Modellierung [HEL97]. Verkehrsflusssimulation untersucht das Verhalten auf Strecken, welche sich gut als Materialfluss einer Anlage approximieren lassen. *Makroskopische Modelle* behandeln in diesem Kontext das Zeitverhalten von betrachteten Systemen. Einzelne, diskrete Fahrzeuge werden nicht beachtet, stattdessen wird der Fahrzeugstrom durch Größen wie Fahrzeugdichte oder durchschnittliche Geschwindigkeit über geeignet gewählte zeitliche und/oder räumliche Intervalle näherungsweise abgebildet [HEL97]. Die Anforderungen an die Modellbildung sind hier als eher moderat zu bezeichnen.

Um die Betrachtung einzelner Fahrzeuge und deren Eigenschaften wie gegenseitiger Wechselwirkung oder individuellem Fahrverhalten zu ermöglichen, werden *mikroskopische Modelle* eingesetzt. Sie beschreiben das Flussverhalten im Detail und betrachten diskrete Fahrzeuge, deren (parametrisierbare) Eigenschaften sowie auch einzelne Fahrbahnsuren. Mikroskopische Modelle bilden simulierte Objekte und Ressourcen in hohem Detaillierungsgrad ab [HEL97]. Sie stellen die höchsten Anforderungen an die Modellbildung.

Als dritte Modellierungsart ist die *mesoskopische Modellierung* gebräuchlich. Der Begriff der *mesoskopischen Modellierung* kann in der Verkehrsflusssimulation zwar als etabliert [SAV08], jedoch nicht als standardisiert bezeichnet werden. Zumeist ist die Bildung einer Zwischenebene und das Vereinen von Teilen der makroskopischen und mikroskopischen Sichtweise gemeint. Damit sollen die Nachteile dieser Modellarten einerseits minimiert, andererseits die Stärken beider Modellarten hervorgehoben werden, ohne den Modellierungs- wie Rechenaufwand der entstehenden Modelle stark zu erhöhen.

Überträgt man diese Sichtweise und Terminologie auf die Automatisierungstechnik und dort im Speziellen auf die Definition von Modellarten in den Abschnitten 7.2.1 bis 7.2.4, ergeben sich viele Überschneidungen. Der *makroskopische Detaillierungsgrad* kann so näherungsweise durch

einfache Totzeitmodelle (Abschnitt 7.2.1) dargestellt werden. Unterschiede ergeben sich bei der Betrachtung einzelner Stückgüter, deren Durchlaufzeit bei Verkehrsflusssimulationen in dieser Detaillierungsstufe nicht angegeben werden kann, nach der getroffenen Definition für *einfache Totzeitmodelle* mit diesen aber schon. Stück- sowie Schüttgüter (respektive Fahrzeug analog zur Verkehrssimulation) an sich werden nicht betrachtet, das Verhalten rein zeitbasiert abgebildet.

Komplexe Geräte- und Dynamikmodelle (Abschnitt 7.2.4) können dagegen in die mikroskopische Detaillierungsstufe eingeordnet werden. Sie stellen die höchsten Anforderungen an die Modellbildung, bieten aber auch die größte Realitätstreue und ermöglichen ein detailliertes Testen der zu automatisierenden Funktion.

Die Zwischenebene der mesoskopischen Modelle kann für die Sichtweise der Fertigungstechnik in zwei Stufen aufgeteilt werden: mesoskopisch/nicht dynamisch (nd) und mesoskopisch/dynamisch (d). Diese stellen einerseits die *einfachen Geräte- und Kinematikmodelle* (Abschnitt 7.2.2, Tendenz zum makroskopischen Detaillierungsgrad) sowie die *Geräte- und einfachen Dynamikmodelle* (Abschnitt 7.2.2, Tendenz zum mikroskopischen Detaillierungsgrad) dar. Die mesoskopischen Detaillierungsstufen vereinen die Vorteile der einfachen Totzeitmodelle mit denen einer physikalisch/dynamischen Simulation, wenn auch nicht so detailliert wie in der mikroskopischen Detailstufe, und beschränken sich auf die vom Grundsystem vorgegebenen Freiheitsgrade.

7.3.2 Zusammenfassende Darstellung von Modellarten und Detaillierungsstufen

Tabelle 10 gibt einen Überblick über die vorgestellten Modellarten und Detaillierungsstufen hinsichtlich derjenigen Aspekte, die in den einzelnen Detaillierungsstufen berücksichtigt werden. Die hier dargestellten Aspekte sind aus den Aussagen vorheriger Abschnitte und Kapitel akkumuliert.

Aspekt:	Makroskopische Modellierung Totzeitmodelle	Mesoskopische Modellierung				Mikroskopische Modellierung	
		Gerät		Grundsystem		Gerät	Grundsystem
		(nd)	(d)	(nd)	(d)		
Berücksichtigung des gesamten Prozessabbildes	ja	ja	ja	-	-	ja	-
Rein zeitbasiertes Verhalten	ja	ja	nein	ja	nein	nein	nein
Detailliertes Geräteverhalten	nein	nein	ja	-	-	ja	-
Physikalisch/Dynamisches Verhalten	nein	nein	ja	nein**	ja	ja	ja
Fehlerzustände (Paarfehler, Laufzeit, ...)****	ja	ja	ja	ja	ja	ja	ja
Berücksichtigung des transportierten Guts*:	nein	-	-	ja	ja	-	ja
- Eigenschaften von Gütern (Masse, Maße, ...)	nein	-	-	nein**	ja	-	ja
- Bewegung von Gütern auf festgelegten Bahnen (= Freiheitsgraden)	nein	-	-	ja**	ja	-	nein
- Freie Bewegung von Gütern im Raum	nein	-	-	nein	nein	-	ja
- Kollision von Gütern möglich	nein	-	-	nein	ja***	-	ja

* Gut ist synonym zu Stückgut und Schüttgut ** Beschleunigungsfreie Bewegung *** Nur möglich im simulierten Freiheitsgrad

**** Fehlerfunktionen sind in Abhängigkeit des Detaillierungsgrads unterschiedlich abgebildet (vgl. Abschnitte 7.2.1 bis 7.2.4)

nd: nicht-dynamisch d: dynamisch -: nicht relevant

Tabelle 10: Vergleich der definierten Detaillierungsstufen

7.4 Fertigungstechnische Simulationsmodellbibliothek

Nach Definition einzelner Detaillierungsstufen und der beinhalteten Modellarten wird anhand eines konkreten Beispiels beschrieben, wie eine fertigungstechnische Simulationsmodellbibliothek unter Beachtung der bisher beschriebenen Gestaltungsprinzipien und Definitionen aufgebaut sein kann.

7.4.1 Ableitung spezifischer Anforderungen an Simulationswerkzeuge

Das Simulationswerkzeug respektive die genutzte Modellierungssprache muss spezifische Anforderungen erfüllen, um für eine virtuelle Inbetriebnahme eingesetzt werden zu können. Als grundlegende Anforderung kann zunächst identifiziert werden, dass das Werkzeug in der Lage sein muss, die Abbildung typischer Automatisierungskomponenten zu ermöglichen. Typische Automatisierungskomponenten sind Sensoren und Aktoren, aber auch z. B. mechanische Grundsysteme. Diese können gemäß [VDI 2206-1] mechanische, elektromechanische, hydraulische oder pneumatische Strukturen oder Kombinationen aus selbigen sein. Ebenso sollte das Simulationswerkzeug respektive die Modellierungssprache in der Lage sein, die zu simulierende Anlage in unterschiedlichen Detaillierungsstufen, wie sie beispielhaft in Abschnitt 7.2 beschrieben wurden, abbilden und auch simulieren zu können. Um die Verbindung zwischen diesen Teilmodellen zu ermöglichen, wie auch die Verbindung mit anderen Modellen (z. B. des Prozessabbildes der SPS) zu realisieren, muss das Simulationswerkzeug als auch die Modellierungssprache die Einführung von standardisierten Schnittstellen ermöglichen.

Das Simulationswerkzeug sollte zudem die Anforderungen an das *Datenmodell der Digitalen Fabrik* erfüllen. Ein wichtiges Merkmal ist in diesem Zusammenhang die durchgehende Objektorientiertheit, die ein Simulationswerkzeug aufweisen sollte. Von Vorteil ist nicht nur ein hoher Wiedererkennungswert der Simulationsobjekte für den Ingenieur, sondern im Verbund mit einem geeigneten, objektorientierten Datenaustauschformat auch „[...] die Beibehaltung der ursprünglichen Datenmodellierung im CAE-System sowie deren vollständige Übertragung in [einen] Modellgenerator“ [BAR11]. Die Objektorientiertheit der Modellierungssprache kann daher vor allem immer dann von Vorteil sein, wenn Anlagensimulationsmodelle mit dem Ziel der Aufwandsminimierung aus einer geeigneten Datenquelle heraus automatisch generiert werden sollen. Sie stellt aber auch eine wichtige Voraussetzung für die Modellerstellung im Allgemeinen dar (→ Abschnitt 6.1.1). Vorteilhaft bei der Verwendung von Komponentenmodellen bei der Simulation fertigungstechnischer Anlagen ist ferner die Verwendung eines Baukastens respektive einer Modellbibliothek, die abstrakte Objekte in Form allgemeingültiger Modelle als Klassen beinhaltet. Das Simulationswerkzeug sollte daher ein Bibliothekskonzept explizit unterstützen.

Das Simulationswerkzeug sollte zudem die Erstellung von hart-echtzeitfähigen Simulationsmodellen ermöglichen. Im Konkreten bedeutet dies, dass der Einsatz von Solvern ermöglicht werden muss, die es erlauben, die Berechnung der Simulationsergebnisse immer rechtzeitig und mit hinreichender Genauigkeit zu jedem Zeitschritt abzuschließen. Um dies sicherzustellen, können Anpassungen an den Simulationsmodellen notwendig sein. Ein Beispiel hierfür wird in Abschnitt 7.4.2.2 beschrieben.

7.4.2 Fertigungstechnische Simulationsmodellbibliothek in Modelica

In den folgenden Abschnitten wird ausschnittsweise die während des Projekts „HIL-IMA“ (ZIM Förderkennzeichen KF2625802KM2) in der Modellierungssprache [MODELICA[®]] als Prototyp entstandene Simulationsmodellbibliothek vorgestellt. Die Betrachtungen konzentrieren sich auf Stückgut-Förderprozesse. Die Simulationsmodellbibliothek wurde in den Detaillierungsstufen makroskopisch (Abschnitt 7.2.1) und mesoskopisch/dynamisch (Abschnitt 7.2.3) umgesetzt, um repräsentativ die Machbarkeit unterschiedlicher Detaillierungsstufen zu zeigen. Informationen über die Modellierung von Schüttgut-Förderprozessen sind in [KPR⁺14[#]] einsehbar.

7.4.2.1 Grundlegendes Prinzip bei der Umsetzung einer Fördertechnikapplikation

Während des Projekts „HIL-IMA“ entstand eine Simulationsmodellbibliothek, die typische Komponenten einer automatisierten Anlage wie Sensoren, Aktoren und mechanische Grundsysteme wie Förderbänder und Drehtische enthält. Des Weiteren wurden standardisierte Schnittstellen definiert, mit denen sich diese Modelle verbinden lassen. Grundsätzlich ist die Modellierung von Stückgut nicht mit typischen Anwendungen in Modelica vergleichbar, sondern weist einige Besonderheiten auf, wie folgende Einschränkungen zeigen. So können in Modelica prinzipiell keine Objekte ...

- ... zur Laufzeit erzeugt oder gelöscht werden
- ... zwischen einzelnen Teilmodellen ausgetauscht werden

Teilmodelle können im gegebenen Kontext typische Förderelemente wie ein Bandförderer oder auch ein Kettenförderer sein. Eine Übergabe von Stückgütern zwischen Teilmodellen (d. h. Austausch von Stückgut-Objekten) ist in Modelica ausgeschlossen, was einer Fördertechnikapplikation natürlich diametral entgegensteht. Um die Realisierung solcher Modelle dennoch zu ermöglichen, wurden diese Punkte daher bei der Modellierung explizit berücksichtigt, wie nachfolgend beschrieben.

Eine mögliche Lösung stellt die Trennung der Berechnung von Stückgütern und des Verhaltens des Förderbands dar [KPR⁺14[#]]. Das bedeutet, dass die Bewegung von Stückgütern in einem vom jeweiligen Förderband getrennten Modell, Stückgutarray genannt, betrachtet wird. Die relevanten Größen, die für die Beschreibung und Realisierung dieser Bewegung notwendig sind, werden über eine Schnittstelle zwischen diesen Modellen ausgetauscht. Die Zuordnung eines Stückguts zu einem Förderband findet über einen *ID* genannten Parameter statt. Jedes Förderband innerhalb der Simulation muss hierfür eine eindeutige *ID* besitzen, die z. B. aus dem Betriebsmittelkennzeichen abgeleitet werden kann. Stückgüter können im Verhältnis *n* zu 1 entsprechend dieser *ID* einem Förderband zugeordnet werden; auf einem Förderband können sich daher mehrere Stückgüter befinden. Bei Übergang von einem Förderband auf ein anderes ändert sich die im Stückgutarray dem jeweiligen Stückgut zugewiesene *ID* und somit das Förderband, auf dem sich das Stückgut befindet.

Das Förderbandmodell beschreibt in dieser Konstellation die Randbedingungen einer Bewegung [KPR⁺14[#]]. Wechselt das Stückgut von einem Förderband auf ein anderes, können sich diese Randbedingungen aufgrund der Eigenschaften des folgenden Förderbandmodells ändern. Das Verhalten der Stückgüter wird äquivalent zum gewählten Detaillierungsgrad des mittels *ID* zugewiesenen Förderbands berechnet [KPR⁺14[#]]. Beispiele für Randbedingungen sind bei einem mesoskopisch/dynamischen Detaillierungsgrad die maximal mögliche Fahrgeschwindigkeit oder rele-

vante Reibungskoeffizienten, da durch die Reibung zwischen Stückgut und Förderband und die somit dort stattfindende Kraftübertragung letztendlich die Bewegung realisiert wird. Die eigentliche Bewegung der Stückgüter wird im Stückgutarray vorgenommen. Abgeleitet aus den vom Förderbandmodell vorgegebenen Randbedingungen wird dort berechnet, wie sich das Stückgut auf dem jeweiligen Förderband bewegt und wo genau es sich aufhält. Im makroskopischen Detaillierungsgrad werden dagegen keine Stückgüter betrachtet. Ein Stückgut wird im makroskopischen Detaillierungsgrad daher nur zugewiesen.

Die Verbindung zwischen einzelnen Förderbandmodellen und dem Stückgutarray erfolgt über eine in der Modelica-Simulationsbibliothek hinterlegte Schnittstelle „ConveyorBus“. Diese Schnittstelle wurde für die hier beschriebene Anwendung entworfen und bündelt alle Signale, die zwischen Förderbandmodellen und Stückgutarray ausgetauscht werden. Sie ist als äquivalent zur in Abschnitt 6.1.5 beschriebenen Schnittstelle „PPR_Interface“ im Datenmodell zu sehen. Die Zuordnung spezifischer Schnittstellensignale wird durch die ID ermöglicht. Abbildung 7-7 zeigt das Prinzip der Aufteilung der Modelle *Förderband* ↔ *Stückgüter* in makroskopischem Detaillierungsgrad. Im hier gezeigten Beispiel befindet sich Stückgut 1 auf einem Förderband mit der ID 1. Im Stückgutarray erfolgt keine Berechnung der Position des Stückguts, nur eine Zuweisung, dass sich Stückgut 1 auf dem Förderband mit ID 1 befindet. Über die Schnittstelle „ConveyorBus“ wird daher nur der Parameter *ID* ausgetauscht. Die Zuweisung des Parameters *ID* ändert sich immer dann, wenn ein Stückgut das Ende eines Förderbands erreicht hat. In diesem Fall teilt das Förderbandmodell dem Stückgut mit, dass das Stückgut auf ein anderes Förderband übergeben wurde. Die „neue“ ID ergibt sich aus dem in Materialflussrichtung nachfolgenden Förderbandmodell (im Anlagenstrukturmodell als Schnittstelle abgebildet, vgl. Abschnitt 6.1.3.2 und 6.1.3.4). Die Zuweisung von IDs zu einzelnen Stückgütern erfolgt daher immer durch das Förderband.

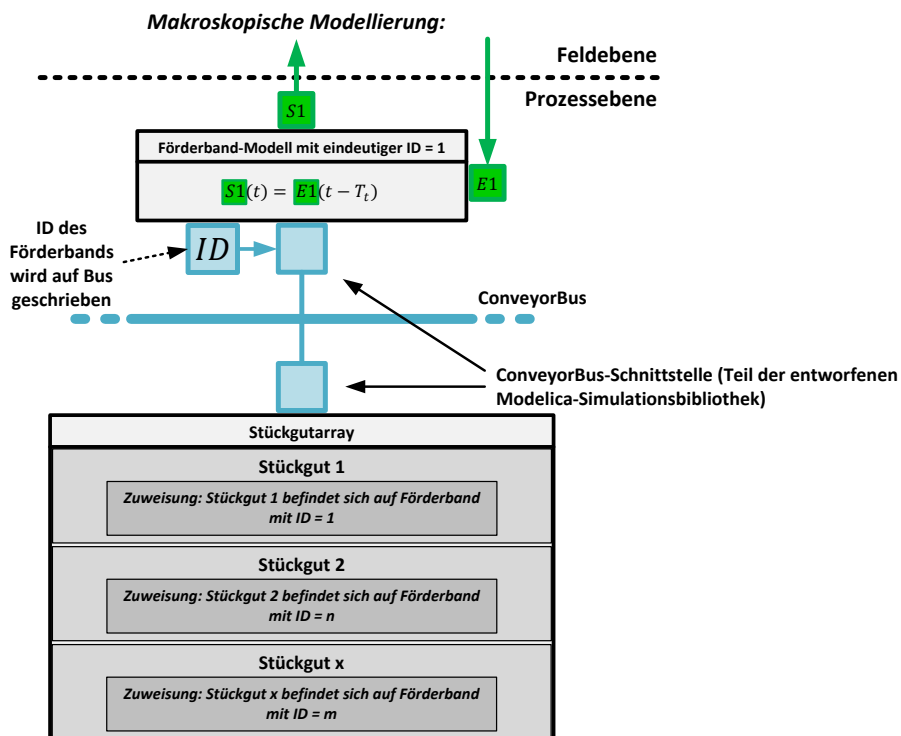


Abbildung 7-7: Funktionsweise der Trennung Stückgut/Förderband in makroskopischem Detaillierungsgrad

Abbildung 7-8 zeigt das Prinzip der Aufteilung der verschiedenen Modelle *Förderband* ↔ *Stückgüter* in mesoskopisch/dynamischem Detaillierungsgrad. Das Förderband entspricht dem in Abschnitt 7.2.3 beschriebenen (Anm.: Zur Vereinfachung wird die Bewegung in y-Richtung hier nicht betrachtet). Im hier gezeigten Beispiel befindet sich Stückgut 1 auf einem Förderband mit der ID 1. Im Stückgutarray erfolgt die Berechnung der Position des Stückguts anhand der über die Schnittstelle „ConveyorBus“ ausgetauschten Größen.

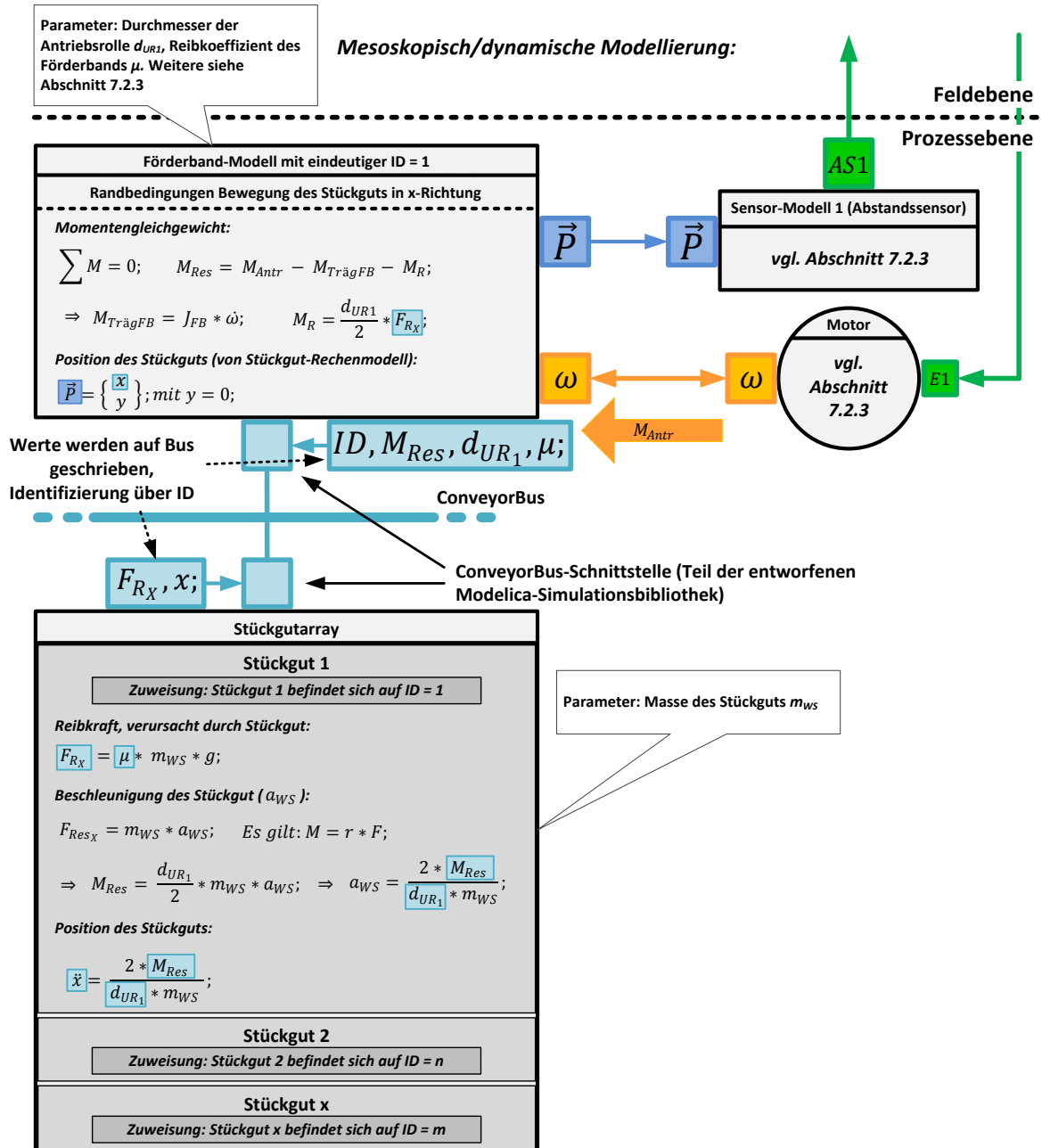


Abbildung 7-8: Funktionsweise der Trennung Stückgut/Förderband in mesoskopisch/dynamischem Detaillierungsgrad

Die entstandene Modelica-Simulationsmodellbibliothek ist mit der hier beschriebener Aufteilung der Betrachtung von Förderbandmodellen und Stückgütern in der Lage, auch mit Fällen umzugehen, bei denen Objekte zur Laufzeit der Simulation nur einmalig vorkommen und von Förderband-Teilmodell zu Förderband-Teilmodell übergeben werden. „Einmalig vorkommen“ bedeutet in diesem Zusammenhang, dass sich ein einzelnes Stückgut nur einmal durch die Anlage bewegt.

Die Anzahl der zu simulierenden Stückgüter muss vor Beginn der Simulation fest definiert werden. Die simulierten Stückgüter werden am Beginn der Simulation erzeugt und sind einem speziellen Modell, das eine Quelle darstellt, zugewiesen. Nach Durchlauf des Stückguts durch die Anlage werden die Stückgüter, äquivalent zur Quelle, einer Senke zugewiesen. In Anhang 1 findet sich ein Beispiel, wie diese Modelle in Modelica konkret umgesetzt sind.

7.4.2.2 Notwendige Anpassungen von Simulationsmodellen aufgrund der geforderten Echtzeitfähigkeit

Zwingende Voraussetzung für den Betrieb von Steuer- und Regelungsalgorithmen auf echtzeitfähiger Hardware sowie das Betreiben von Hardware-in-the-Loop-Prüfständen (wie in Abschnitt 5.3.3 beschrieben) ist, dass die Berechnungen der Simulationsergebnisse immer rechtzeitig und mit hinreichender Genauigkeit zu jedem Zeitschritt abgeschlossen sein müssen. Hierfür stehen verschiedene Möglichkeiten zur Verfügung. So könnte man ...

- ... eine gröbere Detaillierung wählen. Dies ist jedoch nicht immer möglich, da mit abnehmender Detaillierung auch die Aussagefähigkeit der Simulationsmodelle abnimmt und somit die Testdurchführung beeinträchtigt werden könnte.
- ... einen kleineren Anlagenausschnitt wählen. Hierfür könnten jedoch Änderungen am Steuerungscode notwendig sein um nicht genutzte Anlagenteile „abzuschalten“, was der Intention einer virtuellen Inbetriebnahme entgegensteht.
- ... einen schnelleren Rechner wählen. Ein schnellerer Rechner alleine ist jedoch keine Garantie zur Sicherstellung der Echtzeitfähigkeit einer Simulation [LB14].

Eine weitere Möglichkeit besteht darin, die Rechenzeit der genutzten Simulationsmodelle ist so weit zu reduzieren, dass die Bedingung jederzeit erfüllt werden kann. Echtzeitfähigkeit ist in diesem Fall eine Eigenschaft der Simulationsplattform sowie der zum Einsatz kommenden Simulationsmodelle.

Die Sicherstellung einer harten Echtzeitfähigkeit stellt zusätzliche Anforderungen an die Modelica-Simulationsmodelle. Tegethoff et al. [TSG⁺11] beschreiben, welche Eingriffe in Simulationsmodellen notwendig sein können, um eine Echtzeitfähigkeit der Simulation zu gewährleisten. Insbesondere Modelle mit mechanischen Anschlägen, wie sie in automatisierten Anlagen häufig vorkommen, müssen für die Simulation mit einem echtzeit-geeigneten Solver angepasst werden. Echtzeitfähige Solver basieren in der Regel auf einer festen Integrationsschrittweite und auf Verwendung von einfachen, expliziten Integrationsverfahren wie z. B. dem Vorwärts-Euler. Die feste Schrittweite limitiert die zeitliche Auflösungsgenauigkeit von diskreten Ereignissen (z. B. mechanischen Anschlägen). Die Verwendung expliziter Verfahren erfordert die Sicherstellung der numerischen Stabilität der Simulation. Am Beispiel der Simulation von Stückgut auf einem Fördererlelement soll verdeutlicht werden, welche Maßnahmen beispielhaft zu treffen sind.

Während die Zuordnung vom jeweiligen Stückgut zum entsprechenden Förderer anhand der *ID* des Förderers erfolgt, wird die physikalische Kopplung in mesoskopisch/dynamischem Detaillierungsgrad über die Reibkraft F_r umgesetzt [KPR⁺14[#]]. Unter Berücksichtigung von μ , dem Reibkoeffizient des Förderers, m – als Masse des Stückguts – und g – der Gravitationskonstanten – beschleunigt die Reibkraft F_r das Stückgut solange, bis die Differenzgeschwindigkeit zwischen Förderband und Stückgut gleich Null ist ($v_{\text{Förderer}} - v_{\text{Stückgut}} = 0$, Gleichung 1, [HoQu14]).

$$(1) \quad F_r = \mu \cdot m \cdot g \cdot \text{sign}(v_{\text{Förderband}} - v_{\text{Stückgut}})$$

Diese unstetige Vorzeichen-Funktion erfordert es, dass das eingesetzte numerische Lösungsverfahren den Gleichgewichtspunkt ($v_{\text{Förderband}} = v_{\text{Stückgut}}$) genau trifft. Wird in einem Simulationsschritt der Gleichgewichtspunkt nicht genau getroffen, kehrt die Vorzeichenfunktion die Wirkrichtung der Reibkraft F_r um. Das Stückgut erfährt eine Gegenkraft, die im darauffolgenden Simulationsschritt wieder zu einem Vorzeichenwechsel der Reibkraft F_r führt. Diese muss vom Solver im nächsten Simulationsschritt mit einer entsprechenden Gegenkraft erst wieder kompensiert werden, usw. Infolge dessen tritt sogenanntes *Chattering* [FYS13] auf, was einem Oszillieren um den Gleichgewichtspunkt entspricht. Dies kann zur numerischen Instabilität der Simulation führen [KPR⁺14[#]].

Lösungsverfahren mit variabler Schrittweite können derartige Funktionen zwar mit Hilfe einer Null-Durchgangserkennung lösen, benötigen aber viel Rechenaufwand und sind damit nur begrenzt echtzeitfähig. Lösungsverfahren, die eine harte Echtzeitfähigkeit ermöglichen, weisen aufgrund der begrenzten Rechenzeit eine feste Simulationsschrittweite auf. Sie sind nicht in der Lage, ohne zusätzliche Modifikationen derartige Funktionen zu lösen. Zur Vermeidung des *Chattering-Effekts* beim Einsatz eines Lösungsverfahrens mit fester Simulationsschrittweite besteht eine mögliche Lösung dieses Problems darin, unstetige Funktionen durch stetige Funktionen anzunähern [KPR⁺14[#]]. Gleichung 2 zeigt die in diesem Kontext mit einem Reduktionsfaktor erweiterte Gleichung 1, in der die unstetige Vorzeichenfunktion um die Gleichgewichtslage durch einen stetigen, exponentiellen Verlauf mit der Zeitkonstante T_r angenähert wird.

$$(2) \quad F_r = \mu \cdot m \cdot g \cdot \text{sign}(v_{\text{Förderband}} - v_{\text{Stückgut}}) \cdot \left(1 - e^{-\frac{|v_{\text{Stückgut}} - v_{\text{Förderer}}|}{T_r}} \right)$$

Der resultierende zeitliche Verlauf der Reibkraft F_r aus Gleichung 2 um den Gleichgewichtspunkt ist in Abbildung 7-9 dargestellt. Daraus ist ersichtlich, dass die Reibkraft F_r bereits nach wenigen Simulationsschritten (0 bis 4) gegen den Gleichgewichtspunkt konvergiert. Die Erweiterung gemäß Gleichung 2 ist für den Einsatz mit Lösungsverfahren mit fester Schrittweite geeignet und steht stellvertretend für typische Anpassungen, die für die Sicherstellung einer harten Echtzeitfähigkeit von Simulationsmodellen vorgenommen werden müssen.

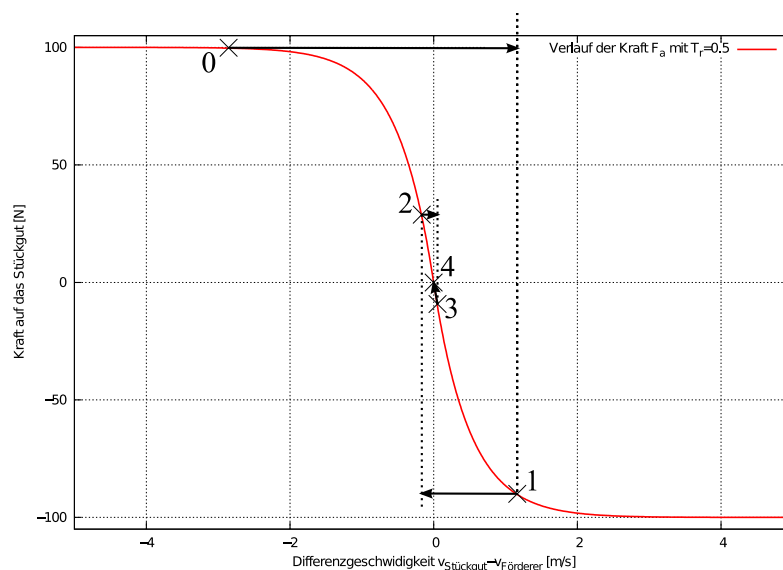


Abbildung 7-9: Verlauf der Konvergenz aufgrund der reduzierten Reibkraft gemäß [KPR⁺14[#]].

8 Applikation der virtuellen Inbetriebnahme

Stehen geeignete Simulationsmodelle in unterschiedlichen Detaillierungsstufen zur Verfügung, stellt sich direkt die Frage, wann und wie diese in einer *virtuellen Inbetriebnahme* eingesetzt werden können. Hierfür gilt, wie bereits im Lösungskonzept in Abschnitt 5.2 formuliert: ein Anlagensimulationsmodell sollte so einfach wie möglich und so komplex wie nötig sein. Daraus lässt sich folgern, dass der größte Teil eines Anlagensimulationsmodells stets im niedrigsten Detaillierungsgrad simuliert werden sollte. Nur einzelne Teile einer Anlage sollten, sofern dies erforderlich ist, mit höherer Detaillierung abgebildet werden. Wie in *Lösungsbaustein 3.1* beschrieben, wird hierfür in den Abschnitten 8.1 und 8.2 zunächst eine Methode vorgestellt, wie sich der notwendige Detaillierungsgrad für einzelne Teile einer Anlage identifizieren lässt. Zudem wird in Abschnitt 8.2.5 untersucht, ob eine dynamische Anpassung des Detaillierungsgrads zur Laufzeit der Simulation möglich ist. Abschnitt 8.3 beschäftigt sich mit der Sicherstellung der Konsistenz heterogen aufgebauter Anlagensimulationsmodelle in unterschiedlichen Detaillierungsstufen. Die Abschnitte 8.1, 8.2 und 8.3 stehen unter Leitfrage III: *Wie sollte man virtuelle Inbetriebnahme umsetzen?*

Abschnitt 8.4 widmet sich der Erstellung eines Anlagensimulationsmodells und damit der eigentlichen Modellbildung. Abschnitt 8.4 steht unter Leitfrage IV: *Wie kann man effizient Simulationsmodelle erstellen?*

8.1 Grundlagen zur Identifizierung des notwendigen Detaillierungsgrads

8.1.1 Gemeinsame Einflussfaktoren auf Systementwurf und Simulation

Das Anlagensimulationsmodell sollte die Realität dahin gehend abbilden, dass die Prozessausführung unter den beim Bauherren vorherrschenden Bedingungen getestet werden kann – so, wie später im realen Betrieb auch. Diese Bedingungen werden exemplarisch durch Verhaltenskenngrößen [JAN10] und die im Engineering-Prozess (unter Nutzung dieser Verhaltenskenngrößen) entstandenen Anlagenplanungsdaten wie Layouts, CAD-Zeichnungen, Tabellen, Skizzen, Steuerungscode repräsentiert (→ Abschnitt 2.4 und 2.5). Es wurde bereits darauf hingewiesen, dass Verhaltenskenngrößen aufgrund ihrer Charakteristik über alle mechatronischen Aspekte hinweg Anlagenteile aufzeigen können, die besonderer Betrachtung bedürfen (→ Abschnitt 2.4).

Die während des Engineerings erstellten Anlagenplanungsdaten enthalten das spezifische Wissen, wie ein Systementwurf umgesetzt wurde, um einen fertigungstechnischen Prozess anforderungsgerecht auszuführen (→ Abschnitt 2.4.2). Wendet man die Prinzipien, die zur Erstellung von Anlagenplanungsdaten geführt haben, konsequent auf die Modellerstellung bei einer *virtuellen Inbetriebnahme* an, können dieselben *Verhaltenskenngrößen*, die zur Auslegung einzelner Anlagenteile geführt haben, auch bei der Modellerstellung von Relevanz sein. *Verhaltenskenngrößen* und die durch sie sinngemäß repräsentierten Beziehungen zwischen fertigungstechnischen Prozessen und technischen Ressourcen (→ Abschnitt 2.4) können im Speziellen der Bestimmung des notwendigen Detaillierungsgrads einzelner Teil-Simulationsmodelle dienen. In diesem Kontext sollte gelten:

Dieselben Prinzipien, die zur Erstellung spezifischer Anlagenplanungsdaten geführt haben, sollten ebenso für die Dimensionierung eines Simulationsmodells (= Wahl des notwendigen Detaillierungsgrads) gelten.

Verhaltenskenngrößen, die gleichermaßen Auswirkungen auf den Systementwurf und eine Simulation haben, können in der Praxis nur fallspezifisch und somit nur für die jeweilig betrachtete Anlage angegeben werden. Eine generische, vollständige Übersicht von möglichen Verhaltenskenngrößen ist aufgrund des heterogenen Charakters von automatisierten Anlagen nicht sinnvoll und wäre nur Stückwerk angesichts der Menge an denkbaren Verhaltenskenngrößen, die einen Einfluss auf die Auswahl und Dimensionierung von technischen Ressourcen haben könnten. Einen Eindruck, welche Verhaltenskenngrößen in Betracht gezogen werden können, vermitteln folgende Beispiele (teilweise aus [JAN10] übernommen):

- **Kleine Sollwertabweichungen:** Ist die erlaubte Abweichung von einem Sollwert sehr klein (Engpass aus Logistiksicht, sehr genaues Positionieren), kann dies ein Hinweis auf eine genauer durchzuführende Simulation sein.
- **Superpositionsprinzip:** Überlagern sich mehrere Kräfte (in einem Freiheitsgrad oder auch in mehreren), kann dies ein Hinweis auf eine genauer durchzuführende Simulation sein.
- **Physikalische Effekte:** Werden Sensorwerte mittels physikalischer Effekte ausgelöst (z. B. Fliehkraftschalter), kann dies ein Hinweis auf eine genauer durchzuführende Simulation sein.
- **Risikoprioritätszahl RPZ:** Eine hohe RPZ kann ein Hinweis auf eine genauer durchzuführende Simulation sein. Hierfür sollte die Aufschlüsselung der RPZ aus Bedeutung, Auftretenswahrscheinlichkeit und Entdeckungswahrscheinlichkeit einzeln beachtet werden.
- **Toleranzen:** Sind sehr genaue Toleranzen angegeben, kann der Aufwand für die Konstruktion entsprechend hoch gewesen sein (z. B. aus CAD-Zeichnung erkennbar). Dies könnte ein Hinweis darauf sein, dass hier viel Aufwand für die Konstruktion betrieben werden musste und möglicherweise eine genauere Simulation sinnvoll ist.
- **Best Practices:** Werden keine Best-Practices (hinsichtlich der konstruktiven Umsetzung mechanischer Grundsysteme, der Verwendung von automatisierungstechnischen Geräten aus Freigabelisten oder hinsichtlich der Verwendung von Standard-SPS-Bausteinen) genutzt, kann dies ein Hinweis auf eine genauer durchzuführende Simulation sein.

Die in den Anlagenplanungsdaten enthaltenen Verhaltenskenngrößen können als Wissensbasis für die Ableitung des für eine Simulation notwendigen Detaillierungsgrads genutzt werden. Diese liegen in der Praxis jedoch nicht formalisiert vor und müssen daher manuell aus den Anlagenplanungsdaten extrahiert werden. In Ergänzung dazu sollten das spezifische Fachwissen und die Erfahrung, die das Gewerk Automatisierungstechnik bei der Automatisierung fertigungstechnischer Prozesse hat, ebenfalls genutzt werden. Der Entwurf einer Methode, die es erlaubt, sich dieses Wissen zur Identifizierung des notwendigen Detaillierungsgrads spezifischer Anlagenteile zu Nutzen zu machen, erscheint sinnvoll.

8.1.2 Beziehung zwischen Prozessen und technischen Ressourcen

In Abschnitt 6.1.3.2 wurde dargelegt, dass die Durchführung einzelner Prozessschritte während einer virtuellen Inbetriebnahme getestet werden muss. Prozessschritte werden in der Fertigungstechnik durch Komponenten ausgeführt, die von einer SPS gesteuert werden. Daraus ergibt sich

folgende Grundidee für die Bestimmung des für eine Simulation notwendigen Detaillierungsgrads:

Abhängig davon, welcher Prozessschritt durchgeführt wird, ist die Komponente, mit der dieser Prozessschritt ausgeführt wird, unterschiedlich detailliert zu simulieren.

Dies bedeutet, den durch Verhaltenskenngößen sinngemäß dargestellten Zusammenhang zwischen einem Prozessschritt und zugehöriger Komponente (→ Abschnitt 2.4) so zu interpretieren, dass eine Bestimmung des für eine Simulation notwendigen Detaillierungsgrads erfolgen kann. Aus Abschnitt 8.1.1 lässt sich folgern, dass sich eine systematisierte Ableitung der für eine Simulation wesentlichen Verhaltenskenngößen auf die Analyse der während des Engineering-Prozesses erstellten Anlagenplanungsdaten konzentrieren sollte. Abbildung 8-1 zeigt im Konkreten, welche Anlagenplanungsdaten analysiert werden sollten. Die Darstellung erfolgt unter Zuhilfenahme der formalisierten Prozessbeschreibung gemäß [VDI 3682-1] (→ Abschnitt 2.4).

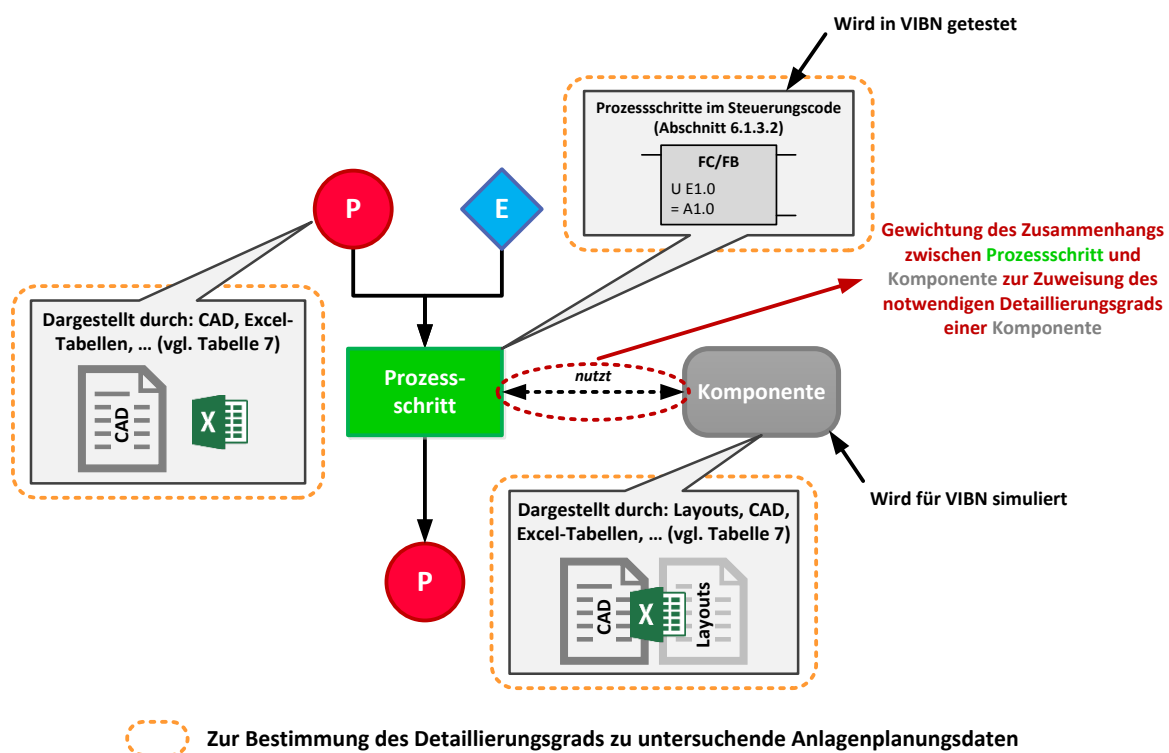


Abbildung 8-1: Gewichtung des Zusammenhangs von Prozessschritten und Komponenten zur Zuweisung des für eine Simulation notwendigen Detaillierungsgrads

Die in den nachfolgenden Abschnitten vorgestellte Methode erzeugt zusätzlichen Aufwand und ist als vorbereitende Maßnahme noch vor der eigentlichen Modellerstellung durchzuführen (→ Abschnitt 8.4). Von Vorteil wäre daher, die Bestimmung des notwendigen Detaillierungsgrads zu automatisieren, z. B. mittels einer Inferenzmaschine [GRS10]. Ein derartiges Expertensystem ist aufgrund der in Abschnitt 2.4.1 identifizierten Gegebenheiten bei KMU in der Praxis jedoch nicht oder nur sehr schwer umsetzbar. Zudem wäre es für eine vollständig automatisierte Lösung notwendig, die in den Anlagenplanungsdaten hinterlegten Verhaltenskenngößen zu erkennen und anschließend automatisiert abzuleiten. Die Verhaltenskenngößen müssten anschließend in das Expertensystem importiert werden. Hierfür ist eine durchgehende, herstellerübergreifende Semantik vorteilhaft. Dies wird in dieser Arbeit nicht betrachtet (→ Abschnitt 6.1.2). Stattdessen sollen hier mögliche Synergieeffekte mit ohnehin stattfindenden Tätigkeiten des Gewerks Automatisie-

rungstechnik ausgenutzt werden. Aus den in Abschnitt 2.4 und 6.1.3.2 getroffenen Aussagen lässt sich folgern, dass das Gewerk Automatisierungstechnik ohnehin die relevanten Prozesse innerhalb einer Anlage betrachten muss, da diese Prozesse eine maßgebliche Grundlage zur Erstellung des Steuerungscode bilden. Das Gewerk Automatisierungstechnik beschäftigt sich daher zwangsläufig mit der Frage, wie Prozessschritte innerhalb des Steuerungscode unter Beachtung der projektierten Komponenten umgesetzt werden können. Die nachfolgend vorgestellte Methode kann dazu genutzt werden, diesen ohnehin stattfindenden Prozess zu systematisieren.

8.2 Systematische Ableitung eines angemessenen Detaillierungsgrads aus Anlagenplanungsdaten

8.2.1 Quantifizierung des Zusammenhangs zwischen Prozessschritt und Komponente

Grundlegend soll die Gewichtung des Zusammenhangs zwischen Prozessschritt und Komponente mittels eines numerischen Werts erfolgen, der den notwendigen Detaillierungsgrad einer Komponente quantifiziert angibt. Hierzu ist es notwendig, ein Maß zu definieren, welcher den notwendigen Detaillierungsgrad eindeutig beschreibt. Konkret wird der Wertebereich durch die Anzahl an verfügbaren Detaillierungsstufen festgelegt. Der Wertebereich reicht bei vier zur Verfügung stehenden Detaillierungsstufen daher von 1 – 4. In dieser Arbeit repräsentiert ein zugewiesener Wert „1“ den makroskopischen Detaillierungsgrad, Wert „2“ den mesoskopisch/nicht dynamischen Detaillierungsgrad, Wert „3“ den mesoskopisch/dynamischen Detaillierungsgrad und Wert „4“ den mikroskopischen Detaillierungsgrad (→ Abschnitt 7.3).

Bei der Zuweisung von Gewichtungen zwischen Prozessschritt und Komponente können sich für Komponenten innerhalb eines Moduls verschiedene Detaillierungsstufen ergeben. Zudem können einer einzelnen Komponente durchaus mehrere Prozessschritte zugeordnet werden, was zur Zuweisung unterschiedlicher Detaillierungsstufen für dieselbe Komponente führen kann. In diesem Kontext ist die in Abschnitt 7.2 aufgestellte Einschränkung zu beachten. Der Komponenten zugewiesene Detaillierungsgrad ergibt sich daher immer aus dem höchsten für eine einzelne Komponente innerhalb eines Moduls identifizierten Detaillierungsgrad ($\max()$). In diesem Fall ist sichergestellt, dass die Simulation immer genügend aussagefähig ist. Unter Beachtung der Aussagen in Abschnitt 7.2 ergibt sich daher folgende Festlegung:

Der Detaillierungsgrad aller Komponenten innerhalb eines Moduls ergibt sich aus dem höchsten Detaillierungsgrad, der einer einzelnen Komponente des Moduls zugewiesen wurde.

Anzumerken ist, dass sich der jeweils notwendige Detaillierungsgrad aus den Anlagenplanungsdaten identifizieren lassen muss – unabhängig davon, welche Simulationsmodelle in welchen Detaillierungsstufen im Konkreten zur Verfügung stehen. Stehen keine geeigneten Simulationsmodelle zur Verfügung, um im notwendigen Detaillierungsgrad simulieren zu können, kann dies zu Einschränkungen hinsichtlich der Durchführung einer virtuellen Inbetriebnahme führen.

8.2.2 Regelbasierte Gewichtung des Zusammenhangs von Prozessschritten und Komponenten

Die eigentliche Gewichtung des Zusammenhangs zwischen Prozessschritt und Komponente kann mit Hilfe eines regelbasierten Ansatzes erfolgen. Die genutzten Regeln können aufgrund des hete-

rogenen Charakters automatisierter Anlagen und der Menge an denkbaren Verhaltenskenngrößen nur auf eine spezifische Anlage bezogen werden, und müssen von Fall zu Fall neu erstellt werden. Sie stellen ein Hilfsmittel für das Gewerk Automatisierungstechnik dar, um Anlagenplanungsdaten interpretieren zu können (→ Abschnitt 2.4.2). Das aus den Anlagenplanungsdaten abgeleitete fallspezifische Wissen, warum eine Komponente ausgesucht oder für den jeweiligen Anwendungszweck durch die Gewerke Mechanik und Elektrotechnik konstruiert/projektiert wurde, sowie das Erfahrungswissen des Gewerks Automatisierungstechnik, wie ein Prozess zu automatisieren ist, sollen berücksichtigt werden. Die Regeln können daher auch als *Heuristiken* gesehen werden - sie sind ein *deduktiver Schluss* aus den Verhaltenskenngrößen, die während des Engineerings aus den Kundenanforderungen verfeinert/destilliert wurden und zur Erstellung konkreter Anlagenplanungsdaten geführt haben. Wie in allen Ansätzen, die Heuristiken enthalten, kann nicht gewährleistet werden, dass die gefundene Lösung auch die bestmögliche Lösung darstellt. Ziel ist es, eine für den individuellen Anwendungsfall *virtuelle Inbetriebnahme* praktikable Lösung (= notwendiger Detaillierungsgrad) zu ermitteln.

Grundsätzlich existieren zwei Möglichkeiten für die Verarbeitung von Regeln [BEKE06]: Einerseits die *Vorwärtsverkettung (Datengetriebene Inferenz)*, bei der das bekannte, fallspezifische Wissen (hier: hinterlegt in Anlagenplanungsdaten) als Ausgangspunkt für den Schlussfolgerungsprozess in Betracht gezogen wird. Andererseits die *Rückwärtsverkettung (Zielorientierte Inferenz)*, bei der das Zielobjekt (hier: die Simulation) dahin gehend untersucht wird, ob sich passende Regeln finden, die das Zielobjekt in der Konklusion enthalten. Die in dieser Betrachtung zu erstellenden Regeln sind nur als *Schlussfolgerungsregeln* definierbar, welche nach dem Prinzip der *Vorwärtsverkettung* ausgewertet werden [BEKE06]:

WENN <Bedingung oder Situation> **DANN** <Schlussfolgerung>

Die Prämisse („**WENN**“) wird unter Beachtung des zur Verfügung stehenden, fallspezifischen Wissens um die konkrete Umsetzung von Komponenten sowie der Ansteuerung/Auswertung von Komponenten im Steuerungscode gebildet. Die Konklusion („**DANN**“) enthält das fachspezifische Wissen, welcher Detaillierungsgrad zum Einsatz kommen sollte, wenn die Prämisse wahr ist.

8.2.3 Vorgehen zur Bestimmung des Detaillierungsgrads

Die Erstellung der zu nutzenden Regeln erfolgt nach einem festgelegten Schema, das grundlegend in zwei Schritte unterteilt werden kann: Der Identifizierung von Zusammenhängen zwischen Prozessschritten und Komponenten (*Herstellung der Beziehung, Schritt I*) sowie einer anschließenden Analyse des Zusammenhangs zwischen Prozessschritt und Komponente, die die Erstellung von Regeln zum Ziel hat (*Analyse der Anlagenplanungsdaten und Regelerstellung, Schritt II*). Die *Schritte I* und *II* werden wie folgt beschrieben:

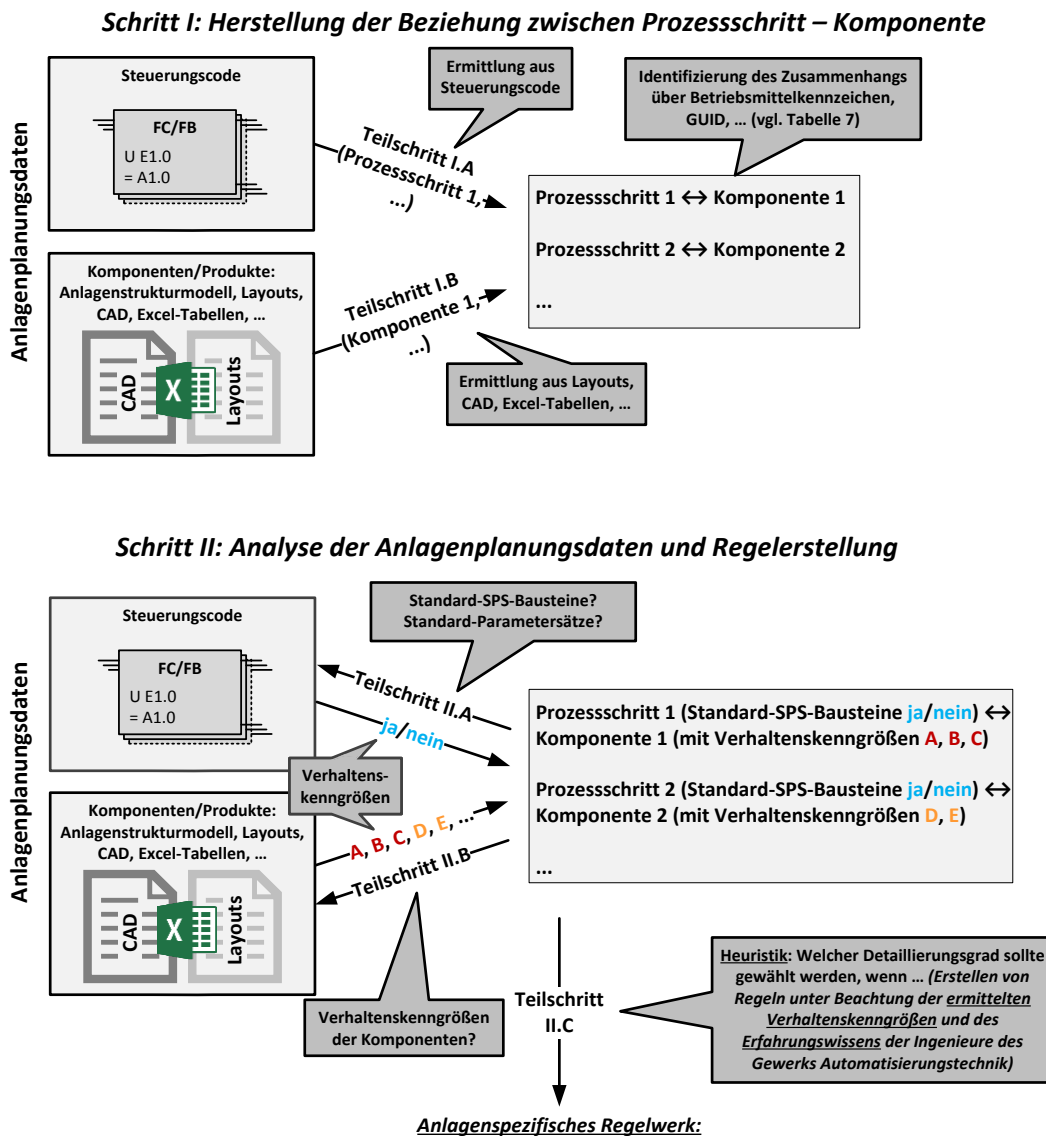
Schritt I: Herstellung der Beziehung zwischen Prozessschritt – Komponente

- I.A** Identifizierung von einzelnen Prozessschritten aus dem Steuerungscode
 - Beispiele: Motor vorwärtslauf, Motor rückwärtslauf, ...
- I.B** Identifizierung der zu den Prozessschritten zugeordnete Komponenten (beispielsweise über Betriebsmittelkennzeichen, GUID, ... → Tabelle 7)
 - Beispiele: Motor =FG01+SC01-A01 vorwärtslauf, Motor =FG01+SC01-A01 rückwärtslauf, ...

Schritt II: Analyse der Anlagenplanungsdaten hinsichtlich abgebildeter Verhaltenskenngrößen und Regelerstellung

- II.A Untersuchung des Steuerungscode hinsichtlich der Frage, ob Standard-SPS-Bausteine und Standard-Parametersätze zur Ansteuerung/Auswertung einer Komponente zur Verfügung stehen und ob diese genutzt werden oder nicht
- II.B Untersuchung der Verhaltenskenngrößen, die in den Anlagenplanungsdaten einer Komponente zugeordnet werden können und die zur spezifischen Konstruktion/Projektierung der Komponente geführt haben
 - Beispiele: Positioniergenauigkeit $\pm 1\text{ mm}$, transportierte Masse 500 kg , Bandgeschwindigkeit $0,6\text{ m/s}$, Stopprampe 2000 ms , ...
- II.C Heuristik: Erstellen von Regeln, die den notwendigen Detaillierungsgrad einer Komponente unter Beachtung der ermittelten Verhaltenskenngrößen und des Fachwissens der Ingenieure des Gewerks Automatisierungstechnik beschreibt
 - **WENN ... DANN** Detaillierungsgrad = x

Abbildung 8-2 zeigt die beiden Schritte sowie die beinhalteten Teilschritte in einer Übersicht.



Für Prozessschritt 1 ↔ Komponente 1 gilt unter Beachtung von Standard-SPS-Bausteine **ja/nein** und Verhaltenskenngrößen **A, B, C**:
Heuristik: WENN (Beachtung von: **ja/nein; A, B, C**) **DANN** Detaillierungsgrad = *x*

Für Prozessschritt 2 ↔ Komponente 2 gilt unter Beachtung von Standard-SPS-Bausteine **ja/nein** und Verhaltenskenngrößen **D, E**:
Heuristik: WENN (Beachtung von: **ja/nein; D, E**) **DANN** Detaillierungsgrad = *y*

...

Abbildung 8-2: Durchzuführende Teilschritte bei der Erstellung einer Regelbasis zur Ermittlung des notwendigen Detaillierungsgrads einzelner Komponentenmodelle

Im Nachfolgenden werden zwei Fallbeispiele vorgestellt, die die Ermittlung des Detaillierungsgrads aus den aus Anlagenplanungsdaten ermittelten Verhaltenskenngrößen beschreiben.

8.2.3.1 Fallbeispiel 1: Zuführ-Fördertechnik

Fallbeispiel 1 behandelt eine einfache Fördertechnik-Applikation, bei der Stückgüter an eine Bearbeitungsmaschine (nicht Teil der Betrachtung) zugeliefert werden sollen.

Ergebnisse der Ermittlung von Zuordnungen Prozessschritt – Komponente (Schritt I.A/I.B):

- **Beschreibung:** Zuführ-Fördertechnik-Applikation
- **Prozessschritte im Steuerungscode:** Motoransteuerung Vorwärtslauf, induktiver Näherungsschalter erfasst Anwesenheit des Stückguts
- **Komponenten:** Motor mit Frequenzumrichter (1), induktiver Näherungsschalter (2), Förderband (3)

Ergebnisse der Auswertung der relevanten Verhaltenskenngrößen aus Anlagenplanungsdaten (Schritte II.A/II.B):

- **Aus Anlagenplanungsdaten ermittelte Verhaltenskenngrößen:**
 - Bandgeschwindigkeit: 0,6 m/s
 - Transportierte Masse: bis 150 kg
 - Keine spezifische Positioniergenauigkeit gefordert
 - Umsetzung der Komponenten (1), (2) und (3) gemäß Best-Practices des Bauherren
- **Standard-SPS-Bausteine/Standard-Parametersatz:** Verfügbar und werden genutzt

Zuführ-Fördertechnik wird häufig auf die einfachste mögliche Weise implementiert. Für die Steuerung werden in der Regel Standard-SPS-Bausteine durch den Auftraggeber zur Verfügung gestellt, um einerseits die Prozesssicherheit hoch, andererseits den Instandhaltungsaufwand niedrig zu halten. Stehen diese nicht zur Verfügung, kommen in der Regel Standardbausteine des Gewerks Automatisierungstechnik zum Einsatz, die in fast jeder Firma vorhanden sein sollten. Da keine spezifische Positioniergenauigkeit gefordert ist, können Standard-Parametersätze der SPS-Bausteine verwendet werden, sofern diese verfügbar sind. Diese Standard-Parametersätze enthalten in der Regel Parameter wie vorgegebene Rampen und sind für Anwendungsfälle wie Zuführfördertechnik, bei der die Positionierung eines Stückguts keine Rolle spielt, geeignet. Es ergeben sich folgende Regeln (**Schritt II.C**):

Komponente Motor mit Frequenzumrichter (1) unter Beachtung des Prozessschritts „Motoransteuerung Vorwärtslauf“:

WENN Best-Practices genutzt werden

UND Standard-SPS-Bausteine genutzt werden

UND Standardparameter (Rampen) der SPS-Bausteine genutzt werden können

DANN nutze den makroskopischen Detaillierungsgrad (Gewichtung = 1)

Komponente Induktiver Näherungsschalter (2) unter Beachtung des Prozessschritts „Anwesenheitserkennung“:

WENN die Anwesenheit eines Stückguts erkannt werden muss

UND die Positioniergenauigkeit keine Rolle spielt

DANN nutze den makroskopischen Detaillierungsgrad (Gewichtung = 1)

Komponente Förderband (3) unter Beachtung aller Prozessschritte (das Förderband ist an allen Prozessschritten beteiligt):

WENN Best-Practices genutzt werden

DANN nutze den makroskopischen Detaillierungsgrad (Gewichtung = 1)

Der für die Simulation der Komponenten Motor (1), induktiver Näherungsschalter (2) und Förderband (3) notwendige Detaillierungsgrad ist daher der *makroskopische grad* ($\max(1, 1, 1) = 1$). Die untersuchten Komponenten bilden gemeinsam ein Modul.

8.2.3.2 Fallbeispiel 2: Positionieren an Pick-and-Place Station

In Fallbeispiel 2 wird eine Fördertechnik-Applikation betrachtet, bei der Stückgüter an einer Pick-and-Place Station sehr genau positioniert werden müssen.

Ergebnisse der Ermittlung von Zuordnungen Prozessschritt – Komponente (Schritte I.A/I.B):

- **Beschreibung:** Positionieren auf Förderband an Pick-and-Place Station
- **Prozessschritte im Steuerungscode:** Motoransteuerung Vorwärtslauf, optisches Lasermesssystem erfasst genaue Position des Stückguts
- **Komponenten:** Motor mit Frequenzumrichter (1), optisches Lasermesssystem (2), Förderband (3)

Ergebnisse der Auswertung der Verhaltenskenngrößen (Schritte II.A/II.B):

- **Aus Anlagenplanungsdaten ermittelte Verhaltenskenngrößen:**
 - Bandgeschwindigkeit: 0,6 m/s
 - Transportierte Masse: bis 500 kg
 - Positioniergenauigkeit $\pm 0,1$ mm
 - Umsetzung der Komponenten (1) und (3) gemäß Best-Practices des Bauherren, Komponente (2) ist nicht in Bauherren-Freigabeliste enthalten
- **Standard-SPS-Bausteine/Standard-Parametersatz:** Bausteine für Motoransteuerung verfügbar und genutzt, Standard-Parametersatz nicht nutzbar, für optisches Lasermesssystem beides nicht verfügbar

Muss an speziellen Stellen einer Anlage sehr genau positioniert werden, kann dies ein Hinweis auf eine genauer durchzuführende Simulation sein (Abschnitt 8.1.1, Verhaltenskenngröße „*kleine Sollwertabweichung*“). Die Positioniergenauigkeit ist immer eine Kombination aus mehreren Faktoren. Naturgemäß muss einerseits die Position sehr genau erfasst werden. Im hier beschriebenen Fall kommt ein optisches Lasermesssystem zum Einsatz, das nicht auf Best-Practices des Bauherren basiert. Es kann daher nicht von einem „Normalfall“ ausgegangen werden (\rightarrow Abschnitt 7.2.2), was ein Hinweis auf eine detailliertere Simulation sein kann. Zudem müssen Parameter wie z. B. Rampen des Motors mit Frequenzumrichter so abgestimmt sein, dass die zu positionierende Masse an der gewünschten Position mit der geforderten Genauigkeit zum Halten kommt. Hierfür ist es zweckmäßig, physikalische Effekte (Trägheit, Reibung, ...) zu beachten, die einen Einfluss auf die Parametrierung einer Rampe haben könnten. Es ergeben sich folgende Regeln (**Schritt II.C**):

Komponente Motor mit Frequenzumrichter (1) unter Beachtung des Prozessschritts „Motoransteuerung Vorwärtslauf“:

WENN Best-Practices genutzt werden

UND Standard-SPS-Bausteine genutzt werden

UND wegen hoher Positioniergenauigkeit und großer Masse Standardparameter des SPS-Bausteins (Rampen, max. Fördergeschwindigkeit, ...) nicht genutzt werden können

DANN nutze den mesoskopisch/dynamischen Detaillierungsgrad (Gewichtung= 3)

Komponente Optisches Lasermesssystem (2) unter Beachtung des Prozessschritts „erfasst genaue Position des Stückguts“:

WENN die Position eines Stückguts sehr genau erfasst werden muss

UND keine Standard-SPS-Bausteine genutzt werden

DANN nutze den mesoskopisch/nicht dynamischen Detaillierungsgrad (Gewichtung = 2)

Komponente Förderband (3) unter Beachtung aller Prozessschritte (das Förderband ist an allen Prozessschritten beteiligt):

WENN Best-Practices genutzt werden

DANN nutze den makroskopischen Detaillierungsgrad (Gewichtung = 1)

Der für die Simulation der Komponenten Motor (1), Optisches Lasermesssystem (2) und Förderband (3) notwendige Detaillierungsgrad ist daher der *mesoskopisch/dynamische Detaillierungsgrad* ($\max(3, 2, 1) = 3$).

Weitere Beispiele, die sich an einem konkreten Praxisbeispiel orientieren, sind in Abschnitt 9.2 im Rahmen einer Fallstudie dargestellt.

8.2.4 Darstellung der ermittelten Gewichtungen mittels Domain-Mapping-Matrix (DMM)

Eine auswertbare Darstellung des Zusammenhangs zwischen Prozessschritt und Komponente ist für eine weitere Verwendung vorteilhaft. Die *Domain-Mapping-Matrix (DMM)* [DABR07] stellt ein Rahmenkonzept dar, um die identifizierten Zusammenhänge abzubilden. Die *DMM* wurde mit der Intention geschaffen, Abhängigkeiten und Verbindungen zwischen verschiedenen Domänen darstellen zu können. Eine *DMM* wird als Rechteckmatrix ($n \times m$) dargestellt, wobei sich die betrachteten Domänen auf die Spalten (n) und Zeilen (m) der *DMM* aufteilen. Eine *DMM* kann genutzt werden, um die ermittelnden Zusammenhänge zwischen Prozessschritten und Komponenten sowie die Gewichtung dieser Zusammenhänge in eine klare und leicht verständliche, zudem rechnergestützt auswertbare Darstellungsform zu bringen.

Die *DMM* wird wie in Abbildung 8-3 dargestellt aufgebaut. Auf die einzelnen Zeilen (m) der *DMM* verteilen sich die Komponenten einer Anlage (Domäne: Ressourcen). In den Spalten (n) sind die einzelnen Prozessschritte aufgelistet (Domäne: Prozesse). Durch das Eintragen von Werten (\rightarrow Abschnitt 8.2.1) in die Zellen kann der quantifizierte Zusammenhang zwischen einem Prozessschritt und einer Komponente (hier: der notwendige Detaillierungsgrad einer Komponente für die Simulation eines Prozessschritts) dargestellt werden.

Das Eintragen von Werten in die *DMM* ergibt sich aus dem beschriebenen Vorgehen der Abschnitte 8.2.2 und 8.2.3:

1. Aus den Schritten **I.A** und **I.B** (Abbildung 8-2) ergeben sich zunächst die Zusammenhänge zwischen einzelnen Prozessschritten und Komponenten. Dieser Zusammenhang kann durch die in Abbildung 8-3 dargestellte farbliche Markierung in den Zellen abgebildet werden.

2. Aus den darauf folgenden Schritten II.A bis II.C (Abbildung 8-2) ergeben sich wiederum die numerischen Werte, die den Detaillierungsgrad einer Komponente bei Ausführung eines Prozessschritts beschreiben. Das Eintragen der Werte in die Zellen geschieht manuell, basierend auf dem mittels Regeln ermittelten Detaillierungsgrad einer einzelnen Komponente.

Abbildung 8-3 zeigt eine beispielhaft mit Werten gefüllten DMM, die die notwendigen Detaillierung einer einzelnen Komponente darstellt.

DMM		Prozessschritt 1	Prozessschritt 2	Prozessschritt 3	Prozessschritt 4	Prozessschritt 5	Prozessschritt 6	Prozessschritt 7	Prozessschritt 8	Prozessschritt 9	Prozessschritt 10	Prozessschritt 11	Prozessschritt 12
		Modul 1	Komponente 1	1									
Komponente 2		2	3										
Komponente 3				2									
Komponente 4				1									
Modul 2	Komponente 5					2	1						
Komponente 6								1					
Komponente 7								1					
Komponente 8									2				
Modul 3	Komponente 9									3	1		
Komponente 10												4	
Komponente 11													2

Detaillierungsgrad festlegen:

Detaillierungsgrad für alle Komponenten in Modul 1:
 $\max(1, 2, 3, 2, 1) = 3$;

Detaillierungsgrad für alle Komponenten in Modul 2:
 $\max(2, 1, 1, 1, 2) = 2$;

Detaillierungsgrad für alle Komponenten in Modul 3:
 $\max(3, 1, 4, 2) = 4$;

Abbildung 8-3: Beispielhafte Darstellung einer DMM bei der Komponenten (Zeilen) mit Prozessschritten (Spalten) in Beziehung gesetzt werden

Der für einzelne Module zugewiesene Detaillierungsgrad muss für eine (teil-)automatische Modellgenerierung jedem relevanten Objekt innerhalb des Anlagenstrukturmodells zugewiesen sein, damit ein Modellgenerierungsalgorithmus diesen erkennt (\rightarrow Abschnitt 8.4.2). Der Detaillierungsgrad ist daher als „simulationsspezifischer“ Parameter jedem Modul im Anlagenstrukturmodell (\rightarrow Abschnitt 6.1.5) zuzuweisen.

8.2.5 Dynamische Anpassung des Detaillierungsgrads zur Laufzeit des Modells

Die in den vorherigen Abschnitten beschriebene Methode ermöglicht die statische Zuweisung eines Detaillierungsgrads zu Komponenten. Komponenten können aber nicht nur einen Prozessschritt ausführen, sondern mitunter mehrere. Ein Beispiel hierfür ist in Abbildung 8-3 ersichtlich: Komponente 2 führt die Prozessschritte 2 und 3 aus, wobei für Prozessschritt 2 und 3 unterschiedliche Detaillierungsgrade ermittelt wurden. Es ergeben sich Situationen, in denen mehrere Detaillierungsstufen für eine Komponente (und damit alle Komponenten innerhalb eines Moduls) möglich sein können. Ein adaptierbarer Detaillierungsgrad, bei denen Teil-Simulationsmodelle abhängig vom durchgeführten Prozessschritt zwischen verschiedenen Detaillierungsstufen wechseln können, erscheint insbesondere zur Senkung der Rechenzeit sinnvoll und erstrebenswert [FSG⁺15]. Kerngedanke ist, dass ein hoher Detaillierungsgrad (viele Events = hohe Rechenzeit) nur dann genutzt wird, wenn er wirklich benötigt wird – in allen anderen Fällen soll nur in einem niedrigen Detaillierungsgrad (wenig Events = wenig Rechenzeit) simuliert werden.

Ein derartiger Ansatz wird von Schmüdderich/Frieben et al. ([ScTr14], [FSG⁺15]) beschrieben. Schmüdderich/Frieben et al. gelingt es, durch die Abschaltung einzelner Teile der Simulation die Berechnungsdauer für ein Anlagensimulationsmodell um insgesamt ~45% zu senken – im Konkreten von 78 Minuten Simulationsdauer auf 36 Minuten für einen in Realzeit 25 Sekunden langen Produktionsprozess [FSG⁺15]. Die von Frieben et al. beschriebene Lösung weist einige Gemeinsamkeiten zu dem in dieser Arbeit vorgestellten Ansatz auf: So wird die Modellierungssprache Modelica genutzt. Die Modelica-Modelle stehen in unterschiedlichen Detaillierungsstufen zur Verfügung. Das Anlagensimulationsmodell wird „heterogen“ (d. h. einzelne Teile in unterschiedlichem Detaillierungsgrad) aufgebaut. Zugleich bestehen jedoch bedeutende Unterschiede hinsichtlich der Implementierung und Nutzung der Simulationsmodelle sowie der zugrunde liegenden Testkonfiguration (für Details siehe Abschnitt 4.2.2.1). Diese Unterschiede haben zur Konsequenz, dass der von Schmüdderich/Frieben et al. beschriebene Ansatz des „Umschaltens“ von Detaillierungsstufen nicht ohne weiteres auf den in dieser Arbeit beschriebenen Ansatz angewendet werden kann. Die Gründe hierfür werden in den folgenden Absätzen dargelegt.

Unterschiede hinsichtlich der Verwendung der Modelica-Simulationsmodelle

Der von Schmüdderich/Frieben et al. beschriebene Ansatz basiert darauf, vorher definierte Teile eines Simulationsmodells (vorzugsweise diejenigen Teile, die viele Events und hohen Rechenaufwand verursachen) zur Laufzeit des Simulationsmodells nur bei „Bedarf“ zu aktivieren. Dies bedeutet, dass „unnütze“ Modellteile zur Laufzeit abschaltbar sein müssen [ScTr14]. Bei der Verwendung von Modelica ist ein derartiges Vorhaben prinzipiell nicht möglich, wie von Schmüdderich/Frieben et al. erkannt wird [ScTr14]. Generierte Modelica-Modelle sind stets gleichungsbasiert – es wird nicht mit Zuweisungen gearbeitet. Mittels bedingter Anweisungen „ausgeblendet“ Teile eines Modelica-Simulationsmodells würden daher trotzdem berechnet werden – die Ergebnisse würden nach der Berechnung schlicht verworfen. Ein Rechenzeitgewinn ist dadurch nicht möglich. Um diese Einschränkung zu umgehen, implementieren Schmüdderich/Frieben et al. die Modelica-Modelle in Matlab/Simulink [ScTr14]. Matlab/Simulink ermöglicht es wiederum, einzelne Teilmodelle zur Laufzeit einer Simulation zu aktivieren oder zu deaktivieren [ScTr14] – wobei deaktiviert bedeutet, dass die Modelle übersprungen und daher nicht berechnet werden (= weniger Rechenaufwand). Dies geschieht mittels vorher definierten Umschaltbedingungen [FSG⁺15].

Im Gegensatz dazu sind die in dieser Arbeit beschriebenen Simulationsmodelle (→ Abschnitt 7.4) „reine“ Modelica-Modelle, die als FMU exportiert werden und in eine HIL-Plattform implementiert werden (→ Abschnitt 5.3.3). Die Abschaltung einzelner Simulationsteile mittels vorher definierter Umschaltbedingungen wie von [FSG⁺15] beschrieben würde aus den beschriebenen Gründen keinen Rechenzeitgewinn bedeuten.

Durch die Umschaltung des Detaillierungsgrads erzeugte Events

Das „Umschalten“ des Detaillierungsgrads erzeugt zusätzliche Events, die die Rechenzeit erhöhen. Bei der von Schmüdderich/Frieben et al. beschriebenen MIL-Simulation (→ Abschnitt 4.2.2.1) spielt dies keine Rolle, da die MIL-Simulation nicht in (harter) Echtzeit ablaufen muss. Zudem ist der „netto“ erzielte Rechenzeitgewinn als deutlich höher einzuschätzen als der für das Umschalten benötigte Rechenaufwand.

Für die hier im Vordergrund stehende HIL-Simulation ist das Erzeugen zusätzlicher Events (und damit ein erhöhter Rechenaufwand) kontraproduktiv, da Situationen denkbar sind, in denen das Umschalten die notwendige harte Echtzeitfähigkeit beeinflussen könnte.

Umschaltbedingungen zu festen Zeitpunkten

Bei dem von Schmüdderich/Frieben et al. beschriebenen Vorgehen steht vorher fest, wann welcher Detaillierungsgrad verwendet werden muss [FSG⁺15]. Bei der Nutzung einer HIL-Simulation mit gekoppelter externer Steuerung ist dies nicht möglich – die Umschaltbedingungen können prinzipiell zu jedem beliebigen Zeitpunkt während eines Simulationslaufs eintreten. Es müsste daher eine Art „Überwachung“ in die Simulation eingebaut werden, die feststellt, wann welcher Prozessschritt ausgeführt wird. Diese Überwachung würde jedoch ihrerseits Events erzeugen und damit zusätzlichen Rechenaufwand erzeugen.

Fazit

Zusammenfassend lässt sich festhalten, dass die von Schmüdderich/Frieben et al. beschriebene Lösung für die in dieser Arbeit betrachtete HIL-Simulation so nicht umsetzbar ist. Die vorgestellte Methode der Umschaltung des Detaillierungsgrads von Simulationsmodellen stellt nur dann eine Alternative dar, wenn der Steuerungscode und das Anlagensimulationsmodell in einem Simulationswerkzeug modelliert werden (MIL-Simulation). In diesem Fall spielt die Berechnungsdauer einer Simulation prinzipiell keine Rolle. Die von Schmüdderich/Frieben et al. beschriebene Methode reduziert in diesem Fall die Zeit, bis Simulationsergebnisse vorliegen.

Derzeit ist nach Kenntnis des Autors kein technischer Lösungsweg bekannt, einen „dynamischen Detaillierungsgrad“ in der in dieser Arbeit beschriebenen Konstellation aus Modelica-Modellen und HIL-Simulation umzusetzen. Eine Umsetzung innerhalb des FMI-Frameworks könnte denkbar sein, scheitert derzeit aber daran, dass ein derartiger Anwendungsfall nicht explizit für das FMI Framework bedacht wurde. Es wäre daher hoher Aufwand am FMI-Framework notwendig, um einen derartigen Anwendungsfall ermöglichen zu können. Dies wird in dieser Arbeit nicht vertieft.

8.3 Sicherstellung der Konsistenz von Schnittstellen zwischen Modellen in verschiedenen Detaillierungsstufen

Ist die Zuweisung von Detaillierungsstufen zu einzelnen Komponenten erfolgt, muss sichergestellt werden, dass in heterogenem Detaillierungsgrad aufgebaute Anlagensimulationsmodelle stets konsistent hinsichtlich ihrer Schnittstellen sind. Simulationsmodelle in unterschiedlichen Detaillierungsgraden können unterschiedliche Schnittstellen aufweisen, die sich nicht zwangsläufig miteinander verbinden lassen. Dies betrifft insbesondere Teil-Simulationsmodelle, die entweder mittels kausaler, signalflussbasierter (→ Abschnitt 7.2.1 und 7.2.2) oder a-kausaler Modellierung (→ Abschnitt 7.2.3 und 7.2.4) umgesetzt wurden und die im Anlagensimulationsmodell miteinander verbunden werden sollen.

8.3.1 Analyse von Schnittstellen zwischen Modul-Simulationsmodellen

Schnittstellen zwischen modulbildenden Komponenten sind im Anlagensimulationsmodell aufgrund der in Abschnitt 7.2 festgelegten Restriktion immer konsistent. Dies gilt jedoch nicht für Schnittstellen zwischen Modulen. Je nach zugewiesenem Detaillierungsgrad kann es daher zur

Verschaltung von kausal oder a-kausal modellierten Simulationsmodellen kommen. Kernproblem ist, dass sich kausale und a-kausale Schnittstellen nicht ohne weiteres miteinander verbinden lassen.

A-kausale Schnittstellen sind vorzugsweise gemäß den aus der Elektrotechnik bekannten Kirchhoffschen Regeln zu betrachten. Sattler [SAT07] verallgemeinert diese wie folgt: „[...] [Die] Potentialdifferenzen zwischen zwei Klemmen treiben Flüsse [durch diese] hindurch“. Eine Liste typischer Potential- und Flussvariablen kann Tabelle 11 entnommen werden.

Aspekt:	Potentialgröße	Flussgröße
Elektrisch	U [V]	I [A]
Translation	v [m/s]	F [N]
Rotation	w_p [rad/s]	M [Nm]
Fluidisch	p [N/m ²]	Q [m ³ /s]
Thermisch	T [K]	Q _{th} [W]

Tabelle 11: Typische, korrespondierende Potential- und Flussgrößen in a-kausalen Modellierungsansätzen

A-kausale Simulationsmodelle sind gleichungsbasiert aufgebaut. Die Flussrichtung wird nicht vorgegeben, sondern ergibt sich aus den Gleichungen in den miteinander verbundenen Teil-Simulationsmodellen. A-kausale Schnittstellen können daher zugleich Eingangs- und Ausgangsschnittstellen sein.

Kausale Schnittstellen werden dagegen in Eingangs- oder Ausgangsschnittstellen unterteilt. Sie enthalten entweder eine Potential- oder eine Flussgröße (oder Kombinationen daraus) und sind stets gerichtet. Dies bedeutet, dass die Flussrichtung fest vorgegeben ist. Eine Ausgangsschnittstelle eines kausalen Simulationsmodells würde beispielsweise einen Volumenstrom Q zur Verfügung stellen. Ein korrespondierendes kausales Simulationsmodell kann diesen Wert aufnehmen und weiter verarbeiten.

Abbildung 8-4 zeigt inkonsistente Schnittstellen abstrakt am Beispiel einer Scheibenklebeanlage. Die betrachtete Anlage besteht aus 2 Modulen: Einer „Pumpenanlage“ (Modul 1), die Klebstoff aus einem Behälter mittels Pumpe an ein Applikationsmodul („Applikationskopf an Roboter“, Modul 2) pumpt. Hier sei das Pumpenmodul in einem mesoskopisch/nicht dynamischen Detaillierungsgrad modelliert (kausaler Modellierungsansatz), das Applikationsmodul in einem mesoskopisch/dynamischen Detaillierungsgrad (a-kausaler Modellierungsansatz).

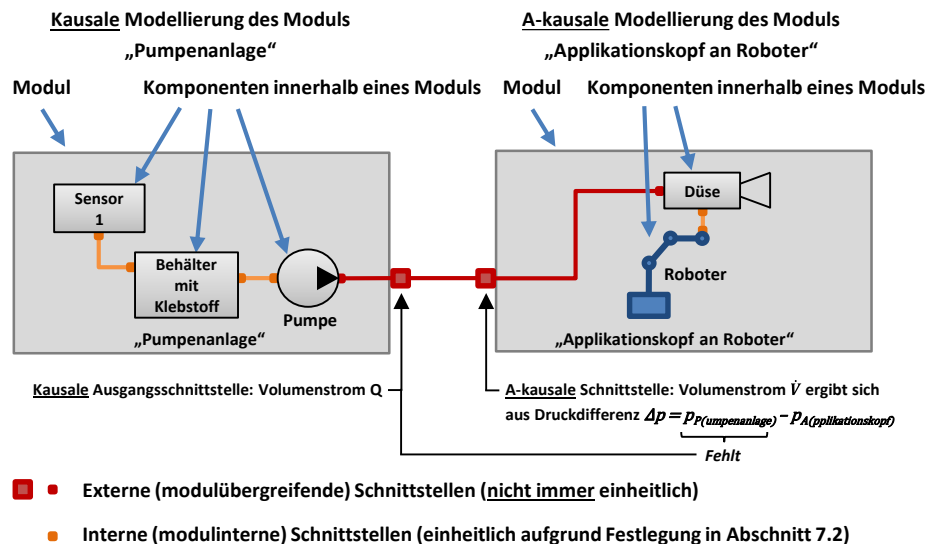


Abbildung 8-4: Mögliche Inkonsistenzen bei der am Beispiel zweier miteinander verbundener Module

Ersichtlich ist, dass die *internen* Schnittstellen (Schnittstellen zwischen Komponenten innerhalb eines Moduls) einheitlich sind (\rightarrow Abschnitt 7.2). Dies trifft jedoch nicht immer auch auf die *externen* Schnittstellen zu (modulübergreifende Schnittstellen). An der Ausgangsschnittstelle des Moduls „Pumpenanlage“ fehlt im Konkreten der Druck p_p . Dieser wird benötigt, damit aus der Druckdifferenz $\Delta p = p_p - p_A$ der Volumenstrom \dot{V} und seine Flussrichtung zwischen den Modulen berechnet werden kann (\rightarrow Abschnitt 8.3.2.1). Ein Anlagensimulationsmodell, bei dem ein kausal modelliertes Modul und ein a-kausal modelliertes Modul verbunden sind, würde daher nicht ohne weiteres funktionsfähig sein.

8.3.2 Nutzung von Modellverbindern zur Identifizierung von inkonsistenten Schnittstellen

Um ein konsistentes Anlagensimulationsmodell erzeugen zu können, muss ein Modellgenerator in der Lage sein, die in Abschnitt 8.3.1 beschriebene Situation und weitere, bei denen es zu Inkonsistenzen bei modulübergreifenden Schnittstellen kommen kann, zu erkennen und zu beheben.

Zur Erkennung von Inkonsistenzen ist es vorteilhaft, Schnittstellen hinsichtlich ihrer Eigenschaften (kausal/a-kausal) darzustellen. Ein Ansatz für die konsistente Darstellung von Schnittstellen wird von Christiansen et al. [CHF14A] vorgeschlagen. Die Autoren beschreiben das Konzept sogenannter *Modellverbinder*. Modellverbinder beschreiben die Kommunikation zwischen verschiedenen Teilen einer Anlage, die als Modelle dargestellt sind (z. B. in einem Anlagenstrukturmodell). Modellverbinder sollen „[...] eine einheitliche und durchgängige Informationsverwaltung [...]“ ermöglichen [CHF14A]. Modellverbinder stellen daher eine Möglichkeit dar, den Informationsfluss zwischen verschiedenen Modellen darzustellen. Jedoch sind im Speziellen die Schnittstellen zwischen (Teil-) Simulationsmodellen (welche in unterschiedlichem Detaillierungsgrad und damit möglicherweise in unterschiedlichen Modellierungsansätzen abgebildet wurden) nicht Teil der Betrachtung in [CHF14A]. Sollen Modellverbinder genutzt werden, ist es notwendig, ergänzende Regeln und auch Vorschriften einzuführen, um bei der Verwendung von Modellverbindern die Erkennung von Inkonsistenzen durch einen Modellgenerator zu ermöglichen.

Das hier vorgestellte Konzept sieht im Konkreten vor, Modellverbinder einzelnen Teil-Simulationsmodellen, die in der Simulationsmodellbibliothek vorliegen (die Simulationsmodellbibliothek ist ein Teil des Anlagenstrukturmodells, → Abschnitt 6.2.1), als virtuelle Objekte zuzuweisen. Modellverbinder werden durch Attribute beschrieben, die Informationen über die Art und Struktur der über diese Schnittstelle zu übermittelnden Signale/Werte beinhalten. In den nachfolgenden Abschnitten wird die genaue Verwendung von Modellverbindern erläutert.

8.3.2.1 Definition von Modellverbindern zur Abbildung in der Simulationsmodellbibliothek

Zunächst wird festgelegt, dass Modellverbinder stets granular so fein aufgelöst sein sollten, dass immer nur eine Potential- und/oder (korrespondierende) Flussgröße beschrieben wird. Ein Modellverbinder, der alle Schnittstellen eines Teil-Simulationsmodells zu einer zusammenfasst, wäre zwar möglich und würde die Betrachtung auf den ersten Blick vereinfachen, gleichzeitig aber den Aufwand zur Kopplung in Situationen, in denen unterschiedliche Schnittstellen existieren, stark erhöhen. Dies wird sich nachfolgend zeigen. Um das Erstellen von konsistenten Simulationsmodellen zu ermöglichen und gleichzeitig die Anforderungen an eine automatische Modellgenerierung zu erfüllen, müssen Modellverbinder so gestaltet sein, dass in einem ersten Schritt Inkonsistenzen identifiziert werden können. In einem zweiten Schritt sollen die erkannten Inkonsistenzen mittels des Modellgenerators behoben werden.

Für den ersten Schritt ist eine grundlegende Klassifizierung von Schnittstellen, die sich durch einen Modellgenerator interpretieren lassen, notwendig. Dies kann mit Hilfe der UML-Klassifikation von Schnittstellen geschehen: Anhand der Einteilung in *angebotene Schnittstellen* und *benötigte Schnittstellen* [BRJ05]. Diese Terminologie soll genutzt werden, um das Modellverbinder-Konzept mit grundlegenden Informationen anzureichern, die von einem Modellgenerator genutzt werden können.

Benötigt meint in diesem Fall, dass die „Erfüllung“ der Schnittstelle obligatorisch ist: Die Schnittstelle *muss* immer angeschlossen sein. Benötigte Schnittstellen sind die Eingangsschnittstellen von kausalen Simulationsmodellen. Es wird angenommen, dass die Aktionen innerhalb eines kausalen Simulationsmodells immer durch einen äußeren Einfluss (z. B. ankommendes Material) angestoßen werden. Eine andere Art von benötigter Schnittstelle sind die Schnittstellen von a-kausalen Simulationsmodellen. A-kausale Schnittstellen müssen immer mit einer korrespondierenden a-kausalen Schnittstelle verbunden sein.

Als *angebotene Schnittstellen* werden die Ausgangsschnittstellen von kausalen, signalflussbasierten Simulationsmodellen definiert. Diese werden bereitgestellt und *können* von anderen Teil-Simulationsmodellen genutzt werden. Angebotene Schnittstellen haben für das Simulationsmodell, welches diese Schnittstelle bereitstellt, keine Funktionsrelevanz (d. h., ob die Ausgangsschnittstelle von einem anderen Teil-Simulationsmodell genutzt wird oder nicht spielt für das Teil-Simulationsmodell, das diese Schnittstelle bereitstellt, keine Rolle).

Ergänzend zur Information, ob eine angebotene oder eine benötigte Schnittstelle vorliegt, sind die Potential- und/oder (korrespondierenden) Flussgrößen, die über die Schnittstelle ausgetauscht werden, als Attribute im Modellverbinder zu hinterlegen. Abbildung 8-5 zeigt anhand des in Abbildung 8-4 gezeigten Beispiels der Scheibenklebeanlage, wie Modellverbinder aufgebaut sind. Die vom Simulationsmodell der Pumpe (Abbildung 8-5 „Pumpe_DG2“) bereitgestellte Schnittstelle ist vom Typ *angeboten*, während die Schnittstelle des Simulationsmodells des Applikationskopfs (Abbildung 8-5 „Applikationskopf_DG3“) als Typ *benötigt* identifiziert werden kann.

Aus der Liste von Attributen der Potential- und Flussgrößen der Schnittstellen ist ersichtlich, dass im Modellverbinder des (kausalen) Simulationsmodells der Pumpe das Attribut „DruckP“ fehlt, welches vom (a-kausalen) Simulationsmodell des Applikationskopfs benötigt wird.

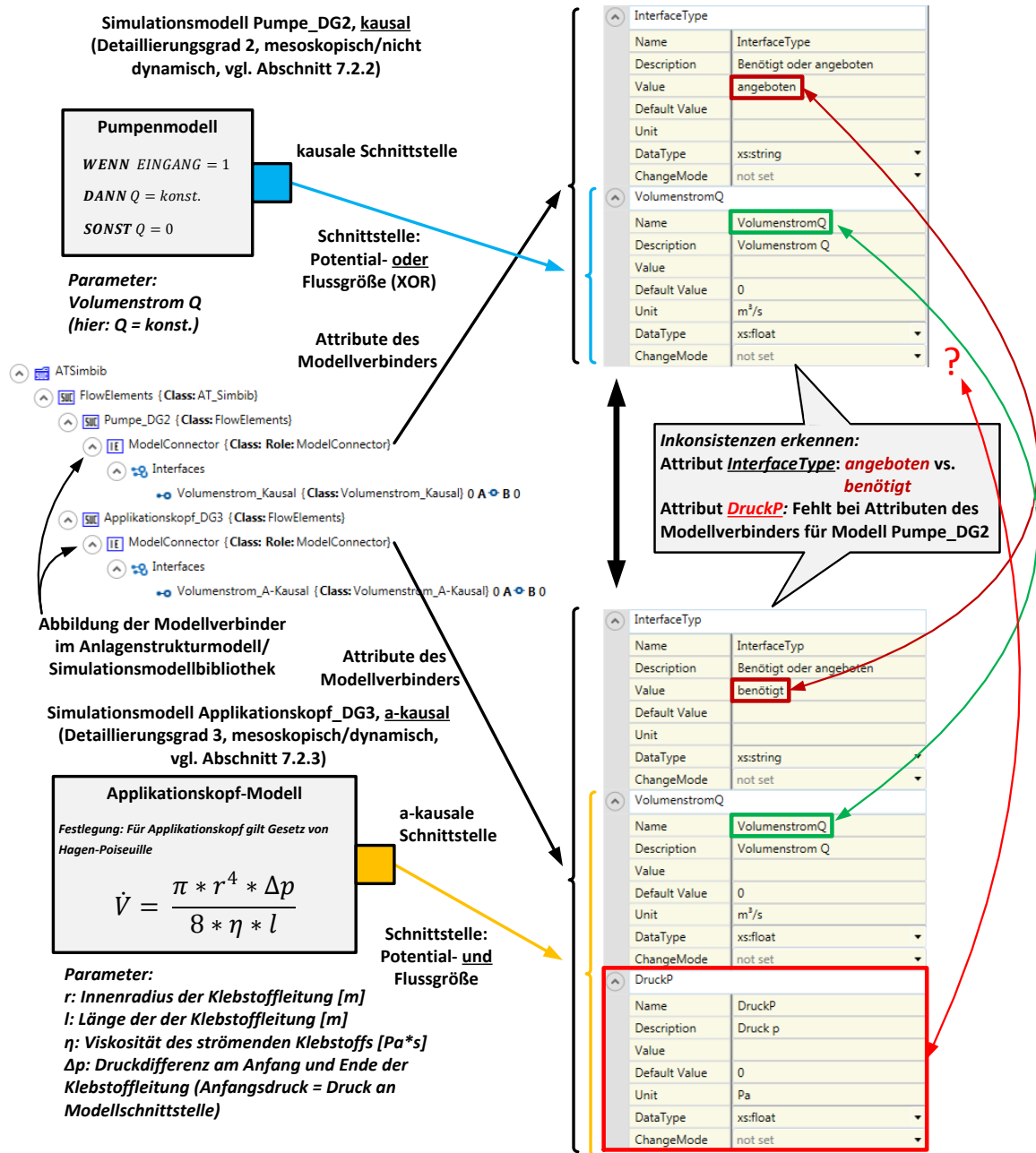


Abbildung 8-5: Prinzip des erweiterten Modellverbinder-Konzepts zur Identifizierung von Inkonsistenzen

8.3.2.2 Regelbasierter Ansatz zur Analyse von Modellverbindern und zur Sicherstellung der Konsistenz von Anlagensimulationsmodellen

Diskrepanzen zwischen Schnittstellen von Teil-Simulationsmodellen können immer dort auftreten, wo Attribute der Schnittstellen, definiert innerhalb der jeweiligen Modellverbinder, nicht zueinander passend sind und angebotene auf benötigte Schnittstellen treffen. Ein regelbasierter Ansatz soll helfen, diese Situationen identifizieren zu können. Nach erfolgter Erkennung ist die betroffene Schnittstelle direkt im Anlagensimulationsmodell so zu modifizieren, dass ein durchge-

hender Informationsfluss zwischen einzelnen Teil-Simulationsmodellen ermöglicht wird und damit die Konsistenz des Anlagensimulationsmodells gewährleistet wird. Hierfür sind zusätzliche Simulationsmodelle, die z. B. die a-kausale Schnittstelle abschließen, nötig. Diese Simulationsmodelle müssen für jede mögliche a-kausale Schnittstelle vorgesehen werden und den Übergang zwischen detaillierten und weniger detaillierten Anlagenteilen und umgekehrt ermöglichen. Dadurch, dass die Granularität der Schnittstellen und der sie repräsentierenden Modellverbinder feingranular gewählt wurde (→ Abschnitt 8.3.2.1), ist nur die Definition von wenigen Kopplungs- und Terminierungssimulationsmodellen notwendig (vgl. auch Liste von Potential- und Flussgrößen in Tabelle 11). Kopplungs- und Terminierungssimulationsmodelle *müssen* in einer Simulationsmodellbibliothek vorgehalten werden. Ein Beispiel für ein derartiges Modell wird in Abbildung 8-7 gegeben.

Die folgenden Regeln zeigen ein allgemeines Schema für diejenigen Fälle, in denen Schnittstellen aufgrund unterschiedlicher Attribute/Typen nicht miteinander kompatibel sind.

- **Regel 1** (Fall 1 Abbildung 8-6):

WENN eine *angebotenen Schnittstelle* mit einer *benötigten Schnittstelle* gekoppelt werden soll

UND die *Schnittstellenattribute* inkompatibel sind

DANN füge ein *Kopplungselement* als ergänzendes Simulationsmodell ein, dass die in den Modellverbindern hinterlegten Schnittstellen aufweist

UND leite die für das *Kopplungselement* notwendigen Parameterwerte aus den Anlagenplanungsdaten ab

- **Regel 2** (Fall 2 Abbildung 8-6):

WENN beide *Schnittstellen* als *benötigte Schnittstellen* klassifiziert sind

UND die *Schnittstellenattribute* inkompatibel sind

DANN füge ein zusätzliches Simulationsmodell als Abschlussmodell (*Terminierungselement*) mit einer Schnittstelle ein, dass dieselben Schnittstellenattribute wie die korrespondierende, benötigte Schnittstelle am ursprünglichen Simulationsmodell aufweist

UND leite die für das *Terminierungselement* notwendigen Parameterwerte aus den Anlagenplanungsdaten ab

UND greife den für das nachfolgende Simulationsmodell notwendigen Wert aus der Schnittstellenverbindung ab (Analog dem Spannungs-Messprinzip)

Abbildung 8-6 zeigt eine Anwendung der vorgestellten Regeln anhand eines einfachen Beispiels. Betrachtet wird die Flussgröße \dot{V} , die zwischen zwei Teil-Simulationsmodellen (hier: Pumpe, Applikationskopf, → Abbildung 8-4) ausgetauscht werden soll.

keine Differentialgleichungen). Werden Druck wie auch Volumenstrom vorgegeben ist das in Abbildung 8-7 dargestellte Gleichungssystem überbestimmt. Bei der Generierung des Modelica-Modells wird dies verhindert, in dem der zugewiesene Druck dem in Modelica stets zu modellierenden Medium zugewiesen wird. Das hier „Kopplungselement“ genannte Modell ist aus Modelica-Sicht daher eine Volumenstrom-Quelle, bei der der Volumenstrom vorgegeben wird.

8.4 Erstellung des Simulationsmodells

Die Betrachtungen in den Kapiteln 6, 7 und 8 beschreiben einerseits notwendige Teilschritte, die während der Simulationsmodellerstellung ausgeführt werden müssen. Andererseits beschreiben sie die Voraussetzungen, die für einen automatischen Modellerstellungsprozess erfüllt sein müssen. Der zeitliche Verlauf einer virtuellen Inbetriebnahme kann in diesem Zusammenhang in mehrere Phasen gegliedert werden [LIN11]. Anfangs stehen zunächst eine Analysephase und in ihr Tätigkeiten wie z. B. die Untersuchung der vorliegenden Anlagenplanungsdaten hinsichtlich des zu wählenden Detaillierungsgrads. Weitere Tätigkeiten wie eine Situationsanalyse oder auch die Versuchsplanung (= Definition durchzuführender Testfälle) können hier ebenso durchgeführt werden (vgl. Abbildung 3-3). In der Phase der Modellerstellung erfolgt die Erstellung des Simulationsmodells unter Beachtung der in der Analysephase erarbeiteten Ergebnisse. In der finalen Phase erfolgt der eigentliche Test des Steuerungscode. Abbildung 8-8 zeigt diesen Verlauf in einer Übersicht.

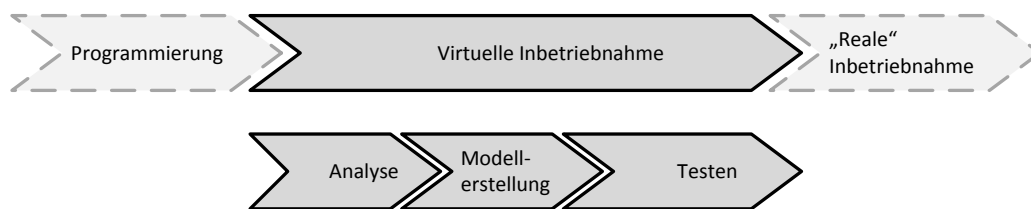


Abbildung 8-8: Durchführung einer virtuelle Inbetriebnahme in Anlehnung an [LIN11]

Die nächsten Abschnitte beschäftigen sich ausschließlich mit der Phase der Modellerstellung und der dort möglichen manuellen, teil-automatischen oder voll-automatischen Modellgenerierung.

8.4.1 Durchzuführende Schritte bei der Modellerstellung und Bedingungen für eine teil-/voll-automatische Modellgenerierung

Die Phase der Modellerstellung lässt sich in einzelne Teilschritte zerlegen, deren Durchführung bei einer Modellerstellung zwingend notwendig ist. Aus den bisher erarbeiteten Lösungen in Kapitel 6, 7 und 8 ergeben sich hierfür insgesamt 4 Teilschritte:

- **Schritt A:** Substitution von Objekten aus Anlagenstruktur mit passenden Simulationsmodellen (Struktur aus Anlagenstrukturmodell in Simulationsmodell übertragen) (→ Kapitel 6 und 7)
- **Schritt B:** Übertragung der relevanten Parameterwerte aus Anlagenstrukturmodell in Simulationsmodell mittels Mapping (→ Kapitel 6)
- **Schritt C:** Übertragung der relevanten Schnittstellen aus Anlagenstrukturmodell in Simulationsmodell mittels Mapping (→ Kapitel 6)
- **Schritt D:** Sicherstellung der Konsistenz des entstandenen Anlagenstrukturmodells (→ Abschnitt 8.3)

Abhängig davon, ob die Modellerstellung rein manuell oder teil-/voll-automatisch durchgeführt werden soll, sind verschiedene Bedingungen hinsichtlich dieser Teilschritte zu beachten. Diese Bedingungen lassen sich aus den bisherigen Beschreibungen in den Kapiteln 6, 7 und 8 akkumulieren. Tabelle 12 zeigt zusammenfassend die identifizierten Schritte der Simulationsmodellerstellung und die dafür notwendigen Bedingungen hinsichtlich einer voll-automatischen Modellgenerierung.

Durchzuführender Schritt:	Beschreibung der Tätigkeit bei Modellerstellung:	Lösungsbaustein	Beschreibung der Kriterien	Bedingung für automatische Modellgenerierung:
A	Substitution von Objekten in Anlagenstruktur mit passenden Simulationsmodellen (Transferieren von Objekten aus Anlagenstruktur in Simulationsmodell)	Baustein 1.1	- Vollständiges und konsistentes Anlagenstrukturmodell hinsichtlich des Vorhandenseins aller zu simulierenden Anlagenstrukturobjekte - Anlagenstruktur muss rechnerauswertbar vorliegen - Simulationsmodelle aus Anlagenstruktur heraus identifizierbar	A_B
		Baustein 2	- Simulationsmodellbibliothek muss vorliegen - Unterschiedliche Detaillierungsstufen von Modellen müssen vorliegen	
		Baustein 3.1	- Detaillierungsgrad festgelegt und im Anlagenstrukturmodell an Objekt als Parameter hinterlegt	
B	Übertragung der relevanten Parameterwerte aus Anlagenstrukturmodell in Simulationsmodell	Baustein 1.2	- Vollständige und konsistente Abbildung aller Parameterwerte in Anlagenstrukturmodell - Mappinginformation muss vorliegen	B_B
C	Übertragung der relevanten Schnittstellen und Verbindungsinformationen aus Anlagenstrukturmodell in Simulationsmodell		- Vollständige und konsistente Abbildung aller Verbindungen zwischen Objekten in Anlagenstrukturmodell - Mappinginformation muss vorliegen	C_B
D	Konsistenzcheck hinsichtlich aus dem Anlagenstrukturmodell übertragener Schnittstellen und ggf. Korrekturen direkt im Simulationsmodell	Baustein 3.2	- Aus Anlagenstrukturdaten abgeleitetes Simulationsmodell inkl. Verbindungen und Schnittstellen liegt vor - Simulationsmodellbibliothek bietet geeignete Kopplungselemente	D_B

Tabelle 12: Durchzuführende Schritte der Modellerstellung und dafür notwendige Bedingungen hinsichtlich einer teil- oder voll-automatischen Modellgenerierung.

Aus den in Tabelle 12 dargestellten Informationen lässt sich zunächst ableiten, dass eine rein manuelle Modellerstellung immer dann notwendig ist, wenn die Bedingung **A_B** für Schritt **A** nicht erfüllt werden kann. In diesem Fall gilt:

Manuelle Modellerstellung notwendig, wenn: $\neg A_B$

Es wird in diesem Zusammenhang angenommen, dass wenn Bedingung **A_B** nicht erfüllt ist, auch die Erfüllung der anderen Bedingungen **B_B**, **C_B** oder **D_B** in der Praxis nicht realistisch erscheint. Zwar könnten unter Beachtung der Bedingung **A_B** durch die manuelle Erstellung eines Anlagenstrukturmodells einzelne Teilschritte „nachträglich“ automatisch ausgeführt werden. In diesem Fall wird angenommen, dass eine (teil-)automatische Modellgenerierung keine wesentlichen Vorteile hinsichtlich einer Aufwandsreduzierung im Vergleich zur konventionellen, rein manuellen Erstellung eines Simulationsmodells ergibt.

Eine teil-automatische Modellgenerierung definiert sich als die Möglichkeit der teilweisen Erstellung eines Simulationsmodells mittels eines Modellgenerators, solange einzelne Bedingungen

erfüllt sind. Als ein wesentlicher zu beachtender Aspekt ist die Sicherstellung der Strukturähnlichkeit von Anlagenstrukturmodell und entstehendem Simulationsmodell, d. h. dem Herstellen einer Ähnlichkeitsbeziehung zwischen beiden (\rightarrow Abschnitt 6.1.1). Dies ist immer dann möglich, wenn zumindest Schritt **A** automatisiert durchgeführt werden kann und damit alle der in Bedingung **A_B** genannten Kriterien erfüllt sind. Für eine teil-automatische Modellgenerierung gilt in diesem Zusammenhang:

Teil-automatische Modellgenerierung möglich, wenn: A_B

Abhängig von möglicherweise zusätzlich erfüllten Bedingungen **B_B**, **C_B** oder **D_B** und der damit zusammenhängen möglichen Umsetzung einzelner Teilschritte **B**, **C** oder **D** kann der bei einer teil-automatischen Modellgenerierung in jedem Fall zusätzlich notwendige manuelle Aufwand bei der Modellerstellung variieren.

Eine voll-automatische Modellgenerierung enthält stets alle Schritte **A**, **B**, **C** und **D**, die zur Erzeugung eines Simulationsmodells notwendig sind. Folglich müssen alle der damit zusammenhängenden Bedingungen erfüllt sein. Es gilt hinsichtlich der einzelnen Bedingungen:

Voll-automatische Modellgenerierung möglich, wenn: $A_B \wedge B_B \wedge C_B \wedge D_B$

Möglicherweise zusätzlich notwendige Schritte wie die Generierung eines in einer Laufzeitumgebung lauffähigen Modells aus dem Simulationswerkzeug heraus (im Rahmen dieser Arbeit: die Generierung einer FMU \rightarrow Abschnitt 5.3.3) werden nicht weiter betrachtet.

8.4.2 Algorithmus zur Erstellung des Simulationsmodells

Aus den beschriebenen Teilschritten **A**, **B**, **C** und **D** kann, unter Beachtung der in dieser Arbeit vorgestellten Lösungsbausteine, ein Algorithmus abgeleitet werden, der helfen kann, den Vorgang der Modellerstellung zu systematisieren. Dieser Algorithmus kann einerseits als Hilfe („Leitfaden“) für die rein manuelle Modellerstellung dienen, andererseits aber auch eine mögliche Grundlage für die Umsetzung einer teil- oder voll-automatischen Modellgenerierung bilden. Die einzelnen Teilschritte des Algorithmus sind im Folgenden beschrieben. Abbildung 8-9 zeigt die einzelnen Teilschritte in einer Übersicht. Anm.: In Klammern erfolgt ein Verweis auf den Abschnitt, in dem die Erklärung zur Erfüllung der Voraussetzung bzw. das Vorgehen beschrieben ist.

Vorbereitende Tätigkeiten (Analysephase):

Ermittlung des notwendigen Detaillierungsgrads von Simulationsmodellen gemäß beschriebener Methode (\rightarrow **Baustein 3.1, Beschreibung in Abschnitt 8.1 und 8.2**)

Voll-automatische Modellerstellung:

- **Voraussetzung zur Erfüllung von Bedingung A_B :**
 - Vollständiges und konsistentes Anlagenstrukturmodell liegt vor (\rightarrow **Baustein 1.1, Abschnitt 6.1.3, 6.1.5**)
 - Simulationsmodellbibliothek muss in Anlagenstrukturmodell vorliegen (\rightarrow **Baustein 1.2, Abschnitt 6.2**)
 - Anlagenstrukturmodell muss rechnerauswertbar vorliegen (\rightarrow **Baustein 1.1, Abschnitt 6.1.4**)

- Simulationsmodellbibliothek muss vorliegen (→ **Baustein 2, Abschnitt 7.4**)
- Simulationsmodelle müssen in unterschiedlichem Detaillierungsgrad vorliegen (→ **Baustein 2, Abschnitt 7.2**)
- Detaillierungsgrad festgelegt und im Anlagenstrukturmodell an Objekt als Parameter hinterlegt (→ **Baustein 3.1, Abschnitt 8.2**)

- **Durchführung Teilschritt A (Abbildung 8-9 blau hinterlegt):**
 - siehe Beschreibungen in **Abschnitt 6.2.1** zu **Abbildung 6-9**

- **Voraussetzung zur Erfüllung von Bedingung B_B:**
 - Vollständige und konsistente Abbildung aller Parameterwerte im Anlagenstrukturmodell (→ **Baustein 1.2, Abschnitt 6.2.2**)
 - Mappinginformation muss vorliegen (→ **Baustein 1.2, Abschnitt 6.2.2**)
- **Durchführung Teilschritt B (Abbildung 8-9 orange hinterlegt):**
 - siehe Beschreibungen in **Abschnitt 6.2.2** zu **Abbildung 6-11**

- **Voraussetzung zur Erfüllung von Bedingung C_B:**
 - Vollständige und konsistente Abbildung aller Verbindungen zwischen Objekten im Anlagenstrukturmodell (→ **Baustein 1.2, Abschnitt 6.1.2**)
 - Mappinginformation muss vorliegen (→ **Baustein 1.1, Abschnitt 6.1.3.4**)
- **Durchführung Teilschritt C (Abbildung 8-9 grün hinterlegt):**
 - siehe Beschreibungen in **Abschnitt 6.2.2** zu **Abbildung 6-12**

- **Voraussetzung zur Erfüllung von Bedingung D_B:**
 - Aus Anlagenstrukturdaten abgeleitetes Simulationsmodell inkl. aller Verbindungen und Schnittstellen liegt vor (*Teilschritte A, B, C ausgeführt*)
 - Simulationsmodellbibliothek bietet geeignete Kopplungselemente (→ **Baustein 3.2, Abschnitt 8.3**)
- **Durchführung Teilschritt D (Abbildung 8-9 rot hinterlegt):**
 - siehe Beschreibungen in **Abschnitt 8.3** zu **Abbildung 8-6/Abbildung 8-7**

Abbildung 8-9 zeigt zusammenfassend den entstandenen Modellgenerierungs-Algorithmus. Dieser ermöglicht eine voll-automatische Modellgenerierung, insofern alle Voraussetzungen dafür erfüllt sind.

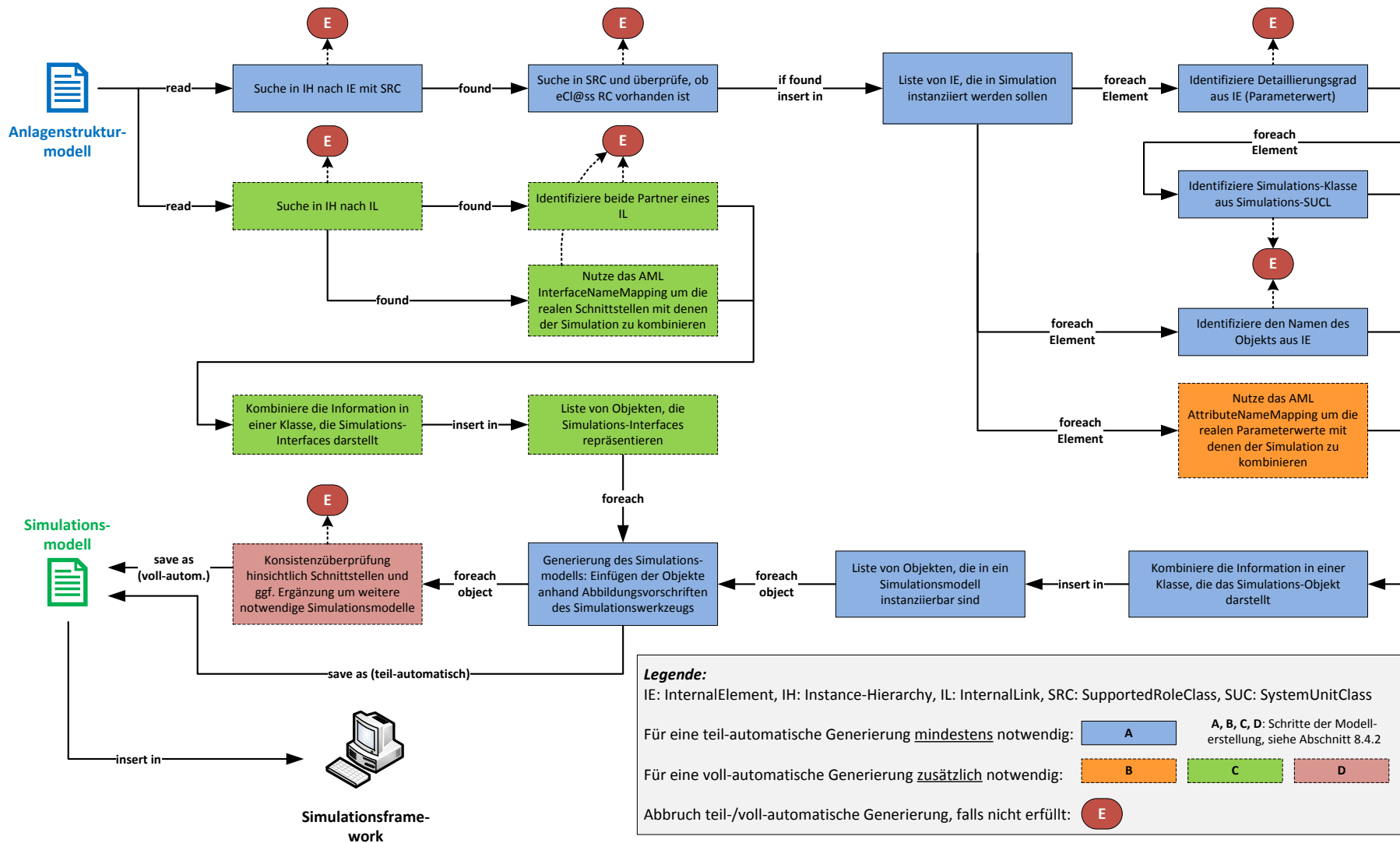


Abbildung 8-9: Generischer Modellgenerierungsalgorithmus

9 Fallstudien

Die diesem Kapitel vorgestellten Fallstudien stellen eine exemplarische Anwendung der in Kapitel 6, 7 und 8 erarbeiteten Lösungsbausteine dar und sollen deren praktische Anwendbarkeit demonstrieren. In diesem Kontext werden 2 Fallstudien durchgeführt.

Fallstudie 1 konzentriert sich auf die prinzipielle Machbarkeit des Konzepts der voll-automatischen Modellgenerierung unter Beachtung der in dieser Arbeit dafür beschriebenen Voraussetzungen. Die Fallstudie erfolgt unter Nutzung einer real existierenden Anlage, die in Forschung und Lehre am Institut für Automatisierungstechnik der Helmut-Schmidt-Universität zum Einsatz kommt. Ferner wird die Fähigkeit des generierten Simulationsmodells demonstriert, für die Durchführung von Testfällen während einer virtuellen Inbetriebnahme eingesetzt werden zu können.

Fallstudie 2 konzentriert sich auf die in Kapitel 8 beschriebene Methode zur Ermittlung eines anwendungsfallspezifischen Detaillierungsgrads (Baustein 3.1). Zu diesem Zweck erfolgt die Analyse einer Elektrohängbahn-Applikation, die in enger Anlehnung an die VDI-Richtlinie [VDI 4441-1] entworfen wurde. Die Betrachtungen erfolgen an einem generalisierten Beispiel und sollen einen Eindruck von der Anwendung der Methode schaffen.

9.1 Fallstudie Modellfabrik

9.1.1 Vorstellung der untersuchten Fördertechnikapplikation

Bei der in Fallbeispiel 1 betrachteten Anlage handelt sich um eine Fördertechnikapplikation. Die Anlage steht beispielhaft für Materialflussapplikationen, wie sie sich in der Handhabungs- und Montagetechnik häufig in der industriellen Praxis wiederfinden. Im Konkreten kann die Anlage als eine Sortieranlage gesehen werden, mit der die Bearbeitungsreihenfolge von Stückgütern verändert werden kann.

Die betrachtete Anlage enthält eine Reihe von Fördertechnikelementen wie Band- und Kettenförderer sowie insgesamt 4 Drehtische. Der in dieser Fallstudie betrachtete Anlagen-Gesamtprozess sieht vor, dass Stückgüter aus einer Quelle bis zu einem Dispositionspunkt transportiert werden, an dem festgelegt wird, durch welchen Pfad das Stückgut durch die Anlage fährt (Abbildung 9-1). Die Stückgüter „verschwinden“ nach Durchfahren der Anlage in einer Senke. Um das Beispiel einerseits einfach zu gestalten, andererseits die Wiederholbarkeit des Prozesses zu gewährleisten, erfolgte für die Konzept-Validierung die Programmierung einer alternierenden Richtungszuweisung auf den oberen (Pfad 1) und unteren (Pfad 2) Pfad der Anlage (siehe Abbildung 9-1), so dass sich eine genaue Verteilung der transportierten Stückgüter von 1:1 ergibt.

Die Ansteuerung der Motoren erfolgt direkt über die zum Einsatz kommende Kommunikationsperipherie durch das Ansteuern von (digitalen) Ausgängen. Selbiges gilt für die Motoren der Drehtische. Die Sensoren zur Positionserkennung der Stückgüter sind als induktive Näherungsschalter ausgeführt, die Sensoren zur Erkennung der Position der Drehtische als mechanische Kontakt-

schalter. Die Stückgüter besitzen gleiche Maße wie auch Masse, etwaige Zustandsänderungen während des Durchfahrens der Anlage können vernachlässigt werden. Abbildung 9-1 zeigt die Anlage in einer vereinfachten Übersicht. Ein detailliertes Anlagenschaubild inkl. Kennzeichnung aller in der Anlage enthaltenen Komponenten mittels Betriebsmittelkennzeichen ist in Anhang 3 dargestellt.

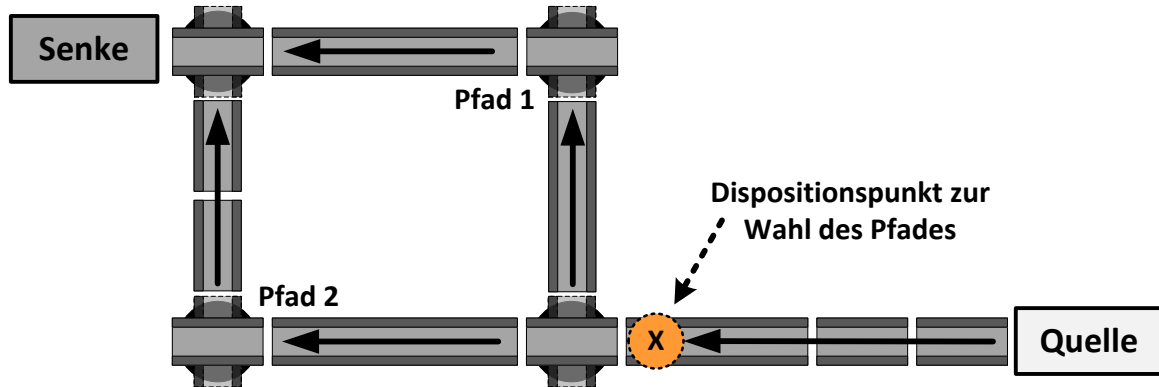


Abbildung 9-1: Übersicht über die betrachtete Anlage

9.1.2 Ermittlung des notwendigen Detaillierungsgrads (Baustein 3.1)

Eine Bestimmung des notwendigen Detaillierungsgrads (Baustein 3.1) kann für Fallstudie 1 stark verkürzt erfolgen. Die in der Anlage beinhalteten Komponenten können in guter Näherung als durchgehend standardisiert beschrieben werden. Daraus lässt sich, auch aufgrund der konsequenten Verwendung von Standard-SPS-Bausteinen zur Steuerung des Prozesses (→ Abschnitt 9.1.5), schließen, dass von einer durchgehend standardisierten Lösung gesprochen werden kann. In diesem Fall genügt es, den niedrigsten Detaillierungsgrad (makroskopischer Detaillierungsgrad) zu wählen.

Um die Machbarkeit einer physik-basierten Simulation einerseits zu demonstrieren, andererseits die Lauffähigkeit einer virtuellen Inbetriebnahme unter Verwendung eines in heterogenem Detaillierungsgrad abgebildeten Anlagensimulationsmodells zu demonstrieren, erfolgten die durchgeführten Tests bewusst in davon abweichenden Detaillierungsstufen. Die Validierung des Gesamtansatzes erfolgte daher im ersten Testdurchlauf mittels eines rein mesoskopisch/dynamischen Modells (Detaillierungsgrad 3). In einem zweiten Testlauf wurde zusätzlich ein gemischt makroskopisch, mesoskopisch/dynamisches Modell der Anlage erzeugt (Detaillierungsgrad 1 und 3). Die genaue Testdurchführung wird in Abschnitt 9.1.5 beschrieben.

9.1.3 Genutztes Anlagenstrukturmodell (Baustein 1.1 und 1.2) und genutzte Simulationsmodellbibliothek (Baustein 2)

Das Anlagenstrukturmodell, welches als Basis der automatischen Modellgenerierung dient, wurde gemäß der in den Lösungsbausteinen 1.1 und 1.2 erarbeiteten Ergebnisse aufgebaut (→ Abschnitt 6.1.5). Abbildung 9-2 zeigt einen Ausschnitt der in AutomationML erstellten Anlagenstruktur inkl. eines Beispiels der hinterlegten Attribute für die mechanische Apparatur eines Förderbands. Für eine ausführliche Darstellung des genutzten Anlagenstrukturmodells sei auf Anhang 4 verwiesen.



Abbildung 9-2: Ausschnitt Anlagenstrukturmodell

Die in Abschnitt 7.4 vorgestellte Simulationsmodellbibliothek wurde für die Simulation des vorgestellten Prozessablaufs genutzt. Die Simulationsmodellbibliothek wurde gemäß den Beschreibungen in Abschnitt 6.2.1 (zweiter Absatz) aus Modelica als SUC-Objekte in AutomationML importiert. Ein Ausschnitt aus der Simulationsmodellbibliothek in Modelica, deren Abbildung in AutomationML (als SUC-Objekte) und eine Darstellung der aus Modelica in AutomationML übernommenen Parameter ist in Abbildung 9-3 dargestellt.

Eine Übersicht über die in AutomationML abgebildete Simulationsmodellbibliothek findet sich in Anhang 5, für die Implementierung der Simulationsmodelle in Modelica sei auf Abschnitt 7.4.2 verwiesen.

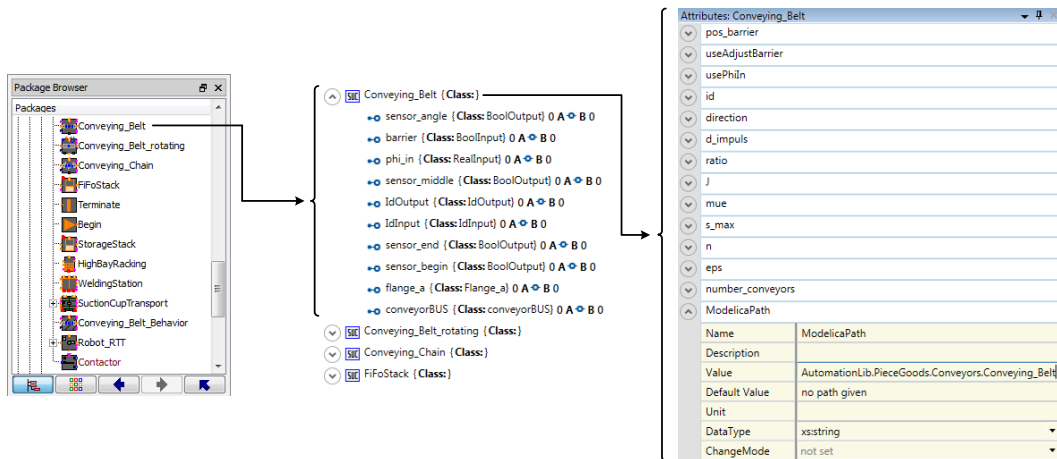


Abbildung 9-3: Ausschnitt Simulationsmodellbibliothek in Modelica (links), Abbildung als SUC inkl. Schnittstellen in AutomationML (mitte) und der aus Modelica in AutomationML übertragenen Parameter (rechts)

9.1.4 Automatische Modellgenerierung (Baustein 4)

Das Simulationsmodell wurde mit Hilfe des in Abschnitt 8.4.2 vorgestellten Modellgenerierungsalgorithmus aus dem vorliegenden Anlagenstrukturmodell voll-automatisch generiert. Die für eine automatische Modellgenerierung notwendige Konsistenzüberprüfung hinsichtlich der im Simulationsmodell vorhandenen Schnittstellen (Baustein 3.2, → Abschnitt 8.3) ergab keine Inkonsistenzen, da in der betrachteten Anlage kein Austausch von physikalischen Größen zwischen einzelnen Modulen stattfindet. Die für die automatische Modellgenerierung notwendigen Daten und deren Abbildung im Anlagenstrukturmodell sind in Abbildung 9-4 beispielhaft für die mechanische Apparatur eines Förderbands dargestellt.

Name des InternalElements (BMK)

```
<InternalElement Name="B1-SC01-CV1" RefBaseSystemUnitPath="Vendor_V0/ACME/Conveyors/ConveyorType1_Belt" ID="{26ba2636-41ba-4249-ba50-6078276eb8d9}">
  <Attribute Name="GlobalIdentifier" Unit="" AttributeDataType="xs:integer"><Value>1</Value></Attribute>
  <Attribute Name="DefaultConvDirection" Unit="" AttributeDataType="xs:Bool"><Value>true</Value></Attribute>
  <Attribute Name="DiameterWheels" Unit="" AttributeDataType="xs:float"><Value>0.03</Value></Attribute>
  <Attribute Name="0173-1#02-AA0991#001 - Transmission ratio of the transmission" Unit="" AttributeDataType="xs:float"><Value>25</Value></Attribute>
  <Attribute Name="InertiaConv" Unit="" AttributeDataType="xs:float"><Value>5</Value></Attribute>
  <Attribute Name="StaticFrictionCoeff" Unit="" AttributeDataType="xs:float"><Value>5</Value></Attribute>
  <Attribute Name="0173-1#02-BAI087#004_Length" Unit="" AttributeDataType="xs:float"><Value>0.125</Value></Attribute>
  <ExternalInterface Name="MaterialIn" RefBaseClassPath="ACME_InterfaceClassLib/ProductFlow/ProductOut" ID="d9adec65-e528-445e-a67e-8df2b4bd02a2" />
  <ExternalInterface Name="MaterialOut" RefBaseClassPath="ACME_InterfaceClassLib/ProductFlow/ProductIn" ID="27406754-068e-49a2-b300-e37621f67f61" />
  <ExternalInterface Name="PPR_Interface" RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/PPRConnector" ID="e57fa300-1fc7-450b-8ab3-2531a56ee903" />
  <ExternalInterface Name="ConvSensorEnd" RefBaseClassPath="ACME_InterfaceClassLib/DataInterfaces/PLCInterfaceInputBool" ID="9de1cf37-c0e6-43c7-bb75-b95728ef4da4" />
  <ExternalInterface Name="MechanicalAttachment" RefBaseClassPath="ACME_InterfaceClassLib/Mechanical/MechanicalFlangeA" ID="488e640b-396b-4bbd-94be-2f5c8d777142" />
  <SupportedRoleClass RefRoleClassPath="eClassRoleClassLib/Machine_Apparatus/Material_Handling/Continuous_Handling/Conveying_Belt">
    <MappingObject>
      <AttributeNameMapping SystemUnitAttributeName="id" RoleAttributeName="GlobalIdentifier" />
      <AttributeNameMapping SystemUnitAttributeName="direction" RoleAttributeName="DefaultConvDirection" />
      <AttributeNameMapping SystemUnitAttributeName="d_impuls" RoleAttributeName="DiameterWheels" />
      <AttributeNameMapping SystemUnitAttributeName="ratio" RoleAttributeName="0173-1#02-AA0991#001 - Transmission ratio of the" />
      <AttributeNameMapping SystemUnitAttributeName="J" RoleAttributeName="InertiaConv" />
      <AttributeNameMapping SystemUnitAttributeName="mue" RoleAttributeName="StaticFrictionCoeff" />
      <AttributeNameMapping SystemUnitAttributeName="s_max" RoleAttributeName="0173-1#02-BAI087#004_Length" />
      <InterfaceNameMapping SystemUnitInterfaceName="idOutput" RoleInterfaceName="MaterialOut" />
      <InterfaceNameMapping SystemUnitInterfaceName="idInput" RoleInterfaceName="MaterialIn" />
      <InterfaceNameMapping SystemUnitInterfaceName="PPR_Interface" RoleInterfaceName="PiecesBus" />
      <InterfaceNameMapping SystemUnitInterfaceName="flange_a" RoleInterfaceName="MechanicalAttachment" />
      <InterfaceNameMapping SystemUnitInterfaceName="sensor_end" RoleInterfaceName="ConvSensorEnd" />
    </MappingObject>
  </SupportedRoleClass>
</InternalElement>
```

Parameter des realen Förderbands

Schnittstellen

Simulations-SUC (Parameter, die von der Simulation benötigt werden)

Hersteller-SUC (Parameter, die der Hersteller zur Verfügung stellt)

Simulations-SUC (Schnittstellen, die in der Simulation vorhanden sind)

Hersteller-SUC (Schnittstellen, die in der Anlage vorhanden sind)

eClass-Referenzierung, Mapping der Parameter und Schnittstellen

Abbildung 9-4: Abbildung in Anlagenstrukturmodell

Abbildung 9-5 zeigt einen Ausschnitt des aus der Anlagenstruktur generierten Modelica-Simulationsmodells nach dem Durchlauf des in Abschnitt 8.4.2 vorgestellten Modellgenerierungsalgorithmus. Erkennbar sind die einzelnen, aus dem Anlagenstrukturmodell übernommenen Parameter gemäß der vordefinierten und im Anlagenstrukturmodell hinterlegten Mappingtabelle, der im Anlagenstrukturmodell hinterlegten Parameterwerte realer Objekte sowie die Abbildung der Schnittstellen des Förderbands gemäß der im Anlagenstrukturmodell hinterlegten Schnittstellen (→ Abschnitt 6.1.5 Abbildung 6-7). Für eine grafische Darstellung des Gesamtmodells sei auf Anhang 6 verwiesen.

```
model HSU_HR_FTFT_Testfall_1
  Conveyors.Conveying_Belt B1_SC01_CV1
  ...
equation
  connect(Source.idOutput, B1_SC01_CV1.idInput)
  connect(B1_SC01_CV1.idOutput, B1_SC02_CV1.idInput)
  connect(B1_SC01_CV1.PPR_Interface, Rectangular_Pieces.PPR_Interface)
  connect(B1_SC01_CV1.flange_a, B1_SC01_A01.flange_b)
  connect(B1_SC01_CV1.sensor_end, B1_SC01_S01.sensor_attachment)
  ...
end HSU_HR_FTFT_Testfall_1;
```

Name des InternalElements (BMK)

Simulationsmodellerkennung mittels eClass-Referenzierung

Übernommene Parameterwerte des realen Förderbands

Mapping der Parameter und Schnittstellen

Schnittstellen

Abbildung 9-5: Teilausschnitt des generierten Modelica Simulationsmodells (vgl. Abbildung 9-4 dazu)

9.1.5 Implementierung des generierten Simulationsmodells in die HIL-Plattform und Testdurchführung

9.1.5.1 Generelles Vorgehen und HIL-Konfiguration

Das generierte Anlagensimulationsmodell wurde mittels der in Abschnitt 5.3.3 beschriebenen HIL-Konfiguration validiert. Im Vordergrund der durchgeführten Validierung stand zunächst nicht die Überprüfung des Steuerungscode, sondern die Validierung des automatisch generierten Simulationsmodells hinsichtlich seiner Fähigkeit, für den Test von Steuerungscode eingesetzt werden zu können. Die Validierung des programmierten Steuerungscode erfolgte daher zunächst „klassisch“ an der realen Anlage. Möglicherweise auftretende Probleme bei der Inbetriebnahme der HIL-Plattform sollten so schneller eingegrenzt werden können, Probleme im Steuerungscode möglichst ausgeschlossen sein. Der vorab getestete Steuerungscode wurde anschließend genutzt, um das in die HIL-Plattform integrierte Anlagensimulationsmodell in verschiedenen Testsituationen zu nutzen.

Der SPS-Code wurde analog zu in der Praxis üblichen Lösungen aufgebaut und hat sich in dieser Form bereits in vielen Anwendungen in der industriellen Praxis bewährt. Die eingesetzte SPS-Standard-Software folgt dem allgemein bekannten *Zone/Belegt* Prinzip, wobei *Zone* als auszuführende Prozessoperation (Übergabe eines Stückguts von einem Förderband zum nächsten) definiert ist. Ein Förderband ist solange belegt, wie es die Zonenmeldung für den Weitertransport bekommt. Die Zonenmeldung entsteht immer dann, wenn ein nachfolgendes Förderband nicht belegt ist und die Betriebsart „Automatik“ aktiv ist.

Für die Visualisierung und Bedienung des Prozesses kam eine in C# programmierte, proprietäre Visualisierungslösung zum Einsatz (HMI). Diese ist in der Lage, beliebige Signale aus der SPS auszulesen bzw. manipulieren zu können. Die Kopplung an die SPS erfolgte unter Nutzung von [LIBNODEAVE[®]]. Die Visualisierungslösung ist in Anhang 7 abgebildet.

9.1.5.2 Testdurchführung

Im Rahmen eines Gut-Prozessablaufs wurden in einem ersten Schritt an der realen Anlage die PROFINET IO Kommunikationstelegramme zwischen SPS und dezentraler Peripherie mit dem Werkzeug Wireshark [WIRESHARK[®]] aufgezeichnet [SCFE07]. Aus den aufgenommenen Daten lässt sich der genaue Verlauf aller gesetzten und aufgenommenen Signale durch die SPS extrahieren. In einem zweiten Schritt wurde das SPS-Programm gegen das automatisch generierte Simulationsmodell der Anlage in der HIL-Experimentierumgebung unter Verwendung von PROFINET IO mit emulierter dezentraler Peripherie getestet. Das SPS-Programm wie auch die Hardwarekonfiguration der SPS blieben für beide Testfälle unverändert. Der Signalverlauf wurde (analog zur Aufnahme der realen Messwerte) aus dem Telegrammverkehr SPS ↔ HIL-Experimentierumgebung mittels Wireshark extrahiert, um eine Gegenüberstellung des Signalverlaufs von realer zu simulierter Anlage zu ermöglichen.

9.1.5.3 Aufgenommene Messwerte und Interpretation

Die genutzten Messpunkte an der Anlage sind in Abbildung 9-6 dargestellt und stehen hier stellvertretend für die aufgenommenen Daten des Prozessabbilds.

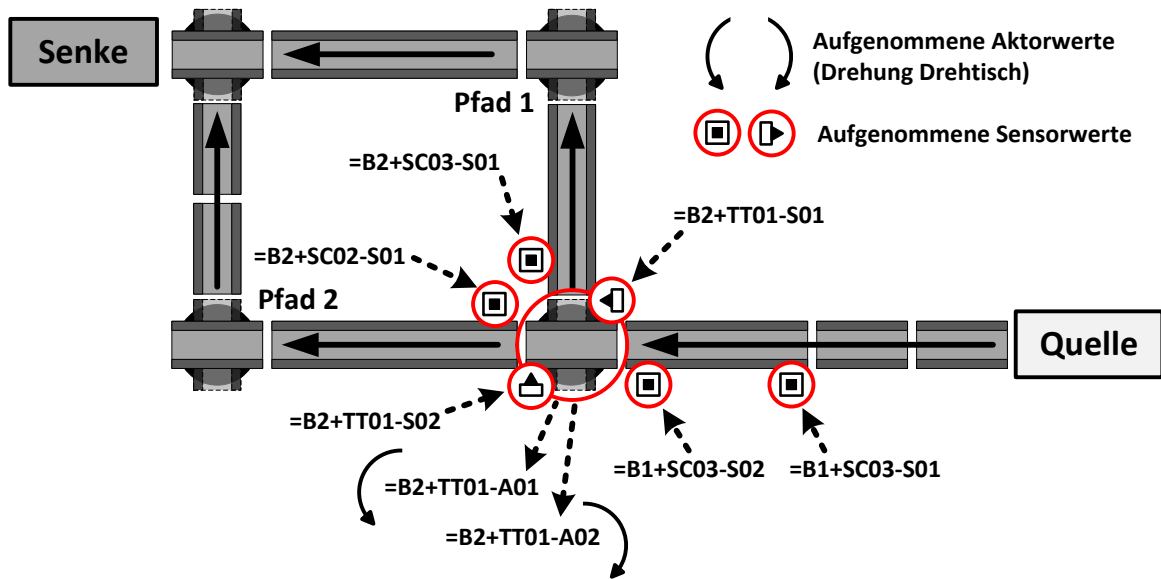
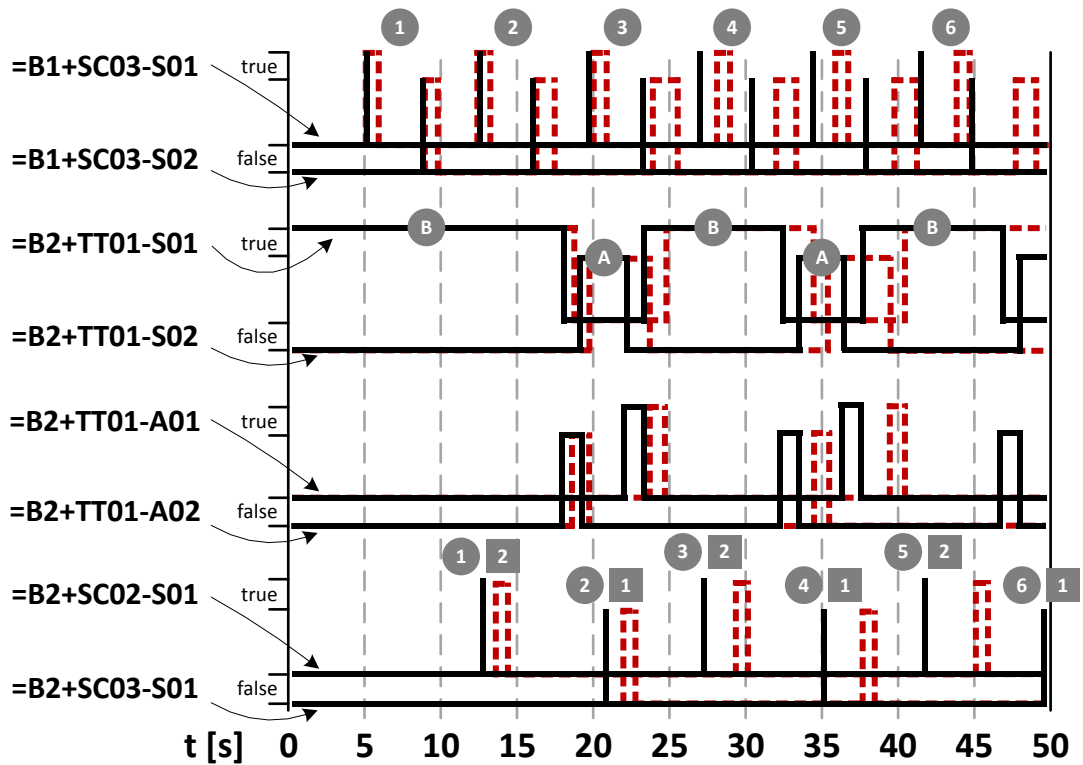


Abbildung 9-6: Messpunkte

Der Signalverlauf ist in Abbildung 9-7 für die ersten 50 Sekunden des durchgeführten Prozesslaufs dargestellt [KPR⁺14[#]]. Die schwarzen, durchgehenden Linien kennzeichnen die an der realen Anlage aufgenommenen Signalwerte, die Signalverläufe an der HIL-Simulation sind rot gestrichelt dargestellt.



Legende:
 Stückgut n auf Pfad 1: n 1 Drehtisch Position Pfad 1: A
 Stückgüter: 1 ... 6 Stückgut n auf Pfad 2: n 2 Drehtisch Position Pfad 2: B

Abbildung 9-7: Aufgenommene Messwerte im Vergleich [KPR⁺14[#]]

9.1.5.4 Interpretation der aufgenommenen Messwerte

Ein Vergleich der aufgenommenen Messwerte zeigt, dass eine Übereinstimmung der Signalverläufe der realen Anlage und der HIL-Simulation hinsichtlich der *Richtigkeit* des Signalverlaufs besteht. So ist z. B. die alternierende Richtungszuweisung nach dem Dispositionspunkt mit den dazugehörigen Abläufen gut zu erkennen. Unterschiede sind insbesondere im zeitlichen Verlauf erkennbar, z. B. bei der Dauer der Einschaltimpulse. Die dort auftretenden Abweichungen können mit Ungenauigkeiten der Parametrierung des Simulationsmodells erklärt werden. Im Konkreten liegen die Abweichungen im Vergleich zur Realität an abweichenden ...

- ... Fördergeschwindigkeiten
- ... Längenangaben der Förderbänder
- ... Positionen der Sensoren an den Förderbändern
- ... Schaltabständen der Sensoren (= Bereich, in dem ein Sensor das Stückgut erkennt)

Die hierfür notwendigen Parameter wurden manuell an der realen Anlage aufgenommen und unterliegen dadurch naturgemäß Ungenauigkeiten. Bestimmte Parameter wie z. B. die Reibung des Förderbands (→ Abschnitt 7.2.3) waren zudem nicht verfügbar oder messbar und mussten daher geschätzt werden. Die Abweichungen könnten durch Anpassung der Parametrierung zwar deutlich reduziert werden, allerdings entsprechen die hier zu beobachtenden Parameterungenauigkeiten durchaus den bei einer tatsächlichen virtuellen Inbetriebnahme zu erwartenden Verhältnissen. Die exakten Parameter der realen Anlage sind häufig nur mit begrenzter Genauigkeit und auch Gültigkeit in den verfügbaren Anlagenplanungsdaten und damit der als Basis der Modellgenerierung genutzten Anlagenstruktur dargestellt. In diesem Kontext ist zu beachten, dass in einer virtuellen Inbetriebnahme nicht auf die finalen Anlagenplanungsunterlagen zurückgegriffen wird (vgl. Abschnitt 2.2 und 2.5.1); vielmehr basiert eine virtuelle Inbetriebnahme auf den in den Anlagenplanungsdaten gemachten Annahmen, wie eine Anlage errichtet werden soll (→ Abschnitt 2.5.1 „Freigabelayout“, → Abschnitt 2.5.2 „Elektroplan-Baustellenstand“).

Die auftretenden Zeitverzögerungen sind kumulativ, d. h. mit fortschreitender Simulationsdauer entfernt sich der Signalverlauf zeitlich immer weiter von dem der realen Anlage (vgl. dazu beispielsweise die Messwerte der realen Anlage und der HIL-Simulation für die Sensoren =B1+SC03-S01 und -S02). Dies lässt sich wie folgt erklären: Der Start eines Stückguts an der Quelle wird mittels eines SPS-Ausgangssignals getriggert, sobald der erste Platz frei (d. h. nicht belegt) ist. Dies entspricht in der Praxis (vereinfacht) der üblichen Handhabung von Materialtransporten zwischen Anlagen. Wird nun die Freigabe (Zonenmeldung) aufgrund der beschriebenen Ungenauigkeiten in der Parametrierung später erteilt, verzögern sich kumulativ alle weiteren Freigaben. Die auftretenden Zeitverzögerungen beeinflussen die Testdurchführung im Sinne der Überprüfung der programmierten Ablauflogik nicht, können aber bei der Überprüfung von programmierten Zeitfunktionen zu Problemen führen (z. B. Einschalt- sowie Ausschaltverzögerungen, Impulse, Laufzeitüberwachungen, etc.). Ggf. sind die im Steuerungscode als Parameter hinterlegten Zeitwerte der Zeitfunktionen für die Durchführung einer virtuellen Inbetriebnahme anzupassen, sollte ungewolltes Verhalten (insbesondere ungewollt auflaufende Störmeldungen) während des Testens resultieren.

Der Test wurde in einem zweiten Durchlauf mit einem Anlagensimulationsmodell in heterogenem Detaillierungsgrad wiederholt. Zu diesem Zweck wurden zwei Förderband-Module in makroskopischem Detaillierungsgrad abgebildet. Der Testlauf erfolgte wie bereits beschrieben. Die resultierenden Messwerte zeigten hinsichtlich der Richtigkeit des Signalverlaufs keinen wesentlichen

Unterschied im Vergleich zu den in Abbildung 9-7 dargestellten, weswegen auf eine gesonderte Darstellung verzichtet wird. Eine Verbesserung des Signalverlaufs im Sinne der Vermeidung der auftretenden kumulativen Zeitverzögerungen konnte nicht beobachtet werden. Dies ist nur dann zu erwarten, wenn das Anlagenmodell gesamtheitlich im makroskopischen Detaillierungsgrad simuliert wird und die parametrisierten Totzeiten sehr genau mit den tatsächlichen Zeiten der einzelnen durch die Förderbänder und Drehtische durchgeführten Prozessoperationen übereinstimmt.

Aus den aufgenommenen Messwerten lässt sich schließen, dass der Ansatz im Rahmen des Einsatzzwecks virtuelle Inbetriebnahme dazu geeignet ist, den Steuerungscode durch zu definierende Testfälle zu validieren.

9.2 Fallstudie Schwenk-EHB

Die zweite Fallstudie zeigt die Anwendung der in den Abschnitten 8.1 und 8.2 beschriebenen Methode zur Identifizierung eines für die virtuelle Inbetriebnahme notwendigen Detaillierungsgrads einer Simulation (Baustein 3.1). Für die Fallstudie wird eine Elektro-Hängebahn beispielhaft untersucht. Das vorgestellte System lehnt sich eng an die in [VDI 4441-1] beschriebenen Anforderungen und Gestaltungsmerkmale derartiger Systeme an.

EHB-Systeme sind in der Regel keine Einzelgeräte, sondern setzen sich aus vielen einzelnen Fahrzeugen zusammen, die auf einem verzweigten Bahnsystem operieren [VDI 4441-1]. Im Folgenden werden einzelne Fahrzeuge auch als Gehänge bezeichnet. Abbildung 9-8 zeigt typische Anwendungen von EHB-Systemen in der Automobil-Endmontage.

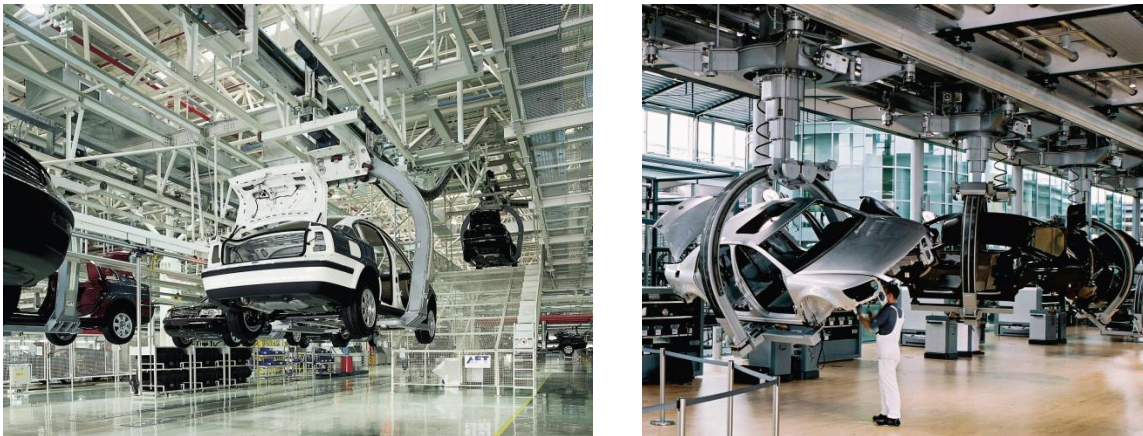


Abbildung 9-8: Typische Anwendungen von Elektro-Hebebühnen im Automobilbau (Bildübernahme aus [VDI 4441-1])

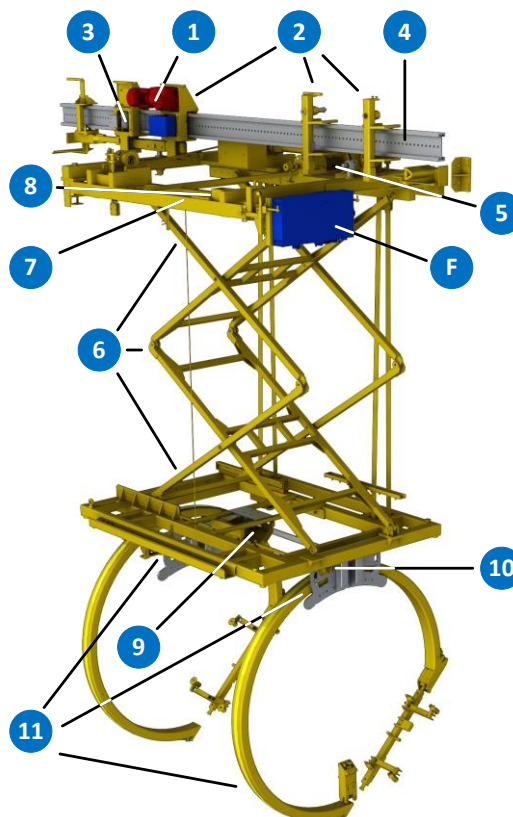
9.2.1 Systembeschreibung

Im Fokus der hier stattfindenden Untersuchung steht die Applikation einer Schwenk-EHB, wie sie in der Automobil-Endmontage zum Einsatz kommen kann (→ Abbildung 9-8 rechts). Das hier betrachtete EHB-System basiert auf dem System KB 240 der Firma ROFA INDUSTRIAL AUTOMATION AG [ROFA®]. Das System kommt insbesondere in der Automobilindustrie zum Ein-

satz und kann in guter Näherung als ein Best-Practice gesehen werden. Es enthält pro Gehänge eine Hub- und Fahrvorrichtung sowie (als wesentliches, charakteristisches Merkmal) eine Schwenkvorrichtung. Die Umsetzung des hier beschriebenen Systems ist als stark vereinfachte Betrachtung von konkret in der Praxis eingesetzten EHB-Systemen zu sehen. Die hier im Fokus stehenden Gehänge unterteilen sich in insgesamt 3 Module, die wie folgt beschrieben werden können:

- **Modul 1: Fahrvorrichtung.** Das Modul besteht aus einem mittels Frequenzumrichter (F) gesteuertem Fahrmotor (1) und mechanischer Aufhängung an Schiene (2). Zur Aufnahme der Position kommt ein Absolutwertgeber (optisch) zum Einsatz (3) (Barcode-Reader). Hierfür sind an der Fahrschiene Barcodes (4) angebracht.
- **Modul 2: Hubvorrichtung.** Das Modul besteht aus einem mittels Frequenzumrichter (F) gesteuerten Motor (5), der mechanischen Hubvorrichtung (6) und multiplen Sensoren: Einem Absolutwertgeber (Seilzuggeber) für die Erfassung der Hubposition (7) sowie einem Sensor zur Erkennung von Hubbereichen, die Aufgrund von Personenschutzgründen nicht untersenkt werden dürfen (8).
- **Modul 3: Schwenkvorrichtung.** Die Schwenkvorrichtung besteht aus einem Motor (Ansteuerung über Wende-Schützschiene) (9), einem Drehgeber zur Erfassung der Drehposition (10) und dem eigentlichen Drehmechanismus (11). Motor und Drehmechanismus sind über einen Riemenantrieb miteinander verbunden (10).

Abbildung 9-9 zeigt die hier beschriebenen Module in einer Gesamtübersicht. Die einzelnen Komponenten sind mit (F) sowie (1)-(11) durchnummeriert.



© ROFA INDUSTRIAL AUTOMATION AG, Kolbermoor, www.rofa-group.com

Abbildung 9-9: EHB Typ KB 240 mit Hub-, Fahr- und Schwenkvorrichtung (Bildübernahme mit freundlicher Genehmigung der ROFA INDUSTRIAL AUTOMATION AG [ROFA®])

Mehrere dieser Gehänge kommen in einem einfachen Rundkurs zum Einsatz, wobei die genaue Anzahl der Gehänge hier keine Rolle spielt. Der Rundkurs unterteilt sich in insgesamt 4 Abschnitte, wie in Abbildung 9-10 dargestellt.

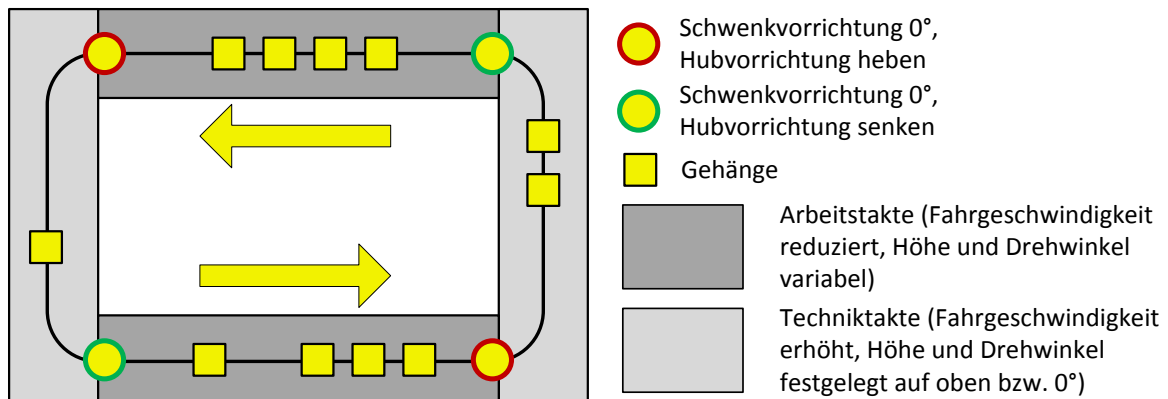


Abbildung 9-10: Gesamtanlagenübersicht

Als zu beachtende Verhaltenskenngrößen der Gehänge werden in diesem Beispiel festgelegt:

- Maximale Geschwindigkeit Fahrvorrichtung: $0,1 \text{ m/s}$ in Arbeitstakten, $0,2 \text{ m/s}$ in Techniktakten (Techniktakte sind Takte, in denen das Fahrzeug keinen Kontakt zu Werkern hat)
- Maximale Geschwindigkeit Hubvorrichtung: $0,1 \text{ m/s}$ in Arbeitstakten (nicht relevant für Techniktakte, da Gehänge stets in oberer Position)
- Maximale Beschleunigung Hubvorrichtung: $0,1 \text{ m/s}^2$
- Zu transportierende/drehende Masse: $550 - 850 \text{ kg}$
- Eigenmasse des Gehänges: 1200 kg , davon 200 kg Drehmechanismus
- Geforderte Positioniergenauigkeit der Gehänge bei Fahren auf Schiene (Bewegung in x-Richtung): kann vernachlässigt werden
- Geforderte Positioniergenauigkeit der Gehänge bei Höhenänderung (Bewegung in y-Richtung): $\pm 1 \text{ mm}$
- Winkelgeschwindigkeit beim Drehen: $0,07 \text{ rad/s}$ ($\approx 4^\circ/\text{s}$)
- Geforderte Positioniergenauigkeit des Drehens: $\pm 1^\circ$

Nicht betrachtet werden rein mechanische Vorrichtungen wie Führungsschienen, Verriegelungen des transportierten PKWs in der Schwenkvorrichtung sowie der Aufbau der Automatisierungslösung in Bezug auf die elektrotechnischen Umfänge. Zudem werden Aufnahme und Abgabe der transportierten/gehandhabten PKWs hier nicht betrachtet.

9.2.2 Ermittlung des notwendigen Detaillierungsgrads

Die Ermittlung des notwendigen Detaillierungsgrads bei der Simulation eines Gehänges wird anhand von **Modul 1: Fahrvorrichtung** sowie **Modul 3: Schwenkvorrichtung** erläutert.

9.2.2.1 Untersuchung des Moduls „Fahrvorrichtung“

Ergebnisse der Ermittlung von Zuordnungen Prozessschritt – Komponente:

- **Beschreibung:** Positionieren des Gehänges in Fahrrichtung

- **Prozessschritte im Steuerungscode:** Motoransteuerung Vorwärtslauf, Absolutwertgeber (optisch) erfasst genaue Position des Gehänges auf Schiene mittels Barcode, Fahren auf Schiene
- **Komponenten:** Motor mit Frequenzumrichter (1), Absolutwertgeber (optisch) (2), mechanische Aufhängung des Gehänges an Schiene (3)

Ergebnisse der Auswertung der Verhaltenskenngrößen:

- **Aus Anlagenplanungsdaten ermittelte Verhaltenskenngrößen:**
 - Geschwindigkeit Fahrvorrichtung: 0,1 m/s in Arbeitstakten, 0,2 m/s in Technikstakten
 - Transportierte Masse: 550 – 850 kg
 - Eigenmasse des Gehänges: 1200 kg
 - Positioniergenauigkeit der Gehänge in x Richtung: kann vernachlässigt werden
 - Umsetzung der Komponente (1) und (3) gemäß Best-Practices des Bauherren/des Gewerks Mechanik, Komponente (2) ist nicht in Bauherren-Freigabeliste enthalten
- **Standard-SPS-Bausteine/Standard-Parametersatz:** Bausteine für Motoransteuerung verfügbar und genutzt, Standard-Parametersatz verfügbar, für Absolutwertgeber (optisch) beides nicht verfügbar

Das Modul Fahrvorrichtung sorgt für die Bewegung des Gehänges in Fahrrichtung. Bei der Ermittlung des notwendigen Detaillierungsgrads kann unter Beachtung der relevanten Verhaltenskenngrößen identifiziert werden, dass das Fahrmodul teilweise aus Standardkomponenten aufgebaut ist. Dies trifft auf den Motor mit Frequenzumrichter zu. Stehen zudem Standard-SPS-Bausteine zur Verfügung und können Standard-Parameter genutzt werden, reicht die Simulation im niedrigsten Detaillierungsgrad aus. Physikalische Effekte wie Trägheit oder Reibung können vernachlässigt werden, da keine genaue Positioniergenauigkeit gefordert ist. Eine Ausnahme bildet der Absolutwertgeber (optisch), der nicht in der bauherrenspezifischen Freigabeliste zu finden ist und daher kein Best-Practice darstellt. Es kann daher nicht von einem „Normalfall“ (= durchgehende Verwendung von Standardkomponenten, Abschnitt 7.2.2) ausgegangen werden. Im Konkreten muss die Simulation in der Lage sein, die vom Absolutwertgeber (optisch) mittels Barcode aufgenommenen Positionswerte genau wiedergeben zu können. Dies kann ein Hinweis auf eine detaillierte Simulation sein. Es ergeben sich folgende Regeln:

Komponente Motor (1) unter Beachtung des Prozessschritts „Motoransteuerung Vorwärtslauf“:

WENN Best-Practices genutzt werden

UND Standard-SPS-Bausteine genutzt werden

UND Standard-Parameter genutzt werden können

DANN nutze den makroskopischen Detaillierungsgrad (Gewichtung = 1)

Komponente Absolutwertgeber (optisch) (2) unter Beachtung des Prozessschritts „erfasst genaue Position des Gehänges auf Schiene mittels Barcode“:

WENN keine Best-Practices genutzt werden

UND die Position eines Gehänges sehr genau erfasst werden muss

UND keine Standard-SPS-Bausteine genutzt werden/verfügbar sind

DANN nutze den mesoskopisch/nicht dynamischen Detaillierungsgrad (Gewichtung = 2)

Komponente mechanische Aufhängung des Gehänges (3) unter Beachtung des Prozessschritts „Fahren auf Schiene“:

WENN Best-Practices genutzt werden

DANN nutze den makroskopischen Detaillierungsgrad (Gewichtung = 1)

Der für die Simulation der Komponenten Motor (1), Absolutwertgeber (optisch) (2) und mechanische Aufhängung des Gehänges (3) notwendige Detaillierungsgrad ist daher der mesoskopisch/nicht dynamische Detaillierungsgrad ($\max(1, 2, 1) = 2$).

9.2.2.2 Betrachtung des Moduls „Schwenkvorrichtung“

Ergebnisse der Ermittlung von Zuordnungen Prozessschritt – Komponente:

- **Beschreibung:** Drehen des Gehänges um x-Achse (x-Achse = Achse in Fahrrichtung)
- **Prozessschritte im Steuerungscode:** Motoransteuerung Links-/Rechtslauf, Drehgeber erfasst genaue Position des Drehmechanismus, Drehen der Karosse
- **Komponenten:** Motor mit Wende-Schützschialtung (1), Drehgeber (2), Drehmechanismus (3)

Ergebnisse der Auswertung der Verhaltenskenngrößen:

- **Aus Anlagenplanungsdaten ermittelte Verhaltenskenngrößen:**
 - Zu drehende Masse: 550 – 850 kg (Änderungen während des Durchfahrens)
 - Eigenmasse des Drehmechanismus: 200 kg
 - Winkelgeschwindigkeit beim Drehen: 0,07 rad/s ($\approx 4^\circ/s$)
 - Geforderte Positioniergenauigkeit des Drehens: $\pm 1^\circ$
 - Umsetzung der Komponente (1) und (3) gemäß Best-Practices des Gewerks Mechanik, Komponente (2) ist in Bauherren-Freigabeliste enthalten
- **Standard-SPS-Bausteine/Standard-Parametersatz:** Bausteine für Motoransteuerung nicht verfügbar (obsolet, da Wende-Schützschialtung und somit nur das Ansteuern von Ausgängen notwendig ist, zu beachtende/einzustellende Parameter gibt es nicht), für Drehgeber nicht verfügbar

Die Drehfunktion ist eine Sonderfunktion und kann als das wesentliche, charakteristische Merkmal der hier betrachteten Gehänge beschrieben werden. Die konkrete Umsetzung weist einige Besonderheiten auf. So kommt kein mittels Frequenzumrichter geregelter Antrieb zum Einsatz, der sich für genaues Positionieren eignet, sondern eine einfache Lösung mit Wende-Schützschialtung. Diese ist naturgemäß nicht für eine Genauhaltfunktion geeignet. Unabhängig davon muss gemäß der definierten Verhaltenskenngrößen die Drehung mit einer Toleranz von $\pm 1^\circ$ erfolgen. Wird nun von einer geforderten Winkelgeschwindigkeit von $\omega = 0,07 \text{ rad/s}$ ausgegangen und eine Reaktionszeit bis zum vollständigen Stopp der Drehfunktion von geschätzten 500 ms zugrunde gelegt (summiert aus SPS-Zykluszeit, Reaktionszeit der Schütze, Zeit bis Einfall der Bremse am Elektromotor sowie hauptsächlich dem Schlupf, verursacht durch das wirkende Trägheitsmoment des transportierten Fahrzeugs bei der Drehung, und der daraus resultierenden Zeitverzögerung bis zum Halt) so entspricht dies einer systemimmanenten Mindesttoleranz der Drehfunktion von $0,5 \text{ s} * 0,07 \text{ rad/s} = 0,035 \text{ rad}$. Dies entspricht einem Winkel von $0,035 \text{ rad} * 180/\pi \approx 2^\circ$. Um die geforderte Toleranz von $\pm 1^\circ$ einzuhalten, ist eine speziell für diesen Zweck umgesetzte Funktion im Steuerungscode notwendig. Dies kann beispielsweise

durch die Verrechnung des Zielwinkels mit einem Vorhaltewinkel umgesetzt werden. Ist ein Winkel von 90° anzufahren, würde beispielsweise bereits bei 88° die Motorabschaltung erfolgen, so dass in den Zielbereich unter Einhaltung der geforderten Toleranz „gerutscht“ wird.

Bei genauerer Betrachtung ist die Bestimmung des Vorhaltewinkels auf der einen Seite von statischen Größen wie Zykluszeiten der SPS oder Reaktionszeit der Schütze abhängig. Auf der anderen Seite aber auch vom Trägheitsmoment (und dem daraus resultierenden Schlupf), welches von der Masse des sich drehenden Autos und der Drehvorrichtung bei negativer Beschleunigung (= Bremsen) erzeugt wird. In diesem Kontext ist zu beachten, dass sich die Masse des Fahrzeugs bei Durchfahren der Arbeitstakte ändern kann. Daraus lässt sich folgern, dass die zu drehende Masse einen Einfluss auf die Genauigkeit der Drehung hat. Der Einfluss physikalischer Größen bei der Drehbewegung sollte daher für die Simulation beachtet werden. Unter diesen Bedingungen lassen sich folgende Regeln ableiten:

Komponente Motor (1) unter Beachtung des Prozessschritts „Motoransteuerung Links-/Rechtslauf“:

***WENN** Best-Practices genutzt werden*

***UND** Standard Bausteine zur Verfügung stehen (hier angenommen, da Standard-SPS-Bausteine in dem hier beschriebenen Fall obsolet sind)*

***UND** Standard-Parameter genutzt werden können (hier angenommen, da Standard-Parameter in dem hier beschriebenen Fall obsolet sind)*

***DANN** nutze den makroskopischen Detaillierungsgrad (Gewichtung = 1)*

Komponente Drehgeber (2) unter Beachtung des Prozessschritts „Erfasst genaue Position des Drehmechanismus“:

***WENN** die Position eines Gehänges sehr genau erfasst werden muss*

***UND** keine Standard-SPS-Bausteine genutzt werden/verfügbar sind*

***DANN** nutze den mesoskopisch/nicht dynamischen Detaillierungsgrad (Gewichtung = 2)*

Komponente Drehmechanismus (3) unter Beachtung des Prozessschritts „Drehen der Karosse“:

***WENN** Best-Practices genutzt werden*

***UND** eine hohe Positioniergenauigkeit gefordert ist*

***UND** die zu drehende Masse als hoch eingeschätzt wird*

***UND** sich die zu drehende Masse verändern kann*

***DANN** nutze den mesoskopisch/dynamischen Detaillierungsgrad (Gewichtung = 3)*

Der für die Simulation der Komponenten Motor (1), Drehgeber (2) und Drehmechanismus des Gehänges (3) notwendige Detaillierungsgrad ist daher der *mesoskopisch/dynamische Detaillierungsgrad* ($\max(1, 2, 3) = 3$).

10 Zusammenfassung und Ausblick

Die in dieser Arbeit vorgestellten Lösungen werden im Folgenden unter Beachtung der im Lösungskonzept genannten und in dieser Arbeit behandelten Leitfragen zusammengefasst:

- **Leitfrage I: Welche Daten werden für eine virtuelle Inbetriebnahme benötigt und wie sollten diese abgebildet sein?**
- **Leitfrage II: Wie kann man virtuelle Inbetriebnahme umsetzen?**
- **Leitfrage III: Wie sollte man virtuelle Inbetriebnahme umsetzen?**
- **Leitfrage IV: Wie kann man effizient Simulationsmodelle erstellen?**

Darauf aufbauend werden in Abschnitt 10.2 Weiterentwicklungsmöglichkeiten des hier vorgestellten Ansatzes aufgezeigt.

10.1 Zusammenfassung

10.1.1 Leitfrage I: Welche Daten werden für eine virtuelle Inbetriebnahme benötigt und wie sollten diese abgebildet sein?

Bei der Durchführung einer virtuellen Inbetriebnahme ist die Anlagenstruktur als substanzielle Datenquelle für die Identifikation von Simulationsmodellen zu nutzen [VDI 3693-1]. Wie in dieser Arbeit gezeigt, kann das *Datenmodell der Digitalen Fabrik* und das in der Digitalen Fabrik definierte *Produkt/Prozess/Ressource-Prinzip* einen wesentlichen Beitrag dazu leisten, die für eine virtuelle Inbetriebnahme notwendigen Informationen in einem Anlagenstrukturmodell bereitzustellen – was alleine aber nicht ausreichend ist. Im Konkreten mangelt es an Abbildungsvorschriften und Definitionen zur strukturell-systematischen Abbildung von Anlagenplanungsdaten. Die in dieser Arbeit erarbeiteten Erweiterungen und Konkretisierungen des *Datenmodells der Digitalen Fabrik* zeigen, wie Anlagenstrukturen hinterlegt werden können, um für Anwendungsfälle wie die virtuelle Inbetriebnahme nutzbar zu sein. Darüber hinausgehend konnte gezeigt werden, dass eine standardisierte Darstellung einer Anlagenstruktur unter Nutzung eines im Engineering verbreiteten Beschreibungsmittels (hier: AutomationML) dazu genutzt werden kann, als substanzielle Datenquelle für eine automatische Modellerstellung zu dienen.

Der im Kontext automatische Modellgenerierung zu beachtende, wesentliche Aspekt der Modellerstellung besteht grundsätzlich aus der Transformation der Anlagenstruktur und in ihr enthaltener Objekte in ein Anlagensimulationsmodell. Mittels der Einbindung von *eCI@ss* in das *Datenmodell der Digitalen Fabrik* ist es gelungen, diese Transformation zu ermöglichen. Die entwickelte *eCI@ss*-Rollenbibliothek ermöglicht durch die konkrete Rollenzuweisung einzelner Objekte in Form eines globalen Identifiers die Verknüpfung von bestehendem Anlagenstrukturmodell und neu entstehendem Simulationsmodell. Die in *eCI@ss* enthaltenen Merkmalleisten konnten ferner dazu genutzt werden, mittels Mapping Parameter realer Objekte auf Simulationsmodelle zu über-

tragen. Dies gilt auch für die im Anlagenstrukturmodell abgebildeten Schnittstellen zwischen einzelnen Objekten.

10.1.2 Leitfrage II: Wie kann man virtuelle Inbetriebnahme umsetzen?

Simulationsmodelle bilden den Kern einer jeden virtuellen Inbetriebnahme. In dieser Arbeit wurde eine Systematik vorgestellt, mit der sich ein heterogener Detaillierungsgrad für Anlagensimulationsmodelle implementieren lässt. Hierzu wurden verschiedene Modellarten entwickelt, die mit aufsteigendem Detaillierungsgrad Prozesse innerhalb einer Anlage immer genauer darstellen. Die erarbeiteten Modellarten wurden in einer prototypischen Simulationsmodellbibliothek umgesetzt und erfolgreich auf ihre Einsetzbarkeit hin untersucht. Die erarbeiteten Modellarten können einen Beitrag dazu leisten, virtuelle Inbetriebnahme zielgerichtet durchzuführen. Die in dieser Arbeit erarbeiteten Modellarten konnten zudem in die VDI-Richtlinie [VDI 3693-1] eingebracht werden und stehen somit potenziell vor einer breiten Anwendung in der industriellen Praxis.

10.1.3 Leitfrage III: Wie sollte man virtuelle Inbetriebnahme umsetzen?

Stehen geeignete Simulationsmodelle in unterschiedlichen Detaillierungsstufen zur Verfügung, stellt sich direkt die Frage, wann diese eingesetzt werden sollten. Hierfür konnte eine den Modellerstellungsprozess unterstützende Methode entwickelt werden, die sich das in den Anlagenplanungsdaten hinterlegte Wissen in Form von Verhaltenskenngößen zu nutzen macht, um diese Frage zu beantworten. Ferner wird explizit das bei dem mit virtueller Inbetriebnahme betrauten Personal des Gewerks Automatisierungstechnik vorhandene spezifische Fachwissen mit einbezogen, um den für den jeweiligen Anwendungsfall möglichst optimalen Detaillierungsgrad einzelner Teil-Simulationsmodelle identifizieren zu können. Das entwickelte Vorgehen ist als Heuristik zu verstehen, da nicht immer davon ausgegangen werden kann, alle für die Ermittlung des notwendigen Detaillierungsgrads benötigten Informationen vollständig und konsistent vorliegen zu haben. Zudem ist die Anzahl an denkbaren Verhaltenskenngößen aufgrund des heterogenen Charakters automatisierter Anlagen als sehr groß einzuschätzen. Vielmehr soll aus den vorliegenden Anlagenplanungsdaten die für die jeweilige Situation bestmögliche Lösung ermittelt werden.

Bei der Wahl des Detaillierungsgrads muss ferner sichergestellt sein, dass die erstellten, in heterogenem Detaillierungsgrad abgebildeten Anlagensimulationsmodelle konsistent hinsichtlich Ihrer Schnittstellen sind. Hierfür wurde eine Methode entwickelt, mit der mögliche Inkonsistenzen im Anlagensimulationsmodell zunächst regelbasiert identifiziert werden können und dann, in einem zweiten Schritt, durch das Ergänzen von Kopplungs- oder Terminierungselementen, die in einer Simulationsmodellbibliothek vorgehalten werden müssen, behoben werden können.

10.1.4 Leitfrage IV: Wie kann man effizient Simulationsmodelle erstellen?

Eine effiziente Durchführung einer virtuellen Inbetriebnahme bedeutet eine Minderung des zur Durchführung benötigten Aufwands – insbesondere hinsichtlich der Modellerstellung, die einen wesentlichen Anteil am Gesamtaufwand einer virtuellen Inbetriebnahme hat. In dieser Arbeit konnte gezeigt werden, dass eine automatische Modellgenerierung für den fertigungstechnischen Anlagenbau möglich und auch realistisch umsetzbar ist. Eine automatische Modellgenerierung ist jedoch nur dann möglich, wenn die in dieser Arbeit genannten Voraussetzungen dafür ausnahmslos erfüllt sind. Hierzu zählen das Vorhandensein eines vollständigen und konsistenten Anlagen-

strukturmodells sowie das Vorhandensein einer Simulationsmodellbibliothek, die Teil-Simulationsmodelle vorzugsweise in jedem Detaillierungsgrad enthält. Ferner sind den Modellerstellungsprozess unterstützende Methoden nötig – wie die Identifikation des notwendigen Detaillierungsgrads und die Sicherstellung der Konsistenz der heterogen aufgebauten Anlagensimulationsmodelle.

Die in dieser Arbeit gezeigte automatische Modellgenerierung kann als Zusammenfassung aller beschriebenen Lösungsbausteine dieser Arbeit gesehen werden. Durch die automatische Modellgenerierung konnte der Aufwand für die Modellerstellung signifikant verringert werden. Ferner konnte der experimentelle Nachweis erbracht werden, dass ein automatisch erzeugtes Anlagensimulationsmodell dazu geeignet ist, den Steuerungscode durch zu definierende Testfälle zu validieren.

10.2 Ausblick auf potenzielle Weiterentwicklungsmöglichkeiten

10.2.1 Effizienter Datenaustausch

Um das in dieser Arbeit entwickelte Konzept für eine automatische Modellgenerierung nutzbringend einsetzbar zu machen, müssen reale Engineering-Daten in das in dieser Arbeit konkretisierte *Datenmodell der Digitalen Fabrik* überführt werden. In Abschnitt 6.1.2 wurden insgesamt drei grundlegende Fragen hinsichtlich eines durchgängigen Datenmanagements identifiziert. In dieser Arbeit konnte die Frage hinsichtlich der möglichen Darstellung einer Anlagenstruktur beantwortet werden. Es bleibt die Frage nach dem *woher* die Anlagenplanungsdaten während des Engineering-Prozesses kommen. Ferner ist die Frage nach der Verarbeitung und Aufbereitung von Engineering-Daten, um sie im objektorientierten *Datenmodell der Digitalen Fabrik* gemäß den in dieser Arbeit definierten Spezifikationen abbilden zu können, offen.

Der effiziente Austausch von Anlagenplanungsdaten zwischen den am Engineering beteiligten Gewerken Mechanik/Elektrotechnik/Automatisierungstechnik und deren spezifischen Software-Werkzeugen stellt, unter Betrachtung einer gewollt durchgängigen Fabrikplanung, einen wichtigen zu erreichenden Meilenstein dar. Methoden wie Interoperabilitätsfunktionen und Kollaborationsobjekte, die es ermöglichen, den Datenaustausch im Sinne einer durchgehenden Fabrikplanung effizient umzusetzen [DFB11], können einen signifikanten Fortschritt im automatisierten iterativen Datenaustausch bedeuten. Die Sicherstellung eines effizienten Datenaustausches unter Beachtung der in dieser Arbeit erarbeiteten Ergebnisse wird derzeit im Projekt „EISIVIO“ (ZIM Förderkennzeichen ZF4063901SS5) behandelt.

10.2.2 Abbildung von Gerätemodellen

Die für eine virtuelle Inbetriebnahme genutzten Gerätemodelle müssen initial aufwändig von Hand erstellt werden. Hinzu kommt, dass insbesondere für detaillierte Modelle das genaue Verhalten der abzubildenden Geräte sehr genau bekannt sein muss. Das entsprechende Wissen ist oftmals nur beim Gerätehersteller vorhanden, der in der Regel ein starkes Interesse am Schutz seines Know-Hows hat. Derzeit beschäftigen sich Fachgruppen, insbesondere die Arbeitsgruppe „Virtuelle Inbetriebnahme“ des VDA Arbeitskreises Digitale Fabrik [VDA AK DF[®]], mit der Schaffung einer Schnittstelle zwischen Hersteller und Anwender. Ziel ist es, das Verhalten von Geräten

in standardisierter Form werkzeugübergreifend verfügbar zu machen. Erste Ergebnisse hierzu wurden bereits in [LBO⁺15] veröffentlicht. Ferner konzipiert der von Süß et al. [SSD15] vorgestellte Ansatz (Abschnitt 4.2.1.2) einen Austausch von Gerätemodellen zwischen Hersteller und Anwender unter Verwendung des Functional-Mockup-Interfaces.

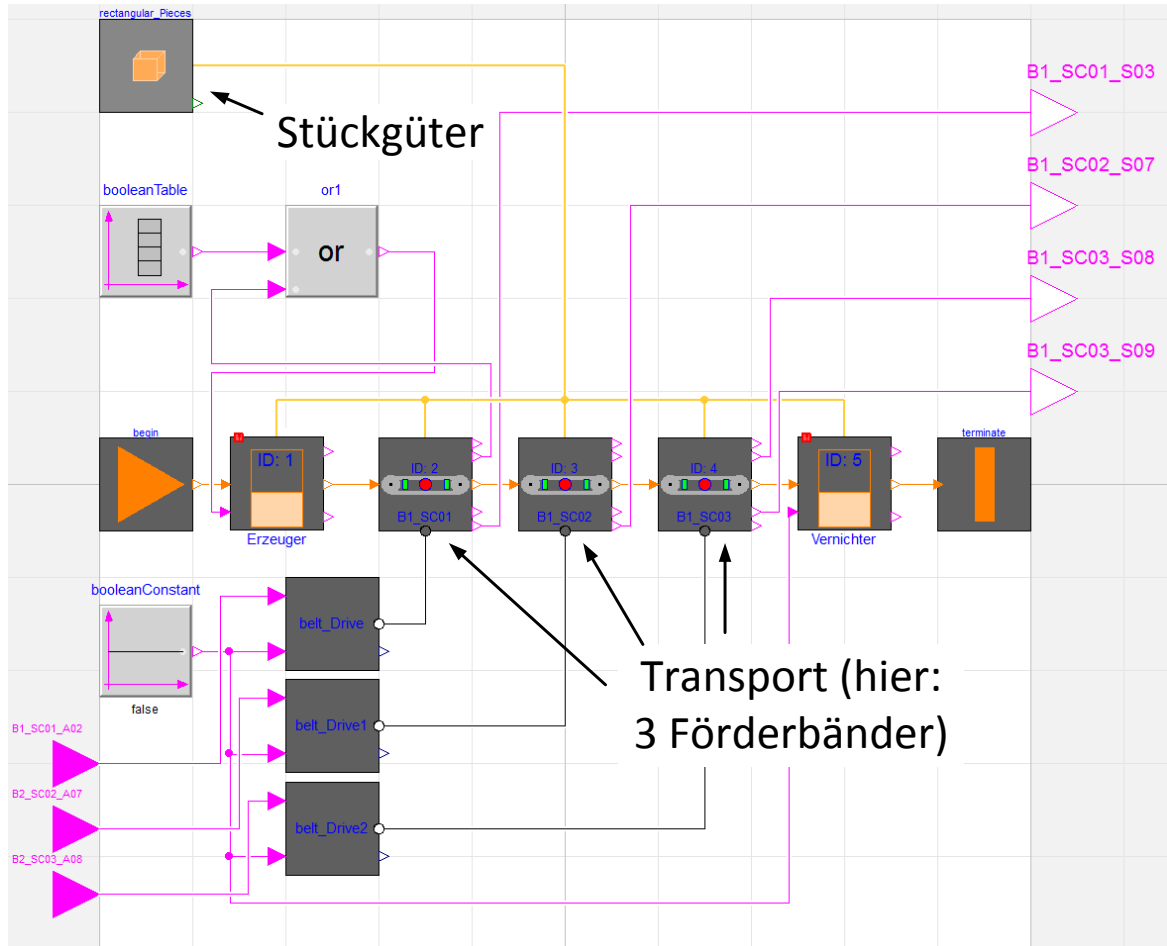
10.2.3 eCl@ss

Auch wenn sich der Einsatz von *eCl@ss* in dieser Arbeit bewährt und als nutzbringend erwiesen hat, sind einige wichtige Fragen offen. So ist die Granularität einzelner *eCl@ss* Untergruppen nicht immer eindeutig. Unklar ist beispielsweise, ob sich ein Objekt „Förderband“ auf ein komplettes Modul (das auch Sensoren und Aktoren beinhalten würde) oder nur auf das mechanische Grundsystem (und damit eine Komponente) bezieht. Ferner sind in *eCl@ss* vorhandene Merkmalleisten nicht immer vollständig und daraus folgend nicht immer für den Gebrauch geeignet (z. B. im Anlagenbau). Im Forschungsprojekt „SemAnz40“ wird derzeit erforscht, wie Industrie 4.0 Anwendungsfälle unter Anwendung deutscher Standards (z. B. *eCl@ss*) realisiert werden können [SEM ANZ 4.0[®]]. Dies wird zu einer Harmonisierung von in Standards bereits vorhandenen und (im Speziellen) für die Domäne Automatisierungstechnik benötigten Merkmalen führen. Der in dieser Arbeit vorgestellte Ansatz kann davon profitieren, in dem initialer Aufwand bei der Ergänzung bzw. Ertüchtigung von *eCl@ss*-Merkmalleisten entfallen könnte.

Anhang

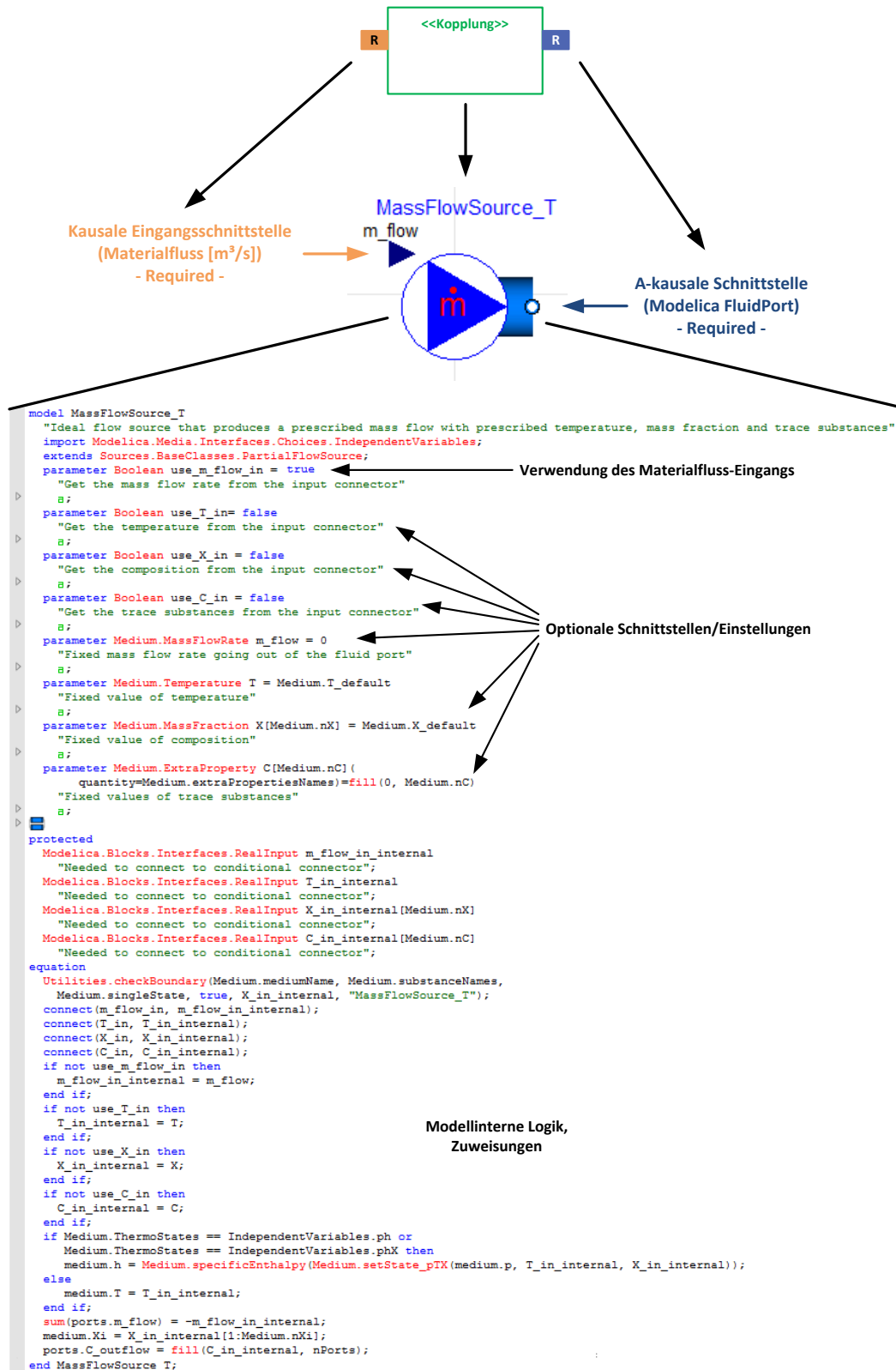
Anhang 1 Trennung von Stückgütern und Transport in Simulation

Trennung der Berechnung von Stückgütern und des Verhaltens des Förderbands in der Simulation anhand eines einfachen Beispiels (→ Abschnitt 7.4.2).



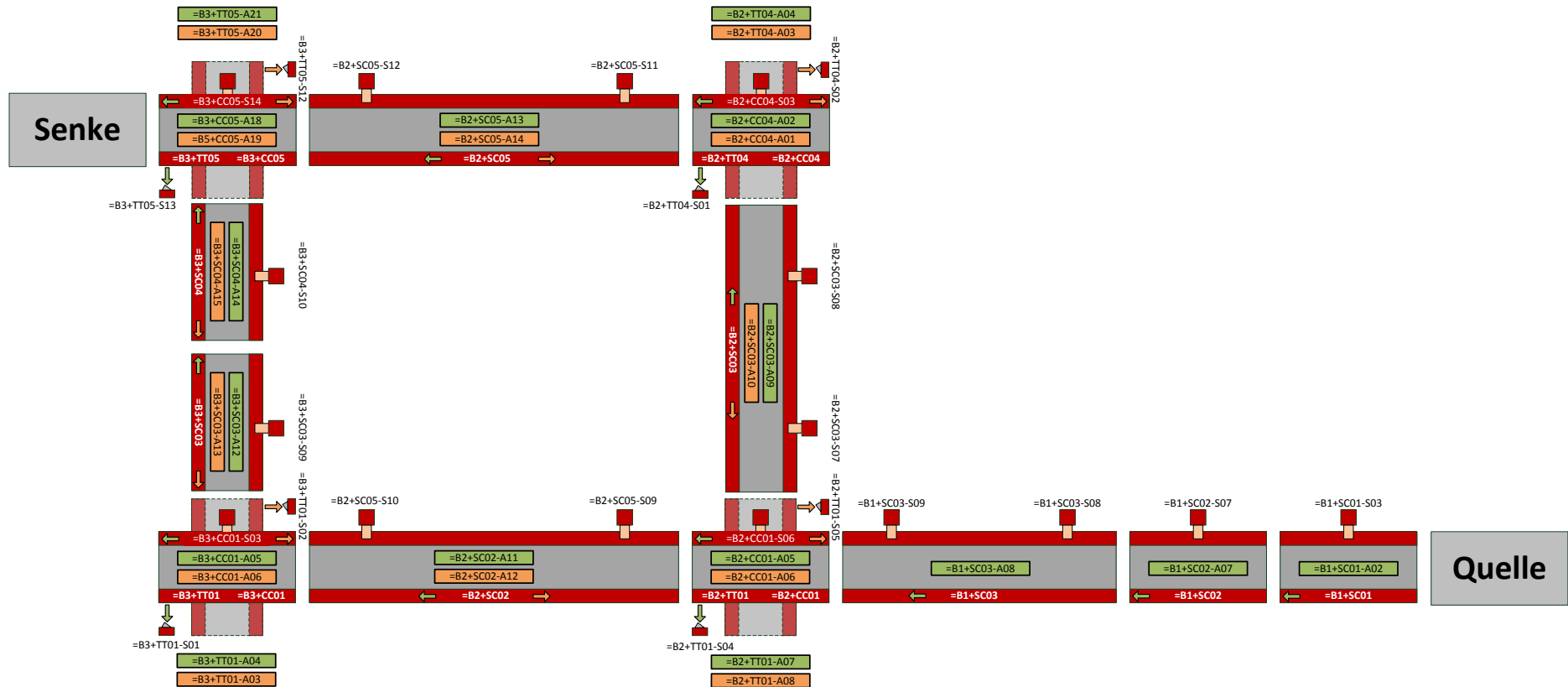
Anhang 2 Kopplungselement

Beispielhafte Darstellung eines Kopplungselements, hier: MassFlowSource_T aus der Modelica Fluid Standardbibliothek (→ Abschnitt 8.3.2.2).



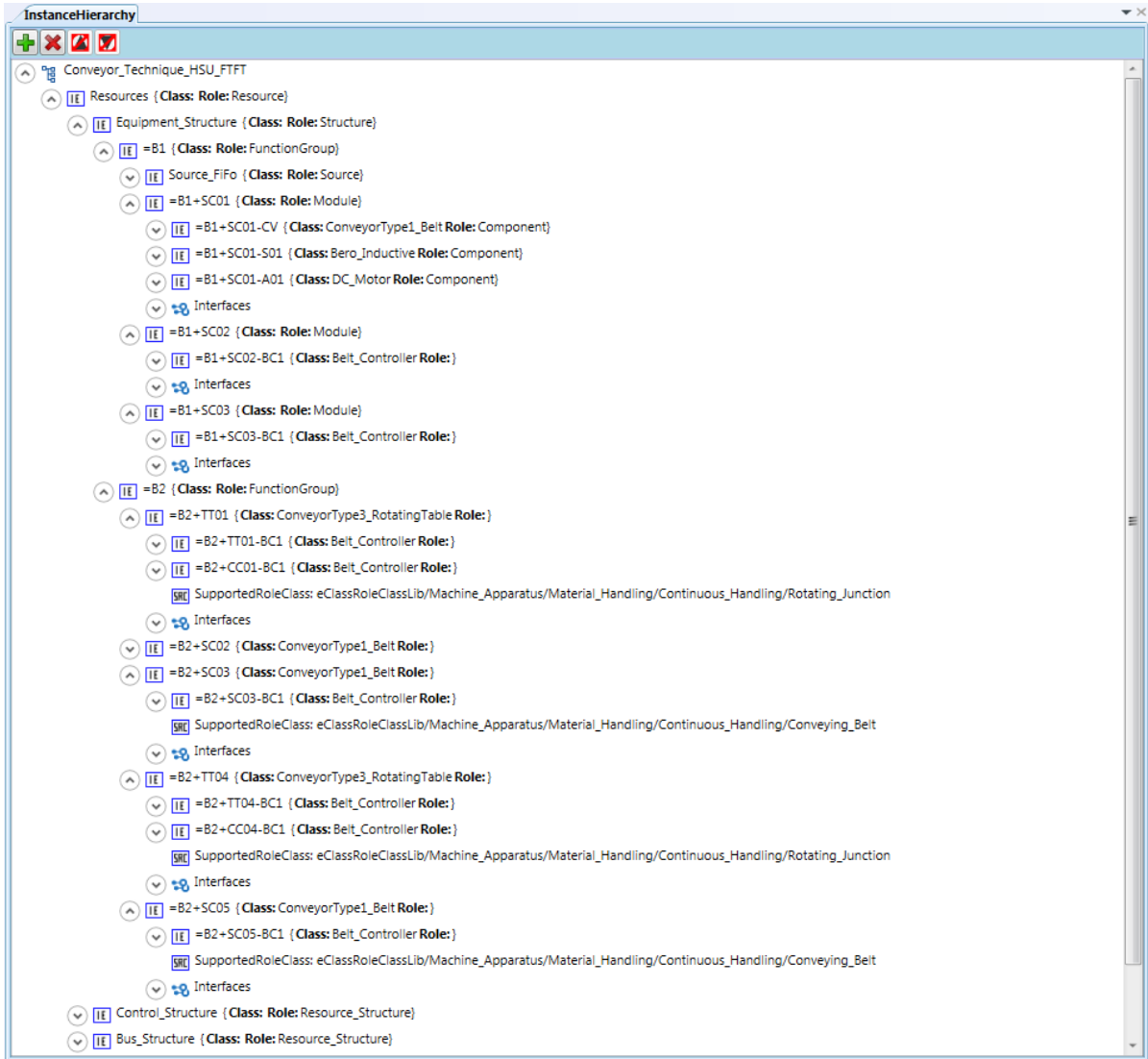
Anhang 3 Anlagenübersicht

Übersicht der genutzten Anlagen inkl. aller relevanten Betriebsmittelkennzeichen (→ Abschnitt 9.1.1).



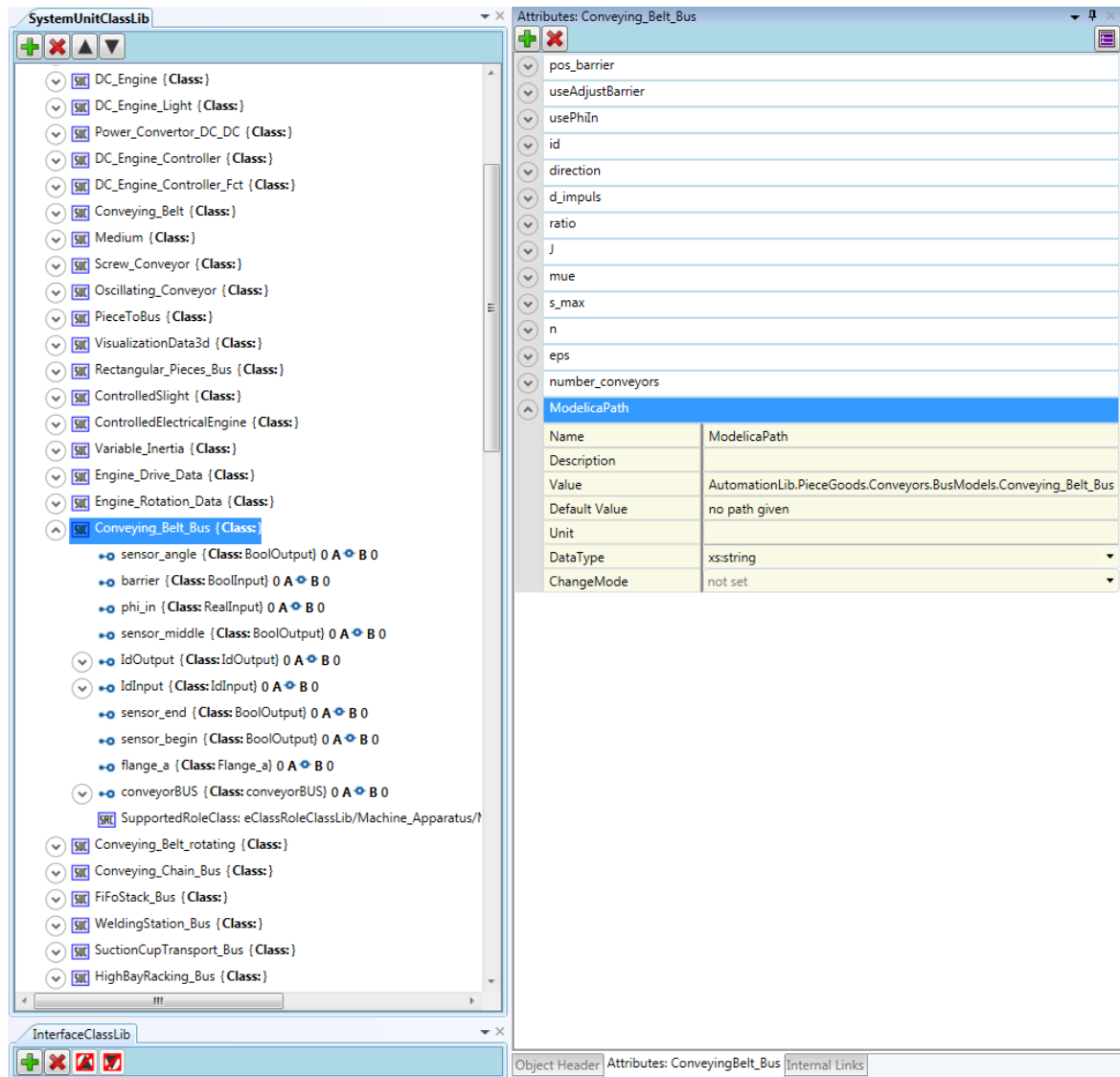
Anhang 4 Anlagenstrukturmodell

Anlagenstrukturmodell in grafischer Abbildung, das für die automatische Modellgenerierung genutzt wurde (→ Abschnitt 9.1.3).



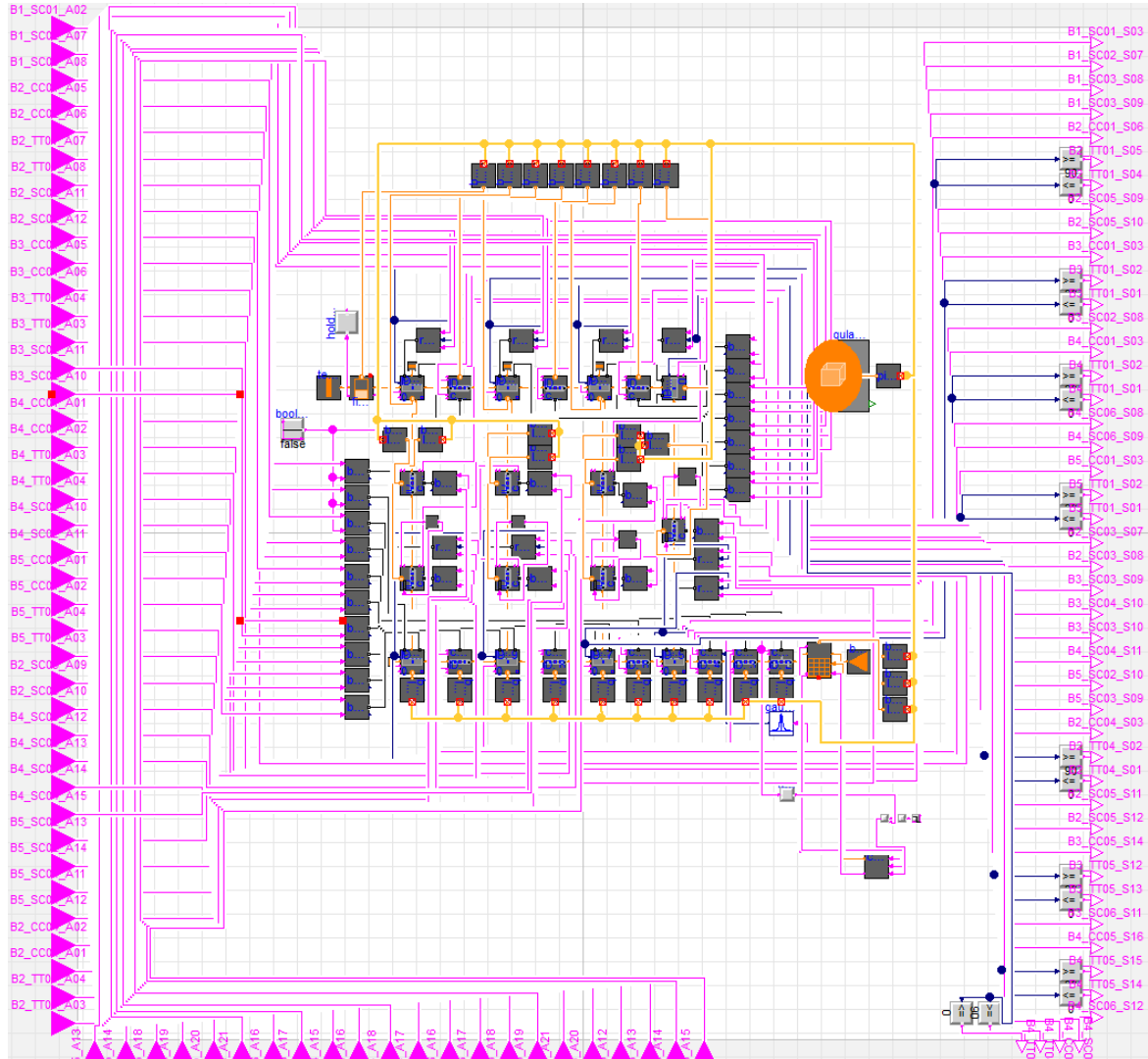
Anhang 5 Simulationsmodellbibliothek als SystemUnitClassLibrary

Ausschnitt aus der im Projekt „HIL-IMA“ erstellten und in AutomationML importierten Simulationsmodellbibliothek (→ Abschnitt 9.1.3).



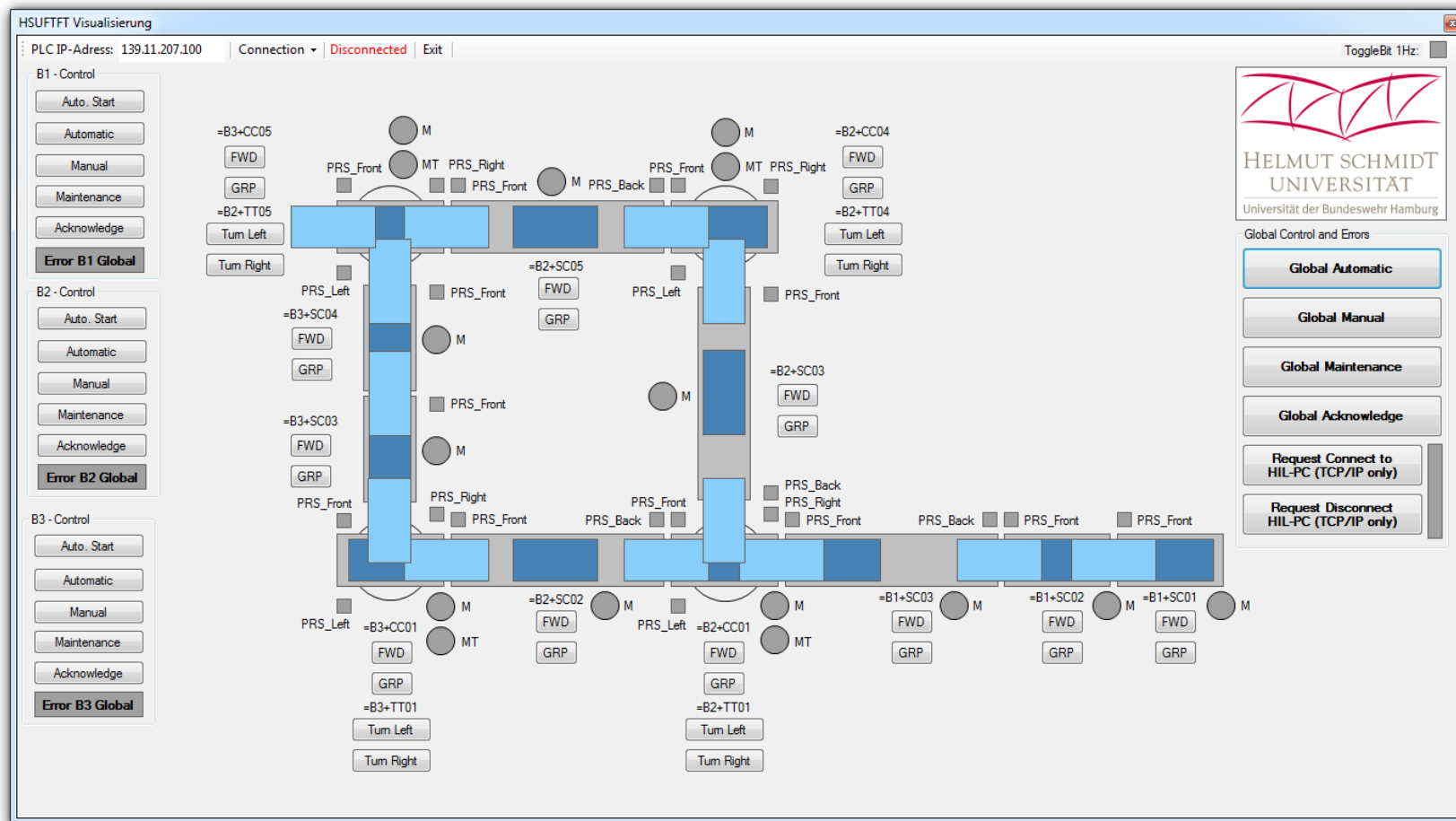
Anhang 6 Generiertes Simulationsmodell in Modelica

Automatisch generiertes Simulationsmodell in Modelica (grafische Ansicht). Die Position der Elemente wurde manuell erstellt, um die Übersichtlichkeit zu gewährleisten und das Debugging zu vereinfachen (→ Abschnitt 9.1.4).



Anhang 7 Visualisierungslösung

Proprietäre Visualisierungslösung zur Steuerung des programmierten Programmablaufs in der SPS (→ Abschnitt 9.1.5.1).



Literaturverzeichnis

Dieses Verzeichnis enthält eine Liste der referenzierten Quellen, die nicht vom Autor stammen. Die Quellen werden mittels [<Kurzbeleg>] gekennzeichnet.

- [ACÉ96] P. Acél: *Methode zur Durchführung betrieblicher Simulationen*. Eidgenössische Technische Hochschule Zürich. Dissertation, 1996.
- [ACÉ97] P. Acél: *Systems-Engineering und Simulation: Leitfaden zum Vorgehen*. In: Acél (Hrsg.): *Methode zur Durchführung betrieblicher Simulationen: Effiziente Optimierung der diskreten Simulation*. Wien: ARGESIM/ASIM-Verlag, 1997.
- [ALB99] J. Albert: *Echtzeitsimulation zum Test von Maschinensteuerungen*. München: Utz Wiss., 1999.
- [AMA94] W. Amann: *Eine Simulationsumgebung für Planung und Betrieb von Produktionssystemen*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 1994.
- [ABV11] AUDI AG, BMW Group, Volkswagen AG: *Werksplanung - Fabrikplanung*, 2011.
- [ABD*08] AUDI AG, BMW Group, Daimler AG, Volkswagen AG: *Basisrichtlinie für die Konstruktion von Anlagen und Vorrichtungen mit CATIA V5*. 2008.
- [BAEB01] H. Balzert, C. Ebert: *Lehrbuch der Softwaretechnik*. Heidelberg: Spektrum Akademischer Verlag, 2008.
- [BAR11] M. Barth: *Automatisch generierte Simulationsmodelle verfahrenstechnischer Anlagen für den Steuerungstest*. Hamburg, Helmut-Schmidt-Universität, Institut für Automatisierungstechnik. Dissertation, 2011 .
- [BAR15] M. Barth: *Methoden und Modelle der virtuellen Inbetriebnahme*. Automatisierungsforum Westküste. Präsentation, 10.02.2015.
- [BAT06] J. Bathelt: *Entwicklungsmethodik für SPS-gesteuerte mechatronische Systeme*. Zürich, Eidgenössischen Technischen Hochschule Zürich, Zentrum für Produkt-Entwicklung. Dissertation, 2006.
- [BJB*03] J. Bathelt, A. Jönsson, C. Bacs, M. Meier: *Modularisierung SPS-gesteuerter mechatronischer Systeme*. In: Gausemeier (Hrsg.): *Intelligente mechatronische Systeme: Grundlagen, Methoden und Werkzeuge, Adaption und Selbstoptimierung, Mechatronik und Mikrosystemtechnik*. Paderborn: HNI-Verlagsschriftenreihe, 2003.
- [BCW03] J. Bayer, T. Collisi, S. Wenzel: *Simulation in der Automobilproduktion*. Berlin, Heidelberg: Springer Verlag, 2003.
- [BEKE06] C. Beierle, G. Kern-Isberner: *Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen*. Berlin: Springer-Verlag, 2006.
- [BKH*09] M. Bergert, J. Kiefer, S. Höme, C. Fedrowitz: *Einsatz der Virtuellen Inbetriebnahme im automobilen Karosserierohrbau – Ein Erfahrungsbericht*. 2009.
- [BMW08] BMW Group - Technologie Montage: *Liefervorschrift CAD-Layoutplanung*, 2008.
- [BKS*09] T. Böhme, M. Kennel, M. Schumann, A. Winge: *Automatisierte Erstellung domänenübergreifender Modelle und echtzeitfähige Kopplung von Simulation, Visualisierung und realen Steuerungen*. In: Gausemeier (Hrsg.): *Augmented & virtual reality in der Produktentstehung: Grundlagen, Methoden und Werkzeuge*. Paderborn: Heinz-Nixdorf-Institut, 2009.

- [BRJ05] G. Booch, J. Rumbaugh, I. Jacobson: *The unified modeling language user guide*. Reading, Mass.: Addison-Wesley-Verlag, 2005.
- [Bos04] H. Bossel: *Systeme, Dynamik, Simulation: Modellbildung, Analyse und Simulation komplexer Systeme*. Norderstedt: Books on Demand, 2004.
- [BHH*10] J. Botaschanjan, T. Hensel, B. Hummel, A. Lindworsky, M.F. Zäh, G. Reinhart, M. Broy: *AutoVIBN – Abschlussbericht: Automatische Generierung von Verhaltensmodellen aus CAD-Daten für die Qualitätsorientierte virtuelle Inbetriebnahme*. München, 2010.
- [BWG09] U. Bracht, S. Wenzel, D. Geckler: *Digitale Fabrik: Methoden und Praxisbeispiele*. Berlin: Springer, 2009.
- [BRPO09] F. Breitenecker, N. Popper: *Classification and Evaluation of Features in Advanced Simulators*. In: Troch, Breitenecker (Hrsg.): MATHMOD Vienna 09: Proceedings. Wien: ARGESIM, 2009.
- [BRTO96] R. Brooks, A. Tobias: *Choosing the best model: Level of detail, complexity, and model performance*. Mathematical and Computer Modelling, Vol. 24 (4), 1996, S. 1–14.
- [CHR15] L. Christiansen: *Wissensgestütztes Diagnosekonzept durch Kombination von Anlagenstruktur- und Prozessmodell*. Hamburg, Helmut-Schmidt-Universität, Institut für Automatisierungstechnik. Dissertation, 2015.
- [CHF14A] L. Christiansen, M. Hoernicke, A. Fay: *Entwicklung eines Modellverbinders zur Vermeidung von Inkonsistenzen zwischen Modellen im Kontext des modellbasierten Engineerings*. In: Automation 2014: 15. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verlag, 2014.
- [CHF14B] L. Christiansen, M. Hoernicke, A. Fay: *Regelbasierte Erstellung von Topologie-Modellen im Kontext der „Automatisierung der Automatisierung“*. In: Jumar, Diedrich (Hrsg.): Entwurf komplexer Automatisierungssysteme: EKA 2014. Magdeburg: Institut für Automation und Kommunikation e.V., 2014.
- [CHR03] F. Christoph: *Durchgängige simulationsgestützte Planung von Fertigungseinrichtungen der Elektronikproduktion*. Erlangen, Friedrich-Alexander-Universität Erlangen Nürnberg, Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik. Dissertation, 2003.
- [CLHE08] M. Clausing, S. Heinrich: *Mensch, Maschine, Material – die Standardisierung der Ablaufsimulation in der Automobilindustrie*. ProduktDaten Journal, Vol. 23-25 (1), 2008.
- [DABR07] M. Danilovic, T.R. Browning: *Managing complex product development projects with design structure matrices and domain mapping matrices*. International Journal of Project Management, Vol. 25 (3), 2007, S. 300–314.
- [DiFA13] C. Diedrich, A. Fay: *Engineering*. at - Automatisierungstechnik, Vol. 61 (6), 2013, S. 379–381.
- [DLH11] C. Diedrich, A. Lüder, L. Hundt: *Bedeutung der Interoperabilität bei Entwurf und Nutzung von automatisierten Produktionssystemen*. at - Automatisierungstechnik, Vol. 59 (7), 2011, S. 426–437.
- [DiMü11] C. Diedrich, M. Mühlhause: *Modellansätze für die digitale Fabrik*. at - Automatisierungstechnik, Vol. 59 (1), 2011, S. 18–25.

- [DOHE07] S. Dominika, C. Heuschmann: *Prozessgutsimulation und Anlagenteilsimulation zur Verkürzung der Inbetriebnahme*. atp - Automatisierungstechnische Praxis, Vol. 49 (7), 2007, S. 55–60.
- [DPA14] dpa: *Autobauer verzichten zunehmend auf einheitliche Werksferien*. Der Westen, 26.05.2014.
- [DRA08] R. Drath: *Die Zukunft des Engineering: Herausforderungen an das Engineering von fertigungs- und verfahrenstechnischen Anlagen*. In: Karlsruher Leittechnisches Kolloquium 2008. Stuttgart: Fraunhofer-IRB-Verlag, 2008.
- [DRA10] R. Drath: *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*. Heidelberg, New York: Springer-Verlag, 2010.
- [DFB11] R. Drath, A. Fay, M. Barth: *Interoperabilität von Engineering-Werkzeugen: Konzepte und Empfehlungen für den Datenaustausch zwischen Engineering-Werkzeugen*. at - Automatisierungstechnik, Vol. 59 (7), 2011, S. 451–460.
- [DLP*08] R. Drath, A. Lüder, J. Peschke, L. Hundt: *AutomationML – the glue for seamless Automation Engineering*. In: 13th IEEE International Conference on Emerging Technologies and Factory Automation: ETFA 2008. Piscataway, N.J.: IEEE Xplore, 2008.
- [DSK*13] B. Drescher, P. Stich, J. Kiefer, A. Strahilov, T. Bär, G. Reinhart: *Physikbasierte Simulation im Anlagenentstehungsprozess – Einsatzpotenziale bei der Entwicklung automatisierter Montageanlagen im Automobilbau*. In: Dangelmaier, Laroque, Klaas (Hrsg.): *Simulation in Produktion und Logistik 2013*. Paderborn: HNI-Verlagsschriftenreihe, 2013.
- [EHR84] H.-D. Ehrich: *Fachgespräche auf der 14. GI-Jahrestagung, Braunschweig, 1-2. Oktober 1984*. Berlin, New York: Springer-Verlag, 1984.
- [EHR03] K. Ehrlenspiel: *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit*. München: Hanser Verlag, 2003.
- [ENHO07] C. Engel, C. Holm: *Ergebnisse der Projektmanagement Studie 2007: Gemeinsame Studie der GPM Deutsche Gesellschaft für Projektmanagement e.V. und PA Consulting Group*, 2007.
- [EC41] N.N.: *Engineers' Council for Professional Development*. Science, Vol. 94 (2446), 1941, S. 456.
- [EPP03] U. Epple: *Austausch von Anlagenplanungsdaten auf der Grundlage von Metamodellen*. atp - Automatisierungstechnische Praxis, Vol. 45 (7), 2003, S. 61–70.
- [FYS13] A.B. Farjadian, M.J. Yazdanpanah, B. Shafai: *Application of Reinforcement Learning in Sliding Mode Control for Chattering Reduction*. In: Ao, Gelman, et al. (Hrsg.): *World Congress on Engineering: WCE 2013*. Hong Kong: Newswood Limited; International Association of Engineers, 2013.
- [FAY05] A. Fay: *Engineering in vernetzten, offenen, durchgängigen Systemen*. at - Automatisierungstechnik, Vol. 53 (4), 2005, S. 205–210.
- [FAY06] A. Fay: *Reduzierung der Engineering-Kosten für Automatisierungssysteme*. Industrie Management, Vol. 22 (2), 2006, S. 29–32.
- [FAY09] A. Fay: *Effizientes Engineering komplexer Automatisierungssysteme*. In: Schnieder, Ständer (Hrsg.): *Wird der Verkehr automatisch sicherer? 20 Jahre - vom IfRA zum iVA - Jubiläumskolloquium*. Braunschweig: iVA, 2009.

- [FAY11] A. Fay: *Wie real ist die virtuelle Inbetriebnahme?* atp - Automatisierungstechnische Praxis, Vol. 53 (6), 2011, S. 3.
- [FSM09] A. Fay, M. Schleipen, M. Mühlhause: *Wie kann man den Engineering-Prozess systematisch verbessern?* atp - Automatisierungstechnische Praxis, Vol. 51 (1), 2009, S. 80–85.
- [FEL97] C. Feldmann: *Eine Methode für die integrierte rechnergestützte Montageplanung*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 1997.
- [FRTH11] G. Frey, K. Thramboulidis: *Einbindung der IEC 61131 in modellgetriebene Entwicklungsprozesse*. In: Automation 2011: 12. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verlag, 2011.
- [FSG+15] T. Friebe, M. Schneider, J. Gausemeier, A. Trächtler: *Virtuelle Inbetriebnahme mit wählbarer Modellierungstiefe*. Zeitschrift für wirtschaftlichen Fabrikbetrieb: ZWF, Vol. 110 (4), 2015, S. 227–232.
- [GAN13] O. Ganschar: *Produktionsarbeit der Zukunft - Industrie 4.0: Studie*. Stuttgart: Fraunhofer Verlag, 2013.
- [GOB+14] S. Gulan, U. Odefey, T. Bär, H. Beesten, H. Hämmerle, B. Kärcher, M. Riedel: *Physikbasierte Simulation im Anlagenbau*. In: Scheible, Bausch-Gall, Deatcu (Hrsg.): ASIM-Workshop STS/GMMS 2014: Treffen der ASIM/GI-Fachgruppen: Simulation technischer Systeme - Grundlagen und Methoden in Modellbildung und Simulation. Reutlingen, 2014.
- [GSA+11] J. Gausemeier, W. Schäfer, H. Anacker, F. Bauer, S. Dziwok: *Semantische Technologien im Entwurf mechatronischer Systeme*. In: Gausemeier, Rammig et al. (Hrsg.): 8. Paderborner Workshop Entwurf Mechatronischer Systeme (EMS): Wissenschaftsforum 2011. Intelligente Technische Systeme. Paderborn: HNI-Verlagsschriftenreihe, 2011.
- [GIE05] L. Gielsok: *Analyse und Konzeption serviceorientierter Middleware zur Kopplung von Qualitätsapplikationen mit der Anwendung PQS*. Dresden, TU Dresden, Fakultät für Informatik. Diplomarbeit, 2005.
- [GLÖ14] M. Glöckler: *Simulation mechatronischer Systeme: Grundlagen und technische Anwendung*. Wiesbaden: Springer Vieweg, 2014.
- [GRS10] G. Görz, C.-R. Rollinger, J. Schneeberger: *Handbuch der Künstlichen Intelligenz*. München: Oldenbourg Wissenschaftsverlag, 2010.
- [GÜHo10] W. Günthner, M. ten Hompel: *Internet der Dinge in der Intralogistik*. Berlin, Heidelberg: Springer-Verlag, 2010.
- [GÜT12] K. Güttel: *Konzept zur Generierung von Steuerungscode für Fertigungsanlagen unter Verwendung wissensbasierter Methoden*. Hamburg, Helmut-Schmidt-Universität, Institut für Automatisierungstechnik. Dissertation, 2012.
- [GÜFA08] K. Güttel, A. Fay: *Beschreibung von fertigungstechnischen Anlagen mittels CAEX*. atp - Automatisierungstechnische Praxis, Vol. 50 (5), 2008, S. 34–39.
- [HWF+12] R. Haberfellner, O. de Weck, E. Fricke, S. Vössner, R. Haberfellner: *Systems Engineering: Grundlagen und Anwendungen*. Zürich: Orell Füssli Verlag, 2012.
- [HAP08] M. Happacher: *Der Weg zum Standard*. Computer & Automation Sonderdruck, 2008.

- [HTY11] W.S. Harrison, D.M. Tilbury, C. Yuan: *From Hardware-in-the-Loop to Hybrid Process Simulation: An Ontology for the Implementation Phase of a Manufacturing System*. IEEE Transactions on Automation Science and Engineering, Vol. 1 (9), 2011.
- [HAS11] C. Haskins: *Systems engineering handbook: A guide for system life cycle processes and activities*. San Diego, California: International Council of Systems Engineering, 2011.
- [HEH11] P. Hehenberger: *Computerunterstützte Fertigung: Eine kompakte Einführung*. Berlin, Heidelberg: Springer Verlag, 2011.
- [HEL97] D. Helbing: *Verkehrsdynamik: Neue physikalische Modellierungskonzepte; mit 6 Tabellen*. Berlin: Springer Verlag, 1997.
- [HEL13] D. Hellenbrand: *Transdisziplinäre Planung und Synchronisation mechatronischer Produktentwicklungsprozesse*. München, TU München, Lehrstuhl für Produktentwicklung. Dissertation, 2013.
- [HFB15] M. Hoernicke, A. Fay, M. Barth: Virtual Plants for Brown-Field Projects - Automated generation of simulation models based on existing engineering data. In: Tagungsband der 19. Tagung "IEEE International Conference on Emerging Technologies and Factory Automation" (ETFA), Luxembourg, 8.-11. September 2015.
- [HJS⁺15] S. Heymann, J. Jasperneite, S. Schröck, A. Fay: *Beschreibung von modularisierten Produktionsanlagen in Industrie 4.0*. In: Automation 2015: 16. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verlag, 2015.
- [HIR12] A. Hirzle: *AutomationML – ein Überblick: Ein Standard für die Verbesserung des Datenaustauschs von Engineeringwerkzeugen*. SPS Magazin (12), 2012, S. 65–66.
- [HOQU14] J. Hoffmann, F. Quint: *Simulation technischer linearer und nichtlinearer Systeme mit MATLAB/Simulink*. München: De Gruyter Oldenbourg, 2014.
- [HOVE12] S. Hoher, A. Verl: *Multi-Simulator-Materialflusssimulation für die virtuelle Inbetriebnahme*. In: Frey, Schumacher, Verl (Hrsg.): *SPS IPC DRIVES: Tagungsband; internationale Fachmesse und Kongress Nürnberg, 27. - 29. November 2012*. Berlin, Offenbach: VDE-Verl, 2012.
- [HRD97] V. Hrdliczka: *Leitfaden für Simulationsbenutzer in Produktion und Logistik: ASIM Mitteilungen 58*, 1997.
- [JCS⁺12] T. Jäger, L. Christiansen, M. Strube, A. Fay: *Durchgängige Werkzeugunterstützung von der Anforderungserhebung bis zur Anlagenstrukturbeschreibung mittels formalisierter Prozessbeschreibung und AutomationML*. In: Diedrich, Jumar, Schnieder (Hrsg.): *Entwurf komplexer Automatisierungssysteme: EKA 2012*. Magdeburg, ifak, 2012.
- [JAN10] K. Janschek: *Systementwurf mechatronischer Systeme: Methoden, Modelle, Konzepte*. Berlin, Heidelberg: Springer-Verlag, 2010.
- [KKS⁺06] W. Keijzer, M. Kreimeyer, R. Schack, U. Lindemann, M.F. Zäh: *Isolierung versus Vernetzung – wie arbeitet die Branche zusammen?* 3. Fachkongress „Digitale Fabrik“, Ludwigsburg. 2006.
- [KIE07] J. Kiefer: *Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau*. Saarbrücken, Universität des Saarlandes, Lehrstuhl für Fertigungstechnik/CAM. Dissertation, 2007.

- [KIE95] J.U. Kiess: *Objektorientierte Modellierung von Automatisierungssystemen: Software Engineering für Embedded Systems*. Dissertation, 1995.
- [KLU10] F. Klug: *Logistikmanagement in der Automobilindustrie: Grundlagen der Logistik im Automobilbau*. Berlin, Heidelberg: Springer-Verlag, 2010.
- [KOH93] A. Kohring: *Systematisches Projektieren und Testen von Steuerungssoftware für Werkzeugmaschinen*. Aachen, RWTH Aachen. Dissertation, 1993.
- [KRO14] F. Kronthaler: *Statistik angewandt: Datenanalyse ist (k)eine Kunst*. Berlin: Springer Spektrum, 2014.
- [KWV10] B. Kormann, D. Witsch, B. Vogel-Heuser: *Automatische Testfallgenerierung mittels Model-Checking für Steuerungsprogramme*. In: Automation 2010: 11. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verlag, 2010.
- [Köv10] L. Kövari: *Konzeption und Realisierung eines neuen Systems zur produktbegleitenden virtuellen Inbetriebnahme komplexer Förderanlagen*. Karlsruhe, Karlsruher Institut für Technologie, Institut für Angewandte Informatik/Automatisierungstechnik. Dissertation, 2010.
- [KRNE98] U. Kramer, M. Neculau: *Simulationstechnik*. München, Wien: Hanser Verlag, 1998.
- [KUD00] T. Kudlich: *Optimierung von Materialflusssystemen mit Hilfe der Ablaufsimulation*. München, TU München, Lehrstuhl für Fördertechnik Materialfluß Logistik. Dissertation, 2000.
- [KUF11] A. Kufner: *Automatisierte Erstellung von Maschinenmodellen für die Hardware-in-the-Loop-Simulation von Montagemaschinen*. Stuttgart, Universität Stuttgart, Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen. Dissertation, 2011.
- [KHK10] A. Kufner, K. Haug, P. Klemm: *Modellierung von Montagemaschinen für die Hardware-in-the-Loop-Simulation*. In: Gausemeier (Hrsg.): *Entwurf mechatronischer Systeme Grundlagen, Methoden und Werkzeuge; Adaption, Selbstoptimierung und Verlässlichkeit; Integration Mechanik und Elektronik, Miniaturisierung*. Paderborn: Heinz-Nixdorf-Institut, 2010
- [KUG99] D. Kugelmann: *Aufgabenorientierte Offline-Programmierung von Industrierobotern*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 1999.
- [KÜH06] W. Kühn: *Digitale Fabrik: Fabriksimulation für Produktionsplaner*. München: Hanser Verlag, 2006.
- [KÜRo12] C. Kühnl, M. Rothhöft: *Unzufrieden mit Engineering-Schnittstellen*. computer-automation.de (6), 2012.
- [LAC12] F.-F. Lacour: *Modellbildung für die physikbasierte Virtuelle Inbetriebnahme materialflussintensiver Produktionsanlagen*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 2012.
- [LANO01] L. Lamieri, K. North: *Wissensmanagement in Klein- und Mittelbetrieben*. Praxis Wissensmanagement (6), 2001, S. 103–122.
- [LB14] R. Löhner, J. D. Baum: *On maximum achievable speeds for field solvers*. International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 24 (7), 2014, S. 1537–1544.

- [LHM*99] M. Lasa, H.-M. Heinkel, E. Moser, R. Rothfuss: *Modeling Mechatronic Systems at Different Levels of Abstraction*. In: Kasper (Hrsg.): *Entwicklungsmethoden und Entwicklungsprozesse im Maschinenbau: Tagungsband*. Berlin: Logos-Verlag, 1999.
- [LIN11] A. Lindworsky: *Teilautomatische Generierung von Simulationsmodellen für den entwicklungsbegleitenden Steuerungstest*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 2011.
- [LiFR09] L. Liu, G. Frey: *Efficient Simulation of Hybrid Control Systems in Modelica/Dymola*. In: Troch, Breitenecker (Hrsg.): *MATHMOD Vienna 09: Proceedings*. Wien: ARGE-SIM, 2009.
- [LBO*15] Z. Liu, Bieliaiev, O., C. Diedrich, T. Meyer, B. Völzke: *Komponentenmodelle für die Virtuelle Inbetriebnahme*. In: *Automation 2015: 16. Branchentreff der Mess- und Automatisierungstechnik*. Düsseldorf: VDI-Verlag, 2015.
- [LSB12] M. Lochbichler, T. Schmüdderich, J.T.A. Brökelmann: *Methodology for Selecting the Modeling Depth of Object-Oriented Behavioral Models*. In: *World Academy of Science, Engineering and Technology*, Jul. 2012.
- [LSJ16] A. Lüder, N. Schmidt, M. John: *Lossless Exchange of Automation Project Configuration Data*. In: *21st IEEE International Conference on Emerging Technologies and Factory Automation: ETFA 2016*. Piscataway, N.J.: IEEE, 2016.
- [LÜPE00] A. Lüdecke, G. Pelz: *Top-Down Design of a Mechatronic System*. In: *Proceedings: Third Forum on Design Languages (FDL 2000)*. Tübingen, 2000.
- [MAN09] S. Mandel: *Proaktive Anlaufabsicherung automatisierter Produktionsanlagen*. Hamburg, Helmut-Schmidt-Universität, Institut für Automatisierungstechnik. Dissertation, 2009.
- [MRW04] H. Martin, P. Römisch, A. Weidlich: *Materialflusstechnik: Konstruktion und Berechnung von Transport-, Umschlag- und Lagermitteln*. 8. Aufl. Wiesbaden: Vieweg, 2004.
- [MABU06] G. Mayer, U. Burges: *Virtuelle Inbetriebnahme von Produktionssteuerungssystemen in der Automobilindustrie mittels Emulation*. In: Wenzel (Hrsg.): *Simulation in Produktion und Logistik 2006: Tagungsband zur 12. Fachtagung*. Erlangen: SCS Publ. House, 2006.
- [MEL04] S.J. Mellor: *MDA distilled: principles of model-driven architecture*. Boston: Addison-Wesley, 2004.
- [MBS*11] H. Mersch, D. Behnen, D. Schmitz, U. Epple, C. Brecher, M. Jarke: *Gemeinsamkeiten und Unterschiede der Prozess- und Fertigungstechnik: Interdisziplinäre Aspekte der Produktionsmodellierung*. at - Automatisierungstechnik, Vol. 59 (1), 2011, S. 7–17.
- [MEY14] T. Meyer: *Wirtschaftliche Erstellung von Emulationsmodellen für die Virtuelle Inbetriebnahme*. Illmenau, Technische Universität Illmenau, Fakultät für Wirtschaftswissenschaften und Medien. Dissertation, 2014.
- [NgSc89] Nguyen-huu-chau-Thuy., P. Schnupp: *Wissensverarbeitung und Expertensystem*. München, Wien: Oldenbourg, 1989.
- [NON91] I. Nonaka: *The Knowledge-Creating Company*. Harvard Business Review, Vol. 69 (6), 1991, S. 96–104.

- [OBV15] M. Obermeier, S. Braun, B. Vogel-Heuser: *A Model-Driven Approach on Object-Oriented PLC Programming for Manufacturing Systems with Regard to Usability*. IEEE Transactions on Industrial Informatics, Vol. 11 (3), 2015, S. 790–800.
- [OERE15] C. Oetter, G. Reimann: *IT und Automation im Maschinenbau: VDMA Trendstudie 2015 – Zusammenfassung*. 2015.
- [OWD*14] M. Oppelt, G. Wolf, O. Drumm, B. Lutz, T. Baudisch, J.C. Wehrstedt, A. Krause, L. Urbas: *Automatische Generierung von Simulationsmodellen für die virtuelle Inbetriebnahme auf Basis von Planungsdaten*. In: Automation 2014: 15. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verlag, 2014.
- [OSM98] U. Osmer: *Projektieren Speicherprogrammierbarer Steuerungen mit Virtual Reality*. Karlsruhe, Universität Karlsruhe, Institut für Werkzeugmaschinen und Betriebs-technik. Dissertation, 1998.
- [PHK*13] H. Palm, J. Holzmann, R. Klein, S. Schneider, D. Gerling: *A Novel Approach on Virtual Systems Prototyping Based on a Validated, Hierarchical, Modular Library*. In: Proceedings of the Embedded World 2013. Haar: WEKA Fachmedien, 2013.
- [PAU12] A. Paul: *Verwundbarkeitsanalyse des Industrial-Ethernet Protokolls Profinet IO*. In: Stewin, Mulliner (Hrsg.): Seventh GI SIG SIDAR Graduate Workshop on Reactive Security: SPRING. Berlin, 2012.
- [RWS06] M. Rabe, S. Wenzel, S. Spieckermann (Hrsg.): *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*. 1. Aufl. Berlin: Springer Verlag, 2006.
- [RELA10] G. Reinhart, F.F. Lacour: *Skalierbare Simulation kinematischer Strukturen in der physikbasierten Virtuellen Inbetriebnahme*. In: Gausemeier, Grafe (Hrsg.): 9. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. Paderborn: Heinz-Nixdorf-Institut, Universität Paderborn, 2010.
- [RSF08] M. Riedel, T. Schmidberger, A. Fay: *Wissensbasierte Auswahl geeigneter Messprinzipien auf der Basis von Merkmalleisten*. atp - Automatisierungstechnische Praxis, Vol. 50 (1), 2008, S. 44–49.
- [RöD02] H.-J. Rödiess: *Planungswerkzeuge aus Sicht des Anlagenbaus*. atp - Automatisierungstechnische Praxis, Vol. 44 (1), 2002, S. 40–44.
- [RUD06] H. Rudolf: *Wissensbasierte Montageplanung in der digitalen Fabrik am Beispiel der Automobilindustrie*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 2006.
- [RUN11] S. Runde: *Wissensbasierte Engineeringunterstützung in der Automatisierungstechnik am Beispiel der Gebäudeautomation*. Düsseldorf, Helmut-Schmidt-Universität, Institut für Automatisierungstechnik. Dissertation, 2011.
- [RFS*11] S. Runde, A. Fay, S. Schmitz, U. Epple: *Wissensbasierte Systeme im Engineering der Automatisierungstechnik: Potenziale, Anwendungen, Defizite und zukünftige Herausforderungen*. at - Automatisierungstechnik, Vol. 59 (1), 2011, S. 42–49.
- [SAT07] R. Sattler: *Physikalisch basierte Mixed-Level Modellierung von gedämpften elektromechanischen Mikrosystemen*. München, TU München, Fakultät für Elektrotechnik und Informationstechnik. Dissertation, 2007.
- [SSA10] O. Sauer, M. Schleipen, C. Ammermann: *Digitaler Fabrikbetrieb: Virtual Manufacturing*. In: Zülch, Stock (Hrsg.): Integrationsaspekte der Simulation: Technik, Organisation und Personal. Hannover, Karlsruhe: KIT Scientific Publ., 2010.

- [SAV08] M. Savrasov: *Overview of Flow Systems Investigation and Analyses Methods*. In: Proceedings of the 8th International Conference "Reliability and Statistics in Transportation and Communication". Riga, 2008.
- [SCFE07] I. Schafer, M. Felser: *Precision of Ethernet Measurements based on Software Tools*. In: 12th IEEE International Conference on Emerging Technologies and Factory Automation: ETFA 2007. Piscataway, N.J.: IEEE, 2007.
- [SCDR09] M. Schleipen, R. Drath: *Three-View-Concept for modeling process or manufacturing plants with AutomationML*. In: 14th IEEE International Conference on Emerging Technologies and Factory Automation: ETFA 2009. Piscataway, N.J.: IEEE, 2009.
- [SCH08] T. Schmidberger: *Wissensbasierte Auswertung von Anlagen-Planungsdaten für die Unterstützung des Prozessleittechnik-Ingenieurs: Anwendung einer rollenbasierten Mustersuche*. Hamburg, Helmut-Schmidt-Universität, Institut für Automatisierungstechnik. Dissertation, 2008.
- [SchHo08] U. Schmidt, U. Hoffmann: *Informatik-Seminar WS2008/09: Linux und Netzwerke, Softwareentwicklung mit Eclipse*. FH Wedel, 2008.
- [SSD15] S. Süß, A. Strahilov, C. Diedrich: *Behaviour Simulation for Virtual Commissioning using Co-Simulation*. In: 20th Conference on Emerging Technologies and Factory Automation: ETFA 2015. Piscataway, N.J.: IEEE, 2015.
- [SSE09] D. Schmitz, M. Schluetter, U. Epple: *Automation of Automation – Definition, components and challenges*. In: 14th IEEE International Conference on Emerging Technologies and Factory Automation: ETFA 2009. Piscataway, N.J.: IEEE, 2009.
- [SST16] S. Süß, S. Magnus, M. Thron, H. Zipper, U. Odefey, V. Fäßler, A. Strahilov, A. Klodowski, T. Bär, C. Diedrich: *Test methodology for virtual commissioning based behavior simulation of production systems*. In: 21st IEEE International Conference on Emerging Technologies and Factory Automation: ETFA 2016. Piscataway, N.J.: IEEE, 2016.
- [SLT13] T. Schmüdderich, M. Lochbichler, A. Trächtler: *Methodik zur anforderungsgerechten Wahl der Modellierungstiefe von Verhaltensmodellen für die virtuelle Inbetriebnahme*. In: Bertram (Hrsg.): *Fachtagung Mechatronik 2013*: Aachen, 06.03. - 08.03.2013. Aachen, 2013.
- [SCTR14] T. Schmüdderich, A. Trächtler: *Virtual Commissioning by Means of an Adaptive Selection of the Modeling Depth*. In: ASME 2014 International Mechanical Engineering Congress and Exposition: November 14-20, 2014, Montreal, Quebec, Canada, 2014.
- [STB*13] T. Schmüdderich, A. Trächtler, J. Brökelmann, J. Gausemeier: *Procedural Model for the Virtual Commissioning on the Basis of Model-Based Design*. In: Abramovici, Stark (Hrsg.): *Smart product engineering: Proceedings of the 23rd CIRP Design Conference*. Berlin, London: Springer-Verlag, 2013.
- [STR13] M. Strube: *Modellgestützte Modernisierungsplanung industrieller Automatisierungslösungen*. Hamburg, Helmut-Schmidt-Universität, Institut für Automatisierungstechnik. Dissertation, 2013.
- [SGS*14] M. Schneider, J. Gausemeier, T. Schmüdderich, A. Trächtler: *Approach for Scenario-Based Test Specification for Virtual Commissioning*. In: *International Design Conference - Design 2014*. Dubrovnik, 2014.

- [SCH99A] E. Schnieder: *Methoden der Automatisierung: Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme*. Wiesbaden: Vieweg+Teubner Verlag, 1999.
- [SCH01] E. Schnieder: *Modellkonzepte in der Automatisierungstechnik*. In: Engels (Hrsg.): *Modellierung 2001*. Bonn: Gesellschaft für Informatik, 2001.
- [SCH99B] J. Schöttner: *Produktdatenmanagement in der Fertigungsindustrie: Prinzip, Konzepte, Strategien*. München: Hanser Verlag, 1999.
- [SSS08] G. Schuh, W. Stölzle, F. Straube: *Anlaufmanagement in der Automobilindustrie erfolgreich umsetzen: Ein Leitfaden für die Praxis*. Berlin, Heidelberg: Springer-Verlag, 2008.
- [SWD+13] F. Schumacher, G. Wolf, O. Drumm, A. Fay: *Anforderungen an die Feldgerätesimulation im Lebenszyklus von Anlagen*. In: *Automation 2013: 14. Branchentreff der Mess- und Automatisierungstechnik*. Düsseldorf: VDI-Verlag, 2013.
- [SCH07] R. Schumann: *Virtuelle Inbetriebnahme - Konzept und Realisierungsansätze*. In: *GMA-Kongress 2007: Automation im gesamten Lebenszyklus*. Düsseldorf: VDI-Verlag, 2007.
- [SHA48] C.E. Shannon: *A Mathematical Theory of Communication*. *The Bell System Technical Journal* (27), 1948, S. 379–423, 623–656.
- [SHE94] A. Shenhar: *A new systems engineering taxonomy*. *INCOSE International Symposium*, Vol. 5 (1), S. 723–732, 1995.
- [SNO00] D. Snowden: *The ASHEN Model: an enabler of action*. *Knowledge Management*, Vol. 7 (3), 2000.
- [SMZ07] F. Som, M. Michl, C. Ziegler: *Verkürzte Programmier- und Montagezeiten für kooperierende Industrieroboter*. *Zeitschrift für wirtschaftlichen Fabrikbetrieb: ZWF*, Vol. 102 (10), 2007, S. 686–689.
- [SPLA00] D. Spath, R. Landwehr: *3-D-Projektierung und Simulation von Ablaufsteuerungen*. *wt Werkstattstechnik online*, Vol. 90 (7-8), 2000, S. 292.
- [SPI09] M. Spitzweg: *Methode und Konzept für den Einsatz eines physikalischen Modells in der Entwicklung von Produktionsanlagen*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 2009.
- [STA73] H. Stachowiak: *Allgemeine Modelltheorie*. Wien, New York: Springer-Verlag, 1973.
- [STVÖ05] T. Stahl, M. Völter: *Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management*. 2. Auflage. Heidelberg: Dpunkt-Verlag, 2012.
- [SJF11] M. Strube, T. Jäger, A. Fay: *Integriertes Engineering durch Zusammenführen von Prozess- und Anlagenbeschreibung – Ein Konzept zur ganzheitlichen Beschreibung von Produktionsanlagen*. In: Schenk (Hrsg.): *Sichere und nachhaltige Logistik: 16. Magdeburger Logistiktage*. Magdeburg, Stuttgart: Fraunhofer Verlag, 2011.
- [TSG+11] W. Tegethoff, C. Schulze, M. Gräber, M. Huhn, N. Stulgies: *TEMO: Thermische Echtzeitfähige Modelle. – Abschlussbericht BMBF Verbundvorhaben*. Braunschweig, 2011.
- [UFF+08] A. Ulrich, A. Fay, M. Felleisen, B. Polke, U. Enste: *Richtlinie VDI/VDE 3682 „Formalisierte Prozessbeschreibungen“ – CAE-Werkzeug und Workflow*. In: Diedrich, Jumar, Schneider (Hrsg.): *Entwurf komplexer Automatisierungssysteme: EKA 2008*. Magdeburg: Univ., 2008.

- [VAJ94] S. Vajna: *CAD-CAM für Ingenieure: Hardware, Software, Strategien*. Braunschweig, Wiesbaden: Vieweg-Verlag, 1994.
- [VDM15] Verband Deutscher Maschinen- und Anlagenbau (VDMA): *Investieren in die Zukunft: Gemeinsame wirtschaftspolitische Positionen des deutschen Maschinen- und Anlagenbaus 2015*, Frankfurt am Main. 2015.
- [VDW97] Verein Deutscher Werkzeugmaschinenfabriken e. V. (VDW): *Abteilungsübergreifende Projektierung komplexer Maschinen und Anlagen (VDW-Bericht)*. Aachen, 1997.
- [VBK*11] B. Vogel-Heuser, S. Braun, B. Kormann, F. Friedrich: *Implementation and evaluation of UML as modeling notation in object oriented software engineering for machine and plant automation*. In: Bittanti (Hrsg.): *Proceedings of the 18th IFAC World Congress*. New York: IFAC, 2011.
- [VDF*13] B. Vogel-Heuser, C. Diedrich, A. Fay, P. Göhner: *Anforderungen an das Software-Engineering in der Automatisierungstechnik*. In: Kowalewski (Hrsg.): *Software Engineering 2013: Fachtagung des GI-Fachbereichs Softwaretechnik*. Bonn: Gesellschaft für Informatik, 2013.
- [WAFR07] F. Wagner, G. Frey: *Hardware-in-the-Loop-Simulationen bei kurzfristig zu langsamen Simulationsmodellen*. In: *GMA-Kongress 2007: Automation im gesamten Lebenszyklus*. Düsseldorf: VDI-Verlag, 2007.
- [WALÖ10] T. Wagner, U. Löwen: *Modellierung: Grundlage für integriertes Engineering*. In: *Automation 2010: 11. Branchentreff der Mess- und Automatisierungstechnik*. Düsseldorf: VDI-Verlag, 2010.
- [WEB02] K. Weber: *Inbetriebnahme verfahrenstechnischer Anlagen: Praxishandbuch mit Checklisten und Beispielen*. 2. Aufl. Berlin, Heidelberg: Springer Verlag, 2002.
- [WEBR06] M. Weck, C. Brecher: *Werkzeugmaschinen - Konstruktion und Berechnung*. 8. Aufl. Berlin: Springer Verlag, 2006.
- [WEFR95] R. von der Weth, E. Frankenberger: *Strategies, competence and style—Problem solving in engineering design*. *Learning and Instruction*, Vol. 5 (4), 1995, S. 357–383.
- [WSR92] D. Wünsch, B. Schlecht, W. Rulka: *Mehrkörpersimulation im Maschinen- und Anlagenbau*. *Konstruktion*, Vol. 44 (7-8), 1992, S. 265–273.
- [WÜN08] G. Wünsch: *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. München, TU München, Institut für Werkzeugmaschinen und Betriebswissenschaften. Dissertation, 2008.
- [ZSL08] M.F. Zäh, M. Spitzweg, F.-F. Lacour: *Einsatz eines Physikmodells zur Simulation des Materialflusses einer Produktionsanlage*. *Information technology: IT*, Vol. 50 (3), 2008, S. 192–198.
- [ZÄWÜ05] M.F. Zäh, G. Wünsch: *Schnelle Inbetriebnahme von Produktionssystemen: Qualitätssicherung von automatisierten Maschinen durch Simulation*. *wt Werkstattstechnik online*, Vol. 95 (9), 2005, S. 699–704.
- [ZWH*06] M.F. Zäh, G. Wünsch, T. Hensel, A. Lindworsky: *Feldstudie – Virtuelle Inbetriebnahme*. *wt Werkstattstechnik online*, Vol. 96 (10), 2006, S. 767–771.
- [ZEU98] K. Zeugträger: *Anlaufmanagement für Großanlagen*. Hannover, Universität Hannover, Institut für Fabrikanlagen. Dissertation, 1998.

Normen- und Richtlinienverzeichnis

Dieses Verzeichnis enthält eine Liste der referenzierten Normen, Richtlinien und Empfehlungen. Die Quellen werden mittels [<Typ> <Nummer>] gekennzeichnet.

- [DIN 8580] DIN 8580. *Fertigungsverfahren - Begriffe, Einteilung*, 2003.
- [DIN EN 62424] DIN EN 62424. *CAEX – Datenmodell zum Austausch von maschinell erstellten Informationen*, 2009.
- [DIN EN ISO 12100] DIN EN ISO 12100. *Sicherheit von Maschinen – Allgemeine Gestaltungsleitsätze – Risikobeurteilung und Risikominderung*, 2011.
- [DIN EN ISO 13849-1] DIN EN ISO 13849-1. *Sicherheit von Maschinen - Sicherheitsbezogene Teile von Steuerungen - Teil 1: Allgemeine Gestaltungsleitsätze*, 2008-12.
- [EU 2006/42/EG] EU 2006/42/EG. *Richtlinie 2006/42/EG des europäischen Parlaments und des Rates vom 17. Mai 2006 über Maschinen und zur Änderung der Richtlinie 95/16/EG*, 2006.
- [IEC 61131-3] IEC 61131-3. *Programmable controllers - Part 3: Programming languages*, 2013.
- [IEC 61512-1] IEC 61512-1. *Batch control – Part 1: Models and terminology*, 2000.
- [IEC PAS 62424] IEC PAS 62424. *Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools*, 2009.
- [IEC 62714-1] IEC 62714-1. *Engineering data exchange format for use in industrial automation systems engineering - Automation markup language - Part 1: Architecture and general requirements*, 2014.
- [IEEE STD. 610.3] IEEE STD. 610.3. *IEEE Standard Glossary of Modeling and Simulation Terminology*, 1989.
- [ISO/IEC 2382] ISO/IEC 2382. *Information technology - Vocabulary*, 2015.
- [NA 35] NA 35: *Abwicklung von PLT-Projekten*. 2003.
- [VDA 09] VDA-Empfehlung. *Informationen zum 3D-Datenaustausch in der Fabrikplanung*, 2009.
- [VDI 2206-1] VDI 2206-1. *Entwicklungsmethodik für mechatronische Systeme*, 2004.
- [VDI 2218] VDI 2218. *Informationsverarbeitung in der Produktentwicklung Feature-Technologie*, 2003.
- [VDI 2510] VDI 2510. *Fahrerlose Transportsysteme (FTS)*, 2005.
- [VDI 2860] VDI 2860. *Montage- und Handhabungstechnik*, 1990.
- [VDI 3423] VDI 3423. *Verfügbarkeit von Maschinen und Anlagen - Begriffe, Definitionen, Zeiterfassung und Berechnung*, 2011.
- [VDI 3581] VDI 3581. *Verfügbarkeit von Transport- und Lageranlagen sowie deren Teilsysteme und Elemente*, 2006.
- [VDI 3633-8] VDI 3633-8. *Simulation von Logistik-, Materialfluss- und Produktionssystemen - Blatt 8: Maschinennahe Simulation*, 2005.
- [VDI 3643] VDI 3643. *Elektro Hängebahnen - Oberläufer, Traglastbereich 500kg - Anforderungsprofil an ein kompatibles System*, 1998.

- [VDI 3682-1] VDI 3682-1. *Formalisierte Prozessbeschreibung - Blatt 1: Konzept und grafische Darstellung*, 2015.
- [VDI 3693-1] VDI 3693-1. *Virtuelle Inbetriebnahme – Blatt 1: Modellarten und Glossar*, 2015.
- [VDI 3694] VDI 3694. *Lastenheft/Pflichtenheft für den Einsatz von Automatisierungssystemen*, 2014.
- [VDI 3695-1] VDI 3695-1. *Engineering von Anlagen Evaluieren und optimieren des Engineerings - Blatt 1: Grundlagen und Vorgehensweise*, 2009.
- [VDI 3695-4] VDI 3695-4. *Engineering von Anlagen – Blatt 4: Evaluieren und optimieren des Engineerings, Themenfeld Hilfsmittel*, 2010.
- [VDI 3977] VDI 3977. *Empfehlung für die Abnahme von automatischen Lagersystemen*, 1993.
- [VDI 3979] VDI 3979. *Abnahmeregeln für Stückgut-Fördersysteme*, 1992.
- [VDI 4441-1] VDI 4441-1. *Hängefördertechnik - Elektrohängebahnen (EHB) - Eigenschaften und Anwendungsgebiete*, 2012.
- [VDI 4499-1] VDI 4499-1. *Digitale Fabrik - Blatt 1: Grundlagen.*, 2008.
- [VDI 4499-2] VDI 4499-2. *Digitale Fabrik - Blatt 2: Digitaler Fabrikbetrieb*, 2011.
- [VDI 5600-1] VDI 5600-1. *Fertigungsmanagementsysteme - Manufacturing-Execution-Systems (MES)*, 2007.
- [VDI 5610-1] VDI 5610-1. *Wissensmanagement im Ingenieurwesen - Blatt 1: Grundlagen, Konzepte, Vorgehen*, 2009.
- [VDI 6600-2] VDI 6600-2. *Projektingenieur - Anforderungsprofil an die Qualifikation*, 2009.

Verzeichnis verwendeter Internet-Quellen und erwähnter Software

Dieses Verzeichnis enthält eine Liste der referenzierten Internetquellen und erwähnter Software. Die Quellen werden mittels [<Kurzbeleg>[®]] gekennzeichnet.

- [3DEXPERIENCE[®]] 3DEXPERIENCE Plattform. Url: <https://www.3ds.com/de/ueber-dassault-systemes/3dexperience-plattform/> [Abruf: 23.11.2016]
- [AUTOMATIONML[®]] AutomationML e.V.: AutomationML. Url: <https://www.automationml.org/o.red.c/home.html> [Abruf: 23.11.2016]
- [AVANTI[®]] AVANTI Projekt. Url: <http://avanti-project.de/project.html> [Abruf: 23.11.2016]
- [BENTLEYARCH[®]] Bentley Architecture. Url: <https://www.bentley.com/en/solutions/industries/discrete-manufacturing> [Abruf: 23.11.2016]
- [BMWASL[®]] BMW: AutomatischeS Lager. Url: http://logiplus.de/upload/Consulting_Layout/Referenzen/Success_Stories/pdf/success_bmw_trm_de.pdf [Abruf: 23.11.2016]
- [CATIA[®]] CATIA. Url: www.3ds.com/de/produkte-und-services/catia/ [Abruf: 23.11.2016]
- [CODESYS[®]] CODESYS. Url: <https://de.codesys.com/> [Abruf: 23.11.2016]
- [COLLADA[®]] COLLADA. Url: <https://www.khronos.org/collada/> [Abruf: 23.11.2016]
- [DRA10A[®]] R. Drath: AutomationML: Architekturüberblick. Url: https://www.automationml.org/o.red/uploads/dateien/1317721862-AutomationML_Basic_Architecture.pdf [Abruf: 23.11.2016]
- [DÜRR[®]] Dürr: Applikation Scheibenkleben. Url: <http://www.durr-application-technology.com/de/applikationstechnik-produkte/kleben/> [Abruf: 23.11.2016]
- [E!COCKPIT[®]] e!COCKPIT. Url: <http://www.wago.de/produkte/produktkatalog/software/ecockpit/uebersicht/index.jsp> [Abruf: 23.11.2016]
- [ECL@SSA[®]] eCl@ss e.V.: eCl@ss: Classification and Product Description. Url: <http://www.eclass.eu/eclasscontent/standard/overview.html.de> [Abruf: 23.11.2016]
- [ECL@SSB[®]] eCl@ss e.V.: eCl@ss: Structure and structural elements. Url: http://wiki.eclass.eu/wiki/Category:Structure_and_structural_elements [Abruf: 23.11.2016]
- [ELTIME[®]] ELTIME. <http://www.hos.de/eltime-cae-software/> [Abruf: 23.11.2016]
- [ENTIME[®]] Heinz Nixdorf Institute: ENTIME. [https://www.hni.uni-paderborn.de/en/software-engineering/projekte/project-details/?tx_ttnews\[tt_news\]=97&cHash=28c37a4c694c8e40f687abed738cbe43](https://www.hni.uni-paderborn.de/en/software-engineering/projekte/project-details/?tx_ttnews[tt_news]=97&cHash=28c37a4c694c8e40f687abed738cbe43) [Abruf: 23.11.2016]
- [EPLAN[®]] EPLAN. Url: <https://www.eplan.de/de/loesungen/elektrotechnik/eplan-electric-p8/> [Abruf: 23.11.2016]
- [ESPLAN[®]] ESPLAN. Url: <http://www.esplan.eu/cae/> [Abruf: 23.11.2016]
- [FMI[®]] Modelisar: FMI for Model Exchange 1.0 Specification. Url: https://svn.modelica.org/fmi/branches/public/specifications/v2.0/FMI_for_ModelExchange_and_CoSimulation_v2.0.pdf [Abruf: 23.11.2016]

- [GMA FA 6.11[®]] VDI/GMA Fachausschuss 6.11: Virtuelle Inbetriebnahme. Url: <http://www.vdi.de/technik/fachthemen/mess-und-automatisierungs-technik/fachbereiche/engineering-und-betrieb-automatisierter-anlagen/gma-fa-611-virtuelle-inbetriebnahme/> [Abruf: 23.11.2016]
- [LIBNODAVE[®]] Libnodave. Url: <http://www.proview.se/misc/libnodave/gettingStarted.html> [Abruf: 23.11.2016]
- [MATLAB[®]] Matlab. Url: <http://de.mathworks.com/products/matlab/> [Abruf: 23.11.2016]
- [MODELICA[®]] Modelica. Url: <http://www.modelica.org/> [Abruf: 23.11.2016]
- [MVIBN[®]] DFG: Methodik zur virtuellen Inbetriebnahme auf Basis von objektorientierten Verhaltensmodellen mit wählbarer Modellierungstiefe. Url: <http://gepris.dfg.de/gepris/projekt/183279191> [Abruf: 23.11.2016]
- [NX[®]] NX. Url: http://www.plm.automation.siemens.com/en_us/products/nx/11/index.shtml [Abruf: 23.11.2016]
- [OPCUATSN[®]] I. Näther: TSN – der Turbo für OPC UA? Url: <http://www.elektrotechnik.vogel.de/tsn-der-turbo-fuer-opc-ua-a-530318/> [Abruf: 24.07.2016]
- [PCWORX[®]] PCWORX. Url: <https://www.phoenixcontact.com> [Abruf: 23.11.2016]
- [PLCOPEN XML[®]] PLCopen XML. Url: <http://www.plcopen.org/> [Abruf: 23.11.2016]
- [PRO/ENGINEER[®]] Pro/ENGINEER. Url: <http://www.ptc.com/cad> [Abruf: 23.11.2016]
- [PROFINET[®]] PROFIBUS Nutzerorganisation e. V. (PNO): PROFINET: Systembeschreibung - Technologie und Anwendung. Url: <http://www.profibus.com/nc/download/technical-descriptions-books/downloads/profinet-technology-and-application-system-description/download/20727/> [Abruf: 23.11.2016]
- [ROBSIM[®]] RF:RobSim. Url.: <http://www.eks-intec.de/produkte/rf-suite.html> [Abruf: 23.11.2016]
- [ROFA[®]] ROFA AG: ROFA INDUSTRIAL AUTOMATION AG. Url: www.rofa-ag.de [Abruf: 23.11.2016]
- [SEMANZ 4.0[®]] A. Fay: Semantische Allianz für Industrie 4.0: SemAnz40. Url: <https://www.semanz40.de> [Abruf: 23.11.2016]
- [SIEMENSPLM[®]] Siemens PLM. Url: www.plm.automation.siemens.com/de_de/ [Abruf: 23.11.2016]
- [STEP 7[®]] STEP 7. Url: <http://w3.siemens.com/mcms/automation/de/automatisierungssysteme/automatisierungssoftware/Seiten/Default.aspx> [Abruf: 23.11.2016]
- [TEAMCENTER[®]] Teamcenter. Url: http://www.plm.automation.siemens.com/de_de/products/teamcenter/ [Abruf: 23.11.2016]
- [TRICAD[®]] TRICAD. Url: <https://www.venturisit.de/> [Abruf: 23.11.2016]
- [VDA[®]] VDA: Verband der Automobilindustrie. Url: <https://www.vda.de/de.html> [Abruf: 23.11.2016]

- [VDA AK DF[®]] VDA Arbeitskreis Digitale Fabrik: Arbeitsgruppe Virtuelle Inbetriebnahme. Url: <https://www.vda.de/de/verband/organisation/organisation-ausschuesse/arbeitskreis-digitale-fabrik.html> [Stand: 18.04.2016]
- [VDIA[®]] VDI: Verein Deutscher Ingenieure. Url: <https://www.vdi.de/> [Abruf: 23.11.2016]
- [VDIB[®]] VDI: *Der Ingenieur/ die Ingenieurin in der Mechatronik*. Url: https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gpp_dateien/Berufsbilder/berufsbild_Mechatronik.pdf [Abruf: 23.11.2016]
- [WIKI[®]] N.N.: *Wikipedia: Programming by permutation*. Url: https://en.wikipedia.org/wiki/Programming_by_permutation [Abruf: 23.11.2016]
- [WINCC[®]] WINCC. Url: <http://w3.siemens.com/mcms/human-machine-interface/de/visualisierungssoftware/scada-wincc/seiten/default.aspx> [Abruf: 23.11.2016]
- [WINCCFLEXIBLE[®]] WINCC Flexible. Url: <http://w3.siemens.com/mcms/human-machine-interface/de/visualisierungssoftware/wincc-flexible/seiten/default.aspx> [Abruf: 23.11.2016]
- [WINMOD[®]] WINMOD. Url: <http://winmod.de/de/> [Abruf: 23.11.2016]
- [WIRESHARK[®]] WireShark. Url: <https://www.wireshark.org/#1398253364-1-69> [Abruf: 23.11.2016]
- [WSCAD[®]] WSCAD. Url: <http://www.wscad.de/> [Abruf: 23.11.2016]
- [ZENON[®]] Zenon. Url: <http://www.copadata.com/de/produkte.html> [Abruf: 23.11.2016]

Verzeichnis der Veröffentlichungen des Verfassers

Dieses Verzeichnis enthält eine Liste der Veröffentlichungen des Autors, die mit [<Kurzbeleg>#] gekennzeichnet sind.

- [PFR+14#] P. Puntel Schmidt, A. Fay, W. Riediger, T. Schulte, F. Köslin, S. Diehl: *Validierung von Steuerungscode fertigungstechnischer Anlagen mit Hilfe automatisch generierter Simulationsmodelle*. In: Jumar, Diedrich (Hrsg.): Entwurf komplexer Automatisierungssysteme EKA 2014. Magdeburg: Institut für Automation und Kommunikation e.V., 2014.
- [KPR+14#] F. Köslin, P. Puntel Schmidt, W. Riediger, A. Fay: *Entwurf einer Modelica Simulationsbibliothek für die virtuelle Inbetriebnahme fertigungstechnischer Anlagen*. In: 7th International Symposium on Automatic Control: AUTSYM 2014. Wismar, 2014.
- [RPK+14#] W. Riediger, P. Puntel Schmidt, F. Köslin, T. Schulte: *Hardware-in-the-Loop-Simulation fertigungstechnischer Anlagen*. In: Proceedings of the SPS/IPC/DRIVES: Internationale Fachmesse und Kongress: Elektrische Automatisierung - Systeme und Komponenten. Nürnberg: Aprimus-Verlag, 2014.
- [PSFA15A#] P. Puntel Schmidt, A. Fay: *Levels of Detail and Appropriate Model Types for Virtual Commissioning in Manufacturing Engineering*. In: Breitenecker, Kugi, Troch (Hrsg.): MATHMOD 2015: 8th Vienna International Conference on Mathematical Modelling. Vienna: ARGESIM (ARGESIM Report, no. 44), 2015.
- [PSFA15B#] P. Puntel Schmidt, A. Fay: *Applying the Domain-Mapping-Matrix to Identify the Appropriate Level of Detail of Simulation Models for Virtual Commissioning*. In: 2nd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control: CESCIT. Maribor, 2015.
- [PSFA15C#] P. Puntel Schmidt, A. Fay: *Konsistente Simulationsmodelle für die virtuelle Inbetriebnahme fertigungstechnischer Anlagen mit Hilfe regelbasierter Modellverbinder*. In: Rabe, Clausen (Hrsg.): Simulation in Production and Logistics 2015. Stuttgart: Fraunhofer Verlag (ASIM-Mitteilung, Nr. 157), 2015.
- [PSFA15D#] P. Puntel Schmidt, A. Fay: *Potential of Dynamically Adaptable Simulation Models for Virtual Commissioning*. SNE - Simulation News Europe, Vol. 25 (2), 2015, S. 59–68.
- [PSFA15E#] P. Puntel Schmidt, A. Fay: *Transformation of Continuous Simulation Models of Automated Manufacturing Systems into Discrete Event Models on Different Levels of Detail*. In: IEEE 20th Conference on Emerging Technologies & Factory Automation: ETFA 2015. Piscataway, NJ: IEEE, 2015.
- [PFR+15#] P. Puntel Schmidt, A. Fay, W. Riediger, T. Schulte, F. Köslin, S. Diehl: *Validierung von Steuerungscode mit Hilfe automatisch generierter Simulationsmodelle*. at - Automatisierungstechnik, Vol. 63 (2), 2015.
- [PKF15#] P. Puntel Schmidt, F. Köslin, A. Fay: *Potential of Dynamically Adaptable Simulation Models for Virtual Commissioning*. In: Wahmkow, Roßmanek, Wendorf (Hrsg.): Workshop der ASIM/GI-Fachgruppen "Simulation technischer Systeme" und "Grundlagen und Methoden in Modellbildung und Simulation" 2015. ASIM-Mitteilungen, 154, 2015.
- [BPH+15#] M. Barth, P. Puntel Schmidt, M. Hoernicke, M. Oppelt, G. Wolf, O. Stern, L. Hundt: *Methoden und Modelle der Virtuellen Inbetriebnahme: Eine Übersicht der Richtli-*

- nienarbeit des GMA FA 6.11*. In: Automation 2015: 16. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verlag, 2015.
- [FDT*15A[#]] A. Fay, C. Diedrich, M. Thron, A. Scholz, P. Puntel Schmidt, J. Ladiges, T. Holm: *Wie bekommt Industrie 4.0 Bedeutung?* In: Automation 2015: 16. Branchentreff der Mess- und Automatisierungstechnik. Düsseldorf: VDI-Verlag, 2015.
- [FDT*15B[#]] A. Fay, C. Diedrich, M. Thron, A. Scholz, P. Puntel Schmidt, J. Ladiges, T. Holm: *Wie bekommt Industrie 4.0 Bedeutung?* atp edition, Vol. 57 (8), 2015.
- [DDB*16[#]] P. Diekhake, D. Diaz, U. Becker, M. Günther, A. Scholz, P. Puntel Schmidt, A. Fay: *Bewertung verteilter Gebäudeautomatisierungssysteme auf Basis ihrer Beschreibung mittels AutomationML und ihrer Simulation unter Anwendung automatisiert erzeugter Petrinetze*. In: EKA 2016: 14. Fachtagung Entwurf komplexer Automatisierungssysteme. Magdeburg, 2016.
- [GDS*16[#]] M. Günther, P. Diekhake, A. Scholz, S. Bussenius, P. Puntel Schmidt, U. Becker, A. Fay: *Unterstützung bei der Planung und Auslegung einer Gebäudeautomation*. In: *Automation 2016*, 2016.
- [GSP*16A[#]] M. Günther, A. Scholz, P. Puntel Schmidt, A. Fay, P. Diekhake, D. Diaz, U. Becker: *Anforderungserhebung und -modellierung in der Gebäudeautomation*. In: EKA 2016: 14. Fachtagung Entwurf komplexer Automatisierungssysteme. Magdeburg, 2016.
- [GSP*16B[#]] M. Günther, A. Scholz, P. Puntel Schmidt, A. Fay, P. Diekhake, D. E. Diaz Fuentes, U. Becker: *Requirements engineering and modelling for Building automation systems*. IEEE International Conference on Emerging Technology & Factory Automation (ETFA 2016). Berlin, 2016.
- [FBD*16[#]] E. Diaz Fuentes, U. Becker, P. Diekhake, M. Günther, A. Scholz, P. Puntel Schmidt, A. Fay: *Evaluation and simulation of Building automation systems based on their AutomationML description*. IEEE International Conference on Emerging Technology & Factory Automation (ETFA 2016). Berlin, 2016.
- [PGF*16[#]] P. Puntel Schmidt, M. Goß., A. Fay, H. Bertsch, B. Welling, F. Marek: *Ein Reifegradmodell für Engineering-Daten - Bewertung der Eignung von CAD-Layouts für die Erstellung von Simulationsmodellen*. In: *Werkstattstechnik wt-online*, Heft 9/2016.

Verzeichnis der betreuten studentischen Arbeiten

Dieses Verzeichnis enthält eine Liste der studentischen Arbeiten, die vom Autor betreut wurden. Die Quellen sind mittels [<Kurzbeleg>%] gekennzeichnet.

- [DIE14%] M. Dietrich: *Integration von Verhaltensbeschreibungen in Modelica durch PLCopen XML*. Hamburg, Helmut-Schmidt-Universität/Universität der Bundeswehr, Institut für Automatisierungstechnik. Masterarbeit, 2014.
- [GOß16%] M. Goß: *Ein Reifegradmodell für Engineering-Daten*. Hamburg, Helmut-Schmidt-Universität/Universität der Bundeswehr, Institut für Automatisierungstechnik. Masterarbeit, 2016.

Lebenslauf

Persönliches

Geboren am 20.03.1985
in São Paulo/Brasilien

Familienstand: Verheiratet

Berufserfahrung:

Seit 11/2016	Entwicklungsingenieur bei der Siemens AG, Nürnberg
09/2012 – 10/2016	Wissenschaftlicher Mitarbeiter am Institut für Automatisierungstechnik, Prof. Dr.-Ing. Alexander Fay; Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg
03-2009 – 03-2012	Systemingenieur bei der Kocher Gesellschaft für Industrieautomation und Software mbH, Ismaning

Studium:

03/2011 bis 09/2012	Studium an der Hochschule München Studiengang Systems Engineering Abschluss: M. of Systems Engineering
10/2004 bis 03/2009	Studium an der Hochschule München Studiengang Feinwerk- und Mikrotechnik, Vertiefungsrichtung Produktion und Automatisierung Abschluss: Dipl.-Ing. (FH)
09/2002 bis 07/2004	Fachoberschule München Fachrichtung Technik Abschluss: Fachhochschulreife